LifeTime Deployment API v2

The Deployment API allows you to manage applications, modules, environments and deployments of your OutSystems infrastructure. Version 2 of the API adds support for deployment zones.

Follow the guidelines presented in REST API Authentication to authenticate your API requests.

Check LifeTime Deployment API Examples to learn how to perform common tasks using the LifeTime Deployment API.

Summary

<table>
<thead>
<tr>
<th>API</th>
<th>Base URL</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2</td>
<td>/lifetimeapi/rest/v2</td>
<td>SSL/TLS</td>
</tr>
</tbody>
</table>

v2

<table>
<thead>
<tr>
<th>API Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /applications/</td>
<td>Returns a list of applications that exist in the infrastructure.</td>
</tr>
<tr>
<td>GET /applications/{ApplicationKey}/</td>
<td>Returns the details of a given application.</td>
</tr>
<tr>
<td>GET /applications/{ApplicationKey}/versions/</td>
<td>Returns a list of versions of a given application.</td>
</tr>
<tr>
<td>GET /applications/{ApplicationKey}/versions/{VersionKey}/</td>
<td>Returns the details of a</td>
</tr>
<tr>
<td>API Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| DELETE /applications/{ApplicationKey}/versions/{VersionKey}/ | Discards a given application version if not "InUse".
<p>| GET /applications/{ApplicationKey}/versions/{VersionKey}/content/ | Returns a link where the binary file for a given application version can be downloaded. The link will expire in 60 minutes.                                                                                                                                                                                                               |
| GET /deployments/                                     | Returns a list of deployments ordered by creation date, from newest to oldest. You can filter the list of results by MinDate, MaxDate and TargetEnvironmentKey.                                                                                                                                                                         |
| POST /deployments/                                    | Creates a deployment to a target environment. An optional list of applications to include in the deployment can be specified. The input is a subset of deployment object.                                                                                                                                                              |
| GET /deployments/{DeploymentKey}/                    | Returns the details of a given deployment, validating if there are any conflicts. The returned information contains the applications included in the deployment plan and the possible conflicts that can arise from the deployment of the selected applications.                                                                                                           |
| PUT /deployments/{DeploymentKey}/                    | Updates a given deployment. An optional list of applications to include in the deployment can be specified. The input is a subset of deployment object.                                                                                                                                                                                           |
| DELETE /deployments/{DeploymentKey}/                 | Discards a deployment, if possible. Only                                                                                                                                                                                                                                                                                                      |</p>
<table>
<thead>
<tr>
<th>API Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST /deployments/{DeploymentKey}/{Command}/</td>
<td>deployments whose state is “saved” can be deleted. Executes the given command in a specified deployment. The initiation of a deployment plan will check if it’s valid. The applications to redeploy, if applicable, will also be included in the deployment plan.</td>
</tr>
<tr>
<td>GET /deployments/{DeploymentKey}/status/</td>
<td>Returns the details of a given deployment execution, including the deployment status and messages.</td>
</tr>
<tr>
<td>GET /environments/</td>
<td>Lists all the environments in the infrastructure.</td>
</tr>
<tr>
<td>GET /environments/{EnvironmentKey}/</td>
<td>Returns the details of a given environment.</td>
</tr>
<tr>
<td>GET /environments/{EnvironmentKey}/applications/</td>
<td>Returns information about the running versions of all applications in a given environment.</td>
</tr>
<tr>
<td>GET /environments/{EnvironmentKey}/applications/{ApplicationKey}/</td>
<td>Returns information about the running version of the specified application in a given environment.</td>
</tr>
<tr>
<td>GET /environments/{EnvironmentKey}/applications/{ApplicationKey}/content/</td>
<td>Returns a link where the binary file for a given application can be downloaded. The link will expire in 60 minutes.</td>
</tr>
<tr>
<td>POST /environments/{EnvironmentKey}/applications/{ApplicationKey}/versions/</td>
<td>Creates a new version of the application based on the current running application.</td>
</tr>
<tr>
<td>GET /environments/{EnvironmentKey}/deploymentzones/</td>
<td>Returns information about the deployment zones available in a given environment.</td>
</tr>
<tr>
<td>GET /modules/</td>
<td>Returns a list of modules that exist in the...</td>
</tr>
</tbody>
</table>
### /applications

**GET /applications/**

Returns a list of applications that exist in the infrastructure.

