

Oracle® Fusion Middleware

RESTful Lifecycle Reference for Oracle WebLogic Server

12c (12.2.1)

E57838-01

October 2015

This reference describes a RESTful management API for managing a WebLogic Server domain.

E57838-01

Copyright © 2014, 2015, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
----------------------	-----

1 Resources

/lifecycle.....	1-2
/lifecycle/{version}	1-5
/lifecycle/{version}/environmentCreateForm	1-8
/lifecycle/{version}/environments	1-11
/lifecycle/{version}/environments/{environment-name}	1-16
/lifecycle/{version}/environments/{environment-name}/associatePartitions	1-20
/lifecycle/{version}/environments/{environment-name}/dissociatePartitions	1-22
/lifecycle/{version}/environments/{environment-name}/partitionCreateForm	1-24
/lifecycle/{version}/environments/{environment-name}/partitions.....	1-27
/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name} 1-31	
/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/migrate 1-35	
/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/operation/task/{partitionOrResource}/{task-name} 1-37	
/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/quiesce 1-41	
/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/start 1-43	
/lifecycle/{version}/environments/{environment-name}/sync.....	1-45
/lifecycle/{version}/plugins.....	1-47
/lifecycle/{version}/plugins/{plugin-name}.....	1-51
/lifecycle/{version}/runtimeCreateForm.....	1-55
/lifecycle/{version}/runtimes	1-58
/lifecycle/{version}/runtimes/{runtime-name}.....	1-62
/lifecycle/{version}/runtimes/{runtime-name}/failoverGroupUpdate.....	1-67

/lifecycle/{version}/runtimes/{runtime-name}/partitionCreateForm.....	1-69
/lifecycle/{version}/runtimes/{runtime-name}/partitions	1-72
/lifecycle/{version}/runtimes/{runtime-name}/partitions/{partition-name}	1-78
/lifecycle/{version}/runtimes/{runtime-name}/quiesce	1-83
/lifecycle/{version}/runtimes/{runtime-name}/quiesce/task/{server-name}/{task-name} 1-85	
/lifecycle/{version}/runtimes/{runtime-name}/scaleDown.....	1-89
/lifecycle/{version}/runtimes/{runtime-name}/scaleUp	1-91
/lifecycle/{version}/runtimes/{runtime-name}/scaleUp/task/{cluster-name}/{task-name } 1-93	
/lifecycle/{version}/runtimes/{runtime-name}/start	1-97
/lifecycle/{version}/runtimes/{runtime-name}/start/task/{server-name}/{task-name}	1-99
/lifecycle/{version}/runtimes/{runtime-name}/sync	1-103
/lifecycle/{version}/sync	1-105
/lifecycle/{version}/tenantCreateForm.....	1-108
/lifecycle/{version}/tenants	1-111
/lifecycle/{version}/tenants/{tenant-name}.....	1-115
/lifecycle/{version}/tenants/{tenant-name}/serviceCreateForm.....	1-120
/lifecycle/{version}/tenants/{tenant-name}/services	1-123
/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}.....	1-127
/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}/PDB	1-132
/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}/PDBCreateForm	1-137

2 Entities

[3] AssociatePartitions	2-1
[3] DissociatePartitions	2-2
[3] Environment	2-2
[3] EnvironmentPartition.....	2-3
[3] FailoverGroupUpdate	2-3
[3] MigratePartition	2-3
[3] Orchestration	2-4
[3] PDB.....	2-4
[3] Property	2-5
[3] QuiescePartition	2-5
[3] QuiesceRuntime	2-5
[3] RegisterPlugin	2-6
[3] Runtime	2-6
[3] RuntimePartition.....	2-7
[3] ScaleRuntime	2-7
[3] Service.....	2-7
[3] StartPartition.....	2-8
[3] StartRuntime.....	2-9

[3]	SyncOptions.....	2-9
[3]	Task	2-9
[3]	Tenant	2-10
[3]	Version.....	2-10

Preface

This preface describes the document accessibility features and conventions used in this guide, *RESTful Lifecycle Reference for Oracle WebLogic Server*.

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.

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.
<i>italic</i>	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.

Resources

This documentation describes the Oracle WebLogic Server RESTful life cycle management resources.

For a complete listing of the WLS REST reference documents and descriptions of their use, see *Administering Oracle WebLogic Server with RESTful Management Services*.

/lifecycle

The versions resource contains information about the versions of the lifecycle REST interface that are active and supported in the current the WLS domain.

The resource supports the following methods:

- [GET](#)

GET

The GET method on this resource returns information about each supported version of this REST interface.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a list of [Version](#) entities for the supported versions of this REST interface. The returned information includes which version is the latest and whether or not each version is active.

This method can return the following links:

- `uri=/lifecycle/12.2.1.0 rel=current`

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 List Versions

This example uses the GET method to list available versions.

Example Request

```
curl -v \  
--user admin:admin123 \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "items": [{  
    "links": [  
      {  
        "rel": "canonical",  
        "href": "http://localhost:7001/management/lifecycle/12.2.1.0"  
      },  
      {  
        "rel": "self",  
        "href": "http://localhost:7001/management/lifecycle/12.2.1.0"  
      }  
    ],  
    "version": "12.2.1.0",  
    "isLatest": true,  
    "lifecycle": "active"  
  }],  
  "links": [  
    {
```

```
        "rel": "self",
        "href": "http://localhost:7001/management/lifecycle/"
    },
    {
        "rel": "canonical",
        "href": "http://localhost:7001/management/lifecycle/"
    },
    {
        "rel": "current",
        "href": "http://localhost:7001/management/lifecycle/12.2.1.0"
    }
]
}
```

/lifecycle/{version}

The version resource contains information about the latest version of the lifecycle REST interface that is supported by the WLS domain.

The resource supports the following methods:

- [GET](#)

GET

The GET method on this resource returns information about the supported version of this REST interface.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Version](#) entity that contains information about the latest supported REST lifecycle interface version that is active and supported in the current WLS release.

This method can return the following links:

- **uri**=/lifecycle/{version}/environmentCreateForm **rel**=environmentCreateForm
- **uri**=/lifecycle/{version}/environments **rel**=environments
- **uri**=/lifecycle/{version}/plugins **rel**=plugins
- **uri**=/lifecycle/{version}/runtimeCreateForm **rel**=runtimeCreateForm
- **uri**=/lifecycle/{version}/runtimes **rel**=runtimes
- **uri**=/lifecycle/{version}/tenantCreateForm **rel**=tenantCreateForm
- **uri**=/lifecycle/{version}/tenants **rel**=tenants

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Describe Version

This example uses the GET method to describe a version.

Example Request

```
curl -v \  
--user admin:admin123 \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "version": "12.2.1.0",  
  "isLatest": true,  
  "lifecycle": "active",  
  "links": [  
    {  
      "rel": "parent",  
      "href": "http://localhost:7001/management/"
```

```

    },
    {
      "rel": "self",
      "href": "http://localhost:7001/management/lifecycle/latest/"
    },
    {
      "rel": "canonical",
      "href": "http://localhost:7001/management/lifecycle/latest/"
    },
    {
      "rel": "environmentCreateForm",
      "href":
"http://localhost:7001/management/lifecycle/latest/environmentCreateForm"
    },
    {
      "rel": "environments",
      "href":
"http://localhost:7001/management/lifecycle/latest/environments"
    },
    {
      "rel": "runtimeCreateForm",
      "href":
"http://localhost:7001/management/lifecycle/latest/runtimeCreateForm"
    },
    {
      "rel": "runtimes",
      "href":
"http://localhost:7001/management/lifecycle/latest/runtimes"
    },
    {
      "rel": "tenantCreateForm",
      "href":
"http://localhost:7001/management/lifecycle/latest/tenantCreateForm"
    },
    {
      "rel": "tenants",
      "href":
"http://localhost:7001/management/lifecycle/latest/tenants"
    },
    {
      "rel": "plugins",
      "href":
"http://localhost:7001/management/lifecycle/latest/plugins"
    }
  ]
}

```

/lifecycle/{version}/environmentCreateForm

This resource describes the information that is needed to create the environment.

