Oracle® Fusion Middleware
Administering Oracle WebLogic Server with RESTful Management Services

14c (14.1.2.0.0)
F95844-01
December 2024

Oracle Fusion Middleware Administering Oracle WebLogic Server with RESTful Management Services, 14c (14.1.2.0.0)

F95844-01

Contents

Preface

This preface describes the document accessibility features and conventions used in this guide: *Administering Oracle WebLogic Server with RESTful Management Services*.

The Preface includes the following topics:

- Document Scope and Audience
- Documentation Accessibility
- Diversity and Inclusion
- New and Changed Features in this Release
- Conventions

Document Scope and Audience

This document describes how to use Oracle WebLogic Server RESTful management interfaces for administration, monitoring, deploying, and configuration tasks, which are exposed for developing RESTful clients.

The user communities for this documentation are administrators who might use cURL commands to invoke these resources in administration scripts, and software developers who will use this information when writing code, perhaps in Java, perhaps in other languages, that monitors and manages WLS domains.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

New and Changed Features in this Release

For a comprehensive listing of all the new Oracle WebLogic Server features introduced in this release, see *What's New in Oracle WebLogic Server*.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

This chapter introduces the guide, *Administering Oracle WebLogic Server with RESTful Management Services*, and describes the Oracle WebLogic Server REST resources and the WLS REST API reference documentation.

- Information Roadmap

Information Roadmap

The Oracle WebLogic Server REST resources are based on WLS bean trees and organized according to their corresponding root resources. See Mapping the WLS Beans to REST.

Each REST manual refers to resources accessible to domain user roles. In the reference guides, REST resources:

- Run at the domain level.
- Must be accessed over a URL by a user defined in the domain's default security realm.
- Can be used to manage a domain.
- Can be used to manage all WLS MBeans.

See Table 1-1 for a complete listing of the Oracle WebLogic Server REST reference documents and descriptions of their use.

Table 1-1 WLS RESTful Management Interface Reference Documentation

Book Title	Use these REST Resources to...
RESTful Edit Reference for Oracle WebLogic Server	Edit the WLS configuration.
RESTful Domain Configuration Reference for Oracle WebLogic Server	View the last activated WLS configuration.
RESTful Domain Runtime Reference for Oracle WebLogic Server	Monitor the entire WLS domain.
RESTful Server Configuration Reference for Oracle WebLogic Server	View the WLS configuration that the Administration Server or Managed Server is currently running against.
RESTful Server Runtime Reference for Oracle WebLogic Server	Monitor the Administration Server or a Managed Server. You can monitor a Managed Server either by using the Administration Server's `domainRuntime/serverRuntimes/<managedServerName>/...` resources or the Managed Server's `serverRuntime/...` resources.

2

About the WLS RESTful Management Interface

Learn about the Oracle WebLogic Server RESTful management interface, and how the WLS MBeans are mapped to the REST interfaces.

This chapter describes the RESTful management services supported by Oracle WebLogic Server. This chapter includes the following topics:

- Introduction to the WLS RESTful Management Interface
- Mapping the WLS Beans to REST
  Learn how the WLS beans are mapped to the REST interfaces.
- Returning Error Messages
  Resources use different formats for returning error messages.

Introduction to the WLS RESTful Management Interface

The Oracle WebLogic Server RESTful management interface provides comprehensive support for Oracle WebLogic Server administration through the dynamic generation of REST resources based on WLS MBeans and descriptor interfaces. There are resources to support the configuration and monitoring of domain environments, lifecycle management (LCM) resources, and legacy resources from 12.1.3.
For a guide to the WLS REST reference documentation, see Information Roadmap.

Note:

In Oracle WebLogic Server 14.1.2.0.0:

- All prior versions are now deprecated: 12.2.1.0.0, 12.2.1.1.0, 12.2.1.2.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0.
- 14.1.2.0.0: this is the `latest` version

To summarize the changes in this release:

- All prior versions have been deprecated and you should use the 14.1.2.0.0 REST resources instead.
- The `latest` version has changed from 14.1.1.0.0 to 14.1.2.0.0.
- All new MBean features added in 14.1.2.0.0 will show up in 12.2.1.1.0, 12.2.1.2.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0.
- Any MBean features that were deprecated in 14.1.2.0.0, will still be available using the 12.2.1.1.0, 12.2.1.2.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0 REST URLs. However, they will not available for the 14.1.2.0.0 REST URLs.

- Generated REST API for WLS Bean Trees

- WLS Bean Tree Overview

Generated REST API for WLS Bean Trees

The WLS beans are used extensively by WLS components to manage configuration settings and to monitor and manage running servers.

The WLS beans are derived from Java interfaces. At runtime, WLS constructs internal trees of Java beans that can be used to configure and monitor the system. In prior releases, the bean trees were exposed only through JMX, WLST, and configuration files (for example, `config.xml`).

In this release, WLS dynamically generates the REST resources, incrementally and on-demand at runtime, by using the bean trees and bean infos. These REST resources provide an alternative for managing WLS.

WLS Bean Tree Overview

The following sections provide the background information about WLS beans which provide the foundation for the REST interfaces.

There are two main bean types:

- Configuration: Used to configure WLS.
- Runtime: Used to monitor WLS and for some operations, control WLS (for example, starting and stopping servers, shrinking data source connection pools, and so on).

WLS provides the following bean trees:

- Edit access: Available only on the Administration Server, used to modify the configuration (for example, `config.xml` and system resource files).
- Runtime access: Available on every server, used to view that server's configuration and to access its monitoring data.
- Domain access: Available only on the Administration Server, contains copies of the runtime beans of all of the running servers, provides a single point of access for monitoring, is also used to view the most current configuration that has been persisted.

For more information about WLS MBeans, see Understanding WebLogic Server MBeans in *Developing Custom Management Utilities Using JMX for Oracle WebLogic Server*.

The Oracle WebLogic Scripting Tool (WLST) presents the bean trees as follows:

- edit: Matches the underlying edit access bean tree.
- domainConfig: The configuration MBean half of the domain access bean tree (such as, the last persisted configuration).
- domainRuntime: The runtime MBean half of the domain access bean tree (such as, for monitoring all servers).
- serverConfig: The configuration MBean half of the runtime access bean tree (such as, the configuration the server is using).
- serverRuntime: The runtime MBean half of the runtime access bean tree (such as, for monitoring a specific server).

The REST resources parallel the MBean trees presentation in WLST: edit, domainConfig, domainRuntime, serverConfig, and serverRuntime.

Within the WLS bean trees, there are several types of parent/child (containment) relationships:

- Writable collections: For example, a domain bean has a collection of server beans.
- Mandatory singletons: For example, a server bean always has an SSL bean which is automatically created and cannot be deleted.
- Optional singletons: For example, an overload protection bean can optionally have a server failure trigger bean.

Beans can include properties (generally scalars, strings, and arrays), references to other beans, and operations (for example, to start a server).

With regard to contained collections:

- Each child has a unique identity within the collection (for example, each network channel has a name that's unique within its server).
- Most collections are homogeneous (for example, a domain's applications) though a few are heterogeneous (for example, a security realm's authentication providers).

Mapping the WLS Beans to REST

Learn how the WLS beans are mapped to the REST interfaces.

- General REST Patterns
- About the Root Resources
- Naming Conventions
- Mapping the REST URLs
- JSON Mappings

General REST Patterns

Almost all of the WLS beans (a homogeneous collection of children, a mandatory singleton child, and an optional homogenous singleton child) use the following REST patterns:

WLS Beans/REST Resource	REST Method	Description
Collections: collection resource	GET	Returns the collection.
	POST	Creates a new item in the collection.
Collections: create form resource	GET	Returns a pre-populated entity.
Collections: child resource	GET	Returns an item in the collection.
	POST	Updates an item in the collection.
	DELETE	Removes an item from the collection.
Singletons: singleton resource	GET	Returns the singleton.
	POST	Updates the singleton if it exists; creates it if it does not.
	DELETE	Removes the singleton.
Operations: action resource	POST	Invokes the operation.

About the Root Resources

The Administration Server and each running Managed Server hosts a REST web application that runs on each server's administrative port. The context root for each is `management`. The root REST resources mimic the bean trees in WLST.

Table 2-1 describes the root resources on the Administration Server and lists the corresponding bean tree.

Table 2-1 Administration Server Root Resources

URL	Description	Corresponding Bean Tree
`management/weblogic/<version>/edit`	Edits the WLS configuration.	Administration Server's edit tree domain bean.
`management/weblogic/<version>/serverConfig`	Views the WLS configuration that the Administration Server is currently running against.	Administration Server's server runtime tree domain bean.
`management/weblogic/<version>/serverRuntime`	Monitors the Administration Server.	Administration Server's server runtime tree server runtime bean.
`management/weblogic/<version>/domainConfig`	Views the last activated WLS configuration.	The Administration Server's domain runtime tree domain bean.
`management/weblogic/<version>/domainRuntime`	Monitors the entire WLS domain.	The Administration Server's domain runtime tree domain runtime bean.
`management/weblogic/<version>/domainRuntime/serverRuntimes`	Monitors all the running servers in the WLS domain via the Administration Server.	Each running server's server runtime tree server runtime bean.
`management/weblogic/<version>/domainRuntime/serverRuntimes/<serverName>`	Monitors a specific running server in the WLS domain via the Administration Server.	The specified server's server runtime tree server runtime bean.
`management/lifecycle`	Lifecycle management (LCM) REST resources.	n/a
`management/wls`	12.1.3 (legacy) WLS REST resources.	n/a

Table 2-2 describes the root resources on Managed Servers.

Table 2-2 Managed Server Root Resources

URL	Description	Corresponding Bean Tree
`management/weblogic/<version>/serverConfig`	Views the WLS configuration that a Managed Server is currently running against.	Managed Server's server runtime tree domain bean.
`management/weblogic/<version>/serverRuntime`	Monitors the Managed Server.	Managed Server's server runtime tree server runtime bean.

The URLs on Managed Servers are exactly like the ones on the Administration Server, except that the host and port are different.

For example, the URL to view the Administration Server's server runtime:

```
curl ... -X GET http://adminHost:7001/management/weblogic/latest/serverRuntime
```

The URL to view a Managed Server's server runtime:

```
curl ... -X GET http://managed1Host:7002/management/weblogic/latest/serverRuntime
```

Naming Conventions

The WLS bean property names are mapped to names in the REST URLs and JSON object properties. The WLS property names usually start with an upper case letter (for example, Domain, JDBCDataSource, ServerRuntime) whereas the REST naming conventions use camel case, lower then upper case letters (for example, domain, JDBCDataSource, serverRuntime).

Mapping the REST URLs

Each WLS bean is mapped to a separate REST resource. Contained collections and operations are also mapped to separate resources. All WLS beans are either root resources (for example, the domain), contained collection children (for example, servers) or contained singleton children (for example, a server's SSL configuration).

The URLs for the root resources are listed in Table 2-1 and Table 2-2.

Each contained collection bean property maps to a URL for the entire collection as well as a URL for each child. For example:

URL	Description	Example
`<parent>/<collectionPropertyName>`	Manages the entire collection.	`.../edit/servers`
`<parent>/<collectionPropertyName>/<childName>`	Manages a child in the collection.	`.../edit/servers/Server-0`

Similarly, each contained singleton bean property maps to its own URL. For example, a server's SSL bean maps to `.../edit/servers/<serverName>/SSL`.

If a contained bean property is creatable (for example, you can add a new server to the domain's servers collection, or you can create an `RDBMSSecurityStore` for the domain), then create form resources are also provided which return a template JSON object with default values to help you create the new resource. The general procedure is that you GET the create form, fill in the values, then POST it back to create the new resource. If any fields are not filled in, they retain their current values. The URLs of the create form resources are `<parent>/<singlularCollectionPropertyName>CreateForm`. For example:

- `.../edit/serverCreateForm`
- `.../edit/securityConfiguration/realms/myrealm/RDBMSSecurityStoreCreateForm`

Each bean operation maps to its own URL. For example, `.../domainRuntime/serverRuntimes/<serverName>/shutdown` is used to shut down a specific server.

Most of the WLS bean operations are used to create, delete, list, and find contained beans. These operations are handled separately in REST (versus exposed as REST operation URLs). For information about these beans, see Using the WLS RESTful Management Interface.

JSON Mappings

REST maps the various Java types that the WLS beans use (for example, their properties, operation arguments, and return types) to JSON.

- Strings and Scalars
- Arrays
- Identities
- WLS Bean References
- java.util.Properties
- Encrypted Properties

Strings and Scalars

Java strings and scalars are mapped to their JSON equivalent.

Java	JSON	Example (Java to JSON)
`java.lang.String`	string or null	"Foo" -> "Foo" null -> null
`char`, `java.lang.Character`	string	'a' -> "a"

`int`
`java.lang.Integer`
`long`
`java.lang.Long`
	number	7001 -> 7001

`float`
`java.lang.Float`
`double`
`java.lang.Double`
	number	1.23 -> 1.23

`boolean` `java.lang.Boolean`	boolean	true -> true

Arrays

Non-null Java arrays are mapped to JSON arrays. Null Java arrays are mapped to a JSON null.

Identities

Each WLS bean is uniquely identified within its bean tree by the trailing part of its URL, after the version specifier. For example, `edit/machines/Machine-0`.

This identity is mapped to a JSON string array, with one string for each path segment past the root resource of the tree. For example:

```
[ "machines", "Machine-0" ]
```

WLS Bean References

Some WLS bean properties contain references to other WLS beans (versus a containment relationship). The same is true for operation arguments and return types. For example, a Server bean has a reference to a Machine bean, and a Deployment bean has a reference to an array of Target beans.

Singleton references (for example, a server's machine) map to a property whose value is the identity of the referenced bean, as well as a link. For example:

```
{
  machine: [ "domain", "machines", "Machine-0" ],
  links: [
    { rel: "machine", href: "http://localhost:7001/management/latest/weblogic/edit/
machines/Machine-0" }
  ]
}
```

Collections of references (for example, a server's candidate machines) map to an array property where each element is an object containing the referenced bean's identity as well as a link to the bean. For example:

```
{
  candidateMachines: [
    {
      identity: [ "machines", "Machine-0" ],
      links [ { rel: "canonical", href: "http://localhost:7001/management/weblogic/
latest/edit/machines/Machine-0"
    },
    {
      identity: [ "machines", "Machine-1" ],
      links [ { rel: "canonical", href: "http://localhost:7001/management/weblogic/
latest/edit/machines/Machine-1"
    }
  ]
}
```

A null reference or null reference collection is mapped to a JSON null.

java.util.Properties

`java.util.Properties` holds lists of properties. For example, a CommonLogMBean `LoggerSeverityProperties` property. It is mapped to a JSON object, with a matching string property for each property in the set of properties. For example:

```
{
  "property1": "value1",
  "property2": "value2"
}
```

Null `java.util.Properties` are mapped to a JSON null.

Encrypted Properties

Some WLS bean string properties are encrypted because they hold sensitive data, such as passwords. While clients must be able to set passwords (this is done by passing them in as cleartext strings), other users are not allowed to view them but they might want to know whether the password has a value (versus null, is not set).

The mapping is different for inbound versus outbound encrypted properties.

For outbound encrypted properties, if the password is null, it is mapped to a JSON null. If not, then it is mapped to the JSON string `@Oracle_Confidential_Property_Set_V1.1#`.

For inbound encrypted properties, you would typically perform a `GET` to get the current value of a resource, set the values for the properties that should be changed, leaving the others with their current values, then `POST` the new value back. Therefore, if the value in the `POST` is `@Oracle_Confidential_Property_Set_V1.1#`, then the property is not changed (it retains the old property value). Otherwise, the value is changed to the cleartext string value in the `POST`.

Returning Error Messages

Resources use different formats for returning error messages.

Error Messages with One Error String
If a resource returns one error string, it uses this format:

```
HTTP/1.1 400 Bad Request
{
  type: "http://oracle/TBD/WlsRestMessageSchema",
  title: "FAILURE",
  detail: "Bean already exists:
\"weblogic.management.configuration.ServerMBeanImpl@31fa1656([mydomain]/
Servers[Server-1])\"",
  status: 400
}
```

Error Messages with More Than One Error String
If a resource returns more than one error string, it uses this format:

```
HTTP/1.1 400 Bad Request
{
  type: "http://oracle/TBD/WlsRestMessagesSchema",
  title: "ERRORS",
  status: 400,
  wls:errorsDetails: [
    {
      type: "http://oracle/TBD/WlsRestMessageSchema",
      title: "FAILURE",
      detail: "no-such-protocol is not a legal value for DefaultProtocol.\
      The value must be one of the following: [t3, t3s, http, https, iiop, iiops]",
      o:errorPat: "defaultProtocol"
    },
    {
      type: "http://oracle/TBD/WlsRestMessageSchema",
      title: "FAILURE",
      detail: "Type mismatch. Cannot convert abc to int",
      o:errorPath: "listenPort"
    }
  ]
}
```

3 Using the WLS RESTful Management Interface

Learn how to use the RESTful management services supported by Oracle WebLogic Server.For example scripts that show how to use the WLS REST APIs to perform common domain management and monitoring tasks, see Domain Level REST API Examples.
This chapter includes the following topics:

- Accessing REST Resources
  Each REST resource method documents which user roles can access it: Admin, Deployer, Operator, Monitor.
- Viewing WLS Beans
  To view a WLS bean, invoke the HTTP `GET` method on its corresponding REST URL.
- Viewing Collections of Contains Beans
  To view a collection of WLS beans, invoke the HTTP `GET` method on its corresponding REST URL.
- Retrieving Create Forms
  To retrieve a create form for creating a new resource, invoke the HTTP `GET` method on its corresponding create form REST URL.
- Filtering Results
  Bean, collection, and create form resource `GET` methods support these query parameters to let you omit properties and links from the response.
- Modifying the WLS Configuration
  You can create, modify, and delete beans in the edit tree *only* (`.../management/weblogic/<version>/edit/...`). The other bean trees are read-only.
- Using Multiple Edit Sessions
  In a previous release, WebLogic Server introduced multiple edit sessions, named concurrent edit sessions, which allows more than one administrator to make configuration changes at the same time. These edit sessions are domain scoped. Each scope has a default edit session. Edit session names are unique within a scope, but not across scopes.
- Creating WLS Configuration Beans
  You create a new WLS configuration bean by calling `POST` with a JSON structure containing the new bean's properties.
- Managing Whether a Property is Set
- Invoking Operations
  Each WLS bean operation maps to its own REST URL. In the case of overloaded operations (for example, `shutdown()` versus `shutdown(int, boolean)`), all the overloaded operations map to the same URL and the resource looks at the incoming arguments to determine which operation to invoke.
- Using Queries
  The REST API includes a powerful bulk access capability that lets you dynamically describe a tree of beans that can be returned in one call.

- About Synchronous and Asynchronous Operations
  Several MBean operations (for example, server lifecycle, deployment) are asynchronous. They return job MBeans that must be monitored to determine when the job has completed.
- Deploying Applications and Libraries
  You view deployed applications and libraries in the edit tree. You call `POST` on the collections to deploy them, and `DELETE` to undeploy them.
- Cross-Origin Resource Sharing (CORS) for WebLogic Server REST APIs
  As of 14.1.1.0.0, Oracle WebLogic Server RESTful management services support client-side Cross-Origin Resource Sharing (CORS).

Accessing REST Resources

Each REST resource method documents which user roles can access it: Admin, Deployer, Operator, Monitor.

In general:

- You must be in the Admin, Deployer, Operator, or Monitor role to read resources (use the `GET` method).
- You must be in the Admin role to write resources (use the `POST` and `DELETE` methods) or to invoke operations (using `POST`).
- However, with certain resources, a Deployer can deploy and undeploy applications and libraries, and an Operator can start a server.

For a domain user (for example, a user defined in the domain's default security realm), the URL to access REST resources starts with `http://host:port/management`.

Viewing WLS Beans

To view a WLS bean, invoke the HTTP `GET` method on its corresponding REST URL.

For example, to get the configuration for the server, `Server-0`:

```
GET http://localhost:7001/management/weblogic/latest/edit/servers/Server-0
```

`GET` returns a standard WLS REST response body. It returns a JSON object containing the bean's properties and a `links` property, a JSON array containing links to related resources.

- About WLS Bean Properties
- Self and Canonical Links
- Parent Links
- Self Create Form Links
- Child Bean Links
- Child Create Form Links
- Singleton Bean Reference Links
- Bean Reference Collection Links
- Operation Links

About WLS Bean Properties

The returned JSON object contains the WLS bean's properties (for example, typical properties and references, but not children), using the standard Java to JSON mappings (see JSON Mappings). It also includes an `identity` property that specifies the bean's identity. For example:

```
{
  identity: [ "domain", "servers", "Server-0" ],
  name: 'Server-0',
  listenPort: 7001,
  machine: { identity: [ "domain", "machines", "Machine-0" ] }
}
```

Self and Canonical Links

All resources include a `self` and a `canonical` top level link that refer to the resource. For example, a server contains `self` and `canonical` links that refer to the specified server:

```
{
  links: [
    { rel: "self", href: "http://localhost:7001/management/weblogic/latest/edit/servers/
Server-0" }
    { rel: "canonical", href: "http://localhost:7001/management/weblogic/latest/edit/
servers/Server-0" }
   ]
}
```

The cross-references of these links refer to that REST resource also. Therefore, include the name of the tree in which the resource is a child. For example, `edit`, `domainRuntime`, `serverConfiguration`, and such.

Parent Links

All resources, except for root resources, include a top level link to their parent resource. The link's `rel` property is set to `parent`.

Collection children return links to the collection resource. For example, a server returns a link to the server's collection resource:

```
{
  links: [
    { rel: "parent", href: "http://localhost:7001/management/weblogic/latest/edit/
servers" }
   ]
}
```

Similarly, singleton children return links to their parent resource. For example, an SSL bean returns a link to the server bean:

```
{
  links: [
    { rel: "parent", href: "http://localhost:7001/management/weblogic/latest/edit/
servers/Server-0" }
  ]
}
```

Self Create Form Links

If a bean is a creatable, optional singleton (for example, a realm's `RDBMSSecurityStore`), and the bean currently does not exist, then a link to its corresponding create form resource is also returned. The link's `rel` property is set to `create`. For example, calling `GET` on a security realm's adjudicator also returns:

```
{
  links: [
     {
       rel: "create",
       href: "http://localhost:7001/management/weblogic/latest/edit/
securityConfiguration/realms/myrealm/adjudicatorCreateForm"
     }
  ]
}
```

Child Bean Links

Since a WLS bean's containment properties (for example, children) are mapped to separate REST resources, they are returned as top level links in the JSON response body.

Each link's `rel` property is mapped to the bean property's name. For example, calling `GET` on `Server-0` returns:

```
{
  links: [
    // mandatory singleton child:
    {
      rel: "SSL",
      href: "http://localhost:7001/management/weblogic/latest/servers/Server-0/SSL"
    },
    // writable collection of children:
    {
      rel: "networkAccessPoints",
      href: "http://localhost:7001/management/weblogic/latest/edit/servers/Server-0/
networkAccessPoints"
    }
  ]
}
```

Child Create Form Links

Links to create form resources are returned for creatable containment properties (singletons and collections). The link's `rel` property is set to `<singularPropertyName>CreateForm`. For example, calling `GET` on `Server-0` also returns:

```
{
  links: [
     {
       rel: "networkAccessPointCreateForm",
       href: "http://localhost:7001/management/weblogic/latest/edit/servers/Server-0/
networkAccessPointCreateForm"
     }
  ]
}
```

Singleton Bean Reference Links

WLS beans return top level links for each non-null singleton reference. The link's `rel` property is set to the name of the reference property. For example, if `Server-0` refers to `Machine-0`:

```
{
  machine: [ "machines", "Machine-0" ],
  links: [
    { rel: "machine", href: "http://localhost:7001/management/weblogic/latest/edit/
machines/Machine-0" }
  ]
}
```

If `Server-0` has no machine reference:

```
{
  machine: null
}
```

Bean Reference Collection Links

WLS beans return nested links for each reference in a reference collection. The link's `rel` property is set to `self`.