**Full URL**

GET /lifetimeapi/rest/v2/applications/

**Inputs**

- **IncludeModules**
  - Type: optional, Boolean
  - Located in: URL.
  - When set to true, the modules are also returned. The default value is false.

- **IncludeEnvStatus**
  - Type: optional, Boolean
  - Located in: URL.
  - When set to true, the application status per environment is also returned. The default value is false.

**Outputs**

- **Applications**
  - Type: Application List
  - Located in: Body.
  - A list of Application records including AppStatusInEnv sub-lists, if requested.

**Return Codes**


Updated: Mon, 18 Nov 2019 01:15:53 GMT
200 Application list successfully retrieved.

204 No applications available in the infrastructure.

400 Failed to retrieve applications because IncludeModules was requested but IncludeEnvStatus was not, or invalid request when listing all applications.

500 Failed to list the applications.

GET /applications/{ApplicationKey}/

Returns the details of a given application.

Full URL

GET /lifetimeapi/rest/v2/applications/{ApplicationKey}/

Inputs

ApplicationKey
  Type: mandatory, Text
  Located in: URL.
  The key of the desired application.

IncludeModules
  Type: optional, Boolean
  Located in: URL.
  When set to true, the modules details are also retrieved. The default value is false.

IncludeEnvStatus
  Type: optional, Boolean
  Located in: URL.
  When set to true, the application status per environment is also returned. The default value is false.

Outputs

Application
  Type: Application
  Located in: Body.
  An Application record including an AppStatusInEnv sub-list, if requested.

Return Codes

200 Application details successfully retrieved.
Failed to retrieve applications because IncludeModules and IncludeEnvStatus parameters were incorrect.

Failed listing all applications because user has insufficient permissions.

Failed getting running applications because one of the environments was not found.

GET /applications/{ApplicationKey}/versions/

Returns a list of versions of a given application.

Full URL

GET /lifetimeapi/rest/v2/applications/{ApplicationKey}/versions/

Inputs

ApplicationKey
Type: mandatory, Text
Located in: URL.
The key of the desired application.

MaximumVersionsToReturn
Type: optional, Integer
Located in: URL.
The maximum number of versions to return. The default value is 5.

ChangeLogFilter
Type: optional, Text
Located in: URL.
Returns versions with change logs containing the required keyword. If left empty, all versions are returned.

Outputs

ApplicationVersions
Type: ApplicationVersion List
Located in: Body.
A list of ApplicationVersion records.

Return Codes

200
List of application versions successfully retrieved.

400
Invalid request due to invalid max versions to return (less than 0).
403
Failed to retrieve the application with key <ApplicationKey>. The user does not have the required permissions.

404
Failed to retrieve the application with key <ApplicationKey>.

500
Failed to list the application versions.

GET /applications/{ApplicationKey}/versions/{VersionKey}/

Returns the details of a given version of the specified application.

Full URL

GET /lifetimeapi/rest/v2/applications/{ApplicationKey}/versions/{VersionKey}/

Inputs

ApplicationKey
Type: mandatory, Text
Located in: URL.
The key of the application whose version is being requested.

IncludeModules
Type: mandatory, Boolean
Located in: URL.
When set to true, the modules details are also retrieved. The default value is false.

VersionKey
Type: mandatory, Text
Located in: URL.
The key of the desired application version.

Outputs

ApplicationVersions
Type: ApplicationVersion
Located in: Body.
An ApplicationVersion record.