The resource supports the following methods:

- [GET](#)

GET

The GET method on this resource returns an empty form for the Environment.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes an [Environment](#) entity that contains information about fields that need to be specified when creating the environment.

This method can return the following links:

- **uri**=/environments **rel**=/environments

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Get Environment Create Form

This example uses the GET method to get the environment create form.

Example Request

```
curl -v \  
--user admin:admin123 \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/environmentCreateForm
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "name": null,  
  "links": [  
    {  
      "rel": "parent",  
      "href": "http://localhost:7001/management/lifecycle/latest"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environmentCreateForm"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environmentCreateForm"  
    },  
    {  
      "rel": "environments",  
      "href":
```

GET

```
"http://localhost:7001/management/lifecycle/latest/environments"  
  }  
] }  
}
```

/lifecycle/{version}/environments

This resource manages environments.

The resource supports the following methods:

- [GET](#)
- [POST](#)

GET

The GET method on this resource returns a list of environments.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a collection of [Environment](#) entities.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Viewing Environments

This example uses the GET method to display environments.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/environments
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "items": [{  
    "links": [  
      {  
        "rel": "canonical",  
        "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite"  
      },  
      {  
        "rel": "self",  
        "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite"  
      }  
    ],  
    "name": "sprite"  
  }],  
  "links": [  
    {  
      "rel": "parent",  
      "href": "http://localhost:7001/management/lifecycle/latest"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments"
```

```
    },
    {
      "rel": "canonical",
      "href":
"http://localhost:7001/management/lifecycle/latest/environments"
    },
    {
      "rel": "create-form",
      "href":
"http://localhost:7001/management/lifecycle/latest/environmentCreateForm"
    }
  ]
}
```

POST

The POST method creates an environment.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Creating an Environment

This example uses the POST method to create an empty environment.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"name": "sprite"}' \  
-X POST http://localhost:7001/management/lifecycle/latest/environments
```

Example Response

```
HTTP/1.1 201 Created
```

```
Location: http://localhost:7001/management/lifecycle/latest/environments/sprite
```

```
Response Body:
```

```
{}
```

This example uses the POST method to create an environment based on an orchestration so that partitions are created first and are then added to the environment.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"name": "coke",  
"orchestration": {"name": "orchestration1",  
"args": [  
  {"name": "wls",  
  "properties": [  
    {"name": "partitionName", "value": "wlspartition1"},  
    {"name": "runtimeName", "value": "WLSTestRuntime"},  
    {"name": "partitionProperties",  
      "properties": [  
        {"name": "resourceGroups",
```

```
        "properties" : [
          { "name" : "g1",
            "properties" : [
              { "name" : "resourceGroupTemplate", "value" :
"template1" },
              { "name" : "targets" , "value" :
"VirtualHost-0"}]]}],
          { "name" : "availableTargets" , "value" : "VirtualHost-0"}]
        ]}}}}' \
-X POST http://localhost:7001/management/lifecycle/latest/environments
```

Example Response

HTTP/1.1 201 Created

Location: <http://localhost:7001/management/lifecycle/latest/environments/sprite>

Response Body:

```
{}
```

/lifecycle/{version}/environments/{environment-name}

This resource manages an environment.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)

DELETE

The DELETE method deletes the environment identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Deleting an Environment

This example uses the DELETE method to delete a specific environment.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE http://localhost:7001/management/lifecycle/latest/environments/sprite
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns information about the environment identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes an [Environment](#) entity that contains information about the specified environment.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing an Environment

This example uses the GET method to display information about a specific environment.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/environments/sprite
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{  
  "name": "sprite",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite"  
    },  
    {  
      "rel": "partitioncreateForm",  
      "href":
```

```
"http://localhost:7001/management/lifecycle/latest/environments/test1/partitionCreateForm"
  },
  {
    "rel": "partitions",
    "href":
      "http://localhost:7001/management/lifecycle/latest/environments/test1/partitions"
  },
  {
    "rel": "associatePartitions",
    "href":
      "http://localhost:7001/management/lifecycle/latest/environments/test1/associatePartitions"
  }
]
}
```

/lifecycle/{version}/environments/{environment-name}/associatePartitions

This resource is used to associate two environment partitions with each other.

The resource supports the following methods:

- [POST](#)

POST

The POST method associates two environment partitions with each other.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Associating a WLS Partition with a PDB Partition

This example uses the POST method to associate two partitions with each other.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{  
  "partition1Name": "SpritePartition",  
  "partition1RuntimeName": "WLSRuntime",  
  "partition2Name": "SpritePDB",  
  "partition2RuntimeName": "DBRuntime",  
  "properties": [  
    {"name": "jdbcSystemResource", "properties": [  
      {"name": "resourceGroups", "properties": [  
        {"name": "g1", "value": "SpritePDB"}]}]},  
    {"name": "jdbcSystemResourceOverride", "value": "SpritePDB" }]} ' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/environments/sprite/associatePartitions
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

/lifecycle/{version}/environments/{environment-name}/dissociatePartitions

This resource allows you to dissociate associated partitions.

The resource supports the following methods:

- [POST](#)

POST

The POST method dissociates two partitions.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Dissociating WLS partition and PDB

This example uses the POST method to dissociate two partitions.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{  
  "partition1Name": "SpritePartition",  
  "partition1RuntimeName": "WLSRuntime",  
  "partition2Name": "SpritePDB",  
  "partition2RuntimeName": "DBRuntime",  
  "properties": [  
    {"name": "jdbcSystemResource", "properties": [  
      {"name": "resourceGroups", "properties": [  
        {"name": "g1", "value": "SpritePDB"}]}]},  
    {"name": "jdbcSystemResourceOverride", "value": "SpritePDB" }]} ' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/environments/sprite/dissociatePartitions
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

/lifecycle/{version}/environments/{environment-name}/partitionCreateForm

This resource describes the information that is needed to add a partition to the environment.

The resource supports the following methods:

- [GET](#)

GET

The GET method on this resource returns an empty form for the Partition.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes an [EnvironmentPartition](#) entity that contains information about fields that must be specified when adding a partition to the environment.

This method can return the following links:

- **uri**=/partitions **rel**=/partitions

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Get Partition Create Form

This example uses the GET method to get the partition create form.

Example Request

```
curl -v \  
--user admin:admin123 \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/environments/sprite/partitionCreateForm
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "runtimeName": null,  
  "name": null,  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/partitionCreateForm"  
    },  
    {  
      "rel": "canonical",
```

```
        "href":
"http://localhost:7001/management/lifecycle/latest/environments/sprite/partitionCreateForm"
    },
    {
        "rel": "partitions",
        "href":
"http://localhost:7001/management/lifecycle/latest/environments/sprite/partitions"
    }
]
}
```

/lifecycle/{version}/environments/{environment-name}/partitions

This resource manages environment partitions.