For example, if `Application-0` refers to the targets `Server-0` and `Cluster-0`:

```
{
  targets: [
    {
      identity: ["clusters", "Cluster-0" ],
      links: [ { rel: "self", href: "http://localhost:7001/management/weblogic/latest/
edit/clusters/Cluster-0" } ]
    },
    {
      identity: ["servers", "Server-0" ],
      links: [ { rel: "self", href: "http://localhost:7001/management/weblogic/latest/
edit/servers/Server-0" } ]
    }
}
```

Operation Links

Resources also return top level links to their operation resources. The links' `rel` properties are set to `action` and the links' titles are set to the name of the operation. For example, a `ServerRuntimeMBean` returns:

```
{
  links: [
    {
      rel: "action",
      title: "suspend",
      href: "http://localhost:7001/management/weblogic/latest/domainRuntime/
serverRuntimes/Server-0/suspend"
    },
    {
      rel: "action",
      title: "resume",
      href: "http://localhost:7001/management/weblogic/latest/domainRuntime/
serverRuntimes/Server-0/resume"
    },
    {
      rel: "action",
      title: "shutdown",
      href: "http://localhost:7001/management/weblogic/latest/domainRuntime/
serverRuntimes/Server-0/shutdown"
    }
  ]
}
```

Viewing Collections of Contains Beans

To view a collection of WLS beans, invoke the HTTP `GET` method on its corresponding REST URL.

For example, to get the configuration of all the servers:

```
GET http://localhost:7001/management/weblogic/latest/edit/servers
```

`GET` returns a standard WLS REST response body. `items` contains the children's properties. Each item has embedded `self` and `canonical` links to that child's resource.

Only the immediate children are returned. For example, if you get the servers collection, each server's properties will be returned, but the server's children (such as SSL) are not returned.

- About Collection items
- About Collection Links

About Collection items

The resource returns a JSON object for each child in the collection. These objects contain the same data as the items returned from calling `GET` on the children's resources. For example, getting the domain bean's `servers` collection returns:

```
{
  items: [
    { name: "Server-1", listenPort: 7001, ... },
    { name: "Server-2", listenPort: 7003, ... }
  ]
}
```

About Collection Links

A collection resource returns the following links:

- `self` and `canonical` links to itself.
- A link to its parent.
- A link to its corresponding create form resource if the collection is writable.
- Nested `self` and `canonical` links to each of its children.

For example, getting the domain bean's `servers` collection returns:

```
{
  items: [
    {
      name: "Server-1",
      listenPort: 7001,
      links: [
        { rel: "self", href: "http://localhost:7001/management/weblogic/latest/edit/
servers/Server-1" }
        { rel: "canonical", href: "http://localhost:7001/management/weblogic/latest/edit/
servers/Server-1" }
      ]
    },
    {
      name: "Server-2",
      listenPort: 7005,
      links: [
        { rel: "self", href: "http://localhost:7001/management/weblogic/latest/edit/
servers/Server-1" }
        { rel: "canonical", href: "http://localhost:7001/management/weblogic/latest/edit/
servers/Server-1" }
      ]
    }
  ]
  links: [
    { rel: "self", href: "http://localhost:7001/management/weblogic/latest/edit/
servers" }
    { rel: "canonical", href: "http://localhost:7001/management/weblogic/latest/edit/
servers" }
    { rel: "parent", href: "http://localhost:7001/management/weblogic/latest/edit" }
    { rel: "create-form", href: "http://localhost:7001/management/weblogic/latest/edit/
serverCreateForm" }
  ]
}
```

Retrieving Create Forms

To retrieve a create form for creating a new resource, invoke the HTTP `GET` method on its corresponding create form REST URL.

For example, to retrieve a create form for creating a new server:

```
GET http://localhost:7001/management/weblogic/latest/edit/serverCreateForm
```

`GET` returns a standard WLS REST response body. It returns a JSON object containing the create form's properties and a `links` property which is a JSON array containing links to related resources.

- About Create Form Properties
- About Create Form Links

About Create Form Properties

The returned JSON object contains a property for each writable property (normal properties and references) that may be specified when creating a new resource of that type. The property's value will either be the default value from the type's bean info (if available), or the default value for the property's type (for example, `0` for an `int`). The values for reference properties are always null. For example, getting the domain's `serverCreateForm` returns:

```
{
  name: null, // identity - unique names are not generated
  idleConnectionTimeout: 65, // from the default value in the bean info
  replicationGroup: null, // default value for a String since the bean info does not
provide a default value
  machine: null, // singleton reference
  candidateMachines: null, // reference collection
  ...
}
```

About Create Form Links

A create form returns the following links:

- `self` and `canonical` links to itself.
- A link to its parent.
- A `create` link to the corresponding resource that can be used to create a resource of this type.

For example, getting the domain bean's `serverCreateForm` returns:

```
{
  links: [
    { rel: "parent", href: "http://localhost:7001/management/weblogic/latest/edit" },
    { rel: "self", href: "http://localhost:7001/management/weblogic/latest/edit/
serverCreateForm" },
    { rel: "canonical", href: "http://localhost:7001/management/weblogic/latest/edit/
serverCreateForm" },
    { rel: "create", href: "http://localhost:7001/management/weblogic/latest/edit/
servers" }
  ]
}
```

Filtering Results

Bean, collection, and create form resource `GET` methods support these query parameters to let you omit properties and links from the response.

Parameter Name	Description
`fields`	Return only these properties.
`excludeFields`	Return all properties except for these properties.
`links`	Return only links with these `rel` names.
`excludeLinks`	Return all links except for the ones with these `rel` names.

`fields` and `excludeFields` are mutually exclusive, as are `links` and `excludeLinks`. All the values are comma-separated lists of names.

For example, to only retrieve a server's `self` and `parent` links, and `name` and `listenPort` properties:

```
curl ... -X GET http://localhost:7001/management/weblogic/latest/edit/servers/myserver\
  ?fields=name,listenPort\&links=self,parent
{
  links: [
    { rel: "parent", href: "http://localhost:7001/management/weblogic/latest/edit/
servers" },
    { rel: "self", href: "http://localhost:7001/management/weblogic/latest/edit/servers/
myserver" }
  ],
  name: "myserver",
  listenPort: 7001
}
```

Modifying the WLS Configuration

You can create, modify, and delete beans in the edit tree *only* (`.../management/weblogic/<version>/edit/...`). The other bean trees are read-only.

All WLS bean edits must be performed within a configuration transaction:

- If you have already started a transaction, the REST changes will be made in the same transaction. You will still be responsible for committing or rolling back the transaction.
- If you have not started a transaction, the REST resource will begin a transaction on your behalf, try to make the changes, and either commit or roll back the transaction depending on whether the changes could be made (auto-transactions).
- If someone else has already started a transaction, the REST resource will return an error (instead of modifying the configuration).

Sometimes a configuration transaction cannot be committed unless complementary changes to multiple beans are made in the same transaction. In these cases, you need to begin and end the transaction explicitly versus relying on auto-transactions.

Also, when the client manages the transaction, each REST call saves the changes (but does not activate them). There is some MBean validation that occurs during the save operation which might cause it to fail. For example, when you create a JDBC system resource, the changes cannot be saved until after its child JDBC resource name is set. For cases like this, use the `saveChanges=false` query parameter.

See the changeManager resources in *RESTful Edit Reference for Oracle WebLogic Server*.

- Modifying WLS Configuration Beans
- About the JSON Object Request Body

Modifying WLS Configuration Beans

To modify a WLS bean, construct a JSON object containing the values you want to change, and then invoke the HTTP `POST` method on its corresponding REST URL, passing in that JSON object as the request body.

For example, to change a server's listen port and administration port:

```
curl ... -d "{
  listenPort: 7007,
  administrationPort: 9007
}" -X POST http://localhost:7001/management/weblogic/latest/edit/servers/Server-0
```

This command is similar to an HTTP `PATCH` operation where you modify only a part of the bean, versus needing to pass in all of the bean's properties every time.

About the JSON Object Request Body

You construct a JSON object containing the values you want to change. Some WLS bean properties are read-only (for example, a server's name). Read-only properties are ignored.

You don't have to pass in all of the bean's properties. Any properties not passed in will retain their current values. As was described in Encrypted Properties, `GET` returns the value

`@Oracle_Confidential_Property_Set_V1.1#` for an encrypted string property that has a non-null value. If you `POST` back this value, then the property will retain its current value. If you want to change the encrypted property's value, then set the value to the cleartext string that you want it to be, for example:

```
{ defaultIIOPPassword: "admin123" }
```

To change a reference, pass in its identity. The same is true for reference collections. This replaces the reference collection versus adding references to the collection. For example, to set a server's machine to `Machine-0` and candidate Machines to `Machine-0` and `Machine-1`:

```
{
  machine: [ 'machines', 'Machine-0' ] },
  candidateMachines: [
     { identity: [ 'machines', 'Machine-0' ] },
     { identity: [ 'machines', 'Machine-1' ] }
  ]
}
```

Also, use null to remove references. For example, to remove a server's machine and candidate machines' references:

```
{
  machine: null,
  candidateMachines: null
}
```

If you pass in a mixture of valid and invalid values, the valid ones are written and errors are returned for the invalid ones, and overall, the REST method returns an `OK` status code. For example:

```
curl ... -d "{
  listenPort: 7007,
  administrationPort: 'foo'
}" -X POST http://localhost:7001/management/weblogic/latest/edit/servers/Server-0
HTTP/1.1 200 OK
{
  messages: [
    {
      severity: "FAILURE",
      field: "administrationPort",
      message: "Something about the value needs to be an integer"
    }
  ]
}
```

In this example, the listen port is modified and the administration port is not. The method returned an `OK` status code.

Using Multiple Edit Sessions

In a previous release, WebLogic Server introduced multiple edit sessions, named concurrent edit sessions, which allows more than one administrator to make configuration changes at the same time. These edit sessions are domain scoped. Each scope has a default edit session. Edit session names are unique within a scope, but not across scopes.

For all the REST resources in the `edit` tree, you must specify which edit session to use—the name of the scope and the name of the edit session within that scope.

The edit session scope name is derived from the URL. When you use a domain-scoped REST URL, then REST uses the domain level scope. Within that scope, REST must know which edit session to use. You can either specify a header which states exactly which edit session to use, or you can let REST use defaulting rules to pick one.

- Client Specified Edit Session
- The Default Edit Session

Client Specified Edit Session

You can select the edit session by including a `weblogic.edit.session` header in the request. The header's value is used as the edit session name. For example:

```
curl ... -H weblogic.edit.session=MySession ...
```

Each edit session scope has a default edit session named `default`. To explicitly select the scope's default edit session:

```
curl ... -H weblogic.edit.session=default ...
```

The Default Edit Session

If you did not include the `weblogic.edit.session` header, the REST resources use the following rules to select an edit session:

- If you currently have one edit session locked in the scope, REST will use it.
- Or, if you have created one edit session in the scope, REST will use it.
- Otherwise, REST will use the scope's default edit session.

Creating WLS Configuration Beans

You create a new WLS configuration bean by calling `POST` with a JSON structure containing the new bean's properties.

To make this easier, you can use the corresponding create form resource to retrieve a template JSON structure that is populated with default values for the various writeable properties.

- URLs for Creating WLS Configuration Beans
- Getting a JSON Template
- Creating the Bean
- Deleting WLS Configuration Beans

URLs for Creating WLS Configuration Beans

To create a collection child, call `POST` on the collection's URL. For example, `http://localhost:7001/management/weblogic/latest/edit/servers`.

To create an optional singleton child, call `POST` on the proposed child's URL. For example, `http://localhost:7001/management/weblogic/latest/edit/securityConfiguration/realms/myRealm/adjudicator`.

To retrieve a create form, call `GET` on the corresponding create form resource. For example:

```
http://localhost:7001/management/weblogic/latest/edit/serverCreateForm
```

And

```
http://localhost:7001/management/weblogic/latest/edit/securityConfiguration/realms/
myRealm/adjudicatorCreateForm
```

Getting a JSON Template

The underlying WLS beans have default values for many properties. You typically want to display these default values and perhaps, customize them, then use them to create a new WLS bean. You can get these default values by calling `GET` on the corresponding create form resource. For example:

```
curl ... -X GET http://localhost:7001/management/weblogic/latest/edit/serverCreateForm
HTTP/1.1 200 OK
{
    listenPort: 7001,
    ...
  }
}
```

Creating the Bean

To create the WLS configuration bean, call `POST` on a JSON object containing the new bean's properties.

The JSON object does not need to include all the possible properties. Unspecified properties are set to their default values. All collection children need to be assigned a unique identity within their collection, for example, a server needs a unique name. Therefore, the `identity` property is not optional.

The response contains a `location` header containing the resource's URL. For example:

```
curl ... -d "{
  name: "Server-1",
  defaultProtocol: "t3s"
}" -X POST http://localhost:7001/management/weblogic/latest/edit/servers
HTTP/1.1 201 Created
Location: http://localhost:7001/management/weblogic/latest/edit/servers/Server-1
curl -X GET http://localhost:7001/management/weblogic/latest/edit/servers/id/Server-1
HTTP/1.1 200 OK
{
  item: {
    identity: [ "domain", "servers", "Server-1" ],
    name: "Server-1",
    defaultProtocol: "t3s", // specified by the caller
    listenAddress: 7001     // not specified by the caller, therefore set to its default
value
  }
}
```

If a bean with that name already exists, the resource returns a `BAD_REQUEST` status code along with a failure message. For example:

```
curl ... -d "{
  name: "Server-1"
}" -X POST http://localhost:7001/management/weblogic/latest/edit/servers
HTTP/1.1 400 Bad Request
{
  type: "http://oracle/TBD/WlsRestMessageSchema",
  title: "FAILURE",
  detail: "Bean already exists:
\"weblogic.management.configuration.ServerMBeanImpl@31fa1656([mydomain]/
Servers[Server-1])\"",
  status: 400
}
```

Similar to updating a WLS configuration bean, you can pass in a mixture of valid and invalid values. Read-only properties and properties that the bean does not support are ignored. If there is an exception setting a property, the resource adds a failure message to the response. After processing all of the properties, if there were any errors, the resource attempts to delete the new bean and returns a `BAD_REQUEST` status code.

Example 3-1 Mixture of valid and invalid properties

```
curl ... -d "{
  name: "Server-1",
  listenPort: abc,
  defaultProtocol: "no-such-protocol",
  adminstrationProtocol: "iiop"
}" -X POST http://localhost:7001/management/weblogic/latest/edit/servers
HTTP/1.1 400 Bad Request
{
  type: "http://oracle/TBD/WlsRestMessagesSchema",
  title: "ERRORS",
  status: 400,
  wls:errorsDetails: [
    {
      type: "http://oracle/TBD/WlsRestMessageSchema",
      title: "FAILURE",
      detail: "no-such-protocol is not a legal value for DefaultProtocol.\
      The value must be one of the following: [t3, t3s, http, https, iiop, iiops]",
      o:errorPat: "defaultProtocol"
    },
    {
      type: "http://oracle/TBD/WlsRestMessageSchema",
      title: "FAILURE",
      detail: "Type mismatch. Cannot convert abc to int",
      o:errorPath: "listenPort"
    }
  ]
}
```

Example 3-2 All valid properties

```
curl ... -d "{
  name: "Server-1",
  listenPort: 7003,
  defaultProtocol: "https",
  adminstrationProtocol: "iiop"
}" -X POST http://localhost:7001/management/weblogic/latest/edit/servers
HTTP/1.1 201 Created
Location: http://localhost:7001/management/weblogic/latest/edit/servers/Server-1
```

Deleting WLS Configuration Beans

To delete a WLS bean (both collection children and optional singleton children), invoke the HTTP `DELETE` operation on its corresponding REST URL. Any references to that bean will also be removed. For example, to delete a server:

```
curl ... -X DELETE http://localhost:7001/management/weblogic/latest/edit/servers/Server-0
```

Managing Whether a Property is Set

An MBean property can either be set or unset. If it is set, its value is persisted (for example, to `config.xml`) and locked in. If it is unset, then a default value is used. The value can either be the default value for the property's type, a hard coded default value, or a computed default value that runs some custom Java code.
By default, when you call `GET` on a resource, it returns the property's current value. When you set the value of a `String` property to null or an empty string, it unsets the property (returns it to its default value).

REST lets you determine whether a property has been set, and explicitly set or unset a property.

If you set the `expandedValues` query parameter to `true` when getting a resource, each value is returned as a JSON object with a `set` Boolean property and a `value` property that holds the current value. For example, getting a server returns:

```
curl ... -X GET \
  http://localhost:7001/management/weblogic/latest/edit/servers/myserver?
&expandedValues=true
{
  listenPortEnabled: { set: false, value: true }, // currently not set
  name: { set: true, value: "myserver" }, // currently set
  listenPort: { set: true, value: 7003 } // currently set
}
```

Similarly, you can use the `expandedValues` query parameter to explicitly set or unset values. For example, to unset the listen port and set the listen address to an empty string:

```
curl ... -d "{
  listenPort: { set: false }, // value will be ignored if specified
  listenAddress: { set: true, value: "" }
}" -X POST http://localhost:7001/management/weblogic/latest/edit/servers/myserver?
expandedValues=true
```

Invoking Operations

Each WLS bean operation maps to its own REST URL. In the case of overloaded operations (for example, `shutdown()` versus `shutdown(int, boolean)`), all the overloaded operations map to the same URL and the resource looks at the incoming arguments to determine which operation to invoke.

If the operation requires input arguments, they are specified by passing in a JSON object request body with a property for each argument. The name of the property matches the name of the argument.

If the operation does not take input arguments, you must pass in a JSON object with no properties.

Similarly, if the operation returns a value, then it is returned in a standard REST response body's JSON object `return` property. If the operation is void, the response body does not include an `return` property.

If the underlying MBean operation throws an exception, the REST method returns a `BAD REQUEST` (`404`) response containing the exception's text.

Example 3-3 void operation with no arguments : void shutdown()

```
curl ... -d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/serverRuntimes/
Server-0/shutdown
{
  // response does not include a 'return' property since it's a void operation
}
```

Example 3-4 void operation with multiple arguments : void shutdown(int timeout, boolean ignoreSessions)

```
curl ... -d "{ timeout: 500, ignoreSessions: false }" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/serverRuntimes/
Server-0/shutdown
{
  // response does not include a 'return' property since it's a void operation
}
```

Example 3-5 non-void operation with an argument: String getURL(String protocol)

```
curl ... -d "{ protocol: 'http' }" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/serverRuntimes/
Server-0/getURL
{
  return: "http://localhost:7003"
}
```

Using Queries

The REST API includes a powerful bulk access capability that lets you dynamically describe a tree of beans that can be returned in one call.

Each tree (for example, edit, domain runtime, and such), has a root `search` resource. You can `POST` a query to these `search` resources. The query indicates which beans (and properties and links) should be returned, and, as such, returns a portion ("slice") of the bean tree.

Bulk access can only be used for reading; it cannot be used for writing.

- Search Resources
- Object Queries
- Response Body
- Query Examples
- Limiting the Wait Time for a Delegated Request
- Consolidating Search Results

Search Resources

Each bean tree includes a `search` resource for bulk queries.

On the Administration Server:

URL	Description
`.../management/weblogic/latest/edit/search`	Returns a slice of the edit bean tree (in progress edits that have not been saved to disk yet).

URL	Description
`.../management/weblogic/latest/domainConfig/search`	Returns a slice of the last configuration bean tree that was saved to disk (versus the configuration the servers are currently using).
`.../management/weblogic/latest/domainRuntime/search`	Returns a slice of the Administration Server's domain runtime bean tree (which covers all the servers' runtime bean trees).
`.../management/weblogic/latest/serverConfig/search`	Returns a slice of the Administration Server's configuration bean tree (the configuration the Administration Server is running against).
`.../management/weblogic/latest/serverRuntime/search`	Returns a slice of the Administration Server's runtime bean tree.

On Managed Servers:

URL	Description
`.../management/weblogic/latest/serverConfig/search`	Returns a slice of the Managed Server's configuration bean tree (the configuration the server is running against).
`.../management/weblogic/latest/serverRuntime/search`	Returns a slice of the Managed Server's runtime bean tree.

When you `POST` a query to a search resource, the query starts searching at the root bean of the tree. The resource returns a JSON response containing the results of the query, that "slice" of the bean tree.

Object Queries

An object query describes what data should be returned for a WLS bean (or collection of beans), such as:

- Which of the bean's properties should be returned.
- Which of the bean's links should be returned.
- Which of the bean's children should be returned.
- For a collection, which of its children should be returned.

Note that all searches start at the root bean of the `search` resource's tree. For example, if you `POST` a query to `management/weblogic/latest/domainRuntime/search`, it starts searching at the `DomainRuntimeMBean` in the domain runtime tree.

- Fields and ExcludeFields
- Links and ExcludeLinks
- Children
- Identities

Fields and ExcludeFields

`fields` specifies which bean properties (for example, scalars and references) are returned. It is a JSON string array of property names. For example, to return the domain's `name` and `configurationVersion`:

```
curl ... -d "{ fields: [ 'name', 'configurationVersion' ] }" \
-X POST http://localhost:7001/management/weblogic/latest/edit/search
```

If the query lists properties that the bean does not support, then that part of the query is ignored (instead of returning an error). If `fields` is not specified, then all of the properties are returned.

`excludeFields` specifies a list of fields that should not be returned; all other properties are returned. `fields` and `excludeFields` are mutually exclusive.

Note that a query's `fields` and `excludeFields` properties mirror the `fields` and `excludeFields` query parameters that you can specify when calling `GET` on a resource. The difference is that the query parameters use comma-separated names and queries use JSON arrays of names.

Links and ExcludeLinks

`links` specifies which of the bean's links should be returned. It is a JSON string array of link `rel` names. For example, to return the domain's `self` and `servers` links:

```
curl ... -d "{ links: [ 'self', 'servers' ] }" \
-X POST http://localhost:7001/management/weblogic/latest/edit/search
```

If the query lists links that the bean does not support, then that part of the query is ignored (instead of returning an error).

If `links` is not specified, then all the links are returned (except for collection children, which only return their `self` and `canonical` links by default).

Similarly, `excludeLinks` specifies a list of links that should not be returned; all other links are returned. `links` and `excludeLinks` are mutually exclusive.

To return all of a collection's children's links, use `excludeLinks: []`.

Note that a query's `links` and `excludeLinks` properties mirror the `links` and `excludeLinks` query parameters that you can specify when calling `GET` on a resource.

Children

`children` specifies which child bean properties are returned. It is a JSON object whose property names are the names of the children to return, and whose values are object queries. For example, to get the domain's name, along with the name and listen port of each server:

```
curl ... -d "{
  fields: [ 'name' ], // only return the domain's name
  children: {
    servers: { // fetch the domain's 'servers' collection
      fields: [ 'name', 'listenPort' ] // only return each server's name and listen port
    }
  }
}" -X POST http://localhost:7001/management/weblogic/latest/edit/search
```

If `children` is not specified, then none of the bean's children are returned.

Identities

Sometimes you want to return only certain items in a collection (for example, `myserver` and `Server-0`). Each collection child has a property that specifies its identity. Typically, this is the `name` property. The query uses this property name to specify which children of a collection are returned. It is a JSON string array of identities. `fields` and `links` can also be used to control which properties and links are returned for each of these children. For example, to return the name and listen port for the servers, `Server-0` and `Server-1`:

```
curl ... -d "{
  fields: [ 'name' ], // only return the domain's name
  children: {
    servers: { // fetch the domain's 'servers' collection
      names: [ 'Server-0', 'Server-1' ], // only return the children whose 'name' is
'Server-0' or 'Server-1'
      fields: [ 'name', 'listenPort' ] // only return each server's name and listen port
    }
  }
}" -X POST http://localhost:7001/management/weblogic/latest/edit/search
```

Identities that do not exist are ignored (instead of returning an error). Similarly, if the context is not a collection, then this part of the query is ignored. By default, all collection children are returned.