Return Codes

200
Application version details successfully retrieved.

403
Failed to retrieve the application with key <ApplicationKey>. The user does not have the
required permissions.

404
Failed to retrieve the application with key <ApplicationKey>.

500
Failed to retrieve the application version.

DELETE
/applications/{ApplicationKey}/versions/{VersionKey}/

Discards a given application version if not "InUse".

Full URL

DELETE /lifetimeapi/rest/v2/
applications/{ApplicationKey}/versions/{VersionKey}/

Inputs

ApplicationKey
Type: mandatory, Text
Located in: URL.
The key of the application whose version is to be deleted.

VersionKey
Type: mandatory, Text
Located in: URL.
The key of the application version to be deleted.

Return Codes

204
Application version successfully deleted.

403
Service Account doesn't have permissions to delete the specified application version.

404
Application or application version not found.

500
Failed to delete application version <VersionKey>.
GET /applications/{ApplicationKey}/versions/{VersionKey}/content/

Returns a link where the binary file for a given application version can be downloaded. The link will expire in 60 minutes.

**Full URL**

GET /lifetimeapi/rest/v2/applications/{ApplicationKey}/versions/{VersionKey}/content/

**Inputs**

**ApplicationKey**
- Type: mandatory, Text
- Located in: URL.
- The key of the application for which to get the binary file link.

**VersionKey**
- Type: mandatory, Text Located in: URL.
- The key of the application version for which to get the binary file link.

**Outputs**

**DownloadLink**
- Type: DownloadLink
- Located in: Body.
- The link for the application version binary file.

**Expires**
- Type: Date Time Located in: Header.
- The expiration date and time of the returned link.

**Return Codes**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Binary file download link successfully retrieved.</td>
</tr>
<tr>
<td>204</td>
<td>No binary available for given keys.</td>
</tr>
<tr>
<td>400</td>
<td>The request is invalid for the given keys (Application:&lt;ApplicationKey&gt;; ApplicationVersionKey:&lt;ApplicationVersionKey&gt;).</td>
</tr>
<tr>
<td>403</td>
<td>User doesn’t have permissions for the given keys (Application:&lt;ApplicationKey&gt;; ApplicationVersionKey:&lt;ApplicationVersionKey&gt;).</td>
</tr>
<tr>
<td>404</td>
<td>Failed to retrieve the application with key &lt;ApplicationKey&gt; or the application version with key &lt;ApplicationVersionKey&gt;.</td>
</tr>
</tbody>
</table>
Failed to download the oap of the application version.

/getdeployments/

Returns a list of deployments ordered by creation date, from newest to oldest. You can filter the list of results by MinDate, MaxDate and TargetEnvironmentKey.

Full URL

GET /lifetimeapi/rest/v2/deployments/

Inputs

MinDate
Type: optional, Date
Located in: URL.
The minimum creation date of the deployments to return. The default value is 1 week before the current date.

MaxDate
Type: optional, Date
Located in: URL.
The maximum creation date of the deployments to return. The default value is the current date.

TargetEnvironmentKey
Type: optional, Text
Located in: URL.
Target environment unique identifier. If the user does not have access to the environment, the list returned will be empty. If no environment key is passed, the list will not be filtered by any target environment.

Outputs

Deployments
Type: Deployment List
Located in: Body.
A list of Deployment records.

Return Codes

200
Deployments list successfully retrieved.

Updated: Mon, 18 Nov 2019 01:15:53 GMT
POST /deployments/

Creates a deployment to a target environment. An optional list of applications to include in the deployment can be specified. The input is a subset of deployment object.

Full URL

POST /lifetimeapi/rest/v2/deployments/

Inputs

ApplicationOperations
Type: ApplicationOperation List
Located in: Body.
List of Application Operations included in the deployment.
You should only supply the ApplicationVersionKey and DeploymentZoneKey structure attributes of the ApplicationOperation structure. To deploy an application to the default deployment zone, provide an empty string in the DeploymentZoneKey attribute.