The resource supports the following methods:

- [GET](#)
- [POST](#)

GET

The GET method on this resource returns a list of partitions for the environment identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a collection of [EnvironmentPartition](#) entities.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing Partitions

This example uses the GET method to display the partitions for a specific environment.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/environments/sprite/partitions
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:
```

```
{  
  "items": [  
    {  
      "links": [  
        {  
          "rel": "canonical",  
          "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/run  
times/WLSRuntime/partitions/SpritePartition"  
        },  
        {  
          "rel": "self",  
          "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/run  
times/WLSRuntime/partitions/SpritePartition"  
        }  
      ],  
      "runtimeName": "WLSRuntime",  
      "name": "SpritePartition"  
    }  
  ],  
}
```

```
"links": [  
  {  
    "rel": "parent",  
    "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite"  
  },  
  {  
    "rel": "self",  
    "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/partitions"  
  },  
  {  
    "rel": "canonical",  
    "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/partitions"  
  },  
  {  
    "rel": "create-form",  
    "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/partitionCreateForm"  
  }  
]
```

POST

The POST method adds a partition to the environment identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Adding a Partition to the Environment

This example uses the POST method to add a partition to an environment.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"name": "SpritePartition", "runtimeName": "WLSRuntime"}' \  
-X POST \  
http://localhost:7001/management/lifecycle/latest/environments/sprite/partitions
```

Example Response

```
HTTP/1.1 201 Created
```

Location:

```
http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLS  
Runtime/partitions/SpritePartition
```

Response Body:

```
{}
```

/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}

This resource manages the environment partition.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)

DELETE

The DELETE method deletes the partition identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Deleting a Partition

This example uses the DELETE method to delete a specific partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLS  
Runtime/partitions/SpritePartition
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```


GET

The GET method on this resource returns information about the partition identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes an [EnvironmentPartition](#) entity that contains information about the specified partition.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Displaying Information About a Partition

This example uses the GET method to display information about a specific partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLS \  
Runtime/partitions/SpritePartition
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "runtimeName": "WLSRuntime",  
  "name": "SpritePartition",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/partitions"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLSRuntime/partitions/SpritePartition"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/run
```

```
times\WLSRuntime\partitions\SpritePartition"  
    }  
  ]  
}
```

/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/migrate

This resource migrates a partition.

The resource supports the following methods:

- [POST](#)

POST

The POST method on this resource migrates the partition identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Migrating a Partition

This example uses the POST method to migrate a specific partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{ "runtimeName": "WLSRuntime", "phase": "", "properties": [] }' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/environments/sprite/partitions/S  
pritePartition
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/{operation}/task/{partitionOrResource}/{task-name}

This resource manages a partition task.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)

DELETE

The DELETE method cancels the task identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Canceling a Task

This example uses the DELETE method to cancel a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE  
http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLS  
Runtime/partitions/SpritePartition/quiesce/task/quiescePartition/SpritePartition/_  
1_STOP
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:
```

```
{}
```

GET

The GET method on this resource returns the status of the task identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Task](#) entity that contains information about the specified task.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing a Task

This example uses the GET method to display information about a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET  
http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLS  
Runtime/partitions/SpritePartition/quiesce/task/quiescePartition/_  
1_STOP
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{  
  "status": "TASK COMPLETED",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/run  
times/WLSRuntime/partitions/SpritePartition"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/environments/sprite/run  
times/WLSRuntime/partitions/SpritePartition/quiesce/task/quiescePartition/_  
1_STOP"  
    },  
    {  
      "rel": "canonical",
```

```
        "href":  
        "http://localhost:7001/management/lifecycle/latest/environments/sprite/run  
times/WLSRuntime/partitions/SpritePartition/quiesce/task/quiescePartition/S  
pritePartition/_1_STOP"  
    }  
]  
}
```


/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/quiesce

This resource quiesces a partition.

The resource supports the following methods:

- [POST](#)

POST

The POST method performs a quiesce operation on the specified partition.

Roles

Administrator

Response Body

The response body returned contains a link to the corresponding task resource.

Response Codes

The method returns one of the standard HTTP status codes.

Example

Example 1 Quiescing a Partition

This example uses the POST method to quiesce a partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"phase": "", "properties" : []}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLS  
Runtime/partitions/SpritePartition/quiesce
```

Example Response

```
HTTP/1.1 202 Accepted
```

Response Body:

```
{"links": [{"  
  "rel": "task",  
  "href":  
  "http://localhost:7001/management/lifecycle/latest/environments/sprite/run  
times/WLSRuntime/partitions/SpritePartition/quiesce/task/quiescePartition/S  
pritePartition/_1_STOP"  
}]}
```

/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name}/start

This resource is used to start a partition. When you start a partition, all application deployments and resources become active and responsive to end users and other components.

The resource supports the following methods:

- [POST](#)

POST

The POST method performs a start operation on the specified partition.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Starting a Partition

This example uses the POST method to start partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"phase": "", "properties" : []}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/environments/sprite/runtimes/WLS  
Runtime/partitions/SpritePartition/start
```

Example Response

```
HTTP/1.1 202 Accepted
```

Response Body:

```
{"links": [{"  
  "rel": "task",  
  "href":  
  "http://localhost:7001/management/lifecycle/latest/environments/sprite/run  
times/WLSRuntime/partitions/SpritePartition/start/task/startPartition/Sprite  
Partition/_2_START"  
}]}
```

/lifecycle/{version}/environments/{environment-name}/sync

This resource synchronizes the environment.

The resource supports the following methods:

- [POST](#)

POST

The POST method synchronizes the environment.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Synchronizes the environment

This example uses the POST method to synchronize the environment.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '' \  
-X POST http://localhost:7001/management/lifecycle/latest/environments/sprite/sync
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:
```

```
{}
```

/lifecycle/{version}/plugins

This resource manages plugins.

The resource supports the following methods:

- [GET](#)
- [POST](#)

GET

The GET method on this resource returns a list of registered plugins.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a collection of [RegisterPlugin](#) entities.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing a List of Plugins

This example uses the GET method to display a list of registered plugins.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/plugins
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "items": [  
    {  
      "links": [  
        {  
          "rel": "canonical",  
          "href":  
"http://localhost:7001/management/lifecycle/latest/plugins/com.oracle.weblo  
gic.lifecycle.plugin.wls_1.0.0.0.jar"  
        },  
        {  
          "rel": "self",  
          "href":  
"http://localhost:7001/management/lifecycle/latest/plugins/com.oracle.weblo  
gic.lifecycle.plugin.wls_1.0.0.0.jar"  
        }  
      ],  
      "name": "com.oracle.weblogic.lifecycle.plugin.wls_1.0.0.0.jar",  
      "path": "\\opt\\wls\\src1214_build\\Oracle_  
Home\\wlserver\\modules\\com.oracle.weblogic.lifecycle.plugin.wls_1.0.0.0.jar"  
    }  
  ],  
  "links": [  
    {  
      "rel": "parent",
```



