This reference describes a RESTful management API for managing lifecycle resources in a WebLogic Server domain and includes JRF lifecycle resources.
Contents

Preface .................................................................................................................................................................... v

1 Resources ........................................................................................................................................................................... 1-2
/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}/deprovision ................................................................. 1-22
/lifecycle/{version}/environments/{environment-name}/dissociatePartitions .................................................. 1-25
/lifecycle/{version}/environments/{environment-name}/partitionCreateForm .................................................. 1-27
/lifecycle/{version}/environments/{environment-name}/partitions ........................................................................ 1-30
/lifecycle/{version}/environments/{environment-name}/runtimes/{runtime-name}/partitions/{partition-name} 1-34
/lifecycle/{version}/runtimeCreateForm .................................................................................................................... 1-38
/lifecycle/{version}/runtimes ........................................................................................................................................ 1-41
/lifecycle/{version}/runtimes/{runtime-name} ............................................................................................................. 1-45
/lifecycle/{version}/runtimes/{runtime-name}/partitionCreateForm ........................................................................ 1-50
/lifecycle/{version}/runtimes/{runtime-name}/partitions ............................................................................................. 1-53
/lifecycle/{version}/runtimes/{runtime-name}/partitions/{partition-name} ............................................................ 1-59
/lifecycle/{version}/tenantCreateForm .......................................................................................................................... 1-64
/lifecycle/{version}/tenants ............................................................................................................................................ 1-67
/lifecycle/{version}/tenants/{tenant-name} .................................................................................................................. 1-71
/lifecycle/{version}/tenants/{tenant-name}/serviceCreateForm .................................................................................. 1-76
/lifecycle/{version}/tenants/{tenant-name}/services ...................................................................................................... 1-79
/lifecycle/{version}/tenants/{tenant-name}/services/{service-name} ........................................................................ 1-83
/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}/PDB ................................................................ 1-88
/lifecycle/{version}/tenants/{tenant-name}/services/{service-name}/PDBCreateForm ........................................ 1-93
2 Entities

AssociatePartitions ................................................................. 2-1
Deprovision ............................................................................. 2-2
DissociatePartitions ............................................................... 2-2
Environment ........................................................................... 2-3
EnvironmentPartition ............................................................. 2-3
Orchestration .......................................................................... 2-3
PDB .......................................................................................... 2-4
Property .................................................................................... 2-4
Runtime ..................................................................................... 2-4
RuntimePartition ................................................................. 2-5
Service ..................................................................................... 2-6
Tenant ...................................................................................... 2-7
Version ..................................................................................... 2-7
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:

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

**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": false,
    "lifecycle": "deprecated"
  },
  {
    "links": []
  }
}
```
GET

```
{
    "rel": "canonical",
    "href": "http:\/\localhost:7001\management\lifecycle\12.2.1.3.0"
},
{
    "rel": "self",
    "href": "http:\/\localhost:7001\management\lifecycle\12.2.1.3.0"
}
"version": "12.2.1.3.0",
"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\12.2.1.3.0"
    }
]}
```
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}/runtimeCreateForm rel=runtimeCreateForm
- uri=/lifecycle/{version}/runtimes rel=runtimes

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 \
  -u admin:admin123 \
  -H X-Requested-By:MyClient \
  -H Accept:application/json \

Example Response

HTTP/1.1 200 OK

Response Body:

```json
{
  "version": "12.2.1.3.0",
  "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'}
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': "http://\localhost:7001\management\lifecycle\latest\environments"  
    }  
  ]  
}
GET

"http://\localhost:7001\management\lifecycle\latest\environments"
} 
} 
} 

1-10  RESTful Lifecycle Reference for Oracle WebLogic Server
/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 \

Example Response

HTTP/1.1 200 OK

Response Body:

```json
{
  "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"
    }
  ]
}
```
POST

The POST method creates an environment.

Roles

Administrator

Request Body

The request body must include a fully populated Environment entity.

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

Example Response

HTTP/1.1 201 Created


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 \n--user username:password \n-H X-Requested-By:MyClient \n-H Accept:application/json \n-d '{ "name": "coke", "orchestration": {"name": "orchestration1", "args": [ {"name": "wls", "properties": [ {"name": "partitionName", "value": "wlspartition1"}, } ] } }' \n-X POST http://localhost:7001/management/lifecycle/latest/environments
Example Response

HTTP/1.1 201 Created


Response Body:

{}
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 \

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

```bash
curl -v \
--user username:password \
-H X-Requested-By:MyClient \
-H Accept:application/json \
```