Notes
Type: optional, Text
Located in: Body.
Deployment notes.

SourceEnvironmentKey
Type: mandatory, Text
Located in: Body.
Source environment unique identifier.

TargetEnvironmentKey
Type: mandatory, Text
Located in: Body.
Target environment unique identifier.
Outputs
DeploymentKey
  Type: Text
  Located in: Body.
  The key of the newly created deployment.

Return Codes
201
  Deployment successfully created.

400
  Invalid request.

403
  Invalid user permissions.

404
  Source or target environment not found.

500
  Failed to create deployment from environment <SourceEnvironmentKey> to environment <TargetEnvironmentKey>.

Example Request Body
{
  "ApplicationOperations": [
    {
      "ApplicationVersionKey": "702df59d-3700-41b8-907e-a629e05a68f1",
      "DeploymentZoneKey": "d91a7e60-dea3-4150-8da2-39593a780507"
    }
  ],
  "Notes": "WebPortal 1.1 - QA to PRD Deployment",
  "SourceEnvironmentKey": "10061715-16bb-491a-86bc-595b465eaffb",
  "TargetEnvironmentKey": "55c430ee-4783-43e6-a2d4-6eeceed1d90f"
}

GET /deployments/{DeploymentKey}/

Returns the details of a given deployment, validating if there are any conflicts. The returned information contains the applications included in the deployment plan and the possible conflicts that can arise from the deployment of the selected applications.

Full URL
GET /lifetimeapi/rest/v2/deployments/{DeploymentKey}/
Inputs

DeploymentKey
- Type: mandatory, Text
- Located in: URL.
  The key of the desired deployment.

Outputs

ApplicationConflicts
- Type: ApplicationConflict List
- Located in: Body.
  List of conflicts between applications in the deployment.

ApplicationsToRedeploy
- Type: Text List
- Located in: Body.
  List of applications that will be redeployed.

Deployment
- Type: Deployment
- Located in: Body.
  The deployment details.

Return Codes

200
Deployment details successfully retrieved.

403
User doesn’t have permissions to the deployment with key <DeploymentKey>.

404
Deployment with key <DeploymentKey> not found.

500
Failed to access the details of deployment with key <DeploymentKey>.

PUT /deployments/{DeploymentKey}/

Updates a given deployment. An optional list of applications to include in the deployment can be specified. The input is a subset of deployment object.

Full URL

PUT /lifetimeapi/rest/v2/deployments/{DeploymentKey}/

Inputs
DeploymentKey
Type: mandatory, Text
Located in: URL.
The key of the deployment to update.

ApplicationOperations
Type: optional, ApplicationOperation List.
Located in: Body.
List of Application Operations included in the deployment.
You should only supply the ApplicationVersionKey and DeploymentZoneKey structure attributes of
the ApplicationOperation structure. To deploy an application to the default deployment zone, provide
an empty string in the DeploymentZoneKey attribute.

Notes
Type: optional, Text
Located in: Body.
Deployment notes.

SourceEnvironmentKey
Type: optional, Text
Located in: Body.
Source environment unique identifier.

TargetEnvironmentKey
Type: optional, Text
Located in: Body.
Target environment unique identifier.

Outputs
Deployment
Type: DeploymentData
Located in: Body.
A DeploymentData record containing the updated information.

Return Codes
200
Deployment successfully updated.

400
Invalid request.

403
Invalid user permissions.

404
Deployment plan not found.

500
Failed to update deployment with key <DeploymentKey>.

Example Request Body

```json
{
    "ApplicationOperations": [
```
DELETE /deployments/{DeploymentKey}/

Discards a deployment, if possible. Only deployments whose state is “saved” can be deleted.