```
        "href": "http:\\\\localhost:7001\\management\\lifecycle\\latest"
    },
    {
        "rel": "self",
        "href":
"http:\\\\localhost:7001\\management\\lifecycle\\latest\\plugins"
    },
    {
        "rel": "canonical",
        "href":
"http:\\\\localhost:7001\\management\\lifecycle\\latest\\plugins"
    }
    ]
}
```

POST

The POST method registers a plugin.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Registering a Plugin

This example uses the POST method to register a plugin.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{ "path": "/opt/wls/src1214_build/Oracle_  
Home/wlserver/modules/com.oracle.weblogic.lifecycle.plugin.wls_1.0.0.0.jar" }' \  
-X POST http://localhost:7001/management/lifecycle/latest/plugins
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

/lifecycle/{version}/plugins/{plugin-name}

This resource manages a plugin.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)

DELETE

The DELETE method unregisters the plugin identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Unregistering a Plugin

This example uses the DELETE method to unregister a specific plugin.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE  
http://localhost:7001/management/lifecycle/latest/plugins/com.oracle.weblogic.life  
cycle.plugin.wls_1.0.0.0.jar
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns information about the plugin identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [RegisterPlugin](#) entity that contains information about the specified plugin.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing a Plugin

This example uses the GET method to display information about a specific plugin.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/plugins/com.oracle.weblogic.life  
cycle.plugin.wls_1.0.0.0.jar
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "name": "com.oracle.weblogic.lifecycle.plugin.wls_1.0.0.0.jar",  
  "path": "\opt\wls\src1214_build\Oracle_  
Home\wlserver\modules\com.oracle.weblogic.lifecycle.plugin.wls_1.0.0.0.jar",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/plugins"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/plugins/com.oracle.weblo  
gic.lifecycle.plugin.wls_1.0.0.0.jar"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/plugins/com.oracle.weblo
```

```
gic.lifecycle.plugin.wls_1.0.0.0.jar"  
    }  
  ]  
}
```

/lifecycle/{version}/runtimecreateForm

This resource describes the information needed to register a runtime.

The resource supports the following methods:

- [GET](#)

GET

The GET method on this resource returns an empty form for a runtime.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Runtime](#) entity that contains information about fields which must be specified when registering a runtime.

This method can return the following links:

- **uri**=/runtimes **rel**=/runtimes

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Get Runtime Create Form

This example uses the GET method to get the runtime create form.

Example Request

```
curl -v \  
--user admin:admin123 \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/runtimecreateForm
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "name": null,  
  "properties": [],  
  "type": null,  
  "port": null,  
  "hostName": null,  
  "links": [  
    {  
      "rel": "parent",  
      "href": "http://localhost:7001/management/lifecycle/latest"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimecreateForm"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimecreateForm"    }  
  ]  
}
```



```
    },  
    {  
      "rel": "runtimes",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes"  
    }  
  ]  
}
```

/lifecycle/{version}/runtimes

This resource manages runtimes.

The resource supports the following methods:

- [GET](#)
- [POST](#)

GET

The GET method on this resource returns a list of runtimes.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a collection of [Runtime](#) entities.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing a List of Runtimes

This example uses the GET method to display a list of runtimes.

Example Request

```
curl -v \
--user username:password \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/lifecycle/latest/runtimes
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{
  "items": [
    {
      "links": [
        {
          "rel": "canonical",
          "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"
        },
        {
          "rel": "self",
          "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"
        }
      ],
      "name": "WLSRuntime",
      "properties": [
        {
          "name": "password",
          "value": "password"
        },
        {
          "name": "username",
          "value": "username"
        }
      ]
    }
  ]
}
```

```
    ],
    "type": "wls",
    "protocol": "http",
    "port": "7001",
    "hostName": "localhost"
  }
],
"links": [
  {
    "rel": "parent",
    "href": "http://localhost:7001/management/lifecycle/latest"
  },
  {
    "rel": "self",
    "href":
"http://localhost:7001/management/lifecycle/latest/runtimes"
  },
  {
    "rel": "canonical",
    "href":
"http://localhost:7001/management/lifecycle/latest/runtimes"
  },
  {
    "rel": "create-form",
    "href":
"http://localhost:7001/management/lifecycle/latest/runtimecreateForm"
  }
]
}
```

POST

The POST method registers a runtime.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Registering a Runtime

This example uses the POST method to register a runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"name": "WLSRuntime", "type": "wls", "protocol": "http", "hostName":  
"localhost", "port": "7001", "properties": [{"name": "username", "value":  
"username"}, {"name": "password", "confidentialValue": "password"}]}'  
-X POST http://localhost:7001/management/lifecycle/latest/runtimes
```

Example Response

```
HTTP/1.1 201 Created
```

```
Location: http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime
```

```
Response Body:
```

```
{}
```

/lifecycle/{version}/runtimes/{runtime-name}

This resource manages a runtime.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)
- [POST](#)

DELETE

The DELETE method unregisters the runtime identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Unregistering a Runtime

This example uses the DELETE method to unregister a specific runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns information about the runtime identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Runtime](#) entity that contains information about the specified runtime.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing Runtime Information

This example uses the GET method to display information about a specific runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "name": "WLSRuntime",  
  "properties": [  
    {  
      "name": "password",  
      "confidentialValue": "@_Oracle_Confidential_Property_Set_V1.1_#"  
    },  
    {  
      "name": "username",  
      "value": "username"  
    }  
  ],  
  "type": "wls",  
  "protocol": "http",  
  "port": "7001",  
  "hostName": "localhost",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes"  
    }  
  ],  
}
```



```
{
  "rel": "self",
  "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"
},
{
  "rel": "canonical",
  "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"
},
{
  "rel": "partitionCreateForm",
  "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitionCreateForm"
},
{
  "rel": "partitions",
  "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions"
}
]
}
```

POST

The POST method updates the runtime identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Updating a Runtime

This example uses the POST method to update a specific runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"hostName": "localhost", "port": "7001", "properties": [{"name": "username",  
"value": "username"}, {"name": "password", "value": "password"}]}'  
-X POST http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

/lifecycle/{version}/runtimes/{runtime-name}/failoverGroupUpdate

This resource performs a failoverGroup update.

The resource supports the following methods:

- [POST](#)

POST

The POST method performs a failoverGroup update.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Performing a failoverGroup Update

This example uses the POST method to perform a failoverGroup update.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"clusterName" : "wls-cluset", "groupLists": []}' \  
-X POST \  
http://localhost:7001/management/lifecycle/latest/runtimes/runtime/WLSRuntime/failoverGroupUpdate
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{}
```

/lifecycle/{version}/runtimes/{runtime-name}/partitionCreateForm

This resource describes the information needed to create a partition.

The resource supports the following methods:

- [GET](#)

GET

The GET method on this resource returns an empty form for a partition.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [RuntimePartition](#) entity that contains information about fields which must be specified when creating partition.

This method can return the following links:

- **uri**=/partitions **rel**=/partitions

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Get Partition Create Form

This example uses the GET method to get the partition create form.

Example Request

```
curl -v \  
--user admin:admin123 \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitionCreateForm
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "name": null,  
  "properties": [],  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitionCreateForm"  
    },  
    {  
      "rel": "canonical",  
      "href":
```