Response Body

The response body follows the usual pattern (inline properties or `items`, depending on whether the URL is for a bean or a collection). The child beans are returned as nested properties. For example:

```
curl ... -d "{
  fields: [], // don't return any domain level properties
  links: [], // don't return any domain level links
  children: {
    servers: { // fetch the domain's 'servers' collection
      names: [ 'Server-0', 'Server-1' ], // only return the children whose 'name' is
'Server-0' or 'Server-1'
      fields: [ 'name' ], // only return each server's name
      links: [], // don't return any per-server links
      children: {
        SSL: {
          fields: [ 'listenPort' ], // only return each server's SSL listen port
          links: [] // don't return any SSL level links
        }
      }
    }
  }
}" -X POST http://localhost:7001/management/weblogic/latest/edit/search
{code:JavaScript}
HTTP/1.1 200 OK
{
  servers: {
    items: [
      {
        name: "myserver",
        SSL: { listenPort: 7002}
      },
      {
        name: "AnotherServer",
        SSL: { listenPort: 7002}
      }
    ]
  }
}
```

Query Examples

This example gets the component runtimes of specific applications on all running servers. It returns only the name for the server runtimes and application runtime parents and returns all of the component runtimes' properties.

```
curl ... -d "{
  fields: [], links: [], // don't return any domain runtime level properties or links
  children: {
    serverRuntimes: {
      fields: [ 'name' ], links: [], // return each server's name.  don't return any 
server level links
      children: {
        applicationRuntimes: {
          name: [ 'myapp', 'BasicApp' ], // only return apps 'myapp' and 'BasicApp'
          fields: [ 'name' ], links: [], // return each app's name but no per-app links
          children: {
            componentRuntimes: { links: [] } // return all component runtime properties, 
but no links
          }
        }
      }
    }
  }
}" -X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search
```

This example gets all of the servlet runtime and EJB runtime information for a set of applications across all running servers.

```
curl ... -d "{
  links: [],
  fields: [],
  children: {
    serverRuntimes: {
      links: [],
      fields: [ 'name', 'state' ],
      children: {
        applicationRuntimes: {
          name: [ 'myapp', 'BasicApp' ],
          links: [],
          fields: [ 'name', 'healthState' ],
          children: {
            componentRuntimes: {
              links: [],
              fields:[
                'name',
                'healthState',
                'contextRoot',
                'openSessionsCurrentCount',
                'sessionsOpenedTotalCount'
              ],
              children: {
                EJBRuntimes: {
```

```
    links: [],
    fields: [
      'EJBName',
      'type'
    ],
    children: {
      transactionRuntime: {
        links: [],
        fields: [
          'transactionsCommittedTotalCount',
          'transactionsRolledBackTotalCount',
          'transactionsTimedOutTotalCount'
        ]
      },
      poolRuntime: {
        links: [],
        fields: [
          'accessTotalCount',
          'missTotalCount',
          'destroyedTotalCount',
          'pooledBeansCurrentCount',
          'beansInUseCurrentCount',
          'waiterCurrentCount',
          'timeoutTotalCount'
        ]
      },
      cacheRuntime: {
        links: [],
        fields: [
          'cachedBeansCurrentCount',
          'cacheAccessCount',
          'cacheMissCount',
          'activationCount',
          'passivationCount'
        ]
      },
      lockingRuntime: {
        links: [],
        fields: [
          'lockEntriesCurrentCount',
          'lockManagerAccessCount',
          'waiterTotalCount',
          'waiterCurrentCount',
          'timeoutTotalCount'
        ]
      },
      timerRuntime: {
        links: [],
        fields: [
          'timeoutCount',
          'cancelledTimerCount',
          'activeTimerCount',
          'disabledTimerCount'
        ]
      }
    }
  },
  servlets: {
    links: [],
    fields: [
      'servletName',
      'contextPath',
                    'reloadTotalCount',
                    'invocationTotalCount',
                    'executionTimeTotal',
                    'executionTimeHigh',
                    'executionTimeLow'
                  ]
                }
              }
            }
          }
        }
      }
    }
  }
}" -X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search
```

Limiting the Wait Time for a Delegated Request

When you make a GET request or a search POST call in the domainRuntime tree that gets delegated to one or more servers, you can specify the `requestMaxWaitMillis` query parameter which sets the maximum amount of time you are willing to wait for a response from each server. If not specified, it uses the RestfulManagementServicesMBean's `DelegatedRequestMaxWaitMillis` value that the administrator configured for the domain.

For servers that do not respond fast enough, the response body includes bad per-server HTTP status codes.

Examine the following examples:

```
# GET all the server runtimes' names, give up on a server if it doesn't respond within
10 milliseconds:
curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-X GET http://127.0.0.1:7001/management/weblogic/latest/domainRuntime/serverRuntimes?
links=none\&fields=name\&requestMaxWaitMillis=10

< HTTP/1.1 200 OK
{"items": [
    {
        "httpStatus": 504,
        "name": "Cluster1Server1",
        "identity": []
    },
    {"name": "AdminServer"},
    {
        "httpStatus": 504,
        "name": "Cluster1Server2",
        "identity": []
    }
]}

# GET all the server runtimes' application's names, give up on a server if it doesn't
respond within 50 milliseconds:
curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-X POST http://127.0.0.1:7001/management/weblogic/latest/domainRuntime/search?
requestMaxWaitMillis=50 \ -d "{
   links: [], fields: ['name'],
   children: {
     serverRuntimes: {
       links: [], fields: ['name'],
       children: {
         applicationRuntimes: { links: [], fields: ['name'] }
       }
     }
   }
}"

< HTTP/1.1 200 OK
{
    "name": "bean-ex",
    "serverRuntimes": {"items": [
        {
            "httpStatus": 504,
            "name": "Cluster1Server1",
            "identity": []
        },
        {
            "name": "AdminServer",
            "applicationRuntimes": {"items": [
                {"name": "bea_wls_management_internal2"},
                {"name": "jms-internal-xa-adp"},
                {"name": "mejb"},
                {"name": "jms-internal-notran-adp"},
                {"name": "bea_wls_internal"},
                {"name": "wls-management-services"},
                {"name": "bea_wls_deployment_internal"}
            ]}
        },
        {
            "httpStatus": 504,
            "name": "Cluster1Server2",
            "identity": []
        }
    ]}
}
```

Consolidating Search Results

You can consolidate the domainRuntime search results of REST resources that span multiple servers in a domain based on criteria you specify, for example, you can retrieve the maximum `openSocketsCurrentCount` of all the running servers.

Like any search, you use the request body to specify a slice of the bean tree to return. To enable this feature, you specify additional fields to indicate that the results should be consolidated across the servers and how each property should be consolidated.

Review the syntax for the following consolidated search examples:

```
----------------------------------------------------------------------
Get the total number of open sessions across each application's component runtimes
across all servers
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      children: {
        applicationRuntimes: {
          mergeOn: 'name',
          fields: [ { name: 'name', sameValue: true } ],
          children: {
            componentRuntimes: {
              mergeCollection: true,
              fields: [ { name: 'openSessionsCurrentCount', total: true } ]
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search

HTTP/1.1 200 OK

Response Body:
{"serverRuntimes": {"items": [{"applicationRuntimes": {"items": [
    {
        "name": "JDBCDataSource1",
        "componentRuntimes": {"items": [{}]}
    },
    {
        "name": "bea_wls_deployment_internal",
        "componentRuntimes": {"items": [{"openSessionsCurrentCount": {
            "total": 0,
            "count": 3
        }}]}
    },
    {
        "name": "wls-management-services",
        "componentRuntimes": {"items": [{"openSessionsCurrentCount": {
            "total": 202,
            "count": 3
        }}]}
    },
    {
        "name": "bea_wls_cluster_internal",
        "componentRuntimes": {"items": [{"openSessionsCurrentCount": {
            "total": 0,
            "count": 2
        }}]}
    },
    {
        "name": "jms-internal-xa-adp",
        "componentRuntimes": {"items": [{}]}
    },
    {
        "name": "fairShare",
        "componentRuntimes": {"items": [{"openSessionsCurrentCount": {
            "total": 0,
            "count": 2
        }}]}
    },
    {
        "name": "bea_wls_internal",
        "componentRuntimes": {"items": [{"openSessionsCurrentCount": {
            "total": 0,
            "count": 3
        }}]}
    },
    {
        "name": "JMSSystemResource1",
        "componentRuntimes": {"items": [{}]}
    },
    {
        "name": "basicapp",
        "componentRuntimes": {"items": [{"openSessionsCurrentCount": {
            "total": 0,
            "count": 2
        }}]}
    },
    {
        "name": "jms-internal-notran-adp",
        "componentRuntimes": {"items": [{}]}
    },
    {
        "name": "bea_wls_management_internal2",
        "componentRuntimes": {"items": [{"openSessionsCurrentCount": {
            "total": 0,
            "count": 1
        }}]}
    },
    {
        "name": "mejb",
        "componentRuntimes": {"items": [{}]}
    }
]}}]}}

----------------------------------------------------------------------
Get the total number of invocations of the servlets of each component runtime of the
fairShare and wls-management-services applications across all servers
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      children: {
        applicationRuntimes: {
          name: [ 'fairShare', 'wls-management-services' ],
          mergeOn: 'name',
          fields: [ { name: 'name', sameValue: true } ],
          children: {
            componentRuntimes: {
              mergeOn: 'moduleId',
              fields: [ { name: 'contextRoot', sameValue: true } ],
              children: {
                servlets: {
                  mergeCollection: true,
                  fields: [ { name: 'invocationTotalCount', total: true } ]
                }
              }
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search

HTTP/1.1 200 OK

Response Body:
{"serverRuntimes": {"items": [{"applicationRuntimes": {"items": [
    {
        "name": "fairShare",
        "componentRuntimes": {"items": [{
            "contextRoot": "\/fairShare",
            "servlets": {"items": [{"invocationTotalCount": {
                "total": 0,
                "count": 8
            }}]}
        }]}
    },
    {
        "name": "wls-management-services",
        "componentRuntimes": {"items": [{
            "contextRoot": "\/management",
            "servlets": {"items": [{"invocationTotalCount": {
                "total": 202,
                "count": 9
            }}]}
        }]}
    }
]}}]}}
```

The first example, "Get the total number of open sessions across each application's component runtimes across all servers," keeps each application separate (based on the application's `name`), merging each application's components together across all the servers.

This is achieved by specifying:

```
serverRuntimes - mergeCollection: true
  applicationRuntimes - mergeOn: 'name'
    componentRuntimes - mergeCollection: true
```

For example, if the bean tree has:

```
server1
  app1 : comp1, comp2
  app2 : comp3, comp4
  app3 : comp5, comp6

server2
  app1 : comp1, comp2
  app2 : comp3, comp4
  app3 : comp5, comp6
```

The response will return:

```
app1
    comp
      merged values from server1.app1.comp1, server1.app1.comp2, server2.app1.comp1,
server2.app1.comp2

app2
    comp
      merged values from server1.app2.comp3, server1.app2.comp4, server2.app2.comp3,
server2.app2.comp4

app3
    comp
      merged values from server1.app3.comp5, server1.app3.comp6, server2.app3.comp5,
server2.app3.comp6
```

The second example, "Get the total number of invocations of the servlets of each component runtime of the fairShare and wls-management-services applications across all servers," keeps each application separate (based on the application's `name`), and keeps each application's components separate (based on the component's `moduleId`), merging each application's component's servlets together across all the servers. It also specifies to return results only for `app1` and `app2` (versus all the applications).

This is achieved by specifying:

```
  serverRuntimes - mergeCollection: true
    applicationRuntimes - mergeOn: 'name', name: [ 'app1', 'app2' ]
      componentRuntimes - mergeOn: 'moduleId'
        servlets - mergeCollection: true
```

For example, if the bean tree has:

```
  server1
    app1
      comp1 : servlet1, servlet2
      comp2 : servlet3, servlet4
    app2
      comp3 : servlet5, servlet6
      comp4 : servlet7, servlet8
    app3
      comp5 : servlet9, servlet10
      comp6 : servlet11, servlet12

  server2
    app1
      comp1 : servlet1, servlet2
      comp2 : servlet3, servlet4
    app2
      comp3 : servlet5, servlet6
      comp4 : servlet7, servlet8
    app3
      comp5 : servlet9, servlet10
      comp6 : servlet11, servlet12
```

The response will return (notice that `app3` is not returned):

```
  app1
    comp1
      merged values from server1.app1.comp1.servlet1, server1.app1.comp1.servlet2,
server2.app1.comp1.servlet1, server2.app1.comp1.servlet2
    comp2
```

```
      merged values from server1.app1.comp2.servlet3, server1.app1.comp2.servlet4,
server2.app1.comp2.servlet3, server2.app1.comp2.servlet4
  app2
    comp3
      merged values from server1.app2.comp3.servlet5, server1.app2.comp3.servlet6,
server2.app2.comp3.servlet5, server2.app2.comp3.servlet6
    comp4
      merged values from server1.app2.comp4.servlet7, server1.app2.comp4.servlet8,
server2.app2.comp4.servlet7, server2.app2.copm4.servlet8
```

For more examples of consolidating search results in domains, see Domain Level REST API Examples.

Table 3-1 describes the request body merge-related search fields. If none are specified, then the search results will not be consolidated.

Table 3-1 Request Body Merge-Related Fields

Field	Description
`mergeCollection`	Indicates that a collection of MBeans (and their trees of sub-MBeans) should be merged together into a single consolidated MBean tree (and tree of sub-MBeans). Its value is a Boolean.
`mergeOn`	Indicates when child MBeans in a merged collection should be merged together (for example, that app1 in server1 should be merged with app1 in server2). Its value is a string, naming a property. `mergeOn` must be specified for all child collections of a merged collection (unless `mergeCollection: true` is specified for the child collection instead). Similarly, it must not be specified on collections that are not parented by a merged collection. Returns `BAD REQUEST` if either condition is violated.
`fields`	Specifies which properties of an MBean should be returned. `fields` contains a list of *per-property merge rules*. If not specified, no properties are returned. Each entry in the list of fields is a JSON object containing the name of the property and its merge rules. For example: `fields: [ { name: 'openSessionCurrentCount', total: true, min:true, max:true }, { name: 'type', sameValue: true } ]` Multiple merge rules can be specified for the same property (for example, return the min, max, and total). Returns `BAD REQUEST` if no merge rules are specified for a property. See Table 3-2 in this section.
`excludeFields`	Specifies which properties should not be returned when doing a non-consolidated search (for example, a list of what not to return instead of what to return). It is not supported when doing consolidated searches. Returns `BAD REQUEST` if present.
`links`	`links` and `excludeLinks` specify which links to related REST resources are returned when doing a non-consolidated search. They are not supported when doing a consolidated search. Returns `BAD REQUEST` if either is present.

Table 3-2 Per-Property Merge Rules

Field	Description
`name`	Contains the name of the property to merge. It is a mandatory string field. Returns `BAD REQUEST` if it is not specified.

`total`
`min`
`max`
	The `total` field indicates that the sum and number of the property values should be returned. Similarly, the `min` and `max` fields indicate that the minimum or maximum property value should be returned. These are optional Boolean fields which default to false.
	Returns `BAD REQUEST` if the REST API finds a value that is not a number (or a string that can be converted to a number).

`sameValue`	Some property values should be the same on each merged MBean, for example, a component's type and moduleId. The `sameValue` field indicates that the value should be the same for each merged MBean. It is an optional Boolean field which defaults to false. Returns `BAD REQUEST` if the REST API finds different values while merging. Similarly, returns `BAD REQUEST` if the property value is not a string.
`values`	Indicates that the list of property values should be returned. They are unordered. The field is an optional Boolean which defaults to false. It can be used for any type of MBean property.

About Synchronous and Asynchronous Operations

Several MBean operations (for example, server lifecycle, deployment) are asynchronous. They return job MBeans that must be monitored to determine when the job has completed.

Asynchronous MBean operations return a `200 OK`, `201 Created` or `400 Bad Request` if the operation completed or failed immediately. Otherwise, they return a `202 Accepted` and you must poll the returned job resource to find out when the work is done. By default, REST makes a best effort attempt to wait for the work to complete, but returns after about five minutes. You can specify the `Prefer` header to control how long REST waits for the work to complete.

Table 3-3 describes using the `Prefer` header.

Table 3-3 Using the Prefer Header

Header	Description
`-X Prefer:respond-async`	The client polls a returned job resource. REST returns a `200 OK`, `201 Created`, or `400 Bad Request` if the asynchronous MBean operation finishes immediately; otherwise it returns a `202 Accepted`.
`-X Prefer:wait=#` For example, `-X Prefer:wait=10`	The REST resource internally polls the job for up to the specified number of seconds and returns a `200 OK`, `201 Created`, or `400 Bad Request` if the asynchronous MBean operation finishes within that time; otherwise it returns a `202 Accepted`, along with a `Location` header containing the URL of a REST task resource that the client can poll (through `GET`) to find out when the work is done.

If you do not specify the `Prefer` header, REST will return a `200 OK`, 201 `Created`, or `400 Bad Request` if the asynchronous MBean operation finishes within approximately five minutes, otherwise it returns a `202 Accepted`.

If you specify both `respond-async` and `wait`, `respond-async` is ignored.

For examples of synchronous and asynchronous operations, see Domain Level REST API Examples.

Deploying Applications and Libraries

You view deployed applications and libraries in the edit tree. You call `POST` on the collections to deploy them, and `DELETE` to undeploy them.

Similarly, the deployment MBeans take server relative pathnames. In addition, you can upload files from the client to the server then deploy them and use create form resources to inspect deployments (for example, to determine their preferred name and version numbers). For examples of deploying domain-scoped applications, see Domain Level REST API Examples.

Cross-Origin Resource Sharing (CORS) for WebLogic Server REST APIs

As of 14.1.1.0.0, Oracle WebLogic Server RESTful management services support client-side Cross-Origin Resource Sharing (CORS).

CORS is a security mechanism that uses additional HTTP headers to allow a web application running in one domain or origin to access a resource or retrieve data from a different domain, thus enabling a cross-domain request. The RESTful management service endpoints of the Oracle WebLogic REST API include new headers that are used for CORS control. You can configure the CORS HTTP header values to handle HTTP requests that are made to the Oracle WebLogic Server REST endpoints. See Modify This Restful Management Service in *RESTful Edit Management Interface for Oracle WebLogic Server*.

Table 3-4 describes the CORS configuration parameters that you can use to handle HTTP requests:

Table 3-4 CORS Configuration Parameters

CORS Parameters	Description
`CORSEnabled`	Specifies if the support for CORS (Cross-Origin Resource Sharing) processing in the RESTful management services web application is enabled. The default value is `false`.
`CORSAllowedOrigins`	Specifies a list of allowed origins for CORS requests. When specified, the HTTP `Origin` header must exactly match one of the values configured as allowed. If it does not match the `Origin` header, then the browser rejects the request. When the list is empty or not specified and CORS support is enabled, then all origins are accepted by default. CORS origin values include protocol, domain name, and may also include port numbers.

Table 3-4 (Cont.) CORS Configuration Parameters

CORS Parameters	Description
`CORSAllowedCredentials`	Determines if your server allows the use of credentials by using cookies. The default value is `false`.
`CORSAllowedHeaders`	Specifies a list of HTTP header names that are allowed for CORS requests. The default setting allows all headers.
`CORSAllowedMethods`	Specifies a list of HTTP methods that are allowed for CORS requests. The default setting allows all methods.
`CORSExposedHeaders`	Specifies a comma-separated list of HTTP header names to be exposed to the browser. The default setting does not specify any specific headers.
`CORSMaxAge`	Indicates the time, in seconds, for which the CORS preflight request results can be cached. The default setting does not specify any value.

4

Domain Level REST API Examples

Examine example scripts for users in domain level roles using the Oracle WebLogic Server REST APIs to perform common domain management and monitoring tasks. For information about the user roles which can access a REST resource, see Accessing REST Resources.

This chapter includes the following topics:

- Adding Users
  Review an example script that demonstrates how a System Administrator adds users such as Operators, Deployers, and Monitors.
- Setting Up Servers
  Review an example script that demonstrates how a System Administrator creates a cluster, machine, and dynamic server targeted to the cluster.
- Configuring System Resources
  Review an example script that demonstrates how a Deployer configures JDBC and JMS system resources.
- Deploying Domain-Scoped Applications
  Review an example script that demonstrates how a Deployer deploys domain-scoped applications.
- Monitoring Domain Resources
  Review an example script that demonstrates how an Operator monitors the entire domain.
- Starting and Stopping Domain-Scoped Applications
  Review an example script that demonstrates how an Operator starts and stops domain-scoped applications.
- Starting and Stopping Servers
  Review an example script that demonstrates how an Operator starts and stops servers.

Adding Users

Review an example script that demonstrates how a System Administrator adds users such as Operators, Deployers, and Monitors.

Note:

To view long URLs, use the scroll bar located beneath the section.

```
----------------------------------------------------------------------
Demonstrate a domain admin configuring domain level users
----------------------------------------------------------------------

----------------------------------------------------------------------
Create a deployer
----------------------------------------------------------------------
```

```
curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  userName: 'deployer',
  password: 'deployer123',
  description: 'A domain level deployer'
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverConfig/
securityConfiguration/realms/myrealm/authenticationProviders/DefaultAuthenticator/
createUser


HTTP/1.1 200 OK

Response Body:
{}


curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  groupName: 'Deployers',
  memberUserOrGroupName: 'deployer'
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverConfig/
securityConfiguration/realms/myrealm/authenticationProviders/DefaultAuthenticator/
addMemberToGroup


HTTP/1.1 200 OK

Response Body:
{}




----------------------------------------------------------------------
Create an operator
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  userName: 'operator',
  password: 'operator123',
  description: 'A domain level operator'
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverConfig/
securityConfiguration/realms/myrealm/authenticationProviders/DefaultAuthenticator/
createUser
```

```
HTTP/1.1 200 OK

Response Body:
{}


curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  groupName: 'Operators',
  memberUserOrGroupName: 'operator'
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverConfig/
securityConfiguration/realms/myrealm/authenticationProviders/DefaultAuthenticator/
addMemberToGroup


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
Create a monitor
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  userName: 'monitor',
  password: 'monitor123',
  description: 'A domain level monitor'
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverConfig/
securityConfiguration/realms/myrealm/authenticationProviders/DefaultAuthenticator/
createUser


HTTP/1.1 200 OK

Response Body:
{}


curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  groupName: 'Monitors',
  memberUserOrGroupName: 'monitor'
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverConfig/
securityConfiguration/realms/myrealm/authenticationProviders/DefaultAuthenticator/
addMemberToGroup


HTTP/1.1 200 OK

Response Body:
{}
```

Setting Up Servers

Review an example script that demonstrates how a System Administrator creates a cluster, machine, and dynamic server targeted to the cluster.