Full URL

DELETE /lifetimeapi/rest/v2/deployments/{DeploymentKey}/

Inputs

DeploymentKey
  Type: mandatory, Text
  Located in: URL.
  The key of the deployment to delete.

Return Codes

204
  Deployment successfully deleted.

400
  Deployment with key <DeploymentKey> cannot be deleted

403
  Service Account doesn't have permissions to the deployment with key <DeploymentKey>.

404
  Deployment with key <DeploymentKey> not found.

500
  Failed to delete deployment <DeploymentKey>. 
POST /deployments/{DeploymentKey}/{Command}/

Executes the given command in a specified deployment. The initiation of a deployment plan will check if it's valid. The applications to redeploy, if applicable, will also be included in the deployment plan.

Full URL

POST /lifetimeapi/rest/v2/deployments/{DeploymentKey}/{Command}/

Inputs

**DeploymentKey**
- Type: mandatory, Text
- Located in: URL.
- The key of the deployment where the command will be executed.

**Command**
- Type: mandatory, Text
- Located in: URL.
- The command to execute. One of “start”, “continue” or “abort”.

**RedeployOutdated**
- Type: optional, Boolean
- Located in: URL.
- If True, outdated applications in the target environment will be redeployed.

Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>202</td>
<td>Command &lt;Command&gt; executed successfully for deployment &lt;DeploymentKey&gt;.</td>
</tr>
<tr>
<td>400</td>
<td>Command &lt;Command&gt; can't be executed for deployment &lt;DeploymentKey&gt;.</td>
</tr>
<tr>
<td>403</td>
<td>Service Account doesn't have permissions to the deployment with key &lt;DeploymentKey&gt;.</td>
</tr>
<tr>
<td>404</td>
<td>Deployment with key &lt;DeploymentKey&gt; not found, or command not found.</td>
</tr>
<tr>
<td>500</td>
<td>Failed to execute command &lt;Command&gt; for deployment with key &lt;DeploymentKey&gt;.</td>
</tr>
</tbody>
</table>

GET /deployments/{DeploymentKey}/status/

Returns the details of a given deployment execution, including the deployment status and messages.
GET /lifetimeapi/rest/v2/deployments/{DeploymentKey}/status/

**Inputs**

**DeploymentKey**
- Type: mandatory, Text
- Located in: URL.
  - The key of the deployment whose status is being requested.

**Outputs**

**DeploymentLog**
- Type: DeploymentMessage List
- Located in: Body.
  - List of deployment messages.

**DeploymentStatus**
- Type: Text
- Located in: Body.
  - Status of the deployment. [saved | running | needs_user_intervention | aborted | aborting | finished_successful | finished_with_warnings | finished_with_errors]

**Info**
- Type: Text
- Located in: Body.
  - Contains more information on the kind of user intervention needed when the DeploymentStatus value is "needs_user_intervention". For every other status, Info will contain an empty string ("""). [business_process_impact | deploy_prepared | needs_configuration_or_confirmation | deployment_suspended]

**Return Codes**

**200**
- Deployment status successfully retrieved.

**403**
- User doesn't have permissions to the deployment with key <DeploymentKey>.

**404**
- Deployment with key <DeploymentKey> not found.

**500**
- Failed to retrieve the status of the deployment with key <DeploymentKey>.

/environments
GET /environments/

Lists all the environments in the infrastructure.

Full URL

GET /lifetimeapi/rest/v2/environments/

Outputs

Environments
  Type: Environment List
  Located in: Body.
  A list of Environment records.

Return Codes

200  Environments list successfully retrieved.

204  No environments found.

500  Failed to list the environments.

GET /environments/{EnvironmentKey}/

Returns the details of a given environment.

Full URL

GET /lifetimeapi/rest/v2/environments/{EnvironmentKey}/

Inputs

EnvironmentKey
  Type: mandatory, Text
  Located in: URL.
  The key of the desired environment.