```
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitionCreateForm"
  },
  {
    "rel": "partitions",
    "href":
      "http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions"
  }
]
```

/lifecycle/{version}/runtimes/{runtime-name}/partitions

This resource manages runtime partitions.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)
- [POST](#)

DELETE

The DELETE method unregisters a partition identified by name.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Unregistering partition

This example uses the DELETE method to unregister partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions?name=SpritePartition
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns a list of partitions for the runtime identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a collection of [RuntimePartition](#) entities.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Viewing a List of Partitions

This example uses the GET method to display a list of partitions for a specific runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSTestRuntime/partitions
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "items": [{  
    "links": [  
      {  
        "rel": "canonical",  
        "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions/SpritePartition"  
      },  
      {  
        "rel": "self",  
        "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions/SpritePartition"  
      }  
    ],  
    "name": "SpritePartition",  
    "id": "429aaa5a-058d-452d-b256-ce874d6e8583"  
  }],  
  "links": [  
    {  
      "rel": "parent",
```

```
        "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"
    },
    {
        "rel": "self",
        "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions"
    },
    {
        "rel": "canonical",
        "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions"
    },
    {
        "rel": "create-form",
        "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitionCreateForm"
    }
    ]
}
```

POST

The POST method creates or registers a partition in the runtime identified by the resource URL. If partition id is passed in model properties, the partition is registered. Otherwise it is created.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Creating a Partition in a Runtime

This example uses the POST method to create a partition in a runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"name":"SpritePartition",  
"properties" : [  
  { "name" : "resourceGroups",  
    "properties" : [  
      { "name" : "g1",  
        "properties" : [  
          { "name" : "useDefaultTarget", "value" : "false" },  
          { "name" : "resourceGroupTemplate", "value" : "template1" },  
          { "name" : "targets" , "values" : ["VirtualHost-0"]}}]},  
      {"name" : "availableTargets" , "values" : ["VirtualHost-0"]}]}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions
```

Example Response

```
HTTP/1.1 201 Created  
  
Location:  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSTestRuntime/partitions/SpritePartition  
  
Response Body:  
{}
```

Example 2 Registering existing Partition in a Runtime

This example uses the POST method to register existing partition in a runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"name":"SpritePartition",  
"id":"4f138249-5e6b-40fe-9c42-a675f027cd9b"}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions
```

Example Response

HTTP/1.1 201 Created

Location:

http://localhost:7001/management/lifecycle/latest/runtimes/WLSTestRuntime/partitions/SpritePartition

Response Body:

```
{}
```

/lifecycle/{version}/runtimes/{runtime-name}/partitions/{partition-name}

This resource manages a runtime partition.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)
- [POST](#)

DELETE

The DELETE method deletes the partition identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Deleting a Partition

This example uses the DELETE method to delete a specific partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions/S  
pritePartition
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns information about the partition identified by the resource URL.

Roles

Administrator

Response Body

The response body returned includes a [RuntimePartition](#) entity that contains information about the specified partition.

This method can return the following links:

- **uri**=/lifecycle/{version}/environments/{environment-name}/partitions/{partition-name} **rel**=environment
- **uri**=/lifecycle/{version}/tenants/{tenant-name} **rel**=tenant

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Viewing a Partition

This example uses the GET method to display information about a specific partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSTestRuntime/partitions/SpritePartition
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "name": "SpritePartition",  
  "id": "429aaa5a-058d-452d-b256-ce874d6e8583",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/par
```



```
titions\SpritePartition"
    },
    {
      "rel": "canonical",
      "href":
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions\SpritePartition"
    },
    {
      "rel": "environment",
      "href":
"http://localhost:7001/management/lifecycle/latest/environments/srpite",
      "title": "name"
    }
  ]
}
```

POST

The POST method updates the partition identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Updating a Partition

This example uses the POST method to update a specific partition.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"properties" : [  
  { "name" : "resourceGroups",  
    "properties" : [  
      { "name" : "g1",  
        "properties" : [  
          { "name" : "useDefaultTarget", "value" : "false" }  
        ]  
      }  
    ]  
  }  
]}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/partitions/S  
pritePartition
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

/lifecycle/{version}/runtimes/{runtime-name}/quiesce

This resource quiesces a runtime.

The resource supports the following methods:

- [POST](#)

POST

The POST method performs a quiesce operation on the specified runtime.

Roles

Administrator

Response Body

The response body returned contains a link to the corresponding task resource.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Quiescing a Runtime

This example uses the POST method to quiesce a runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"phase": "", "properties" : [{"name": "managedserver", "value":  
"managedserver"}]}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/quiesce
```

Example Response

```
HTTP/1.1 202 Accepted
```

Response Body:

```
{"links": [{"  
  "rel": "task",  
  "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/quiesce/task/managedserver/_0_shutdown"  
}]}
```

/lifecycle/{version}/runtimes/{runtime-name}/quiesce/task/{server-name}/{task-name}

This resource manages a quiesce runtime task.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)

DELETE

The DELETE method cancels the task identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Canceling a Task

This example uses the DELETE method to cancel a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/quiesce/task/  
/managedserver/_0_shutdown
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns the status of the task identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Task](#) entity that contains information about the specified task.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Viewing Task Status

This example uses the GET method to display information about a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/quiesce/task/  
/managedserver/_0_shutdown
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{  
  "status": "TASK COMPLETED",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/star  
t/task/managedserver/_0_shutdown"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/star  
t/task/managedserver/_0_shutdown"
```

```
}  
  ]  
}
```


/lifecycle/{version}/runtimes/{runtime-name}/scaleDown

This resource is used to scale down a runtime.

The resource supports the following methods:

- [POST](#)

POST

The POST method performs a scale down operation on the specified runtime. When you scale down a runtime, origin servers that are associated with the runtime are removed from the origin server pool.

Roles

Administrator

Response Body

The response body returned contains a link to the corresponding task resource.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Scaling Down a Runtime

This example uses the POST method to scale down a runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"scaleFactor": "2", "properties" : [{"name": "cluster-name", "value" : "wls_\  
cluster"}]}' \  
-X POST \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/scaleDown
```

Example Response

```
HTTP/1.1 202 Accepted
```

Response Body:

```
{"links": [{"  
  "rel": "task",  
  "href":  
  "http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/scaleDown/task/wls-cluster/ScaleDown_2"  
}]}
```

/lifecycle/{version}/runtimes/{runtime-name}/scaleUp

This resource is used to scale up a runtime.

The resource supports the following methods:

- [POST](#)

POST

The POST method performs a scale up operation on the specified runtime. When you scale up a runtime, Oracle Traffic Director creates new origin servers and adds them to the existing origin server pool.

Roles

Administrator

Response Body

The response body returned contains a link to the corresponding task resource.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Scaling Up a Runtime

This example uses the POST method to scale up a runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"scaleFactor": "2", "properties" : [{"name": "cluster-name", "value" : "wls_\  
cluster"}]}' \  
-X POST \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/scaleUp
```

Example Response

```
HTTP/1.1 202 Accepted
```

Response Body:

```
{"links": [{"  
  "rel": "task",  
  "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/sca  
leUp/task/wls-cluster/ScaleUp_1"  
}]}
```

/lifecycle/{version}/runtimes/{runtime-name}/scaleUp/task/{cluster-name}/{task-name}

This resource manages a scaled up task.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)

DELETE

The DELETE method cancels the task identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Canceling a Task

This example uses the DELETE method to cancel a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/scaleUp/task  
/wls-cluster/ScaleUp_1
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns the status of the task identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Task](#) entity that contains information about the specified task.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Viewing Task Status

This example uses the GET method to display status information about a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/ScaleUp/task/  
wls-cluster/ScaleUp_1
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{  
  "status": "TASK COMPLETED",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/sca  
leDown/task/wls-cluster/ScaleUp_1"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/sca  
leDown/task/wls-cluster/ScaleUp_1"
```

```
}  
  ]  
}
```


/lifecycle/{version}/runtimes/{runtime-name}/start

This resource is used to start a runtime.