> **Note:**
>
> To view long URLs, use the scroll bar located beneath the section.

```
----------------------------------------------------------------------
Demonstrate a domain admin configuring dynamic servers
----------------------------------------------------------------------

----------------------------------------------------------------------
Start editing
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/changeManager/startEdit


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
View the default values for a new cluster
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/clusterCreateForm?links=none
```

```
HTTP/1.1 200 OK

Response Body:
{
    "sessionStateQueryRequestTimeout": 30,
    "notes": null,
    "sessionFlushInterval": 180,
    "txnAffinityEnabled": false,
    "fencingGracePeriodMillis": 30000,
    "serviceActivationRequestResponseTimeout": 0,
    "autoMigrationTableCreationDDLFile": null,
    "databaseLeasingBasisConnectionRetryCount": 1,
    "rebalanceDelayPeriods": 2,
    "autoMigrationTableCreationPolicy": "Disabled",
    "millisToSleepBetweenAutoMigrationAttempts": 180000,
    "migrationBasis": "database",
    "oneWayRmiForReplicationEnabled": false,
    "secureReplicationEnabled": false,
    "WANSessionPersistenceTableName": "WLS_WAN_PERSISTENCE_TABLE",
    "asyncSessionQueueTimeout": 30,
    "clusterType": "none",
    "databaseLeasingBasisConnectionRetryDelay": 1000,
    "defaultLoadAlgorithm": "round-robin",
    "frontendHTTPPort": 0,
    "singletonServiceRequestTimeout": 30000,
    "sessionFlushThreshold": 10000,
    "httpTraceSupportEnabled": false,
    "tags": null,
    "replicationTimeoutEnabled": true,
    "maxSecondarySelectionAttempts": 0,
    "serviceAgeThresholdSeconds": 180,
    "additionalAutoMigrationAttempts": 3,
    "multicastBufferSize": 64,
    "weblogicPluginEnabled": false,
    "healthCheckIntervalMillis": 10000,
    "jobSchedulerTableName": "WEBLOGIC_TIMERS",
    "concurrentSingletonActivationEnabled": false,
    "memberDeathDetectorEnabled": false,
    "multicastTTL": 1,
    "frontendHost": null,
    "clusterAddress": null,
    "interClusterCommLinkHealthCheckInterval": 30000,
    "remoteClusterAddress": null,
    "greedySessionFlushInterval": 3,
    "replicationChannel": "ReplicationChannel",
    "multicastAddress": "239.192.0.0",
    "numberOfServersInClusterAddress": 3,
    "persistSessionsOnShutdown": false,
    "healthCheckPeriodsUntilFencing": 6,
    "sessionStateQueryProtocolEnabled": false,
    "clusterBroadcastChannel": null,
    "multicastSendDelay": 3,
    "multicastDataEncryption": false,
    "messageOrderingEnabled": true,
    "autoMigrationTableName": "ACTIVE",
    "idlePeriodsUntilTimeout": 3,
    "clientCertProxyEnabled": false,
    "multicastPort": 7001,
    "clusterMessagingMode": "unicast",
    "unicastReadTimeout": 15000,
    "frontendHTTPSPort": 0,
    "dataSourceForSessionPersistence": null,
    "dataSourceForJobScheduler": null,
    "dataSourceForAutomaticMigration": null,
    "coherenceClusterSystemResource": null,
    "candidateMachinesForMigratableServers": [],
    "name": null
}




----------------------------------------------------------------------
Configure a new cluster
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{ name: 'Cluster1' }" \
-X POST http://localhost:7001/management/weblogic/latest/edit/clusters


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/clusters/Cluster1

Response Body:
{}




----------------------------------------------------------------------
View the new cluster
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/clusters/Cluster1?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "clusters",
        "Cluster1"
    ],
    "sessionStateQueryRequestTimeout": 30,
    "notes": null,
    "sessionFlushInterval": 180,
    "txnAffinityEnabled": false,
    "fencingGracePeriodMillis": 30000,
    "serviceActivationRequestResponseTimeout": 0,
    "replicationTimeoutMillis": 30000,
    "type": "Cluster",
    "autoMigrationTableCreationDDLFile": null,
    "databaseLeasingBasisConnectionRetryCount": 1,
    "rebalanceDelayPeriods": 2,
    "autoMigrationTableCreationPolicy": "Disabled",
    "millisToSleepBetweenAutoMigrationAttempts": 180000,
    "migrationBasis": "database",
    "oneWayRmiForReplicationEnabled": false,
    "id": 0,
    "secureReplicationEnabled": false,
    "WANSessionPersistenceTableName": "WLS_WAN_PERSISTENCE_TABLE",
    "asyncSessionQueueTimeout": 30,
    "clusterType": "none",
    "databaseLeasingBasisConnectionRetryDelay": 1000,
    "defaultLoadAlgorithm": "round-robin",
    "frontendHTTPPort": 0,
    "singletonServiceRequestTimeout": 30000,
    "sessionFlushThreshold": 10000,
    "httpTraceSupportEnabled": false,
    "tags": [],
    "replicationTimeoutEnabled": true,
    "maxSecondarySelectionAttempts": 0,
    "serviceAgeThresholdSeconds": 180,
    "additionalAutoMigrationAttempts": 3,
    "name": "Cluster1",
    "sessionLazyDeserializationEnabled": false,
    "multicastBufferSize": 64,
    "weblogicPluginEnabled": false,
    "healthCheckIntervalMillis": 10000,
    "jobSchedulerTableName": "WEBLOGIC_TIMERS",
    "concurrentSingletonActivationEnabled": false,
    "memberDeathDetectorEnabled": false,
    "multicastTTL": 1,
    "siteName": null,
    "frontendHost": null,
    "clusterAddress": null,
    "interClusterCommLinkHealthCheckInterval": 30000,
    "remoteClusterAddress": null,
    "greedySessionFlushInterval": 3,
    "memberWarmupTimeoutSeconds": 0,
    "replicationChannel": "ReplicationChannel",
    "multicastAddress": "239.192.0.0",
    "dynamicallyCreated": false,
    "numberOfServersInClusterAddress": 3,
    "persistSessionsOnShutdown": false,
    "healthCheckPeriodsUntilFencing": 6,
    "sessionStateQueryProtocolEnabled": false,
    "clusterBroadcastChannel": null,
    "multicastSendDelay": 3,
    "multicastDataEncryption": false,
    "messageOrderingEnabled": true,
    "autoMigrationTableName": "ACTIVE",
    "idlePeriodsUntilTimeout": 3,
    "clientCertProxyEnabled": false,
    "multicastPort": 7001,
    "clusterMessagingMode": "unicast",
    "unicastReadTimeout": 15000,
    "frontendHTTPSPort": 0,
    "dataSourceForSessionPersistence": null,
    "dataSourceForJobScheduler": null,
    "dataSourceForAutomaticMigration": null,
    "coherenceClusterSystemResource": null,
    "servers": [],
    "migratableTargets": [],
    "candidateMachinesForMigratableServers": []
}





----------------------------------------------------------------------
View the default values for a new machine
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/machineCreateForm?links=none


HTTP/1.1 200 OK

Response Body:
{
    "notes": null,
    "tags": null,
    "name": null
}





----------------------------------------------------------------------
Configure a new machine
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{ name:'Machine1' }" \
-X POST http://localhost:7001/management/weblogic/latest/edit/machines


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/machines/Machine1

Response Body:
{}





----------------------------------------------------------------------
View the new machine
----------------------------------------------------------------------
```

```
curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/machines/Machine1?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "machines",
        "Machine1"
    ],
    "notes": null,
    "name": "Machine1",
    "id": 0,
    "dynamicallyCreated": false,
    "type": "Machine",
    "tags": []
}





----------------------------------------------------------------------
View the default values for the machine's node manager configuration
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/machines/Machine1/
nodeManager?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "machines",
        "Machine1",
        "nodeManager"
    ],
    "adapter": null,
    "notes": null,
    "NMType": "SSL",
    "debugEnabled": false,
    "userName": null,
    "type": "NodeManager",
    "tags": [],
    "shellCommand": null,
    "NMSocketCreateTimeoutInMillis": 15000,
    "password": null,
    "listenAddress": "localhost",
    "name": "Machine1",
    "nodeManagerHome": null,
    "adapterVersion": null,
    "adapterName": null,
    "id": 0,
    "dynamicallyCreated": false,
    "listenPort": 5556
}




----------------------------------------------------------------------
Customize the machine's node manager configuration
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  NMType:        'Plain',
  listenAddress: 'localhost'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/machines/Machine1/
nodeManager


HTTP/1.1 200 OK

Response Body:
{}




----------------------------------------------------------------------
View the modified node manager configuration
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/machines/Machine1/
nodeManager?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "machines",
        "Machine1",
        "nodeManager"
    ],
    "adapter": null,
    "notes": null,
    "NMType": "Plain",
    "debugEnabled": false,
    "userName": null,
    "type": "NodeManager",
    "tags": [],
    "shellCommand": null,
    "NMSocketCreateTimeoutInMillis": 15000,
    "password": null,
    "listenAddress": "localhost",
    "name": "Machine1",
    "nodeManagerHome": null,
    "adapterVersion": null,
    "adapterName": null,
    "id": 0,
    "dynamicallyCreated": false,
    "listenPort": 5556
}




----------------------------------------------------------------------
View the default values for a new server template
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/serverTemplateCreateForm?
links=none


HTTP/1.1 200 OK

Response Body:
{
    "maxOpenSockCount": -1,
    "interfaceAddress": null,
    "startupTimeout": 0,
    "idleConnectionTimeout": 65,
    "resolveDNSName": false,
    "ignoreSessionsDuringShutdown": false,
    "adminReconnectIntervalSeconds": 10,
    "preferredSecondaryGroup": null,
    "defaultSecureProtocol": "t3s",
    "logCriticalRemoteExceptionsEnabled": false,
    "transactionPublicChannelName": null,
    "maxMessageSize": 10000000,
    "stagingDirectoryName": null,
    "outboundPrivateKeyEnabled": false,
    "defaultTGIOPPassword": null,
    "cleanupOrphanedSessionsEnabled": false,
    "httpTraceSupportEnabled": false,
    "classpathServletSecureModeEnabled": true,
    "tags": null,
    "useEnhancedIncrementAdvisor": true,
    "completeMessageTimeout": 60,
    "managedServerIndependenceEnabled": true,
    "retryIntervalBeforeMSIMode": 5,
    "nativeIOEnabled": true,
    "startupMode": "RUNNING",
    "externalDNSName": null,
    "JMSConnectionFactoryUnmappedResRefMode": "ReturnDefault",
    "extraEjbcOptions": null,
    "autoMigrationEnabled": false,
    "RMIDeserializationMaxTimeLimit": 0,
    "tunnelingClientPingSecs": 45,
    "instrumentStackTraceEnabled": true,
    "synchronizedSessionTimeoutEnabled": false,
    "customIdentityKeyStorePassPhrase": null,
    "transactionPrimaryChannelName": null,
    "gracefulShutdownTimeout": 0,
    "outboundEnabled": false,
    "javaStandardTrustKeyStorePassPhrase": null,
    "classpathServletDisabled": false,
    "healthCheckStartDelaySeconds": 120,
    "clientCertProxyEnabled": false,
    "defaultInternalServletsDisabled": false,
    "customIdentityKeyStoreType": null,
    "sessionReplicationOnShutdownEnabled": false,
    "restartIntervalSeconds": 3600,
    "notes": null,
    "serverLifeCycleTimeoutVal": 30,
    "httpdEnabled": true,
    "javaCompilerPostClassPath": null,
    "keyStores": "DemoIdentityAndDemoTrust",
    "sitConfigRequired": false,
    "defaultTGIOPUser": "guest",
    "use81StyleExecuteQueues": false,
    "uploadDirectoryName": null,
    "tunnelingClientTimeoutSecs": 40,
    "listenThreadStartDelaySecs": 60,
    "tunnelingEnabled": false,
    "listenAddress": null,
    "acceptBacklog": 300,
    "eagerThreadLocalCleanup": false,
    "connectTimeout": 0,
    "transactionSecureChannelName": null,
    "printStackTraceInProduction": false,
    "useFusionForLLR": false,
    "clusterWeight": 100,
    "customTrustKeyStorePassPhrase": null,
    "restartDelaySeconds": 0,
    "transactionLogFilePrefix": ".\/",
    "maxConcurrentLongRunningRequests": 100,
    "customTrustKeyStoreFileName": null,
    "socketReaders": -1,
    "threadPoolPercentSocketReaders": 33,
    "JDBCLoginTimeoutSeconds": 0,
    "customTrustKeyStoreType": null,
    "loginTimeoutMillis": 5000,
    "messageIdPrefixEnabled": true,
    "healthCheckIntervalSeconds": 180,
    "useEnhancedPriorityQueueForRequestManager": false,
    "reverseDNSAllowed": false,
    "periodLength": 60000,
    "socketBufferSizeAsChunkSize": false,
    "JDBCLLRTableName": null,
    "transactionPublicSecureChannelName": null,
    "weblogicPluginEnabled": false,
    "useDetailedThreadName": false,
    "stuckThreadTimerInterval": 60,
    "TGIOPEnabled": true,
    "listenersBindEarly": false,
    "selfTuningThreadPoolSizeMin": 1,
    "JNDITransportableObjectFactoryList": null,
    "NMSocketCreateTimeoutInMillis": 180000,
    "DGCIdlePeriodsUntilTimeout": 5,
    "defaultIIOPUser": null,
    "logRemoteExceptionsEnabled": false,
    "transactionLogFileWritePolicy": "Direct-Write",
    "defaultProtocol": "t3",
    "selfTuningThreadPoolSizeMax": 400,
    "replicationPorts": null,
    "autoRestart": true,
    "extraRmicOptions": null,
    "customIdentityKeyStoreFileName": null,
    "restartMax": 2,
    "replicationGroup": null,
    "defaultIIOPPassword": null,
    "IIOPEnabled": true,
    "maxConcurrentNewThreads": 100,
    "numOfRetriesBeforeMSIMode": 3,
    "JMSDefaultConnectionFactoriesEnabled": true,
    "sitConfigPollingInterval": 5,
    "allowShrinkingPriorityRequestQueue": true,
    "COMEnabled": false,
    "javaCompilerPreClassPath": null,
    "idlePeriodsUntilTimeout": 4,
    "javaCompiler": "javac",
    "cluster": null,
    "reliableDeliveryPolicy": null,
    "machine": null,
    "XMLEntityCache": null,
    "XMLRegistry": null,
    "coherenceClusterSystemResource": null,
    "candidateMachines": [],
    "name": null
}




----------------------------------------------------------------------
Configure a new server template for the cluster
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:          'ServerTemplate1',
  listenPort:    7100,
  listenAddress: 'localhost',
  machine:       [ 'machines', 'Machine1' ],
  cluster:       [ 'clusters', 'Cluster1' ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/serverTemplates


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/serverTemplates/
ServerTemplate1
```

```
Response Body:
{}





----------------------------------------------------------------------
Turn on resource management for the managed servers
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  arguments: '-XX:+UseG1GC'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/serverTemplates/
ServerTemplate1/serverStart


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
View the new server template
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/serverTemplates/
ServerTemplate1?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "serverTemplates",
        "ServerTemplate1"
    ],
    "stagingMode": "stage",
    "maxOpenSockCount": -1,
    "interfaceAddress": null,
    "hostsMigratableServices": true,
    "startupTimeout": 0,
    "idleConnectionTimeout": 65,
    "resolveDNSName": false,
    "ignoreSessionsDuringShutdown": false,
    "type": "ServerTemplate",
"adminReconnectIntervalSeconds": 10,
"administrationPort": 9002,
"preferredSecondaryGroup": null,
"defaultSecureProtocol": "t3s",
"logCriticalRemoteExceptionsEnabled": false,
"transactionPublicChannelName": null,
"id": 0,
"maxMessageSize": 10000000,
"completeWriteTimeout": 60,
"stagingDirectoryName": "\/domains\/mydomain\/servers\/ServerTemplate1\/stage",
"outboundPrivateKeyEnabled": false,
"defaultTGIOPPassword": null,
"cleanupOrphanedSessionsEnabled": false,
"httpTraceSupportEnabled": false,
"classpathServletSecureModeEnabled": true,
"tags": [],
"useEnhancedIncrementAdvisor": true,
"completeMessageTimeout": 60,
"managedServerIndependenceEnabled": true,
"retryIntervalBeforeMSIMode": 5,
"nativeIOEnabled": true,
"startupMode": "RUNNING",
"externalDNSName": null,
"JMSConnectionFactoryUnmappedResRefMode": "ReturnDefault",
"extraEjbcOptions": null,
"autoMigrationEnabled": false,
"RMIDeserializationMaxTimeLimit": 0,
"tunnelingClientPingSecs": 45,
"instrumentStackTraceEnabled": true,
"synchronizedSessionTimeoutEnabled": false,
"dynamicallyCreated": false,
"customIdentityKeyStorePassPhrase": null,
"transactionPrimaryChannelName": null,
"gracefulShutdownTimeout": 0,
"outboundEnabled": false,
"javaStandardTrustKeyStorePassPhrase": null,
"useConcurrentQueueForRequestManager": false,
"classpathServletDisabled": false,
"healthCheckStartDelaySeconds": 120,
"clientCertProxyEnabled": false,
"defaultInternalServletsDisabled": false,
"customIdentityKeyStoreType": null,
"sessionReplicationOnShutdownEnabled": false,
"restartIntervalSeconds": 3600,
"notes": null,
"serverLifeCycleTimeoutVal": 30,
"httpdEnabled": true,
"javaCompilerPostClassPath": null,
"keyStores": "DemoIdentityAndDemoTrust",
"sitConfigRequired": false,
"use81StyleExecuteQueues": false,
"uploadDirectoryName": ".\/servers\/ServerTemplate1\/upload",
"tunnelingClientTimeoutSecs": 40,
"listenThreadStartDelaySecs": 60,
"tunnelingEnabled": false,
"listenAddress": "localhost",
"acceptBacklog": 300,
"listenPortEnabled": true,
"eagerThreadLocalCleanup": false,
"connectTimeout": 0,
"transactionSecureChannelName": null,
"printStackTraceInProduction": false,
"scatteredReadsEnabled": false,
"muxerClass": "weblogic.socket.NIOSocketMuxer",
"useFusionForLLR": false,
"clusterWeight": 100,
"customTrustKeyStorePassPhrase": null,
"restartDelaySeconds": 0,
"transactionLogFilePrefix": ".\/",
"maxConcurrentLongRunningRequests": 100,
"customTrustKeyStoreFileName": null,
"socketReaders": -1,
"threadPoolPercentSocketReaders": 33,
"JDBCLoginTimeoutSeconds": 0,
"customTrustKeyStoreType": null,
"loginTimeoutMillis": 5000,
"messageIdPrefixEnabled": false,
"healthCheckIntervalSeconds": 180,
"useEnhancedPriorityQueueForRequestManager": false,
"name": "ServerTemplate1",
"reverseDNSAllowed": false,
"periodLength": 60000,
"socketBufferSizeAsChunkSize": false,
"JDBCLLRTableName": null,
"transactionPublicSecureChannelName": null,
"weblogicPluginEnabled": false,
"useDetailedThreadName": false,
"stuckThreadTimerInterval": 60,
"TGIOPEnabled": true,
"listenersBindEarly": false,
"selfTuningThreadPoolSizeMin": 1,
"JNDITransportableObjectFactoryList": [],
"DGCIdlePeriodsUntilTimeout": 5,
"defaultIIOPUser": null,
"logRemoteExceptionsEnabled": false,
"transactionLogFileWritePolicy": "Direct-Write",
"gatheredWritesEnabled": false,
"defaultProtocol": "t3",
"selfTuningThreadPoolSizeMax": 400,
"replicationPorts": null,
"autoRestart": true,
"extraRmicOptions": null,
"customIdentityKeyStoreFileName": null,
"restartMax": 2,
"replicationGroup": null,
"defaultIIOPPassword": null,
"IIOPEnabled": true,
"virtualThreadEnableOption": "disabled",
"maxConcurrentNewThreads": 100,
"numOfRetriesBeforeMSIMode": 3,
"JMSDefaultConnectionFactoriesEnabled": true,
"sitConfigPollingInterval": 5,
"allowShrinkingPriorityRequestQueue": true,
"addWorkManagerThreadsByCpuCount": false,
"javaCompilerPreClassPath": null,
"idlePeriodsUntilTimeout": 4,
"listenPort": 7100,
"javaCompiler": "javac",
"cluster": [
    "clusters",
    "Cluster1"
],
"reliableDeliveryPolicy": null,
"machine": [
            "machines",
            "Machine1"
        ],
        "XMLEntityCache": null,
        "XMLRegistry": null,
        "coherenceClusterSystemResource": null,
        "candidateMachines": []
    }





    ----------------------------------------------------------------------
    View the default values for the cluster's dynamic servers configuration
    ----------------------------------------------------------------------

    curl -v \
    --user admin:admin123 \
    -H X-Requested-By:MyClient \
    -H Accept:application/json \
    -X GET http://localhost:7001/management/weblogic/latest/edit/clusters/Cluster1/
    dynamicServers?links=none


    HTTP/1.1 200 OK

    Response Body:
    {
        "identity": [
            "clusters",
            "Cluster1",
            "dynamicServers"
        ],
        "notes": null,
        "serverNameStartingIndex": 1,
        "dynamicClusterSize": 0,
        "machineNameMatchExpression": null,
        "maxDynamicClusterSize": 8,
        "serverNamePrefix": "Cluster1-",
        "ignoreSessionsDuringShutdown": false,
        "type": "DynamicServers",
        "calculatedMachineNames": false,
        "dynamicClusterShutdownTimeoutSeconds": 0,
        "tags": [],
        "waitForAllSessionsDuringShutdown": false,
        "machineMatchExpression": null,
        "dynamicServerNames": [],
        "calculatedListenPorts": true,
        "name": "Cluster1",
        "id": 0,
        "dynamicallyCreated": false,
        "machineMatchType": "name",
        "minDynamicClusterSize": 1,
        "dynamicClusterCooloffPeriodSeconds": 900,
        "serverTemplate": null
    }
```

```
----------------------------------------------------------------------
Customize the cluster's dynamic servers configuration
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  serverTemplate:        [ 'serverTemplates', 'ServerTemplate1' ],
  dynamicClusterSize:    2,
  serverNamePrefix:      'Cluster1Server'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/clusters/Cluster1/
dynamicServers


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
View the modified dynamic servers configuration
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/clusters/Cluster1/
dynamicServers?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "clusters",
        "Cluster1",
        "dynamicServers"
    ],
    "notes": null,
    "serverNameStartingIndex": 1,
    "dynamicClusterSize": 2,
    "machineNameMatchExpression": null,
    "maxDynamicClusterSize": 8,
    "serverNamePrefix": "Cluster1Server",
    "ignoreSessionsDuringShutdown": false,
    "type": "DynamicServers",
    "calculatedMachineNames": false,
    "dynamicClusterShutdownTimeoutSeconds": 0,
    "tags": [],
    "waitForAllSessionsDuringShutdown": false,
    "machineMatchExpression": null,
    "dynamicServerNames": [
        "Cluster1Server1",
        "Cluster1Server2"
    ],
    "calculatedListenPorts": true,
    "name": "Cluster1",
    "id": 0,
    "dynamicallyCreated": false,
    "machineMatchType": "name",
    "minDynamicClusterSize": 1,
    "dynamicClusterCooloffPeriodSeconds": 900,
    "serverTemplate": [
        "serverTemplates",
        "ServerTemplate1"
    ]
}




----------------------------------------------------------------------
Activate the changes
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/changeManager/activate


HTTP/1.1 200 OK

Response Body:
{}




----------------------------------------------------------------------
Synchronously start the managed servers
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server1/start


HTTP/1.1 200 OK

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
serverLifeCycleRuntimes\/Cluster1Server1\/tasks\/_0_start"
```

```
    }],
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server1",
        "tasks",
        "_0_start"
    ],
    "running": false,
    "systemTask": false,
    "endTimeAsLong": 1726603152444,
    "name": "_0_start",
    "progress": "success",
    "description": "Starting Cluster1Server1 server ...",
    "serverName": "Cluster1Server1",
    "taskError": null,
    "startTimeAsLong": 1726603127179,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "start",
    "taskStatus": "TASK COMPLETED",
    "parentTask": null,
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T15:58:47.179-04:00",
    "endTime": "2024-09-17T15:59:12.444-04:00"
}


curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/start


HTTP/1.1 200 OK

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
serverLifeCycleRuntimes\/Cluster1Server2\/tasks\/_1_start"
    }],
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_1_start"
    ],
    "running": false,
    "systemTask": false,
    "endTimeAsLong": 1726603174545,
    "name": "_1_start",
    "progress": "success",
    "description": "Starting Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603154340,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "start",
    "taskStatus": "TASK COMPLETED",
    "parentTask": null,
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T15:59:14.340-04:00",
    "endTime": "2024-09-17T15:59:34.545-04:00"
}





----------------------------------------------------------------------
Verify that the managed servers are running
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes?links=none


HTTP/1.1 200 OK

Response Body:
{
    "items": [
        {
            "identity": [
                "serverLifeCycleRuntimes",
                "Cluster1Server1"
            ],
            "nodeManagerRestartCount": 0,
            "middlewareHome": "\/Oracle_Home",
            "name": "Cluster1Server1",
            "weblogicHome": "\/Oracle_Home\/wlserver",
            "state": "RUNNING",
            "type": "ServerLifeCycleRuntime"
        },
        {
            "identity": [
                "serverLifeCycleRuntimes",
                "AdminServer"
            ],
            "nodeManagerRestartCount": 0,
            "middlewareHome": "\/Oracle_Home",
            "name": "AdminServer",
            "weblogicHome": "\/Oracle_Home\/wlserver",
            "state": "RUNNING",
            "type": "ServerLifeCycleRuntime"
        },
        {
            "identity": [
                "serverLifeCycleRuntimes",
                "Cluster1Server2"
            ],
            "nodeManagerRestartCount": 0,
            "middlewareHome": "\/Oracle_Home",
            "name": "Cluster1Server2",
            "weblogicHome": "\/Oracle_Home\/wlserver",
            "state": "RUNNING",
            "type": "ServerLifeCycleRuntime"
        }
    ]
}


curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/serverRuntimes?
links=none&fields=name,state


HTTP/1.1 200 OK

Response Body:
{
    "items": [
        {
            "state": "RUNNING",
            "name": "Cluster1Server1"
        },
        {
            "state": "RUNNING",
            "name": "AdminServer"
        },
        {
            "state": "RUNNING",
            "name": "Cluster1Server2"
        }
    ]
}
```

Configuring System Resources

Review an example script that demonstrates how a Deployer configures JDBC and JMS system resources.