Outputs

Environment
  Type: Environment
  Located in: Body.
  An Environment record.
Return Codes

200
Environment details successfully retrieved.

403
Failed to retrieve the environment with key: <EnvironmentKey>. The user does not have the required permissions.

404
Failed to retrieve the environment with key: <EnvironmentKey>.

500
Failed to access the details of environment.

GET /environments/{EnvironmentKey}/applications/

Returns information about the running versions of all applications in a given environment.

Full URL

GET /lifetimeapi/rest/v2/environments/{EnvironmentKey}/applications/

Inputs

EnvironmentKey
Type: mandatory, Text
Located in: URL.
The key of the environment whose list of running applications is being requested.

IncludeModules
Type: optional, Boolean
Located in: URL.
When set to true, the modules details are also retrieved. The default value is false.

IncludeEnvStatus
Type: optional, Boolean
Located in: URL.
When set to true, the applications’ status information in the environment is included in the result. The default value is false.

Outputs

Applications
Type: Application List
Located in: Body.
A list of Application records.

Return Codes

Updated: Mon, 18 Nov 2019 01:15:53 GMT
Applications list for the given environment successfully retrieved

No applications found in environment with key `<EnvironmentKey>`.

Failed to retrieve applications published in environment because `IncludeModules` and `IncludeEnvStatus` parameters were incorrect, or invalid request when getting running applications for environment with key `<EnvironmentKey>`.

Failed to retrieve the running applications for environment with key `<EnvironmentKey>` because user has insufficient permissions.

Failed to retrieve running applications for environment with key `<EnvironmentKey>` because it was not found.

GET
/environments/{EnvironmentKey}/applications/{ApplicationKey}/

Returns information about the running version of the specified application in a given environment.

**Full URL**

GET /lifetimeapi/rest/v2/
environments/{EnvironmentKey}/applications/{ApplicationKey}/

**Inputs**

**EnvironmentKey**
Type: mandatory, Text
Located in: URL.
The key of the environment from which to get the running application details.

**ApplicationKey**
Type: mandatory, Text
Located in: URL.
The key of the application whose details are being requested.

**IncludeEnvStatus**
Type: optional, Boolean
Located in: URL.
When set to true, the applications' status information in the environment is included in the result. The default value is false.
IncludeModules
Type: optional, Boolean
Located in: URL.
When set to true, the modules details are also retrieved. The default value is false.

Outputs
Application
Type: Application
Located in: Body.
An Application record.

Return Codes
200
Application information successfully retrieved.

400
Request asked for Modules but not for Status.

403
User doesn’t have permissions for the given keys (EnvironmentKey:<EnvironmentKey>; Application:<ApplicationKey>).

404
Failed to retrieve the environment with key <EnvironmentKey> or the application with key <ApplicationKey>.

500
Failed to access the running version of an application.

GET
/environments/{EnvironmentKey}/applications/{ApplicationKey}/content/

Returns a link where the binary file for a given application can be downloaded. The link will expire in 60 minutes.

Full URL
GET /lifetimeapi/rest/v2/
environments/{EnvironmentKey}/applications/{ApplicationKey}/content/

Inputs
EnvironmentKey
Type: mandatory, Text
Located in: URL.
The key of the environment from which to get the application binary file link.
**ApplicationKey**
Type: mandatory, Text  
Located in: URL.  
The key of the application for which to get the binary file link.

**Type**
Type: optional, Text  
Located in: URL.  
The type of binary file to return, when applicable. One of “oap”, “apk” or “ipa”.

**Outputs**
**DownloadLink**
Type: DownloadLink  
Located in: Body.  
The link for the application binary file.

**Expires**
Type: Date Time  
Located in: Header.  
The expiration date and time of the returned link.

**Return Codes**

200  
Binary file download link successfully retrieved.

204  
No binary available for given type and keys.

400  
The required type `<Type>` is invalid for given keys (EnvironmentKey:`<EnvironmentKey>`; Application:`<ApplicationKey>`).