Example Response

HTTP/1.1 200 OK

Response Body:

```json
{
  "name": "sprite",
  "links": [
    {
      "rel": "parent",
    },
    {
      "rel": "self",
    },
    {
      "rel": "canonical",
    },
    {
      "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

Request Body

The request body must include a fully populated `AssociatePartitions` entity.

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 \
-X POST


Example Response

HTTP/1.1 200 OK

Response Body:

{}
This resource cleans up and removes the environment which includes deprovisioning, disassociating, and removing partitions.

The resource supports the following methods:

- POST
POST

The POST method deprovisions and disassociates partitions in the named environment. The named environment is then deleted.

Roles

Administrator

Request Body

The request body must include a fully populated Deprovision entity.

Response Body

An empty response body is returned.

Response Codes

This method returns one of the standard HTTP status codes.

Example

Example 1  Environment cleanup, including deprovisioning
This example uses the POST method to clean up the environment which includes deprovisioning the partition, deleting the partition, and deleting the environment.

Example Request

curl -v \
--user username:password \n-H X-Requested-By:MyClient \n-H Accept:application/json \n-H Content-Type:application/json \n-d '{
  "orchestration": {
    "name": "deleteAll",
    "args": [
      {
        "name": "wls",
        "properties": {
          "component": {
            "name": "MockComponent",
            "properties": {
              "configurableAttributes": {
                "name": "MockComponent",
                "properties": {
                  "mockUser": "jennifer"
                }
              }
            }
          }
        }
      },
      {
        "name": "partitionName",
        "value": "wlspartition1"
      }
    ]
  }
} -X POST

Example Response

HTTP/1.1 200 OK
POST

Response Body:
{}

1-24  RESTful Lifecycle Reference for Oracle WebLogic Server
This resource allows you to dissociate associated partitions.
The resource supports the following methods:

- POST
POST

The POST method dissociates two partitions.

Roles

Administrator

Request Body

The request body must include a fully populated DissociatePartitions entity.

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

Example Response

HTTP/1.1 200 OK

Response Body:

{}
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:

```json
{
  "runtimeName": null,
  "name": null,
  "links": [
    {
      "rel": "parent",
    },
    {
      "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'
}
]}

Resources  1-29
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

```bash
curl -v \
--user username:password \
-H X-Requested-By:MyClient \
-H Accept:application/json \
-X GET \
```

Example Response

HTTP/1.1 200 OK

Response Body:

```json
{
  "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"
    }
  ],
  "runtimeName": "WLSRuntime",
  "name": "SpritePartition"
}
```
"links": [  
    {  
      "rel": "parent",  
    },  
    {  
      "rel": "self",  
    },  
    {  
      "rel": "canonical",  
    },  
    {  
      "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

Request Body

The request body must include a fully populated EnvironmentPartition entity.

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

Example Response

HTTP/1.1 201 Created

Location:

Response Body:
()

Resources 1-33
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 \
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

Example Response

HTTP/1.1 200 OK

Response Body:

```json
{
  "runtimeName": "WLSRuntime",
  "name": "SpritePartition",
  "links": [
    {
      "rel": "parent",
    },
    {
      "rel": "self",
    },
    {
      "rel": "canonical",
    }
  ]
}
```
times\WLSRuntime\partitions\SpritePartition"}
]}
)
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'}
    ]
}
```
GET

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

Example Response

HTTP/1.1 200 OK

Response Body:

```json
{
  "items": [
    {
      "links": [
        {
          "rel": "canonical",
        },
        {
          "rel": "self",
        }
      ],
      "name": "WLSRuntime",
      "properties": [
        {
          "name": "password",
          "value": "password"
        },
        {
          "name": "username",
          "value": "username"
        }
      ]
    }
  ]
}
```
"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

Request Body

The request body must include a fully populated Runtime entity.

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

```bash
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"}]}' \
```

Example Response

```
HTTP/1.1 201 Created

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

Response Body:

```json
{}
```
/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 \
```

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 \

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/WLSRuntime'
    }
  ]}
POST

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

Roles

Administrator

Request Body

The request body must include a fully populated Runtime entity.

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"}]}' \

Example Response

HTTP/1.1 200 OK

Response Body:

{}
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:

```json
{
  '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"
}
}
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 \
-X Requested-By:MyClient \
-X Accept:application/json \
-X DELETE \

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

Example Response

HTTP/1.1 200 OK

Response Body:

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


{
   "rel": "self",
}

{
   "rel": "canonical",
}

{
   "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

Request Body

The request body must include a fully populated RuntimePartition entity.

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"]}
],
"name": "SpritePartition"}
}' \
-X POST \

Example Response

HTTP/1.1 201 Created

Location: 

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 \

Example Response

HTTP/1.1 201 Created

Location:

Response Body:
{}


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

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 \ 

Example Response

HTTP/1.1 200 OK

Response Body:

```json
{
   "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"
}
```
GET

```
issions\SpritePartition",
},
{
  "rel": "canonical",
  "href": "http:\\//localhost:7001\management\lifecycle\latest\runtimes\WLSRuntime\par
titions\SpritePartition",
},
{
  "rel": "environment",
  "href": "http:\\//localhost:7001\management\lifecycle\latest\environments\sprite",
  "title": "name"
}
}
```
POST

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

Roles

Administrator

Request Body

The request body must include a fully populated RuntimePartition entity.