Note:

To view long URLs, use the scroll bar located beneath the section.

```
----------------------------------------------------------------------
Demonstrate a domain deployer configuring system resources
----------------------------------------------------------------------

----------------------------------------------------------------------
View the default values for a new global JDBC system resource
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/
```

```
JDBCSystemResourceCreateForm?links=none


HTTP/1.1 200 OK

Response Body:
{
    "notes": null,
    "moduleType": null,
    "deploymentPrincipalName": null,
    "compatibilityName": null,
    "deploymentOrder": 100,
    "tags": null,
    "targets": [],
    "name": null,
    "descriptorFileName": null
}





----------------------------------------------------------------------
Start editing
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/changeManager/startEdit


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
Create a new global JDBC system resource and set its name
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name: 'JDBCDataSource1',
  targets: [ { identity: [ clusters, 'Cluster1' ] } ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources?
saveChanges=false


HTTP/1.1 201 Created
```

```
Location: http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1

Response Body:
{}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name: 'JDBCDataSource1'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
Configure the JDBC system resource's JNDI name
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  JNDINames: [ 'JDBCDataSource1' ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDataSourceParams


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
Configure the JDBC system resource's driver info
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  driverName: 'org.apache.derby.jdbc.ClientXADataSource',
  url:        'jdbc:derby://localhost:1527/demo'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams


HTTP/1.1 200 OK

Response Body:
{}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:  'portNumber',
  value: '1527'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams/properties/properties


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams/properties/properties/portNumber

Response Body:
{}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:  'databaseName',
  value: 'demo;create=true'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams/properties/properties


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams/properties/properties/databaseName

Response Body:
{}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:  'serverName',
  value: 'localhost'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams/properties/properties


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams/properties/properties/serverName

Response Body:
{}





----------------------------------------------------------------------
Activate the changes
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/changeManager/activate


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
View the new JDBC system resource
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "JDBCSystemResources",
        "JDBCDataSource1"
    ],
    "notes": null,
    "moduleType": null,
    "deploymentPrincipalName": null,
    "descriptorFileName": "jdbc\/JDBCDataSource1-6865-jdbc.xml",
    "name": "JDBCDataSource1",
    "compatibilityName": null,
    "id": 0,
    "deploymentOrder": 100,
    "dynamicallyCreated": false,
    "type": "JDBCSystemResource",
    "sourcePath": ".\/config\/jdbc\/JDBCDataSource1-6865-jdbc.xml",
    "tags": [],
    "resource": [
        "JDBCSystemResources",
        "JDBCDataSource1",
        "JDBCResource"
    ],
    "targets": [{
        "identity": [
            "clusters",
            "Cluster1"
        ]
    }]
}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "JDBCSystemResources",
        "JDBCDataSource1",
        "JDBCResource"
    ],
    "datasourceType": null,
    "name": "JDBCDataSource1",
    "id": 0,
    "version": null
}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDataSourceParams?links=none


HTTP/1.1 200 OK
```

```
Response Body:
{
    "identity": [
        "JDBCSystemResources",
        "JDBCDataSource1",
        "JDBCResource",
        "JDBCDataSourceParams"
    ],
    "connectionPoolFailoverCallbackHandler": null,
    "globalTransactionsProtocol": "OnePhaseCommit",
    "algorithmType": "Failover",
    "scope": "Global",
    "failoverRequestIfBusy": false,
    "proxySwitchingCallback": null,
    "JNDINames": ["JDBCDataSource1"],
    "proxySwitchingProperties": null,
    "dataSourceList": null,
    "keepConnAfterGlobalTx": false
}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "JDBCSystemResources",
        "JDBCDataSource1",
        "JDBCResource",
        "JDBCDriverParams"
    ],
    "password": null,
    "driverName": "org.apache.derby.jdbc.ClientXADataSource",
    "usePasswordIndirection": false,
    "url": "jdbc:derby:\/\/localhost:1527\/demo",
    "useXaDataSourceInterface": true
}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JDBCSystemResources/
JDBCDataSource1/JDBCResource/JDBCDriverParams/properties/properties?links=none


HTTP/1.1 200 OK

Response Body:
{
    "items": [
        {
            "identity": [
                "JDBCSystemResources",
                "JDBCDataSource1",
                "JDBCResource",
                "JDBCDriverParams",
                "properties",
                "properties",
                "portNumber"
            ],
            "encryptedValue": null,
            "name": "portNumber",
            "sysPropValue": null,
            "value": "1527"
        },
        {
            "identity": [
                "JDBCSystemResources",
                "JDBCDataSource1",
                "JDBCResource",
                "JDBCDriverParams",
                "properties",
                "properties",
                "databaseName"
            ],
            "encryptedValue": null,
            "name": "databaseName",
            "sysPropValue": null,
            "value": "demo;create=true"
        },
        {
            "identity": [
                "JDBCSystemResources",
                "JDBCDataSource1",
                "JDBCResource",
                "JDBCDriverParams",
                "properties",
                "properties",
                "serverName"
            ],
            "encryptedValue": null,
            "name": "serverName",
            "sysPropValue": null,
            "value": "localhost"
        }
    ]
}




----------------------------------------------------------------------
Search for all of the new JDBC data source's runtimes
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      links: [], fields: [ 'name' ],
      children: {
        JDBCServiceRuntime: {
          links: [], fields: [ 'name' ],
          children: {
            JDBCDataSourceRuntimeMBeans : {
              links: [], fields: [ 'name', 'state' ],
              name: [ 'JDBCDataSource1' ]
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "JDBCServiceRuntime": {
                    "name": "Cluster1Server1",
                    "JDBCDataSourceRuntimeMBeans": {
                        "items": [{
                            "state": "Running",
                            "name": "JDBCDataSource1"
                        }]
                    }
                }
            },
            {
                "name": "AdminServer",
                "JDBCServiceRuntime": {
                    "name": "AdminServer",
                    "JDBCDataSourceRuntimeMBeans": {
                        "items": []
                    }
                }
            },
            {
                "name": "Cluster1Server2",
                "JDBCServiceRuntime": {
                    "name": "Cluster1Server2",
                    "JDBCDataSourceRuntimeMBeans": {
                        "items": [{
                            "state": "Running",
                            "name": "JDBCDataSource1"
                        }]
                    }
                }
            }
        ]
    }
}
```

```
----------------------------------------------------------------------
Start editing
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/changeManager/startEdit


HTTP/1.1 200 OK

Response Body:
{}





----------------------------------------------------------------------
View the default values for a new global JMS file store
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/fileStoreCreateForm?
links=none


HTTP/1.1 200 OK

Response Body:
{
    "notes": null,
    "fileLockingEnabled": true,
    "distributionPolicy": "Distributed",
    "partialClusterStabilityDelaySeconds": 240,
    "deploymentOrder": 1000,
    "directory": null,
    "initialBootDelaySeconds": 60,
    "ioBufferSize": -1,
    "minWindowBufferSize": -1,
    "failbackDelaySeconds": -1,
    "cacheDirectory": null,
    "numberOfRestartAttempts": 6,
    "initialSize": 0,
    "rebalanceEnabled": false,
    "logicalName": null,
    "maxFileSize": 1342177280,
    "synchronousWritePolicy": "Direct-Write",
    "blockSize": -1,
    "tags": null,
    "maxWindowBufferSize": -1,
    "migrationPolicy": "Off",
    "secondsBetweenRestarts": 30,
    "failOverLimit": -1,
    "targets": [],
    "name": null
}




----------------------------------------------------------------------
Create a new global file store
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:               'FileStore1',
  targets:            [ { identity: [ 'clusters', 'Cluster1' ] } ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/fileStores


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/fileStores/FileStore1

Response Body:
{}




----------------------------------------------------------------------
View the default values for a new global JMS server
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JMSServerCreateForm?
links=none


HTTP/1.1 200 OK

Response Body:
{
    "messagesThresholdHigh": -1,
    "hostingTemporaryDestinations": true,
    "temporaryTemplateName": null,
    "notes": null,
    "maximumMessageSize": 2147483647,
    "allowsPersistentDowngrade": false,
    "storeMessageCompressionEnabled": false,
    "deploymentOrder": 1000,
    "pagingMessageCompressionEnabled": false,
    "messageBufferSize": -1,
    "bytesThresholdLow": -1,
    "expirationScanInterval": 30,
    "messagesThresholdLow": -1,
    "blockingSendPolicy": "FIFO",
    "pagingBlockSize": -1,
    "insertionPausedAtStartup": "default",
    "pagingMaxWindowBufferSize": -1,
    "bytesThresholdHigh": -1,
    "pagingMaxFileSize": 1342177280,
    "productionPausedAtStartup": "default",
    "pagingFileLockingEnabled": true,
    "tags": null,
    "bytesMaximum": -1,
    "temporaryTemplateResource": null,
    "messageCompressionOptions": "GZIP_DEFAULT_COMPRESSION",
    "pagingMinWindowBufferSize": -1,
    "pagingIoBufferSize": -1,
    "messagesMaximum": -1,
    "consumptionPausedAtStartup": "default",
    "storeEnabled": true,
    "pagingDirectory": null,
    "persistentStore": null,
    "targets": [],
    "name": null
}




----------------------------------------------------------------------
Create a new global JMS server and hook it up to the cluster and file store
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:            'JMSServer1',
  messagesMaximum: 10000,
  bytesMaximum:    10000000,
  targets:         [ { identity: [ 'clusters', 'Cluster1' ] } ],
  persistentStore: [ 'fileStores', 'FileStore1' ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JMSServers


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JMSServers/JMSServer1

Response Body:
{}
```

```
----------------------------------------------------------------------
View the default values for a new global JMS system resource
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JMSSystemResourceCreateForm?
links=none


HTTP/1.1 200 OK

Response Body:
{
    "notes": null,
    "moduleType": null,
    "deploymentPrincipalName": null,
    "compatibilityName": null,
    "deploymentOrder": 100,
    "tags": null,
    "targets": [],
    "name": null,
    "descriptorFileName": null
}





----------------------------------------------------------------------
Create a new global JMS system resource and hook it up to the cluster
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:    'JMSSystemResource1',
  targets: [ { identity: [ 'clusters', 'Cluster1' ] } ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1

Response Body:
{}





----------------------------------------------------------------------
View the default values for a new JMS subdeployment
----------------------------------------------------------------------
```

```
curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/subDeploymentCreateForm?links=none


HTTP/1.1 200 OK

Response Body:
{
    "notes": null,
    "moduleType": null,
    "compatibilityName": null,
    "untargeted": false,
    "tags": null,
    "targets": [],
    "name": null
}





----------------------------------------------------------------------
Create a new JMS subdeployment and hook it up to the JMS server
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:    'JMSSubDeployment1',
  targets: [ { identity: [ 'JMSServers', 'JMSServer1' ] } ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/subDeployments


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/subDeployments/JMSSubDeployment1

Response Body:
{}





----------------------------------------------------------------------
View the default values for a new JMS connection factory
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/JMSResource/connectionFactoryCreateForm?links=none


HTTP/1.1 200 OK

Response Body:
{
    "notes": null,
    "JNDIName": null,
    "defaultTargetingEnabled": false,
    "localJNDIName": null,
    "name": null
}






----------------------------------------------------------------------
Create a new JMS connection factory and hook it up to the JMS subdeployment
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:              'ConnectionFactory1',
  subDeploymentName: 'JMSSubDeployment1'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/JMSResource/connectionFactories


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/JMSResource/connectionFactories/ConnectionFactory1

Response Body:
{}






----------------------------------------------------------------------
View the default values for a new JMS distributed queue
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/JMSResource/uniformDistributedQueueCreateForm?links=none


HTTP/1.1 200 OK
```

```
Response Body:
{
    "notes": null,
    "JNDIName": null,
    "unitOfOrderRouting": "Hash",
    "resetDeliveryCountOnForward": true,
    "defaultUnitOfOrder": false,
    "defaultTargetingEnabled": false,
    "incompleteWorkExpirationTime": -1,
    "loadBalancingPolicy": "Round-Robin",
    "forwardDelay": -1,
    "JMSCreateDestinationIdentifier": null,
    "localJNDIName": null,
    "template": null,
    "quota": null,
    "name": null
}




----------------------------------------------------------------------
Create a new JMS uniform distributed queue and hook it up to the JMS subdeployment
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:              'UniformDistributedQueue1',
  subDeploymentName: 'JMSSubDeployment1'
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/JMSResource/uniformDistributedQueues


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/JMSSystemResources/
JMSSystemResource1/JMSResource/uniformDistributedQueues/UniformDistributedQueue1

Response Body:
{}




----------------------------------------------------------------------
Activate the changes
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/changeManager/activate


HTTP/1.1 200 OK

Response Body:
{}






----------------------------------------------------------------------
View the file stores
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/fileStores?links=none


HTTP/1.1 200 OK

Response Body:
{
    "items": [{
        "identity": [
            "fileStores",
            "FileStore1"
        ],
        "notes": null,
        "fileLockingEnabled": true,
        "distributionPolicy": "Distributed",
        "partialClusterStabilityDelaySeconds": 240,
        "deploymentOrder": 1000,
        "type": "FileStore",
        "directory": null,
        "initialBootDelaySeconds": 60,
        "ioBufferSize": -1,
        "minWindowBufferSize": -1,
        "failbackDelaySeconds": -1,
        "cacheDirectory": null,
        "id": 0,
        "dynamicallyCreated": false,
        "XAResourceName": null,
        "numberOfRestartAttempts": 6,
        "initialSize": 0,
        "rebalanceEnabled": false,
        "logicalName": null,
        "maxFileSize": 1342177280,
        "synchronousWritePolicy": "Direct-Write",
        "blockSize": -1,
        "tags": [],
        "maxWindowBufferSize": -1,
        "name": "FileStore1",
        "migrationPolicy": "Off",
        "secondsBetweenRestarts": 30,
        "restartInPlace": false,
        "failOverLimit": -1,
        "targets": [{
            "identity": [
                "clusters",
                "Cluster1"
            ]
        }]
    }]
}





----------------------------------------------------------------------
View the JMS servers
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/JMSServers?links=none


HTTP/1.1 200 OK

Response Body:
{
    "items": [{
        "identity": [
            "JMSServers",
            "JMSServer1"
        ],
        "messagesThresholdHigh": -1,
        "hostingTemporaryDestinations": true,
        "temporaryTemplateName": null,
        "notes": null,
        "maximumMessageSize": 2147483647,
        "allowsPersistentDowngrade": false,
        "storeMessageCompressionEnabled": false,
        "deploymentOrder": 1000,
        "type": "JMSServer",
        "pagingMessageCompressionEnabled": false,
        "messageBufferSize": -1,
        "bytesThresholdLow": -1,
        "expirationScanInterval": 30,
        "messagesThresholdLow": -1,
        "blockingSendPolicy": "FIFO",
        "id": 0,
        "dynamicallyCreated": false,
        "pagingBlockSize": -1,
        "insertionPausedAtStartup": "default",
        "pagingMaxWindowBufferSize": -1,
        "bytesThresholdHigh": -1,
        "pagingMaxFileSize": 1342177280,
        "productionPausedAtStartup": "default",
        "pagingFileLockingEnabled": true,
        "tags": [],
        "bytesMaximum": 10000000,
        "temporaryTemplateResource": null,
        "messageCompressionOptions": "GZIP_DEFAULT_COMPRESSION",
        "pagingMinWindowBufferSize": -1,
        "pagingIoBufferSize": -1,
        "messagesMaximum": 10000,
        "name": "JMSServer1",
        "consumptionPausedAtStartup": "default",
        "storeEnabled": true,
        "pagingDirectory": null,
        "persistentStore": [
            "fileStores",
            "FileStore1"
        ],
        "targets": [{
            "identity": [
                "clusters",
                "Cluster1"
            ]
        }]
    }]
}




----------------------------------------------------------------------
View the JMS system resources and their children
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    JMSSystemResources: {
      links: [],
      children: {
        JMSResource: {
          links: [], fields: [],
          children: {
            connectionFactories: {
              links: []
            },
            distributedQueues: {
              links: []
            },
          }
        },
        subDeployments: {
          links: []
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/search


HTTP/1.1 200 OK

Response Body:
{
```

```
"JMSSystemResources": {
    "items": [{
        "identity": [
            "JMSSystemResources",
            "JMSSystemResource1"
        ],
        "notes": null,
        "moduleType": null,
        "deploymentPrincipalName": null,
        "descriptorFileName": "jms\/jmssystemresource1-jms.xml",
        "name": "JMSSystemResource1",
        "compatibilityName": null,
        "id": 0,
        "deploymentOrder": 100,
        "dynamicallyCreated": false,
        "type": "JMSSystemResource",
        "sourcePath": ".\/config\/jms\/jmssystemresource1-jms.xml",
        "tags": [],
        "resource": [
            "JMSSystemResources",
            "JMSSystemResource1",
            "JMSResource"
        ],
        "targets": [{
            "identity": [
                "clusters",
                "Cluster1"
            ]
        }],
        "subDeployments": {
            "items": [{
                "identity": [
                    "JMSSystemResources",
                    "JMSSystemResource1",
                    "subDeployments",
                    "JMSSubDeployment1"
                ],
                "notes": null,
                "moduleType": null,
                "name": "JMSSubDeployment1",
                "compatibilityName": null,
                "untargeted": false,
                "id": 0,
                "dynamicallyCreated": false,
                "type": "SubDeployment",
                "tags": [],
                "targets": [{
                    "identity": [
                        "JMSServers",
                        "JMSServer1"
                    ]
                }]
            }]
        },
        "JMSResource": {
            "connectionFactories": {
                "items": [{
                    "identity": [
                        "JMSSystemResources",
                        "JMSSystemResource1",
                        "JMSResource",
                        "connectionFactories",
                            "ConnectionFactory1"
                        ],
                        "notes": null,
                        "JNDIName": null,
                        "defaultTargetingEnabled": false,
                        "name": "ConnectionFactory1",
                        "subDeploymentName": "JMSSubDeployment1",
                        "id": 0,
                        "localJNDIName": null
                    }]
                }
            }
        }]
    }
}





-----------------------------------------------------------------------
Search for all of the JMS related runtimes
-----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      links: [], fields: [ 'name' ],
      children: {
        JMSRuntime: {
          links: [], fields: [ 'name', 'healthState' ],
          children: {
            JMSServers: {
              links: [], fields: [ 'name', 'healthState' ],
              children: {
                destinations: {
                  links: [], fields: [ 'name', 'state' ],
                }
              }
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "JMSRuntime": {
                    "healthState": {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    "name": "Cluster1Server1.jms",
                    "JMSServers": {
                        "items": [{
                            "healthState": {
                                "state": "ok",
                                "subsystemName": "JMSServer.JMSServer1@Cluster1Server1",
                                "symptoms": []
                            },
                            "name": "JMSServer1@Cluster1Server1",
                            "destinations": {
                                "items": [{
                                    "state": "started",
                                    "name": "JMSSystemResource1!
JMSServer1@Cluster1Server1@UniformDistributedQueue1"
                                }]
                            }
                        }]
                    }
                }
            },
            {
                "name": "AdminServer",
                "JMSRuntime": {
                    "healthState": {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    "name": "AdminServer.jms",
                    "JMSServers": {
                        "items": []
                    }
                }
            },
            {
                "name": "Cluster1Server2",
                "JMSRuntime": {
                    "healthState": {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    "name": "Cluster1Server2.jms",
                    "JMSServers": {
                        "items": [{
                            "healthState": {
                                "state": "ok",
                                "subsystemName": "JMSServer.JMSServer1@Cluster1Server2",
                                "symptoms": []
                            },
                            "name": "JMSServer1@Cluster1Server2",
                            "destinations": {
                                "items": [{
                                    "state": "started",
                                    "name": "JMSSystemResource1!
JMSServer1@Cluster1Server2@UniformDistributedQueue1"
                                }]
                            }
                        }]
                    }
                }
            }
        ]
    }
}
```

Deploying Domain-Scoped Applications

Review an example script that demonstrates how a Deployer deploys domain-scoped applications.