403  
User doesn’t have permissions for the given keys (EnvironmentKey:`<EnvironmentKey>`; Application:`<ApplicationKey>`).

404  
Failed to retrieve the environment with key `<EnvironmentKey>` or the application with key `<ApplicationKey>`.

500  
Failed to download `<Type>` of an application.

**POST**
/environments/{EnvironmentKey}/applications/{ApplicationKey}/versions

Creates a new version of the application based on the current running application.
**Full URL**

POST /lifetimeapi/rest/v2/environments/{EnvironmentKey}/applications/{ApplicationKey}/versions/

**Inputs**

**EnvironmentKey**
- Type: mandatory, Text
- Located in: URL.
- The key of the environment from which to get the application.

**ApplicationKey**
- Type: mandatory, Text
- Located in: URL.
- The key of the application for which to generate a new version.

**ApplicationVersionCreate**
- Type: mandatory, ApplicationVersionCreate
- Located in: Body.
- A structure holding the new version name for the application and for its native applications, if applicable.

**Outputs**

**ApplicationVersionKey**
- Type: Text
- Located in: Body.
- The key of the newly created application version.

**Return Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Application version successfully created.</td>
</tr>
<tr>
<td>400</td>
<td>Invalid request.</td>
</tr>
<tr>
<td>403</td>
<td>Invalid user permissions.</td>
</tr>
<tr>
<td>404</td>
<td>Environment or application not found.</td>
</tr>
<tr>
<td>500</td>
<td>Failed to tag an application, or Failed to create a new version for application <code>&lt;ApplicationName&gt;</code>.</td>
</tr>
</tbody>
</table>

**Example Request Body**

```json
{
    "ChangeLog": "First release of iOS mobile app",
    "Version": "1.0.0",
    "MobileVersions": [
        
```
GET /environments/{EnvironmentKey}/deploymentzones/

Returns information about the deployment zones available in a given environment.

Full URL

GET /lifetimeapi/rest/v2/environments/{EnvironmentKey}/deploymentzones/

Inputs

EnvironmentKey
Type: mandatory, Text
Located in: URL.
The key of the environment from which to get the running application details.

Outputs

DeploymentZones
Type: DeploymentZone List
Located in: Body.
A list of Deployment Zones.

Return Codes

200
Deployment zone information successfully retrieved.

400
Failed to access the deployment zones of environment.

403
Failed to retrieve the deployment zones of environment <EnvironmentName> (key: <EnvironmentKey>). Error: The user does not have the required permissions, or Feature not Licensed.

404
Failed to retrieve the deployment zones of environment with key: <EnvironmentKey>.

500
Failed to access the deployment zones of environment.
GET /modules/

Returns a list of modules that exist in the infrastructure.

Full URL
GET /lifetimeapi/rest/v2/modules/

Inputs
IncludeEnvStatus
Type: optional, Boolean
Located in: URL.
When set to true, the module status per environment is also returned. The default value is false.

Outputs
ModuleList
Type: Module List
Located in: Body.
List of Module records.

Return Codes
200
Modules list successfully retrieved.

204
No modules found in the infrastructure.

500
Failed to list modules.

GET /modules/{ModuleKey}/

Returns the details of a given module.

Full URL
GET /lifetimeapi/rest/v2/modules/{ModuleKey}/
Inputs

**ModuleKey**
Type: mandatory, Text  
Located in: URL.  
Key of the module to list the details from.

**IncludeEnvStatus**
Type: optional, Boolean  
Located in: URL.  
Boolean to indicate if status per env should be returned. Default is false.

Outputs

**Module**
Type: Module  
Located in: Body.  
Module record.

Return Codes

200
Module details successfully retrieved.

403
Failed to retrieve the module with key: `<ModuleKey>`. The user does not have the required permissions.

404
Failed to retrieve the module with key: `<ModuleKey>`.

500
Failed to retrieve the module with key `<ModuleKey>`.

GET /modules/{ModuleKey}/versions/

Returns a list of versions of a given module.