The resource supports the following methods:

- [POST](#)

POST

The POST method performs a start operation on the specified runtime.

Roles

Administrator

Response Body

The response body returned contains a link to the corresponding task resource.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Starting A Runtime

This example uses the POST method to start a runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"phase": "", "properties" : [{"name": "managedserver", "value":  
"managedserver"}]}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/start
```

Example Response

```
HTTP/1.1 202 Accepted
```

Response Body:

```
{"links": [{"  
  "rel": "task",  
  "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/start/task/managedserver/_1_start"  
}]}
```

/lifecycle/{version}/runtimes/{runtime-name}/start/task/{server-name}/{task-name}

This resource manages a scaled up task.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)

DELETE

The DELETE method cancels the task identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Canceling a Task

This example uses the DELETE method to cancel a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/start/task/managedserver/_1_start
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns the status of the task identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Task](#) entity that contains information about the specified task.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Viewing the Status of a Task

This example uses the GET method to display information about a specific task.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET  
http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/start/task/managedserver/_1_start
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{  
  "status": "TASK COMPLETED",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/start/task/managedserver/_1_start"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/start/task/managedserver/_1_start"
```

```
}  
  ]  
}
```

/lifecycle/{version}/runtimes/{runtime-name}/sync

This resource synchronizes the environment.

The resource supports the following methods:

- [POST](#)

POST

The POST method synchronizes the runtime.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Synchronizes the runtime

This example uses the POST method to synchronize the runtime.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '' \  
-X POST http://localhost:7001/management/lifecycle/latest/runtimes/WLSRuntime/sync
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:
```

```
{}
```


/lifecycle/{version}/sync

This resource synchronizes all environments.

The resource supports the following methods:

- [POST](#)

POST

The POST method synchronizes or configures periodic synchronization of all environments.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the Standard HTTP Status codes.

Example

Example 1 Synchronizes all environments.

Synchronizes all environments. This example uses the POST method to synchronize all environments.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '' \  
-X POST http://localhost:7001/management/lifecycle/latest/sync
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

Example 2 Configure periodic synchronization of all environments.

Configure periodic synchronization of all environments. This example uses the POST method to configure periodic synchronization of all environments.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{  
    "sync": "true",  
    "syncInterval": "1"}' \  
-X POST http://localhost:7001/management/lifecycle/latest/sync
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:
{}

/lifecycle/{version}/tenantcreateForm

This resource describes the information needed to register a tenant.

The resource supports the following methods:

- [GET](#)

GET

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Tenant](#) entity that contains information about fields that must be specified when registering a tenant.

This method can return the following links:

- **uri**=/tenants **rel**=/tenants

The GET method on this resource returns an empty form for a tenant.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Getting a Create Form

This example uses the GET method to get a tenant create form.

Example Request

```
curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET http://localhost:7001/management/lifecycle/latest/tenantCreateForm
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{
  "topLevelDir": null,
  "name": null,
  "id": null,
  "links": [
    {
      "rel": "parent",
      "href": "http://localhost:7001/management/lifecycle/latest"
    },
    {
      "rel": "self",
      "href":
"http://localhost:7001/management/lifecycle/latest/tenantCreateForm"
    },
    {
      "rel": "canonical",
      "href":
"http://localhost:7001/management/lifecycle/latest/tenantCreateForm"
    },
    {
      "rel": "tenants",
```

GET

```
        "href":  
        "http://localhost:7001/management/lifecycle/latest/tenants"  
    }  
    ]  
}
```

/lifecycle/{version}/tenants

This resource manages tenants.

The resource supports the following methods:

- [GET](#)
- [POST](#)

GET

The GET method on this resource returns a list of tenants.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a collection of [Tenant](#) entities.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing Tenants

This example uses the GET method to display tenants.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/lifecycle/latest/tenants
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "items": [{  
    "links": [  
      {  
        "rel": "canonical",  
        "href":  
"http://localhost:7001/lifecycle/latest/tenants/Sprite"  
      },  
      {  
        "rel": "self",  
        "href":  
"http://localhost:7001/lifecycle/latest/tenants/Sprite"  
      }  
    ],  
    "topLevelDir": "Sprite/top/level/dir",  
    "name": "Sprite",  
    "id": "123"  
  }],  
  "links": [  
    {  
      "rel": "parent",  
      "href": "http://localhost:7001/lifecycle/latest"  
    },  
    {  
      "rel": "self",
```



```
    "href": "http://localhost:7001/lifecycle/latest/tenants"
  },
  {
    "rel": "canonical",
    "href": "http://localhost:7001/lifecycle/latest/tenants"
  },
  {
    "rel": "create-form",
    "href": "http://localhost:7001/lifecycle/latest/tenantcreateForm"
  }
]
}
```

POST

The POST method registers a tenant.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Registering a Tenant

This example uses the POST method to register a tenant.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"uuid" : "55ec0a13-7152-4040-8352-ad1e7726bad9", "name" : "Sprite",  
"topLevelDir": "Sprite/top/level/dir" }' \  
-X POST http://localhost:7001/lifecycle/latest/tenants
```

Example Response

```
HTTP/1.1 201 Created
```

```
Location: http://localhost:7001/lifecycle/latest/tenants/Sprite
```

```
Response Body:
```

```
{}
```

/lifecycle/{version}/tenants/{tenant-name}

This resource manages tenants.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)
- [POST](#)

DELETE

The DELETE method unregisters the tenant identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Unregistering a Tenant

This example uses the DELETE method to unregister a specific tenant.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE http://localhost:7001/management/lifecycle/latest/tenants/Sprite
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns information about the tenant identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Tenant](#) entity that contains information about the specified tenant.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing a Tenant

This example uses the GET method to display information about a specific tenant.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/tenants/Sprite
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "topLevelDir": "Sprite\top\level\dir",  
  "name": "Sprite",  
  "id": "123",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/sprite"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/sprite"  
    },  
    {  
      "rel": "serviceCreateForm",
```

```
        "href":
"http://localhost:7001/management/lifecycle/latest/tenants/sprite/serviceC
reateForm"
    },
    {
        "rel": "services",
        "href":
"http://localhost:7001/management/lifecycle/latest/tenants/sprite/services
"
    }
]
}
```

POST

The POST method updates the tenant identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Updating a Tenant

This example uses the POST method to update a specific tenant.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"topLevelDir": "Sprite/top/level/dir" }' \  
-X POST http://localhost:7001/management/lifecycle/latest/tenants/Sprite
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:
```

```
{}
```

/lifecycle/{version}/tenants/{tenant-name}/servicecreateForm

This resource describes information needed to on-board a tenant to a service.

The resource supports the following methods:

- [GET](#)

GET

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Service](#) entity that contains information about fields that must be specified when on-boarding a tenant to a service.

This method can return the following links:

- **uri**=/services **rel**=/services

The GET method on this resource returns an empty form for a service.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Getting a Create Form

This example uses the GET method to get a service create form.

Example Request

```
curl -v \
--user admin:admin123 \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET
http://localhost:7001/management/lifecycle/latest/tenants/sprite/serviceCreateForm
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{
  "environmentRef": null,
  "name": null,
  "type": null,
  "links": [
    {
      "rel": "parent",
      "href":
"http://localhost:7001/management/lifecycle/latest/tenants/sprite"
    },
    {
      "rel": "self",
      "href":
"http://localhost:7001/management/lifecycle/latest/tenants/sprite/serviceC
reateForm"
    },
    {
      "rel": "canonical",
      "href":
"http://localhost:7001/management/lifecycle/latest/tenants/sprite/serviceC
```