> **Note:**
>
> To view long URLs, use the scroll bar located beneath the section.

```
----------------------------------------------------------------------
Demonstrate a domain deployer deploying apps
----------------------------------------------------------------------

----------------------------------------------------------------------
Synchronously deploy a domain-scoped server-side application to the cluster
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  name:       'fairShare',
  sourcePath: '/deployments/fairShare.war',
  targets:    [ { identity: [ 'clusters', 'Cluster1' ] } ]
}" \
-X POST http://localhost:7001/management/weblogic/latest/edit/appDeployments


HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/edit/appDeployments/fairShare

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
deploymentManager\/deploymentProgressObjects\/fairShare"
    }],
    "identity": [
        "deploymentManager",
        "deploymentProgressObjects",
        "fairShare"
    ],
    "rootExceptions": [],
    "endTimeAsLong": 1726603188497,
    "deploymentMessages": [
        "[Deployer:149191]Operation \"deploy\" on application \"fairShare\" is
initializing on \"Cluster1Server2\".",
        "[Deployer:149191]Operation \"deploy\" on application \"fairShare\" is
initializing on \"Cluster1Server1\".",
        "[Deployer:149192]Operation \"deploy\" on application \"fairShare\" is in
progress on \"Cluster1Server1\".",
        "[Deployer:149192]Operation \"deploy\" on application \"fairShare\" is in
progress on \"Cluster1Server2\".",
        "[Deployer:149194]Operation \"deploy\" on application \"fairShare\" has
succeeded on \"Cluster1Server1\".",
        "[Deployer:149194]Operation \"deploy\" on application \"fairShare\" has
succeeded on \"Cluster1Server2\"."
    ],
    "name": "fairShare",
    "operationType": 3,
    "startTimeAsLong": 1726603188156,
    "state": "STATE_COMPLETED",
    "id": "0",
    "type": "DeploymentProgressObject",
    "targets": ["Cluster1"],
    "applicationName": "fairShare",
    "failedTargets": [],
    "progress": "success",
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T15:59:48.156-04:00",
    "endTime": "2024-09-17T15:59:48.497-04:00"
}




----------------------------------------------------------------------
Asynchronously upload a domain-scoped application from the client and deploy it to the
cluster
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:multipart/form-data \
-F "model={
  name:    'basicapp',
  targets: [ { identity: [ 'clusters' , 'Cluster1' ] } ]
}" \
-F "sourcePath=@/deployments/BasicApp/app/BasicApp.ear" \
-F "planPath=@/deployments/BasicApp/plan/Plan.xml" \
-H "Prefer:respond-async" \
-X POST http://localhost:7001/management/weblogic/latest/edit/appDeployments


HTTP/1.1 202 Accepted

Location: http://localhost:7001/management/weblogic/latest/domainRuntime/
deploymentManager/deploymentProgressObjects/basicapp
```

```
Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
deploymentManager\/deploymentProgressObjects\/basicapp"
    }],
    "identity": [
        "deploymentManager",
        "deploymentProgressObjects",
        "basicapp"
    ],
    "rootExceptions": [],
    "deploymentMessages": [],
    "name": "basicapp",
    "operationType": 3,
    "startTimeAsLong": 1726603189541,
    "state": "STATE_RUNNING",
    "id": "1",
    "type": "DeploymentProgressObject",
    "targets": ["Cluster1"],
    "applicationName": "basicapp",
    "failedTargets": [],
    "progress": "processing",
    "completed": false,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T15:59:49.541-04:00"
}




----------------------------------------------------------------------
Get status for job domainRuntime/deploymentManager/deploymentProgressObjects/basicapp
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
deploymentProgressObjects/basicapp?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "deploymentManager",
        "deploymentProgressObjects",
        "basicapp"
    ],
    "rootExceptions": [],
    "deploymentMessages": [
        "[Deployer:149191]Operation \"deploy\" on application \"basicapp\" is
initializing on \"Cluster1Server2\".",
        "[Deployer:149191]Operation \"deploy\" on application \"basicapp\" is
initializing on \"Cluster1Server1\"."
    ],
    "name": "basicapp",
    "operationType": 3,
    "startTimeAsLong": 1726603189541,
    "state": "STATE_RUNNING",
    "id": "1",
    "type": "DeploymentProgressObject",
    "targets": ["Cluster1"],
    "applicationName": "basicapp",
    "failedTargets": [],
    "progress": "processing",
    "completed": false,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T15:59:49.541-04:00"
}




----------------------------------------------------------------------
Get status for job domainRuntime/deploymentManager/deploymentProgressObjects/basicapp
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
deploymentProgressObjects/basicapp?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "deploymentManager",
        "deploymentProgressObjects",
        "basicapp"
    ],
    "rootExceptions": [],
    "endTimeAsLong": 1726603190343,
    "deploymentMessages": [
        "[Deployer:149191]Operation \"deploy\" on application \"basicapp\" is
initializing on \"Cluster1Server2\".",
        "[Deployer:149191]Operation \"deploy\" on application \"basicapp\" is
initializing on \"Cluster1Server1\".",
        "[Deployer:149191]Operation \"deploy\" on application \"basicapp\" is
initializing on \"Cluster1Server2\".",
        "[Deployer:149191]Operation \"deploy\" on application \"basicapp\" is
initializing on \"Cluster1Server1\".",
        "[Deployer:149192]Operation \"deploy\" on application \"basicapp\" is in
progress on \"Cluster1Server1\".",
        "[Deployer:149192]Operation \"deploy\" on application \"basicapp\" is in
progress on \"Cluster1Server2\".",
        "[Deployer:149194]Operation \"deploy\" on application \"basicapp\" has succeeded
on \"Cluster1Server1\".",
        "[Deployer:149194]Operation \"deploy\" on application \"basicapp\" has succeeded
on \"Cluster1Server2\"."
    ],
    "name": "basicapp",
    "operationType": 3,
    "startTimeAsLong": 1726603189541,
    "state": "STATE_COMPLETED",
    "id": "1",
    "type": "DeploymentProgressObject",
    "targets": ["Cluster1"],
    "applicationName": "basicapp",
    "failedTargets": [],
    "progress": "success",
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T15:59:49.541-04:00",
    "endTime": "2024-09-17T15:59:50.343-04:00"
}




----------------------------------------------------------------------
View the new applications' configurations
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/appDeployments/fairShare?
links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "appDeployments",
        "fairShare"
    ],
    "stagingMode": null,
    "absoluteSourcePath": "\/deployments\/fairShare.war",
    "notes": null,
    "absoluteAltDescriptorPath": null,
    "deploymentOrder": 100,
    "type": "AppDeployment",
    "installDir": null,
    "id": 0,
    "altDescriptorDir": null,
    "dynamicallyCreated": false,
    "sourcePath": "\/deployments\/fairShare.war",
    "applicationName": "fairShare",
    "absoluteAltDescriptorDir": null,
    "moduleType": "war",
    "planStagingMode": null,
    "cacheInAppDirectory": false,
    "absoluteInstallDir": null,
    "compatibilityName": null,
    "absolutePlanPath": null,
    "untargeted": false,
    "planDir": null,
    "validateDDSecurityData": false,
    "applicationIdentifier": "fairShare",
    "tags": [],
    "planPath": null,
    "versionIdentifier": null,
    "deploymentPrincipalName": null,
    "absolutePlanDir": null,
    "name": "fairShare",
    "parallelDeployModules": false,
    "securityDDModel": "DDOnly",
    "targets": [{
        "identity": [
            "clusters",
            "Cluster1"
        ]
    }]
}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/edit/appDeployments/basicapp?
links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "appDeployments",
        "basicapp"
    ],
    "stagingMode": null,
    "absoluteSourcePath": "\/domains\/mydomain\/servers\/AdminServer\/upload\/basicapp\/
app\/BasicApp.ear",
    "notes": null,
    "absoluteAltDescriptorPath": null,
    "deploymentOrder": 100,
    "type": "AppDeployment",
    "installDir": null,
    "id": 0,
    "altDescriptorDir": null,
    "dynamicallyCreated": false,
    "sourcePath": "servers\/AdminServer\/upload\/basicapp\/app\/BasicApp.ear",
    "applicationName": "basicapp",
    "absoluteAltDescriptorDir": null,
    "moduleType": "ear",
    "planStagingMode": null,
    "cacheInAppDirectory": false,
    "absoluteInstallDir": null,
    "compatibilityName": null,
    "absolutePlanPath": "\/domains\/mydomain\/servers\/AdminServer\/upload\/basicapp\/
plan\/Plan.xml",
    "untargeted": false,
    "planDir": null,
    "validateDDSecurityData": false,
    "applicationIdentifier": "basicapp",
    "tags": [],
    "planPath": "servers\/AdminServer\/upload\/basicapp\/plan\/Plan.xml",
    "versionIdentifier": null,
    "deploymentPrincipalName": null,
    "absolutePlanDir": null,
    "name": "basicapp",
    "parallelDeployModules": false,
    "securityDDModel": "DDOnly",
    "targets": [{
        "identity": [
            "clusters",
            "Cluster1"
        ]
    }]
}




----------------------------------------------------------------------
View the new applications' appDeploymentRuntimes
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
appDeploymentRuntimes/fairShare?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "deploymentManager",
        "appDeploymentRuntimes",
        "fairShare"
    ],
    "applicationVersion": null,
    "name": "fairShare",
    "type": "AppDeploymentRuntime",
    "applicationName": "fairShare",
    "modules": ["fairShare"]
}


curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
appDeploymentRuntimes/basicapp?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "deploymentManager",
        "appDeploymentRuntimes",
        "basicapp"
    ],
    "applicationVersion": null,
    "name": "basicapp",
    "type": "AppDeploymentRuntime",
    "applicationName": "basicapp",
    "modules": [
        "BasicAuth",
        "BasicEJB.jar"
    ]
}




----------------------------------------------------------------------
Search for all of the new applications' applicationRuntimes
----------------------------------------------------------------------

curl -v \
--user deployer:deployer123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      links: [], fields: [ 'name' ],
      children: {
        applicationRuntimes: {
          links: [],
          name: [ 'fairShare', 'basicapp' ]
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "applicationRuntimes": {
                    "items": [
                        {
                            "identity": [
                                "applicationRuntimes",
                                "fairShare"
                            ],
                            "applicationVersion": null,
                            "internal": false,
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "fairShare",
                            "type": "ApplicationRuntime",
                            "activeVersionState": 2,
                            "applicationName": "fairShare"
                        },
                        {
                            "identity": [
                                "applicationRuntimes",
                                "basicapp"
                            ],
                            "applicationVersion": null,
                            "internal": false,
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "basicapp",
                            "type": "ApplicationRuntime",
                            "activeVersionState": 2,
                            "applicationName": "basicapp"
                        }
                    ]
                }
            },
            {
                "name": "AdminServer",
                "applicationRuntimes": {
                    "items": []
                }
            },
            {
                "name": "Cluster1Server2",
                "applicationRuntimes": {
                    "items": [
                        {
                            "identity": [
                                "applicationRuntimes",
                                "fairShare"
                            ],
                            "applicationVersion": null,
                            "internal": false,
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "fairShare",
                            "type": "ApplicationRuntime",
                            "activeVersionState": 2,
                            "applicationName": "fairShare"
                        },
                        {
                            "identity": [
                                "applicationRuntimes",
                                "basicapp"
                            ],
                            "applicationVersion": null,
                            "internal": false,
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "basicapp",
                            "type": "ApplicationRuntime",
                            "activeVersionState": 2,
                            "applicationName": "basicapp"
                        }
                    ]
                }
            }
        ]
    }
}
```

Monitoring Domain Resources

Review an example script that demonstrates how an Operator monitors the entire domain.

The example script also shows how to monitor data sources and JMS, capture and download diagnostic images, search logs, and return consolidated search results.

Note:

To view long URLs, use the scroll bar located beneath the section.

```
----------------------------------------------------------------------
Demonstrate a domain monitor monitoring the domain
----------------------------------------------------------------------



----------------------------------------------------------------------
Monitor the servers
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/serverRuntimes?
links=none&fields=name,overallHealthState,healthState,state,openSocketsCurrentCount,activ
ationTime


HTTP/1.1 200 OK

Response Body:
{
    "items": [
        {
            "overallHealthState": {
                "state": "ok",
                "subsystemName": null,
                "symptoms": []
            },
            "state": "RUNNING",
            "activationTime": 1726603152406,
            "openSocketsCurrentCount": 4,
            "healthState": {
                "state": "ok",
                "subsystemName": null,
                "symptoms": []
            },
            "name": "Cluster1Server1"
        },
        {
            "overallHealthState": {
                "state": "ok",
                "subsystemName": null,
                "symptoms": []
            },
            "state": "RUNNING",
            "activationTime": 1726603094766,
            "openSocketsCurrentCount": 9,
            "healthState": {
                "state": "ok",
                "subsystemName": null,
                "symptoms": []
            },
            "name": "AdminServer"
        },
        {
            "overallHealthState": {
                "state": "ok",
                "subsystemName": null,
                "symptoms": []
            },
            "state": "RUNNING",
            "activationTime": 1726603174108,
            "openSocketsCurrentCount": 4,
            "healthState": {
                "state": "ok",
                "subsystemName": null,
                "symptoms": []
            },
            "name": "Cluster1Server2"
        }
    ]
}
```

```
----------------------------------------------------------------------
Get the number of open sockets of the server that has the highest number of open sockets
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      fields: [ { name: 'openSocketsCurrentCount', max:true } ]
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [{
            "openSocketsCurrentCount": {
                "max": 7
            }
        }]
    }
}




----------------------------------------------------------------------
Get the total number of open sockets of the running servers
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      fields: [ { name: 'openSocketsCurrentCount', total:true } ]
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search
```

```
HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [{
            "openSocketsCurrentCount": {
                "total": 15,
                "count": 3
            }
        }]
    }
}





----------------------------------------------------------------------
Get a list of the running servers' overall health states
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      fields: [ { name: 'overallHealthState', values: true } ]
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [{
            "overallHealthState": {
                "values": [
                    {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    }
                ]
            }
        }]
    }
}






----------------------------------------------------------------------
Get a list of the running servers' JVM statistics and links except for the
threadStackDump and processCpuLoad fields and the parent and canonical links
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  'links': [], 'fields': [],
  'children': {
    'serverRuntimes': {
      'links': [], 'fields': [ 'name' ],
      'children': {
        'JVMRuntime': {
          'excludeFields': [ 'threadStackDump', 'processCpuLoad' ],
          'excludeLinks': [ 'parent', 'canonical' ]
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "JVMRuntime": {
                    "links": [
                        {
                            "rel": "self",
                            "href": "http:\/\/localhost:7001\/management\/weblogic\/
latest\/domainRuntime\/serverRuntimes\/Cluster1Server1\/JVMRuntime"
                        },
                        {
                            "rel": "action",
                            "href": "http:\/\/localhost:7001\/management\/weblogic\/
latest\/domainRuntime\/serverRuntimes\/Cluster1Server1\/JVMRuntime\/runGC",
                            "title": "runGC"
                        }
                    ],
                    "identity": ["JVMRuntime"],
                    "javaVersion": "17.0.12",
                    "javaVMVendor": "Oracle Corporation",
                    "OSName": "Mac OS X",
                    "javaVendor": "Oracle Corporation",
                    "type": "JVMRuntime",
                    "uptime": 58247,
                    "heapSizeCurrent": 268435456,
                    "heapFreeCurrent": 65370816,
                    "name": "Cluster1Server1",
                    "javaVendorVersion": null,
                    "OSVersion": "14.6.1",
                    "heapSizeMax": 536870912,
                    "heapFreePercent": 62
                }
            },
            {
                "name": "AdminServer",
                "JVMRuntime": {
                    "links": [
                        {
                            "rel": "self",
                            "href": "http:\/\/localhost:7001\/management\/weblogic\/
latest\/domainRuntime\/serverRuntimes\/AdminServer\/JVMRuntime"
                        },
                        {
                            "rel": "action",
                            "href": "http:\/\/localhost:7001\/management\/weblogic\/
latest\/domainRuntime\/serverRuntimes\/AdminServer\/JVMRuntime\/runGC",
                            "title": "runGC"
                        }
                    ],
                    "identity": ["JVMRuntime"],
                    "javaVersion": "17.0.12",
                    "javaVMVendor": "Oracle Corporation",
                    "OSName": "Mac OS X",
                    "javaVendor": "Oracle Corporation",
                    "type": "JVMRuntime",
                    "uptime": 115868,
                    "heapSizeCurrent": 309329920,
                    "heapFreeCurrent": 119229200,
                    "name": "AdminServer",
                    "javaVendorVersion": null,
                    "OSVersion": "14.6.1",
                    "heapSizeMax": 536870912,
                    "heapFreePercent": 64
                }
            },
            {
                "name": "Cluster1Server2",
                "JVMRuntime": {
                    "links": [
                        {
                            "rel": "self",
                            "href": "http:\/\/localhost:7001\/management\/weblogic\/
latest\/domainRuntime\/serverRuntimes\/Cluster1Server2\/JVMRuntime"
                        },
                        {
                            "rel": "action",
                            "href": "http:\/\/localhost:7001\/management\/weblogic\/
latest\/domainRuntime\/serverRuntimes\/Cluster1Server2\/JVMRuntime\/runGC",
                            "title": "runGC"
                        }
                    ],
                    "identity": ["JVMRuntime"],
                    "javaVersion": "17.0.12",
                    "javaVMVendor": "Oracle Corporation",
                    "OSName": "Mac OS X",
                    "javaVendor": "Oracle Corporation",
                    "type": "JVMRuntime",
                    "uptime": 36545,
                    "heapSizeCurrent": 268435456,
                    "heapFreeCurrent": 80823592,
                    "name": "Cluster1Server2",
                    "javaVendorVersion": null,
                    "OSVersion": "14.6.1",
                    "heapSizeMax": 536870912,
                    "heapFreePercent": 65
                }
            }
        ]
    }
}




----------------------------------------------------------------------
Get the admin server's JVM statistics and links except for the threadStackDump and
processCpuLoad fields and the parent and canonical links
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/serverRuntime/JVMRuntime?
excludeFields=threadStackDump,processCpuLoad&excludeLinks=parent,canonical


HTTP/1.1 200 OK

Response Body:
{
    "links": [
        {
            "rel": "self",
            "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/
serverRuntime\/JVMRuntime"
        },
        {
            "rel": "action",
            "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/
serverRuntime\/JVMRuntime\/runGC",
            "title": "runGC"
        }
    ],
    "identity": ["JVMRuntime"],
    "javaVersion": "17.0.12",
    "javaVMVendor": "Oracle Corporation",
    "OSName": "Mac OS X",
    "javaVendor": "Oracle Corporation",
    "type": "JVMRuntime",
    "uptime": 116095,
    "heapSizeCurrent": 309329920,
    "heapFreeCurrent": 111208976,
    "name": "AdminServer",
    "javaVendorVersion": null,
    "OSVersion": "14.6.1",
    "heapSizeMax": 536870912,
    "heapFreePercent": 63
}




----------------------------------------------------------------------
Monitor the JDBC system resources
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      links: [], fields: [ 'name' ],
      children: {
        JDBCServiceRuntime: {
          links: [], fields: [ 'name' ],
          children: {
            JDBCDataSourceRuntimeMBeans : { links: [], excludeFields: [ 'properties' ] }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "JDBCServiceRuntime": {
                    "name": "Cluster1Server1",
                    "JDBCDataSourceRuntimeMBeans": {
                        "items": [{
                            "identity": [
                                "JDBCServiceRuntime",
                                "JDBCDataSourceRuntimeMBeans",
                                "JDBCDataSource1"
                            ],
                            "connectionsTotalCount": 1,
                            "waitingForConnectionSuccessTotal": 0,
                            "highestNumUnavailable": 0,
                            "reserveRequestCount": 0,
                            "type": "JDBCDataSourceRuntime",
                            "waitingForConnectionTotal": 0,
                            "enabled": true,
                            "currCapacityHighCount": 1,
                            "resolvedAsNotCommittedTotalCount": 0,
                            "prepStmtCacheHitCount": 0,
                            "prepStmtCacheMissCount": 0,
                            "databaseProductName": "Apache Derby",
                            "commitOutcomeRetryTotalCount": 0,
                            "failedRepurposeCount": 0,
                            "state": "Running",
                            "moduleId": "JDBCDataSource1",
                            "prepStmtCacheAddCount": 0,
                            "failuresToReconnectCount": 0,
                            "repurposeCount": 0,
                            "databaseProductVersion": "10.16.1.1 - (1905560-wls)",
                            "leakedConnectionCount": 0,
                            "waitingForConnectionFailureTotal": 0,
                            "activeConnectionsHighCount": 1,
                            "connectionDelayTime": 782,
                            "waitingForConnectionHighCount": 0,
                            "waitSecondsHighCount": 0,
                            "versionJDBCDriver":
"org.apache.derby.jdbc.ClientXADataSource",
                            "failedReserveRequestCount": 0,
                            "prepStmtCacheDeleteCount": 0,
                            "numAvailable": 1,
                            "deploymentState": 2,
                            "unresolvedTotalCount": 0,
                            "prepStmtCacheAccessCount": 0,
                            "driverVersion": "10.16.1.1 - (1905560-wls)",
                            "resolvedAsCommittedTotalCount": 0,
                            "prepStmtCacheCurrentSize": 0,
                            "name": "JDBCDataSource1",
                            "activeConnectionsCurrentCount": 0,
                            "currCapacity": 1,
                            "driverName": "Apache Derby Network Client JDBC Driver",
                            "activeConnectionsAverageCount": 0,
                            "numUnavailable": 0,
                            "waitingForConnectionCurrentCount": 0,
                            "highestNumAvailable": 1,
                            "lastTask": null
                        }]
                    }
                }
            },
            {
                "name": "AdminServer",
                "JDBCServiceRuntime": {
                    "name": "AdminServer",
                    "JDBCDataSourceRuntimeMBeans": {
                        "items": []
                    }
                }
            },
            {
                "name": "Cluster1Server2",
                "JDBCServiceRuntime": {
                    "name": "Cluster1Server2",
                    "JDBCDataSourceRuntimeMBeans": {
                        "items": [{
                            "identity": [
                                "JDBCServiceRuntime",
                                "JDBCDataSourceRuntimeMBeans",
                                "JDBCDataSource1"
                            ],
                            "connectionsTotalCount": 1,
                            "waitingForConnectionSuccessTotal": 0,
                            "highestNumUnavailable": 0,
                            "reserveRequestCount": 0,
                            "type": "JDBCDataSourceRuntime",
                            "waitingForConnectionTotal": 0,
                            "enabled": true,
                            "currCapacityHighCount": 1,
                            "resolvedAsNotCommittedTotalCount": 0,
                            "prepStmtCacheHitCount": 0,
                            "prepStmtCacheMissCount": 0,
                            "databaseProductName": "Apache Derby",
                            "commitOutcomeRetryTotalCount": 0,
                            "failedRepurposeCount": 0,
                            "state": "Running",
                            "moduleId": "JDBCDataSource1",
                            "prepStmtCacheAddCount": 0,
                            "failuresToReconnectCount": 0,
                            "repurposeCount": 0,
                            "databaseProductVersion": "10.16.1.1 - (1905560-wls)",
                            "leakedConnectionCount": 0,
                            "waitingForConnectionFailureTotal": 0,
                            "activeConnectionsHighCount": 1,
                            "connectionDelayTime": 578,
                            "waitingForConnectionHighCount": 0,
                            "waitSecondsHighCount": 0,
                            "versionJDBCDriver":
"org.apache.derby.jdbc.ClientXADataSource",
                            "failedReserveRequestCount": 0,
                            "prepStmtCacheDeleteCount": 0,
                            "numAvailable": 1,
                            "deploymentState": 2,
                            "unresolvedTotalCount": 0,
                            "prepStmtCacheAccessCount": 0,
                            "driverVersion": "10.16.1.1 - (1905560-wls)",
                            "resolvedAsCommittedTotalCount": 0,
                            "prepStmtCacheCurrentSize": 0,
                            "name": "JDBCDataSource1",
                            "activeConnectionsCurrentCount": 0,
                            "currCapacity": 1,
                            "driverName": "Apache Derby Network Client JDBC Driver",
                            "activeConnectionsAverageCount": 0,
                            "numUnavailable": 0,
                            "waitingForConnectionCurrentCount": 0,
                            "highestNumAvailable": 1,
                            "lastTask": null
                        }]
                    }
                }
            }
        ]
    }
}
```

```
----------------------------------------------------------------------
Test a domain level data source
```

```
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/serverRuntimes/
Cluster1Server2/JDBCServiceRuntime/JDBCDataSourceRuntimeMBeans/JDBCDataSource1/testPool


HTTP/1.1 200 OK

Response Body:
{
    "return": null
}





----------------------------------------------------------------------
Monitor the JMS system resources
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      links: [], fields: [ 'name' ],
      children: {
        JMSRuntime: {
          links: [],
          children: {
            JMSServers: {
              links: [],
              children: {
                destinations: {
                  links: [],
                }
              }
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "JMSRuntime": {
                    "identity": ["JMSRuntime"],
                    "JMSServersHighCount": 1,
                    "connectionsHighCount": 0,
                    "connectionsTotalCount": 0,
                    "healthState": {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    "name": "Cluster1Server1.jms",
                    "JMSServersCurrentCount": 1,
                    "type": "JMSRuntime",
                    "JMSServersTotalCount": 1,
                    "connectionsCurrentCount": 0,
                    "JMSServers": {
                        "items": [{
                            "identity": [
                                "JMSRuntime",
                                "JMSServers",
                                "JMSServer1@Cluster1Server1"
                            ],
                            "messagesReceivedCount": 0,
                            "bytesThresholdTime": 0,
                            "bytesHighCount": 0,
                            "insertionPausedState": "Insertion-Enabled",
                            "destinationsCurrentCount": 1,
                            "pagingAllocatedIoBufferBytes": 0,
                            "type": "JMSServerRuntime",
                            "messagesPagedInTotalCount": 0,
                            "consumptionPaused": false,
                            "bytesPagedInTotalCount": 0,
                            "pagingPhysicalWriteCount": 0,
                            "pendingTransactions": null,
                            "bytesPagedOutTotalCount": 0,
                            "consumptionPausedState": "Consumption-Enabled",
                            "sessionPoolsHighCount": 0,
                            "bytesReceivedCount": 0,
                            "messagesHighCount": 0,
                            "productionPausedState": "Production-Enabled",
                            "pagingAllocatedWindowBufferBytes": 0,
                            "destinationsTotalCount": 1,
                            "sessionPoolsTotalCount": 0,
                            "messagesThresholdTime": 0,
                            "bytesCurrentCount": 0,
                            "transactions": null,
                            "messagesPagedOutTotalCount": 0,
                            "messagesCurrentCount": 0,
                            "destinationsHighCount": 1,
                            "insertionPaused": false,
                            "healthState": {
                                "state": "ok",
                                "subsystemName": "JMSServer.JMSServer1@Cluster1Server1",
                                "symptoms": []
                            },
                            "messagesPageableCurrentCount": 0,
                            "sessionPoolsCurrentCount": 0,
                            "name": "JMSServer1@Cluster1Server1",
                            "bytesPendingCount": 0,
                            "productionPaused": false,
                            "bytesPageableCurrentCount": 0,
                            "messagesPendingCount": 0,
                            "destinations": {
                                "items": [{
                                    "identity": [
                                        "JMSRuntime",
                                        "JMSServers",
                                        "JMSServer1@Cluster1Server1",
                                        "destinations",
                                        "JMSSystemResource1!
JMSServer1@Cluster1Server1@UniformDistributedQueue1"
                                    ],
                                    "messagesReceivedCount": 0,
                                    "bytesThresholdTime": 0,
                                    "bytesHighCount": 0,
                                    "insertionPausedState": "Insertion-Enabled",
                                    "type": "JMSDestinationRuntime",
                                    "consumptionPaused": false,
                                    "messagesDeletedCurrentCount": 0,
                                    "destinationType": "Queue",
                                    "consumptionPausedState": "Consumption-Enabled",
                                    "state": "started",
                                    "bytesReceivedCount": 0,
                                    "messagesHighCount": 0,
                                    "productionPausedState": "Production-Enabled",
                                    "subscriptionMessagesLimit": -1,
                                    "consumersTotalCount": 0,
                                    "consumersHighCount": 0,
                                    "messagesThresholdTime": 0,
                                    "bytesCurrentCount": 0,
                                    "messagesMovedCurrentCount": 0,
                                    "destinationID":
"rO0ABXNyACN3ZWJsb2dpYy5qbXMuY29tbW9uLkRlc3RpbmF0aW9uSW1wbFSmyJ1qZfv8DAAAeHB38bhBAEZKTVNT
eXN0ZW1SZXNvdXJjZTEhSk1TU2VydmVyMUBDbHVzdGVyMVNlcnZlcjFAVW5pZm9ybURpc3RyaWJ1dGVkUXVldWUxA
BpKTVNTZXJ2ZXIxQENsdXN0ZXIxU2VydmVyMQASSk1TU3lzdGVtUmVzb3VyY2UxAQADQWxsAgJ\/
dUGSAZEbXwAAAAwBAA9DbHVzdGVyMVNlcnZlcjH171sHAn91QZIBkRtfAAAADgEAD0NsdXN0ZXIxU2VydmVyMfXvW
wcAAwAaRmlsZVN0b3JlMUBDbHVzdGVyMVNlcnZlcjEACkpNU1NlcnZlcjF4",
                                    "messagesCurrentCount": 0,
                                    "insertionPaused": false,
                                    "name": "JMSSystemResource1!
JMSServer1@Cluster1Server1@UniformDistributedQueue1",
                                    "bytesPendingCount": 0,
                                    "productionPaused": false,
                                    "messagesPendingCount": 0,
                                    "consumersCurrentCount": 0
                                }]
                            }
                        }]
                    }
                }
            },
            {
                "name": "AdminServer",
                "JMSRuntime": {
                    "identity": ["JMSRuntime"],
                    "JMSServersHighCount": 0,
                    "connectionsHighCount": 0,
                    "connectionsTotalCount": 0,
                    "healthState": {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    "name": "AdminServer.jms",
                    "JMSServersCurrentCount": 0,
                    "type": "JMSRuntime",
                    "JMSServersTotalCount": 0,
                    "connectionsCurrentCount": 0,
                    "JMSServers": {
                        "items": []
                    }
                }
            },
            {
                "name": "Cluster1Server2",
                "JMSRuntime": {
                    "identity": ["JMSRuntime"],
                    "JMSServersHighCount": 1,
                    "connectionsHighCount": 0,
                    "connectionsTotalCount": 0,
                    "healthState": {
                        "state": "ok",
                        "subsystemName": null,
                        "symptoms": []
                    },
                    "name": "Cluster1Server2.jms",
                    "JMSServersCurrentCount": 1,
                    "type": "JMSRuntime",
                    "JMSServersTotalCount": 1,
                    "connectionsCurrentCount": 0,
                    "JMSServers": {
                        "items": [{
                            "identity": [
                                "JMSRuntime",
                                "JMSServers",
                                "JMSServer1@Cluster1Server2"
                            ],
                            "messagesReceivedCount": 0,
                            "bytesThresholdTime": 0,
                            "bytesHighCount": 0,
                            "insertionPausedState": "Insertion-Enabled",
                            "destinationsCurrentCount": 1,
                            "pagingAllocatedIoBufferBytes": 0,
                            "type": "JMSServerRuntime",
                            "messagesPagedInTotalCount": 0,
                            "consumptionPaused": false,
                            "bytesPagedInTotalCount": 0,
                            "pagingPhysicalWriteCount": 0,
                            "pendingTransactions": null,
                            "bytesPagedOutTotalCount": 0,
                            "consumptionPausedState": "Consumption-Enabled",
                            "sessionPoolsHighCount": 0,
                            "bytesReceivedCount": 0,
                            "messagesHighCount": 0,
                            "productionPausedState": "Production-Enabled",
                            "pagingAllocatedWindowBufferBytes": 0,
                            "destinationsTotalCount": 1,
                            "sessionPoolsTotalCount": 0,
                            "messagesThresholdTime": 0,
                            "bytesCurrentCount": 0,
                            "transactions": null,
                            "messagesPagedOutTotalCount": 0,
                            "messagesCurrentCount": 0,
                            "destinationsHighCount": 1,
                            "insertionPaused": false,
                            "healthState": {
                                "state": "ok",
                                "subsystemName": "JMSServer.JMSServer1@Cluster1Server2",
                                "symptoms": []
                            },
                            "messagesPageableCurrentCount": 0,
                            "sessionPoolsCurrentCount": 0,
                            "name": "JMSServer1@Cluster1Server2",
                            "bytesPendingCount": 0,
                            "productionPaused": false,
                            "bytesPageableCurrentCount": 0,
                            "messagesPendingCount": 0,
                            "destinations": {
                                "items": [{
                                    "identity": [
                                        "JMSRuntime",
                                        "JMSServers",
                                        "JMSServer1@Cluster1Server2",
                                        "destinations",
                                        "JMSSystemResource1!
JMSServer1@Cluster1Server2@UniformDistributedQueue1"
                                    ],
                                    "messagesReceivedCount": 0,
                                    "bytesThresholdTime": 0,
                                    "bytesHighCount": 0,
                                    "insertionPausedState": "Insertion-Enabled",
                                    "type": "JMSDestinationRuntime",
                                    "consumptionPaused": false,
                                    "messagesDeletedCurrentCount": 0,
                                    "destinationType": "Queue",
                                    "consumptionPausedState": "Consumption-Enabled",
                                    "state": "started",
                                    "bytesReceivedCount": 0,
                                    "messagesHighCount": 0,
                                    "productionPausedState": "Production-Enabled",
                                    "subscriptionMessagesLimit": -1,
                                    "consumersTotalCount": 0,
                                    "consumersHighCount": 0,
                                    "messagesThresholdTime": 0,
                                    "bytesCurrentCount": 0,
                                    "messagesMovedCurrentCount": 0,
                                    "destinationID":
"rO0ABXNyACN3ZWJsb2dpYy5qbXMuY29tbW9uLkRlc3RpbmF0aW9uSW1wbFSmyJ1qZfv8DAAAeHB38bhBAEZKTVNT
eXN0ZW1SZXNvdXJjZTEhSk1TU2VydmVyMUBDbHVzdGVyMVNlcnZlcjJAVW5pZm9ybURpc3RyaWJ1dGVkUXVldWUxA
BpKTVNTZXJ2ZXIxQENsdXN0ZXIxU2VydmVyMgASSk1TU3lzdGVtUmVzb3VyY2UxAQADQWxsAgJPYyGSAZGHcAAAAA
wBAA9DbHVzdGVyMVNlcnZlcjIw9USGAk9jIZIBkYdwAAAADgEAD0NsdXN0ZXIxU2VydmVyMjD1RIYAAwAaRmlsZVN
0b3JlMUBDbHVzdGVyMVNlcnZlcjIACkpNU1NlcnZlcjF4",
                                    "messagesCurrentCount": 0,
                                    "insertionPaused": false,
                                    "name": "JMSSystemResource1!
JMSServer1@Cluster1Server2@UniformDistributedQueue1",
                                    "bytesPendingCount": 0,
                                    "productionPaused": false,
                                    "messagesPendingCount": 0,
                                    "consumersCurrentCount": 0
                                }]
                            }
                        }]
                    }
                }
            }
```

```
        ]
    }
}




----------------------------------------------------------------------
Monitor the applications
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      links: [], fields: [ 'name' ],
      children: {
        applicationRuntimes: {
          links: [], fields: [ 'name', 'healthState', 'overallHealthState' ]
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "applicationRuntimes": {
                    "items": [
                        {
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "bea_wls_deployment_internal"
                        },
                        {
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
        "healthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "name": "basicapp"
    },
    {
        "overallHealthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "healthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "name": "bea_wls_cluster_internal"
    },
    {
        "overallHealthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "healthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "name": "JMSSystemResource1"
    },
    {
        "overallHealthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "healthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "name": "bea_wls_internal"
    },
    {
        "overallHealthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "healthState": {
            "state": "ok",
            "subsystemName": null,
            "symptoms": []
        },
        "name": "jms-internal-xa-adp"
    },
    {
        "overallHealthState": {
            "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "wls-management-services"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "fairShare"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "jms-internal-notran-adp"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "JDBCDataSource1"
            }
        ]
    }
},
{
    "name": "AdminServer",
    "applicationRuntimes": {
        "items": [
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "jms-internal-xa-adp"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "bea_wls_deployment_internal"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "mejb"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "wls-management-services"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "bea_wls_internal"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "jms-internal-notran-adp"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "bea_wls_management_internal2"
            }
        ]
    }
},
{
    "name": "Cluster1Server2",
    "applicationRuntimes": {
        "items": [
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "jms-internal-notran-adp"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "bea_wls_cluster_internal"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "JDBCDataSource1"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "JMSSystemResource1"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "fairShare"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "basicapp"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "healthState": {
                    "state": "ok",
                    "subsystemName": null,
                    "symptoms": []
                },
                "name": "bea_wls_deployment_internal"
            },
            {
                "overallHealthState": {
                    "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "bea_wls_internal"
                        },
                        {
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "wls-management-services"
                        },
                        {
                            "overallHealthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "healthState": {
                                "state": "ok",
                                "subsystemName": null,
                                "symptoms": []
                            },
                            "name": "jms-internal-xa-adp"
                        }
                    ]
                }
            }
        ]
    }
}



----------------------------------------------------------------------
Monitor the applications' servlets
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      links: [], fields: [ 'name' ],
      children: {
        applicationRuntimes: {
          links: [], fields: [ 'name' ],
          name: [ 'fairShare', 'basicapp' ],
          children: {
            componentRuntimes: {
              links: [], fields: [ 'name', 'type' ],
              children: {
                servlets: {
                  links: [],
                  fields: [
                    'name',
                    'executionTimeHigh',
                    'executionTimeLow',
                    'executionTimeAverage',
                    'invocationTotalCount'
                  ]
                }
              }
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [
            {
                "name": "Cluster1Server1",
                "applicationRuntimes": {
                    "items": [
                        {
                            "name": "fairShare",
                            "componentRuntimes": {
                                "items": [{
                                    "type": "WebAppComponentRuntime",
                                    "name": "Cluster1Server1_\/fairShare",
                                    "servlets": {
                                        "items": [
                                            {
                                                "executionTimeHigh": 0,
                                                "invocationTotalCount": 0,
                                                "executionTimeLow": 0,
                                                "name": "JspServlet",
                                                "executionTimeAverage": 0
                                            },
                                            {
                                                "executionTimeHigh": 0,
                                                "invocationTotalCount": 0,
                                                "executionTimeLow": 0,
                                                "name": "_WL_FileServlet",
                                                "executionTimeAverage": 0
                                            },
                                            {
                                                "executionTimeHigh": 0,
                        "invocationTotalCount": 0,
                        "executionTimeLow": 0,
                        "name": "SimpleFastServlet",
                        "executionTimeAverage": 0
                    },
                    {
                        "executionTimeHigh": 0,
                        "invocationTotalCount": 0,
                        "executionTimeLow": 0,
                        "name": "SimpleSlowServlet",
                        "executionTimeAverage": 0
                    }
                ]
            }
        }]
    }
},
{
    "name": "basicapp",
    "componentRuntimes": {
        "items": [
            {
                "name": "BasicEJB.jar",
                "type": "EJBComponentRuntime"
            },
            {
                "type": "WebAppComponentRuntime",
                "name": "Cluster1Server1_\/BasicAuth",
                "servlets": {
                    "items": [
                        {
                            "executionTimeHigh": 0,
                            "invocationTotalCount": 0,
                            "executionTimeLow": 0,
                            "name": "JspServlet",
                            "executionTimeAverage": 0
                        },
                        {
                            "executionTimeHigh": 0,
                            "invocationTotalCount": 0,
                            "executionTimeLow": 0,
                            "name": "Servlet3",
                            "executionTimeAverage": 0
                        },
                        {
                            "executionTimeHigh": 0,
                            "invocationTotalCount": 0,
                            "executionTimeLow": 0,
                            "name": "_WL_FileServlet",
                            "executionTimeAverage": 0
                        },
                        {
                            "executionTimeHigh": 0,
                            "invocationTotalCount": 0,
                            "executionTimeLow": 0,
                            "name": "Servlet2",
                            "executionTimeAverage": 0
                        },
                        {
                            "executionTimeHigh": 0,
                            "invocationTotalCount": 0,
                            "executionTimeLow": 0,
                                        "name": "Servlet1",
                                        "executionTimeAverage": 0
                                    }
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    }
},
{
    "name": "AdminServer",
    "applicationRuntimes": {
        "items": []
    }
},
{
    "name": "Cluster1Server2",
    "applicationRuntimes": {
        "items": [
            {
                "name": "fairShare",
                "componentRuntimes": {
                    "items": [{
                        "type": "WebAppComponentRuntime",
                        "name": "Cluster1Server2_\/fairShare",
                        "servlets": {
                            "items": [
                                {
                                    "executionTimeHigh": 0,
                                    "invocationTotalCount": 0,
                                    "executionTimeLow": 0,
                                    "name": "JspServlet",
                                    "executionTimeAverage": 0
                                },
                                {
                                    "executionTimeHigh": 0,
                                    "invocationTotalCount": 0,
                                    "executionTimeLow": 0,
                                    "name": "_WL_FileServlet",
                                    "executionTimeAverage": 0
                                },
                                {
                                    "executionTimeHigh": 0,
                                    "invocationTotalCount": 0,
                                    "executionTimeLow": 0,
                                    "name": "SimpleFastServlet",
                                    "executionTimeAverage": 0
                                },
                                {
                                    "executionTimeHigh": 0,
                                    "invocationTotalCount": 0,
                                    "executionTimeLow": 0,
                                    "name": "SimpleSlowServlet",
                                    "executionTimeAverage": 0
                                }
                            ]
                        }
                    }]
                }
```

```
                        },
                        {
                            "name": "basicapp",
                            "componentRuntimes": {
                                "items": [
                                    {
                                        "name": "BasicEJB.jar",
                                        "type": "EJBComponentRuntime"
                                    },
                                    {
                                        "type": "WebAppComponentRuntime",
                                        "name": "Cluster1Server2_\/BasicAuth",
                                        "servlets": {
                                            "items": [
                                                {
                                                    "executionTimeHigh": 0,
                                                    "invocationTotalCount": 0,
                                                    "executionTimeLow": 0,
                                                    "name": "JspServlet",
                                                    "executionTimeAverage": 0
                                                },
                                                {
                                                    "executionTimeHigh": 0,
                                                    "invocationTotalCount": 0,
                                                    "executionTimeLow": 0,
                                                    "name": "Servlet3",
                                                    "executionTimeAverage": 0
                                                },
                                                {
                                                    "executionTimeHigh": 0,
                                                    "invocationTotalCount": 0,
                                                    "executionTimeLow": 0,
                                                    "name": "_WL_FileServlet",
                                                    "executionTimeAverage": 0
                                                },
                                                {
                                                    "executionTimeHigh": 0,
                                                    "invocationTotalCount": 0,
                                                    "executionTimeLow": 0,
                                                    "name": "Servlet2",
                                                    "executionTimeAverage": 0
                                                },
                                                {
                                                    "executionTimeHigh": 0,
                                                    "invocationTotalCount": 0,
                                                    "executionTimeLow": 0,
                                                    "name": "Servlet1",
                                                    "executionTimeAverage": 0
                                                }
                                            ]
                                        }
                                    }
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

```
----------------------------------------------------------------------
Get the total number of open sessions across each application's component runtimes
across all servers
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      children: {
        applicationRuntimes: {
          mergeOn: 'name',
          fields: [ { name: 'name', sameValue: true } ],
          children: {
            componentRuntimes: {
              mergeCollection: true,
              fields: [ { name: 'openSessionsCurrentCount', total: true } ]
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [{
            "applicationRuntimes": {
                "items": [
                    {
                        "name": "bea_wls_deployment_internal",
                        "componentRuntimes": {
                            "items": [{
                                "openSessionsCurrentCount": {
                                    "total": 0,
                                    "count": 3
                                }
                            }]
                        }
                    },
                    {
                        "name": "basicapp",
                        "componentRuntimes": {
                            "items": [{
                                "openSessionsCurrentCount": {
                                    "total": 0,
                                    "count": 2
                                }
                            }]
                        }
                    },
                    {
                        "name": "bea_wls_cluster_internal",
                        "componentRuntimes": {
                            "items": [{
                                "openSessionsCurrentCount": {
                                    "total": 0,
                                    "count": 2
                                }
                            }]
                        }
                    },
                    {
                        "name": "JMSSystemResource1",
                        "componentRuntimes": {
                            "items": [{}]
                        }
                    },
                    {
                        "name": "bea_wls_internal",
                        "componentRuntimes": {
                            "items": [{
                                "openSessionsCurrentCount": {
                                    "total": 0,
                                    "count": 3
                                }
                            }]
                        }
                    },
                    {
                        "name": "jms-internal-xa-adp",
                        "componentRuntimes": {
                            "items": [{}]
                        }
                    },
                    {
                        "name": "wls-management-services",
                        "componentRuntimes": {
                            "items": [{
                                "openSessionsCurrentCount": {
                                    "total": 3,
                                    "count": 3
                                }
                            }]
                        }
                    },
                    {
                        "name": "fairShare",
                        "componentRuntimes": {
                            "items": [{
                                "openSessionsCurrentCount": {
                                    "total": 0,
                                    "count": 2
                                }
                            }]
                        }
                    },
                    {
```

```
                        "name": "jms-internal-notran-adp",
                        "componentRuntimes": {
                            "items": [{}]
                        }
                    },
                    {
                        "name": "JDBCDataSource1",
                        "componentRuntimes": {
                            "items": [{}]
                        }
                    },
                    {
                        "name": "mejb",
                        "componentRuntimes": {
                            "items": [{}]
                        }
                    },
                    {
                        "name": "bea_wls_management_internal2",
                        "componentRuntimes": {
                            "items": [{
                                "openSessionsCurrentCount": {
                                    "total": 0,
                                    "count": 1
                                }
                            }]
                        }
                    }
                ]
            }
        }]
    }
}




----------------------------------------------------------------------
Get the total number of invocations of the servlets of each component runtime of the
fairShare and wls-management-services applications across all servers
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      children: {
        applicationRuntimes: {
          name: [ 'fairShare', 'wls-management-services' ],
          mergeOn: 'name',
          fields: [ { name: 'name', sameValue: true } ],
          children: {
            componentRuntimes: {
              mergeOn: 'moduleId',
              fields: [ { name: 'contextRoot', sameValue: true } ],
              children: {
                servlets: {
                  mergeCollection: true,
                  fields: [ { name: 'invocationTotalCount', total: true } ]
                }
              }
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [{
            "applicationRuntimes": {
                "items": [
                    {
                        "name": "fairShare",
                        "componentRuntimes": {
                            "items": [{
                                "contextRoot": "\/fairShare",
                                "servlets": {
                                    "items": [{
                                        "invocationTotalCount": {
                                            "total": 0,
                                            "count": 8
                                        }
                                    }]
                                }
                            }]
                        }
                    },
                    {
                        "name": "wls-management-services",
                        "componentRuntimes": {
                            "items": [{
                                "contextRoot": "\/management",
                                "servlets": {
                                    "items": [{
                                        "invocationTotalCount": {
                                            "total": 167,
                                            "count": 9
                                        }
                                    }]
                                }
                            }]
                        }
                    }
                ]
            }
        }]
    }
}
```

```
----------------------------------------------------------------------
Get the information displayed by the console's webapp monitoring page
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  links: [], fields: [],
  children: {
    serverRuntimes: {
      mergeCollection: true,
      children: {
        applicationRuntimes: {
          mergeOn: 'name',
          fields: [
            { name: 'name',     sameValue: true },
            { name: 'internal', sameValue: true }
          ],
          children: {
            componentRuntimes: {
              mergeOn: 'moduleId',
              fields: [
                { name: 'contextRoot',               sameValue: true },
                { name: 'type',                      sameValue: true },
                { name: 'sourceInfo',                sameValue: true },
                { name: 'deploymentState',           values:    true },
                { name: 'openSessionsHighCount',     max:       true },
                { name: 'openSessionsCurrentCount',  total:     true },
                { name: 'sessionsOpenedTotalCount',  total:     true }
              ],
              children: {
                servlets: {
                  mergeCollection: true,
                  fields: [ { name: 'invocationTotalCount', total: true } ]
                }
              }
            }
          }
        }
      }
    }
  }
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/search


HTTP/1.1 200 OK

Response Body:
{
    "serverRuntimes": {
        "items": [{
            "applicationRuntimes": {
                "items": [
                    {
```

```
    "internal": true,
    "name": "bea_wls_deployment_internal",
    "componentRuntimes": {
        "items": [{
            "type": "WebAppComponentRuntime",
            "sourceInfo": "bea_wls_deployment_internal.war",
            "contextRoot": "\/bea_wls_deployment_internal",
            "openSessionsCurrentCount": {
                "total": 0,
                "count": 3
            },
            "deploymentState": {
                "values": [
                    2,
                    2,
                    2
                ]
            },
            "sessionsOpenedTotalCount": {
                "total": 0,
                "count": 3
            },
            "openSessionsHighCount": {
                "max": 0
            },
            "servlets": {
                "items": [{
                    "invocationTotalCount": {
                        "total": 60,
                        "count": 9
                    }
                }]
            }
        }]
    }
},
{
    "internal": false,
    "name": "basicapp",
    "componentRuntimes": {
        "items": [
            {
                "deploymentState": {
                    "values": [
                        2,
                        2
                    ]
                },
                "type": "EJBComponentRuntime"
            },
            {
                "type": "WebAppComponentRuntime",
                "sourceInfo": "BasicAuth.war",
                "contextRoot": "\/BasicAuth",
                "openSessionsCurrentCount": {
                    "total": 0,
                    "count": 2
                },
                "deploymentState": {
                    "values": [
                        2,
                        2
                    ]
                },
                "sessionsOpenedTotalCount": {
                    "total": 0,
                    "count": 2
                },
                "openSessionsHighCount": {
                    "max": 0
                },
                "servlets": {
                    "items": [{
                        "invocationTotalCount": {
                            "total": 0,
                            "count": 10
                        }
                    }]
                }
            }
        ]
    }
},
{
    "internal": true,
    "name": "bea_wls_cluster_internal",
    "componentRuntimes": {
        "items": [{
            "type": "WebAppComponentRuntime",
            "sourceInfo": "bea_wls_cluster_internal.war",
            "contextRoot": "\/bea_wls_cluster_internal",
            "openSessionsCurrentCount": {
                "total": 0,
                "count": 2
            },
            "deploymentState": {
                "values": [
                    2,
                    2
                ]
            },
            "sessionsOpenedTotalCount": {
                "total": 0,
                "count": 2
            },
            "openSessionsHighCount": {
                "max": 0
            },
            "servlets": {
                "items": [{
                    "invocationTotalCount": {
                        "total": 3,
                        "count": 10
                    }
                }]
            }
        }]
    }
},
{
    "internal": false,
    "name": "JMSSystemResource1",
    "componentRuntimes": {
        "items": [{
```