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

Example Response

HTTP/1.1 200 OK

Response Body:

{ }
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 \n  -H X-Requested-By:MyClient \n  -H Accept:application/json \n  -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

```bash
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\/Sprite"
      }
    ]
}
```
'href': 'http://localhost:7001/lifecycle/latest/tenants'
},
{
'href': 'http://localhost:7001/lifecycle/latest/tenants'
},
{
'href': 'http://localhost:7001/lifecycle/latest/tenantCreateForm'
}]}
POST

The POST method registers a tenant.

Roles

Administrator

Request Body

The request body must include a fully populated Tenant entity.

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 \
   -u 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:

{ }
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 \

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 \

Example Response

HTTP/1.1 200 OK

Response Body:

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

"href": "http://\//localhost:7001\//management\//lifecycle\//latest\//tenants\//sprite\//serviceCreateForm"
},
{
    "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

Request Body

The request body must include a fully populated Tenant entity.

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

Example Response

HTTP/1.1 200 OK

Response Body:

()
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 \n-H X-Requested-By:MyClient \n-H Accept:application/json \n-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',
    },
    {
      'rel': 'self',
      'href': 'http://localhost:7001/management/lifecycle/latest/tenants/sprite/serviceCreateForm',
    },
    {
      'rel': 'canonical',
      'href': 'http://localhost:7001/management/lifecycle/latest/tenants/sprite/serviceCreateForm',
    }
  ]
}
GET

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

Example Response

HTTP/1.1 200 OK

Response Body:

{
   "items": [{
      "links": [
      {
         "rel": "canonical",
      },
      {
         "rel": "self",
      }
   ],
   "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''
}
]
The POST method on-boards a tenant to a service.

Roles

Administrator

Request Body

The request body must include a fully populated Service entity.

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-ad1e7726bad9", "name" : "HCMProd", 'type' : "HCMService", 'environmentRef' : "sprite", 'topLevelDir': "sprite/top/level/dir"}' \
```

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  

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 \n-H X-Requested-By:MyClient \n-H Accept:application/json \n-X GET 

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/HCMProd"
        }
    ]
}
```
POST

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

Roles

Administrator

Request Body

The request body must include a fully populated Service entity.

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

Example Response

HTTP/1.1 200 OK

Response Body:

{}
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

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 \

Example Response

HTTP/1.1 200 OK

Response Body:

{  
"status": "ACTIVE",  
"name": "SpriteHCMProdDB",  
"id": "444",  
"links": [  
{  
"rel": "parent",  
"href":  
"http://localhost:7001/management/lifecycle/latest/tenants/Sprite/services/HCMProd"  
},  
{  
"rel": "self",  
"href":  
},  
{  
"rel": "canonical",  
"href":  
}  
]  
}
'href':
'http:\/\//localhost:7001\management\lifecycle\latest\tenants\Sprite\services\HCMP Prod\PDB'
},
{
  'rel': 'create-form',
  'href':
  'http:\/\//localhost:7001\management\lifecycle\latest\tenants\Sprite\services\HCMP Prod\PDBCreateForm'
}
POST

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

Roles

Administrator

Request Body

The request body must include a fully populated PDB entity.

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" : "spriteHCMPDBProd", "id" : 444, "status" : "ACTIVE"}' \
   -X POST 

Example Response

HTTP/1.1 201 Created

Location:

Response Body:

{ }
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
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/CRMProd"
    },
    {
      "rel": "self",
      "href": "http://localhost:7001/management/lifecycle/latest/tenants/sprite/services/CRMProd/PDBCreateForm"
    },
    {
      "rel": "canonical",
      "href": "http://localhost:7001/management/lifecycle/latest/tenants/sprite/services/CRMProd/PDBCreateForm"
    }
  ]
}
'rel': 'PDB',
'href': 'http:\/\localhost:7001\management\lifecycle\latest\tenants\sprite\services\CRMProd\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
### Deprovision

The Deprovision entity represents the logical container for tenant resources, that is, partitions, PDBs and Services.

The properties of the Deprovision entity are as follows:

- **partition2RuntimeName**
  The runtime name of the other partition to associate.
  
  **Type**: string

- **properties**
  Arbitrary properties. May include jdbcSystemResource and jdbcSystemResourceOverride properties.
  
  **Type**: array of Property

### 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 of Property
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

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 of Property

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 of Property
*Constraints*: Immutable

**values**
Property String values.
*Type*: array of Property

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 of Property*
*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*
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 service'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

**properties**
Arbitrary properties passed to a plugin.
Type: array of Property

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