```
    reateForm"
      },
      {
        "rel": "services",
        "href":
        "http:\\\\localhost:7001\\management\\lifecycle\\latest\\tenants\\sprite\\services
        "
      }
    ]
  }
}
```

/lifecycle/{version}/tenants/{tenant-name}/services

This resource manages a tenant's services.

The resource supports the following methods:

- [GET](#)
- [POST](#)

GET

The GET method on this resource returns a list of services of the tenant identified in the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a collection of [Service](#) entities.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing Services

This example uses the GET method to display services for a tenant.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "items": [{  
    "links": [  
      {  
        "rel": "canonical",  
        "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite"  
      },  
      {  
        "rel": "self",  
        "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite"  
      }  
    ],  
    "topLevelDir": "Sprite/top/level/dir",  
    "name": "Sprite",  
    "id": "123"  
  }],  
  "links": [  
    {  
      "rel": "parent",  
      "href": "http://localhost:7001/management/lifecycle/latest"  
    },  
    {
```

```
        "rel": "self",
        "href":
"http://localhost:7001/management/lifecycle/latest/tenants"
    },
    {
        "rel": "canonical",
        "href":
"http://localhost:7001/management/lifecycle/latest/tenants"
    },
    {
        "rel": "create-form",
        "href":
"http://localhost:7001/management/lifecycle/latest/tenantcreateForm"
    }
    ]
}
```

POST

The POST method on-boards a tenant to a service.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 On-boarding a Tenant

This example uses the POST method to on-board a tenant to a service.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"uuid" : "30ec0a13-7102-4040-8352-adle7726bad9", "name" : "HCMProd", "type" :  
"HCMService", "environmentRef" : "sprite", "topLevelDir": "sprite/top/level/dir"}'  
\   
-X POST http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services
```

Example Response

```
HTTP/1.1 201 Created
```

```
Location:
```

```
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMProd
```

```
Response Body:
```

```
{}
```

/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}

This resource manages a tenant's service.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)
- [POST](#)

DELETE

The DELETE method removes a service for a tenant.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Removing a Service

This example uses the DELETE method to remove a service for a tenant.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMProd
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:
```

```
{}
```


GET

The GET method on this resource returns information about the service identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [Service](#) entity that contains information about the specified service.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing a Service

This example uses the GET method to display information about a specific service.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMProd
```

Example Response

HTTP/1.1 200 OK

Response Body:

```
{  
  "uuid": "30ec0a13-7102-4040-8352-ad1e7726bad9",  
  "topLevelDir": "sprite/top/level/dir",  
  "environmentRef": "sprite",  
  "name": "HCMProd",  
  "type": "HCMService",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/  
/HCMProd"  
    },  
    {  
      "rel": "canonical",
```

```
        "href":
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services
/HCMPProd"
    },
    {
        "rel": "PDBCreateForm",
        "href":
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services
/HCMPProd/PDBCreateForm"
    },
    {
        "rel": "PDB",
        "href":
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services
/HCMPProd/PDB"
    }
]
}
```

POST

The POST method updates the service identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Updating a Service

This example uses the POST method to update specific service.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-H Content-Type:application/json \  
-d '{"topLevelDir": "sprite/top/level/dir"}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMProd
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{}
```

/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}/PDB

This resource manages PDB for a service.

The resource supports the following methods:

- [DELETE](#)
- [GET](#)
- [POST](#)

DELETE

The DELETE method removes PDB from the service identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Removing PDB From a Service

This example uses the DELETE method to remove PDB from a specified service.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X DELETE \  
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMPProd/  
PDB
```

Example Response

```
HTTP/1.1 200 OK
```

```
Response Body:  
{}
```

GET

The GET method on this resource returns information about the PDB for the service identified by the resource URL.

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [PDB](#) entity that contains information about the specified service.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Viewing PDB Information

This example uses the GET method to display PDB information about a specific service.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET  
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMPProd/  
PDB
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "status": "ACTIVE",  
  "name": "SpriteHCMPPDBProd",  
  "id": "444",  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services  
\HCMPProd"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services  
\HCMPProd/PDB"  
    },  
    {  
      "rel": "canonical",
```

```
        "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services  
\HCMProd/PDB"  
    },  
    {  
        "rel": "create-form",  
        "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services  
\HCMProd/PDBcreateForm"  
    }  
  ]  
}
```

POST

The POST method adds PDB for the service identified by the resource URL.

Roles

Administrator

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Adding PDB for a Service

This example uses the POST method to add PDB for a service.

Example Request

```
curl -v \  
--user username:password \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-d '{"name" : "spriteHCMProd", "id" : "444", "status" : "ACTIVE"}' \  
-X POST  
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMProd/  
PDB
```

Example Response

```
HTTP/1.1 201 Created
```

```
Location:
```

```
http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMProd/  
PDB
```

```
Response Body:
```

```
{}
```


/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}/PDBCreateForm

This resource describes information needed to add PDB to a service.

The resource supports the following methods:

- [GET](#)

GET

Roles

Administrator, Deployer, Operator, Monitor

Response Body

The response body returned includes a [PDB](#) entity that contains information about fields that must be specified when adding PDB to a service.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1 Getting a Create Form

This example uses the GET method to get the PDB create form.

Example Request

```
curl -v \  
--user admin:admin123 \  
-H X-Requested-By:MyClient \  
-H Accept:application/json \  
-X GET \  
http://localhost:7001/management/lifecycle/latest/tenants/sprite/services/CRMPProd/  
PDBCreateForm
```

Example Response

```
HTTP/1.1 200 OK
```

Response Body:

```
{  
  "status": "ACTIVE",  
  "name": null,  
  "id": null,  
  "links": [  
    {  
      "rel": "parent",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/sprite/services  
\CRMPProd"  
    },  
    {  
      "rel": "self",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/sprite/services  
\CRMPProd/PDBCreateForm"  
    },  
    {  
      "rel": "canonical",  
      "href":  
"http://localhost:7001/management/lifecycle/latest/tenants/sprite/services  
\CRMPProd/PDBCreateForm"  
    }  
  ]  
}
```

```
        "rel": "PDB",
        "href":
"\"http://localhost:7001/management/lifecycle/latest/tenants/sprite/services
/CRMPProd/PDB\"
    }
  ]
}
```


The following sections describe the data models on which the REST resources operate. These data models describe the information exchanged in REST resources.

The descriptions of these data models in this section include a description of the constraints that apply to each data field.

Constraints on Data Fields

Confidential

Specifies that this value is encrypted and will not be returned by a `GET` method. It may be specified in a `POST` for update.

Immutable

Specifies that the contents of this field can be written once, during creation, and may not be modified thereafter.

Not Null

Specifies that a value must be specified for this field.

Read Only

Specifies that this value may be read by a `GET`, but is ignored during a `POST`.

AssociatePartitions

The `AssociatePartitions` entity includes the names of two partitions to associate with each other and arbitrary properties.

The properties of the `AssociatePartitions` entity are as follows:

partition1Name

The name of one of the partitions to associate.

Type: string

partition1RuntimeName

The runtime name of one of the partitions to associate.

Type: string

partition2Name

The name of the other partition to associate.

Type: string

partition2RuntimeName

The runtime name of the other partition to associate.