```
                    "deploymentState": {
                        "values": [
                            2,
                            2
                        ]
                    },
                    "type": "JMSComponentRuntime"
                }]
            }
        },
        {
            "internal": true,
            "name": "bea_wls_internal",
            "componentRuntimes": {
                "items": [{
                    "type": "WebAppComponentRuntime",
                    "sourceInfo": "bea_wls_internal.war",
                    "contextRoot": "\/bea_wls_internal",
                    "openSessionsCurrentCount": {
                        "total": 0,
                        "count": 3
                    },
                    "deploymentState": {
                        "values": [
                            2,
                            2,
                            2
                        ]
                    },
                    "sessionsOpenedTotalCount": {
                        "total": 0,
                        "count": 3
                    },
                    "openSessionsHighCount": {
                        "max": 0
                    },
                    "servlets": {
                        "items": [{
                            "invocationTotalCount": {
                                "total": 0,
                                "count": 33
                            }
                        }]
                    }
                }]
            }
        },
        {
            "internal": true,
            "name": "jms-internal-xa-adp",
            "componentRuntimes": {
                "items": [{
                    "type": "ConnectorComponentRuntime",
                    "deploymentState": {
                        "values": [
                            2,
                            2,
                            2
                        ]
                    }
                }]
            }
```

```
},
{
    "internal": true,
    "name": "wls-management-services",
    "componentRuntimes": {
        "items": [{
            "type": "WebAppComponentRuntime",
            "sourceInfo": "wls-management-services.war",
            "contextRoot": "\/management",
            "openSessionsCurrentCount": {
                "total": 3,
                "count": 3
            },
            "deploymentState": {
                "values": [
                    2,
                    2,
                    2
                ]
            },
            "sessionsOpenedTotalCount": {
                "total": 173,
                "count": 3
            },
            "openSessionsHighCount": {
                "max": 3
            },
            "servlets": {
                "items": [{
                    "invocationTotalCount": {
                        "total": 170,
                        "count": 9
                    }
                }]
            }
        }]
    }
},
{
    "internal": false,
    "name": "fairShare",
    "componentRuntimes": {
        "items": [{
            "type": "WebAppComponentRuntime",
            "sourceInfo": "fairShare.war",
            "contextRoot": "\/fairShare",
            "openSessionsCurrentCount": {
                "total": 0,
                "count": 2
            },
            "deploymentState": {
                "values": [
                    2,
                    2
                ]
            },
            "sessionsOpenedTotalCount": {
                "total": 0,
                "count": 2
            },
            "openSessionsHighCount": {
                "max": 0
            },
            "servlets": {
                "items": [{
                    "invocationTotalCount": {
                        "total": 0,
                        "count": 8
                    }
                }]
            }
        }]
    }
},
{
    "internal": true,
    "name": "jms-internal-notran-adp",
    "componentRuntimes": {
        "items": [{
            "type": "ConnectorComponentRuntime",
            "deploymentState": {
                "values": [
                    2,
                    2,
                    2
                ]
            }
        }]
    }
},
{
    "internal": false,
    "name": "JDBCDataSource1",
    "componentRuntimes": {
        "items": [{
            "type": "JDBCDataSourceRuntime",
            "deploymentState": {
                "values": [
                    2,
                    2
                ]
            }
        }]
    }
},
{
    "internal": true,
    "name": "mejb",
    "componentRuntimes": {
        "items": [{
            "deploymentState": {
                "values": [2]
            },
            "type": "EJBComponentRuntime"
        }]
    }
},
{
    "internal": true,
    "name": "bea_wls_management_internal2",
    "componentRuntimes": {
        "items": [{
            "type": "WebAppComponentRuntime",
            "sourceInfo": "bea_wls_management_internal2.war",
                                "contextRoot": "\/bea_wls_management_internal2",
                                "openSessionsCurrentCount": {
                                    "total": 0,
                                    "count": 1
                                },
                                "deploymentState": {
                                    "values": [2]
                                },
                                "sessionsOpenedTotalCount": {
                                    "total": 0,
                                    "count": 1
                                },
                                "openSessionsHighCount": {
                                    "max": 0
                                },
                                "servlets": {
                                    "items": [{
                                        "invocationTotalCount": {
                                            "total": 4,
                                            "count": 4
                                        }
                                    }]
                                }
                            }]
                        }
                    }
                ]
            }
        }]
    }
}




----------------------------------------------------------------------
Search the admin server log as the domain monitor, returning the matching records as json
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  limit: 2,
  query: 'SEVERITY = \'Info\''
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFAccessRuntime/WLDFDataAccessRuntimes/ServerLog/search


HTTP/1.1 200 OK

Response Body:
{
"records": [
{
  "RECORDID": 1,
  "DATE": "Sep 17, 2024, 3:58:09,341 PM Eastern Daylight Time",
  "SEVERITY": "Info",
  "SUBSYSTEM": "Default",
  "MACHINE": "machine1",
  "SERVER": "",
  "THREAD": "[STANDBY] ExecuteThread: '3' for queue: 'weblogic.kernel.Default (self-
tuning)'",
  "USERID": "",
  "TXID": "",
  "CONTEXTID": "",
  "TIMESTAMP": "1726603089341",
  "MSGID": "BEA-000000",
  "MESSAGE": "JceConfig is unknown",
  "SUPP_ATTRS": "[severity-value: 64] ",
  "SEVERITY_VALUE": 64,
  "RID": ""
},
{
  "RECORDID": 2,
  "DATE": "Sep 17, 2024, 3:58:09,381 PM Eastern Daylight Time",
  "SEVERITY": "Info",
  "SUBSYSTEM": "Default",
  "MACHINE": "machine1",
  "SERVER": "",
  "THREAD": "[STANDBY] ExecuteThread: '3' for queue: 'weblogic.kernel.Default (self-
tuning)'",
  "USERID": "",
  "TXID": "",
  "CONTEXTID": "",
  "TIMESTAMP": "1726603089381",
  "MSGID": "BEA-000000",
  "MESSAGE": "FIPS compliant operation not available for configuration type OTHER",
  "SUPP_ATTRS": "[severity-value: 64] ",
  "SEVERITY_VALUE": 64,
  "RID": ""
}],
"nextRecordId": 3
}




----------------------------------------------------------------------
Continue searching a log. This example uses the POST method to continue searching for
records in the admin server log as the domain monitor.
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  limit: 2,
  fromId: 3,
  query: 'SEVERITY = \'Info\''
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFAccessRuntime/WLDFDataAccessRuntimes/ServerLog/search


HTTP/1.1 200 OK
```

```
Response Body:
{
"records": [
{
  "RECORDID": 3,
  "DATE": "Sep 17, 2024, 3:58:09,387 PM Eastern Daylight Time",
  "SEVERITY": "Info",
  "SUBSYSTEM": "Default",
  "MACHINE": "machine1",
  "SERVER": "",
  "THREAD": "[STANDBY] ExecuteThread: '3' for queue: 'weblogic.kernel.Default (self-
tuning)'",
  "USERID": "",
  "TXID": "",
  "CONTEXTID": "",
  "TIMESTAMP": "1726603089387",
  "MSGID": "BEA-000000",
  "MESSAGE": "JceConfig is in non-FIPS mode",
  "SUPP_ATTRS": "[severity-value: 64] ",
  "SEVERITY_VALUE": 64,
  "RID": ""
},
{
  "RECORDID": 4,
  "DATE": "Sep 17, 2024, 3:58:09,804 PM Eastern Daylight Time",
  "SEVERITY": "Info",
  "SUBSYSTEM": "WebLogicServer",
  "MACHINE": "machine1",
  "SERVER": "",
  "THREAD": "Thread-6",
  "USERID": "",
  "TXID": "",
  "CONTEXTID": "",
  "TIMESTAMP": "1726603089804",
  "MSGID": "BEA-000377",
  "MESSAGE": "Starting WebLogic Server with Java HotSpot(TM) 64-Bit Server VM Version
17.0.12+8-LTS-286 from Oracle Corporation.",
  "SUPP_ATTRS": "[severity-value: 64] ",
  "SEVERITY_VALUE": 64,
  "RID": ""
}],
"nextRecordId": 5
}




----------------------------------------------------------------------
Search the admin server log as the domain monitor, returning the matching records as
plain text
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:text/plain \
-H Content-Type:application/json \
-d "{
  limit: 2,
  query: 'SEVERITY = \'Info\' AND USERID = \'admin\'',
  lastMinutes: 60
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFAccessRuntime/WLDFDataAccessRuntimes/ServerLog/search


HTTP/1.1 200 OK

Response Body:
####<Sep 17, 2024, 3:58:35,652 PM Eastern Daylight Time> <Info> <Security> <machine1>
<AdminServer> <[ACTIVE] ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-
tuning)'> <admin> <> <f1b2c55a-c88e-4f6e-88c5-9189ce73dc39-0000000e> <1726603115652>
<[severity-value: 64] [rid: 0] > <BEA-090516> <The Authenticator provider has pre-
existing LDAP data.>







----------------------------------------------------------------------
Continue searching a log. This example uses the POST method to continue searching for
records in the admin server log as the domain monitor.
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:text/plain \
-H Content-Type:application/json \
-d "{
  limit: 2,
  fromId: ,
  query:  'SEVERITY = \'Info\' AND USERID = \'admin\'',
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFAccessRuntime/WLDFDataAccessRuntimes/ServerLog/search


HTTP/1.1 400 Bad Request

Response Body:
Bad Request






----------------------------------------------------------------------
Capture a diagnostics image. Prevents capturing further images for the configured
default lockout period. Note: only an admin is allowed to capture an image.
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFImageRuntime/capturedImages
```

```
HTTP/1.1 201 Created

Location: http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFImageRuntime/capturedImages/diagnostic_image_AdminServer_2024_09_17_16_00_14.zip

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/serverRuntime\/
WLDFRuntime\/WLDFImageRuntime\/imageCaptureTasks\/DiagnosticImageCaptureTaskRuntime_1"
    }],
    "identity": [
        "WLDFRuntime",
        "WLDFImageRuntime",
        "imageCaptureTasks",
        "DiagnosticImageCaptureTaskRuntime_1"
    ],
    "running": false,
    "systemTask": false,
    "endTimeAsLong": 1726603221275,
    "imageFileName": "diagnostic_image_AdminServer_2024_09_17_16_00_14.zip",
    "name": "DiagnosticImageCaptureTaskRuntime_1",
    "progress": "success",
    "description": "\/domains\/mydomain\/servers\/AdminServer\/logs\/diagnostic_images\/
diagnostic_image_AdminServer_2024_09_17_16_00_14.zip",
    "taskError": null,
    "startTimeAsLong": 1726603214977,
    "type": "WLDFImageCreationTaskRuntime",
    "taskStatus": "Completed",
    "parentTask": null,
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:14.977-04:00",
    "endTime": "2024-09-17T16:00:21.275-04:00"
}




----------------------------------------------------------------------
List the captured diagnostics images.
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFImageRuntime/capturedImages?links=none


HTTP/1.1 200 OK

Response Body:
{
    "items": [{
        "name": "diagnostic_image_AdminServer_2024_09_17_16_00_14.zip"
    }]
}
```

```
----------------------------------------------------------------------
Download a captured diagnostics image
----------------------------------------------------------------------

curl -v \
--user monitor:monitor123 \
-H X-Requested-By:MyClient \
-H Accept:application/octet-stream \
-X GET http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFImageRuntime/capturedImages//contents


HTTP/1.1 404 Not Found

Response Body:
Not Found






----------------------------------------------------------------------
Remove all of the captured images. Note: only an admin is allowed to remove captured
images.
----------------------------------------------------------------------

curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  age: ''
}" \
-X POST http://localhost:7001/management/weblogic/latest/serverRuntime/WLDFRuntime/
WLDFImageRuntime/purgeCapturedImages


HTTP/1.1 200 OK

Response Body:
{}
```

Starting and Stopping Domain-Scoped Applications

Review an example script that demonstrates how an Operator starts and stops domain-scoped applications.

Note:

To view long URLs, use the scroll bar located beneath the section.

```
----------------------------------------------------------------------
Demonstrate a domain operator starting and stopping a domain scoped app
----------------------------------------------------------------------


----------------------------------------------------------------------
Get the app's state on one of the servers in the cluster
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{ target='Cluster1Server1' }" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
appDeploymentRuntimes/fairShare/getState


HTTP/1.1 200 OK

Response Body:
{
    "return": "STATE_ACTIVE"
}




----------------------------------------------------------------------
Synchronously stop the app
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
appDeploymentRuntimes/fairShare/stop


HTTP/1.1 200 OK

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
deploymentManager\/deploymentProgressObjects\/fairShare"
    }],
    "identity": [
        "deploymentManager",
        "deploymentProgressObjects",
        "fairShare"
    ],
    "rootExceptions": [],
    "endTimeAsLong": 1726603228848,
    "deploymentMessages": [
        "[Deployer:149192]Operation \"stop\" on application \"fairShare\" is in progress
on \"Cluster1Server1\".",
        "[Deployer:149192]Operation \"stop\" on application \"fairShare\" is in progress
on \"Cluster1Server2\".",
        "[Deployer:149194]Operation \"stop\" on application \"fairShare\" has succeeded
on \"Cluster1Server1\".",
        "[Deployer:149194]Operation \"stop\" on application \"fairShare\" has succeeded
on \"Cluster1Server2\"."
    ],
    "name": "fairShare",
    "operationType": 2,
    "startTimeAsLong": 1726603223230,
    "state": "STATE_COMPLETED",
    "id": "2",
    "type": "DeploymentProgressObject",
    "targets": ["Cluster1"],
    "applicationName": "fairShare",
    "failedTargets": [],
    "progress": "success",
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:23.230-04:00",
    "endTime": "2024-09-17T16:00:28.848-04:00"
}




----------------------------------------------------------------------
Get the app's state on one of the servers in the cluster
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{ target='Cluster1Server1' }" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
appDeploymentRuntimes/fairShare/getState


HTTP/1.1 200 OK

Response Body:
{
    "return": "STATE_PREPARED"
}




----------------------------------------------------------------------
Synchronously start the app
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
appDeploymentRuntimes/fairShare/start


HTTP/1.1 200 OK

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
deploymentManager\/deploymentProgressObjects\/fairShare"
    }],
    "identity": [
        "deploymentManager",
        "deploymentProgressObjects",
        "fairShare"
    ],
    "rootExceptions": [],
    "endTimeAsLong": 1726603229343,
    "deploymentMessages": [
        "[Deployer:149192]Operation \"start\" on application \"fairShare\" is in
progress on \"Cluster1Server2\".",
        "[Deployer:149194]Operation \"start\" on application \"fairShare\" has succeeded
on \"Cluster1Server1\".",
        "[Deployer:149194]Operation \"start\" on application \"fairShare\" has succeeded
on \"Cluster1Server2\"."
    ],
    "name": "fairShare",
    "operationType": 1,
    "startTimeAsLong": 1726603229268,
    "state": "STATE_COMPLETED",
    "id": "3",
    "type": "DeploymentProgressObject",
    "targets": ["Cluster1"],
    "applicationName": "fairShare",
    "failedTargets": [],
    "progress": "success",
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:29.268-04:00",
    "endTime": "2024-09-17T16:00:29.343-04:00"
}




----------------------------------------------------------------------
Get the app's state on one of the servers in the cluster
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{ target='Cluster1Server1' }" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/deploymentManager/
appDeploymentRuntimes/fairShare/getState
```

```
HTTP/1.1 200 OK

Response Body:
{
    "return": "STATE_ACTIVE"
}
```

Starting and Stopping Servers

Review an example script that demonstrates how an Operator starts and stops servers.

Note:

To view long URLs, use the scroll bar located beneath the section.

```
----------------------------------------------------------------------
Demonstrate a domain operator starting and stopping servers
----------------------------------------------------------------------



----------------------------------------------------------------------
View the servers' states
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes?links=none&fields=name,state


HTTP/1.1 200 OK

Response Body:
{
    "items": [
        {
            "name": "Cluster1Server1",
            "state": "RUNNING"
        },
        {
            "name": "AdminServer",
            "state": "RUNNING"
        },
        {
            "name": "Cluster1Server2",
            "state": "RUNNING"
        }
    ]
}
```

```
----------------------------------------------------------------------
Synchronously shutdown a server
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{
  timeout: 300, ignoreSessions: true
}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server1/shutdown


HTTP/1.1 200 OK

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
serverLifeCycleRuntimes\/Cluster1Server1\/tasks\/_2_shutdown"
    }],
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server1",
        "tasks",
        "_2_shutdown"
    ],
    "running": false,
    "systemTask": false,
    "endTimeAsLong": 1726603236985,
    "name": "_2_shutdown",
    "progress": "success",
    "description": "Shutting down Cluster1Server1 server ...",
    "serverName": "Cluster1Server1",
    "taskError": null,
    "startTimeAsLong": 1726603230092,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "shutdown",
    "taskStatus": "TASK COMPLETED",
    "parentTask": null,
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:30.092-04:00",
    "endTime": "2024-09-17T16:00:36.985-04:00"
}




----------------------------------------------------------------------
Asynchronously force shutdown a server
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-H "Prefer:respond-async" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/forceShutdown


HTTP/1.1 202 Accepted

Location: http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/tasks/_3_forceShutdown

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
serverLifeCycleRuntimes\/Cluster1Server2\/tasks\/_3_forceShutdown"
    }],
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_3_forceShutdown"
    ],
    "running": true,
    "systemTask": false,
    "name": "_3_forceShutdown",
    "progress": "processing",
    "description": "Forcefully shutting down Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603237645,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "forceShutdown",
    "taskStatus": "TASK IN PROGRESS",
    "parentTask": null,
    "completed": false,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:37.645-04:00"
}




----------------------------------------------------------------------
Get status for job domainRuntime/serverLifeCycleRuntimes/Cluster1Server2/tasks/
_3_forceShutdown
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/tasks/_3_forceShutdown?links=none


HTTP/1.1 200 OK
```

```
Response Body:
{
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_3_forceShutdown"
    ],
    "running": true,
    "systemTask": false,
    "name": "_3_forceShutdown",
    "progress": "processing",
    "description": "Forcefully shutting down Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603237645,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "forceShutdown",
    "taskStatus": "TASK IN PROGRESS",
    "parentTask": null,
    "completed": false,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:37.645-04:00"
}




----------------------------------------------------------------------
Get status for job domainRuntime/serverLifeCycleRuntimes/Cluster1Server2/tasks/
_3_forceShutdown
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/tasks/_3_forceShutdown?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_3_forceShutdown"
    ],
    "running": false,
    "systemTask": false,
    "endTimeAsLong": 1726603244602,
    "name": "_3_forceShutdown",
    "progress": "success",
    "description": "Forcefully shutting down Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603237645,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "forceShutdown",
    "taskStatus": "TASK COMPLETED",
    "parentTask": null,
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:37.645-04:00",
    "endTime": "2024-09-17T16:00:44.602-04:00"
}





----------------------------------------------------------------------
View the servers' states
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes?links=none&fields=name,state


HTTP/1.1 200 OK

Response Body:
{
    "items": [
        {
            "name": "Cluster1Server1",
            "state": "SHUTDOWN"
        },
        {
            "name": "AdminServer",
            "state": "RUNNING"
        },
        {
            "name": "Cluster1Server2",
            "state": "SHUTDOWN"
        }
    ]
}





----------------------------------------------------------------------
Synchronously start a server
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server1/start
```

```
HTTP/1.1 200 OK

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
serverLifeCycleRuntimes\/Cluster1Server1\/tasks\/_4_start"
    }],
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server1",
        "tasks",
        "_4_start"
    ],
    "running": false,
    "systemTask": false,
    "endTimeAsLong": 1726603285236,
    "name": "_4_start",
    "progress": "success",
    "description": "Starting Cluster1Server1 server ...",
    "serverName": "Cluster1Server1",
    "taskError": null,
    "startTimeAsLong": 1726603256493,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "start",
    "taskStatus": "TASK COMPLETED",
    "parentTask": null,
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:00:56.493-04:00",
    "endTime": "2024-09-17T16:01:25.236-04:00"
}




----------------------------------------------------------------------
Asynchronously start a server
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-H Content-Type:application/json \
-d "{}" \
-H "Prefer:respond-async" \
-X POST http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/start


HTTP/1.1 202 Accepted

Location: http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/tasks/_5_start

Response Body:
{
    "links": [{
        "rel": "job",
        "href": "http:\/\/localhost:7001\/management\/weblogic\/latest\/domainRuntime\/
serverLifeCycleRuntimes\/Cluster1Server2\/tasks\/_5_start"
    }],
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_5_start"
    ],
    "running": true,
    "systemTask": false,
    "name": "_5_start",
    "progress": "processing",
    "description": "Starting Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603287005,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "start",
    "taskStatus": "TASK IN PROGRESS",
    "parentTask": null,
    "completed": false,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:01:27.005-04:00"
}




----------------------------------------------------------------------
Get status for job domainRuntime/serverLifeCycleRuntimes/Cluster1Server2/tasks/_5_start
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/tasks/_5_start?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_5_start"
    ],
    "running": true,
    "systemTask": false,
    "name": "_5_start",
    "progress": "processing",
    "description": "Starting Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603287005,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "start",
    "taskStatus": "TASK IN PROGRESS",
    "parentTask": null,
    "completed": false,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:01:27.005-04:00"
}




----------------------------------------------------------------------
Get status for job domainRuntime/serverLifeCycleRuntimes/Cluster1Server2/tasks/_5_start
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/tasks/_5_start?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_5_start"
    ],
    "running": true,
    "systemTask": false,
    "name": "_5_start",
    "progress": "processing",
    "description": "Starting Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603287005,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "start",
    "taskStatus": "TASK IN PROGRESS",
    "parentTask": null,
    "completed": false,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:01:27.005-04:00"
}




----------------------------------------------------------------------
Get status for job domainRuntime/serverLifeCycleRuntimes/Cluster1Server2/tasks/_5_start
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes/Cluster1Server2/tasks/_5_start?links=none


HTTP/1.1 200 OK

Response Body:
{
    "identity": [
        "serverLifeCycleRuntimes",
        "Cluster1Server2",
        "tasks",
        "_5_start"
    ],
    "running": false,
    "systemTask": false,
    "endTimeAsLong": 1726603313739,
    "name": "_5_start",
    "progress": "success",
    "description": "Starting Cluster1Server2 server ...",
    "serverName": "Cluster1Server2",
    "taskError": null,
    "startTimeAsLong": 1726603287005,
    "type": "ServerLifeCycleTaskRuntime",
    "operation": "start",
    "taskStatus": "TASK COMPLETED",
    "parentTask": null,
    "completed": true,
    "intervalToPoll": 1000,
    "startTime": "2024-09-17T16:01:27.005-04:00",
    "endTime": "2024-09-17T16:01:53.739-04:00"
}





----------------------------------------------------------------------
View the servers' states
----------------------------------------------------------------------

curl -v \
--user operator:operator123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/weblogic/latest/domainRuntime/
serverLifeCycleRuntimes?links=none&fields=name,state


HTTP/1.1 200 OK

Response Body:
{
    "items": [
        {
            "name": "Cluster1Server1",
            "state": "RUNNING"
        },
        {
            "name": "AdminServer",
            "state": "RUNNING"
        },
        {
            "name": "Cluster1Server2",
            "state": "RUNNING"
        }
    ]
}
```