Type: string

properties

Arbitrary properties. May include jdbcSystemResource and jdbcSystemResourceOverride properties.

Type: array

DissociatePartitions

The DissociatePartitions entity includes the names of two partitions to dissociate and arbitrary properties.

The properties of the DissociatePartitions entity are as follows:

partition1Name

The name of one of the partitions to dissociate.

Type: string

partition1RuntimeName

The runtime name of one of the partitions to dissociate.

Type: string

partition2Name

The name of the other partition to dissociate.

Type: string

partition2RuntimeName

The runtime name of the other partition to dissociate.

Type: string

properties

Arbitrary properties. These may include jdbcSystemResource and jdbcSystemResourceOverride properties.

Type: array

Environment

The Environment entity includes the environment name and represents the logical container for tenant resources, that is, partitions, PDBs and Services.

The properties of the Environment entity are as follows:

name

The name of the environment.

Type: string

Constraints: Immutable

orchestration

Create an environment from an orchestration.

Type: Orchestration

Constraints: Immutable

EnvironmentPartition

The EnvironmentPartition entity includes the name and runtime name, and represents a logical partition added to the environment.

The properties of the EnvironmentPartition entity are as follows:

name

The name of the partition.

Type: string

Constraints: Immutable

runtimeName

The name of the runtime on which the partition exists.

Type: string

Constraints: Immutable

FailoverGroupUpdate

The FailoverGroup entity has a cluster name and group lists for updateFailoverGroup action.

The properties of the FailoverGroupUpdate entity are as follows:

clusterName

The cluster name.

Type: string

Constraints: Immutable

groupLists

A list of group lists.

Type: array

MigratePartition

A MigratePartition entity includes a runtime name, a phase and arbitrary properties.

The properties of the MigratePartition entity are as follows:

phase

The phase.

Type: string

Constraints: Immutable

properties

Arbitrary properties passed to a plugin.

Type: array

Constraints: Immutable

runtimeName

Name of the runtime on which the partition exists.

Type: string

Constraints: Immutable

Orchestration

The Orchestration entity includes the name and args for the environment to be created.

The properties of the Orchestration entity are as follows:

args

Arbitrary properties. These may include the username and password that are required to connect to the runtime.

Type: array

name

The name of the environment.

Type: string

Constraints: Immutable

PDB

A PDB entity includes name, id, status and create date.

The properties of the PDB entity are as follows:

creationDate

The creation date of the PDB.

Type: string

Constraints: Read Only

id

The id of the PDB.

Type: string

Constraints: Immutable

name

The name of the PDB.

Type: string

Constraints: Immutable

status

The status of the PDB. (Not used, reserved for future use)

Type: string

Constraints: Immutable

Property

A Property entity holds a named property, where the value can be a String, a confidential String, or a list of Properties.

The properties of the Property entity are as follows:

confidentialValue

Property Confidential String value.

Type: string

properties

Property Properties value.

Type: array

Constraints: Immutable

values

Property String values.

Type: array

QuiescePartition

The QuiescePartition entity includes arbitrary properties for the quiescing action.

The properties of the QuiescePartition entity are as follows:

phase

The phase.

Type: string

Constraints: Immutable

properties

Arbitrary properties passed to a plugin.

Type: array

QuiesceRuntime

The QuiesceRuntime entity includes the phase and arbitrary properties for the quiescing action.

The properties of the QuiesceRuntime entity are as follows:

phase

The phase.

Type: string

Constraints: Immutable

properties

Arbitrary properties passed to a plugin.

Type: array

RegisterPlugin

The RegisterPlugin entity includes the name and relative path to a Java archive on a server file system.

The properties of the RegisterPlugin entity are as follows:

name

The name of the plugin.

Type: string

Constraints: Immutable

path

The server relative pathname to the jar file containing the plugin.

Type: string

Runtime

The Runtime entity includes the name, type, host and port for a physical runtime. It can also include arbitrary properties.

The properties of the Runtime entity are as follows:

hostName

The hostname of the Runtime.

Type: string

Constraints: Immutable

name

The name of the Runtime.

Type: string

Constraints: Immutable

port

The port of the Runtime.

Type: string

Constraints: Immutable

properties

Arbitrary properties. These may include the username and password that are required to connect to the runtime.

Type: array

Constraints: Immutable

protocol

The protocol of the Runtime.

Type: string

Constraints: Immutable

type

The type of the Runtime.

Type: string

Constraints: Immutable

RuntimePartition

The RuntimePartitions entity includes the name and ID of the partition on a physical runtime. It can also include arbitrary properties.

The properties of the RuntimePartition entity are as follows:

id

The ID of the partition.

Type: string

Constraints: Read Only

name

The name of the partition.

Type: string

Constraints: Immutable

properties

Arbitrary properties passed to a plugin.

Type: array

type

The type of the partition. (Not used, reserved for future use)

Type: string

Constraints: Read Only

ScaleRuntime

The ScaleRuntime entity includes a factor and arbitrary properties for a scaling action.

The properties of the ScaleRuntime entity are as follows:

properties

Arbitrary properties passed to a plugin.

Type: array

scaleFactor

The scale factor number.

Type: string

Constraints: Immutable

Service

The Service entity includes the name, the type and a reference to the environment.

The properties of the Service entity are as follows:

environmentRef

The environment name of the service.

Type: string

Constraints: Immutable

identityDomain

The identity domain of the service.

Type: string

Constraints: Immutable

name

The name of the service.

Type: string

Constraints: Immutable

topLevelDir

The services's top level directory.

Type: string

twoTask

The service's two task.

Type: string

type

The type of the service.

Type: string

Constraints: Immutable

uuid

Universally unique identifier of the service.

Type: string

Constraints: Immutable

StartPartition

The StartPartition entity includes arbitrary properties for starting an action.

The properties of the StartPartition entity are as follows:

phase

The phase.

Type: string

Constraints: Immutable

properties

Arbitrary properties passed to a plugin.

Type: array

StartRuntime

The StartRuntime entity includes the phase and arbitrary properties for a starting action.

The properties of the StartRuntime entity are as follows:

phase

The phase.

Type: string

Constraints: Immutable

properties

Arbitrary properties passed to a plugin.

Type: array

SyncOptions

The SyncOptions entity includes the sync and interval values.

The properties of the SyncOptions entity are as follows:

sync

Periodic sync on/off value.

Type: string

Constraints: Immutable

syncInterval

Periodic sync interval in hours.

Type: string

Constraints: Immutable

Task

The Task entity includes the status of the task.

The properties of the Task entity are as follows:

error

The task error if available.

Type: string

Constraints: Read Only

progress

The task progress.

Type: string

Constraints: Read Only

status

The task status.

Type: string

Constraints: Read Only

Tenant

The Tenant entity includes the top level directory, ID and name of the tenant.

The properties of the Tenant entity are as follows:

id

The ID of the tenant.

Type: string

Constraints: Immutable

name

The name of the tenant.

Type: string

Constraints: Immutable

topLevelDir

The tenant's top level directory for partitions.

Type: string

uuid

Universally unique identifier of the tenant.

Type: string

Constraints: Immutable

Version

A Version entity describes a version of the lifecycle resources.

The properties of the Version entity are as follows:

isLatest

True if this is the default version.

Type: boolean

Constraints: Read Only

lifecycle

Type: string

state

The lifecycle of this version: active or deprecated

Type: string

Constraints: Read Only

version

The name of this version.

Type: string

Constraints: Read Only