Full URL

GET /lifetimeapi/rest/v2/modules/{ModuleKey}/versions/

Inputs

**ModuleKey**
Type: mandatory, Text  
Located in: URL.  
The module from where to retrieve the versions from.

**MaximumVersionsToReturn**
Type: optional, Integer
Located in: URL.
Maximum number of versions to return. Default is 5.

**Outputs**

**ModuleVersionList**
Type: [ModuleVersion](#) List
Located in: Body.
List of ModuleVersion records.

**Return Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>List of module versions successfully retrieved.</td>
</tr>
<tr>
<td>400</td>
<td>Invalid request due to invalid max versions to return (less than 0).</td>
</tr>
<tr>
<td>403</td>
<td>Failed to retrieve the module with key: <code>&lt;ModuleKey&gt;</code>. The user does not have the required permissions.</td>
</tr>
<tr>
<td>404</td>
<td>Failed to retrieve the module with key: <code>&lt;ModuleKey&gt;</code>.</td>
</tr>
<tr>
<td>500</td>
<td>Failed to list module versions.</td>
</tr>
</tbody>
</table>

**GET /modules/{ModuleKey}/versions/{ModuleVersionKey}/**

Returns the details of a given module version.

**Full URL**

GET `/lifetimeapi/rest/v2/modules/{ModuleKey}/versions/{ModuleVersionKey}/`

**Inputs**

**ModuleKey**
Type: mandatory, Text
Located in: URL.
The module from where to retrieve the versions from.

**ModuleVersionKey**
Type: mandatory, Text
Located in: URL.
Key of the module version to return.

**Outputs**
ModuleVersion
Type: ModuleVersion
Located in: Body.
Record of ModuleVersion.

Return Codes
200
Module version details successfully retrieved.

403
Failed to retrieve the module with key: <ModuleKey>. The user does not have the required permissions.

404
Failed to retrieve the module with key: <ModuleKey>, or Failed to retrieve the module version with key: <ModuleKey>.

500
Failed to access the details of a module version.

Structures

Application
An application with its details and its status in the environments were it is running.

Attributes
Key
Type: Text.
Application unique identifier.

Name
Type: Text (50).
Name of the application.

Kind
Type: RuntimeKind Identifier.
Identifies the kind of application. [Mobile | WebResponsive]
**Description**
Type: Text (50).
Description of the application.

**URLPath**
Type: Text (50).
Relative URL path of the application, starting from the hostname.

**IconHash**
Type: Text (50).
Hash of the application icon. Can be used to detect changes in the application icon.

**IconURL**
Type: Text (50).
The URL for the application icon.

**IsSystem**
Type: Boolean.
Indicates if the application is a built-in component of the OutSystems platform (e.g. ServiceCenter, LifeTime, ...).

**AppStatusInEnvs**
Type: AppStatusInEnv List.
Information about the status of the application in each environment it is running.

---

**ApplicationConflict**

A deployment conflict.

**Attributes**

**Message**
Type: Text.
Description of the conflict.

**ProducerApplicationOperation**
Type: ApplicationOperation.
Operation executed over producer application.

**ConsumerApplicationOperation**
Type: ApplicationOperation.
Operation executed over consumed application.

**ModuleConflict**
Type: ModuleConflict.
Details of the module conflict.
ApplicationOperation

Operation executed in the deployment over the application.

Attributes

ApplicationKey
Type: Text.
Application unique identifier.

ApplicationVersionKey
Type: Text.
Application Version unique identifier.

DeploymentOperation
Type: Text.
Label of the operation to be performed. Example: Deploy 1.5.

DeploymentZoneKey
Type: Text.
Deployment Zone unique identifier.

ApplicationVersion

Information about a specific version of an application and the versions of its modules.

Attributes

Key
Type: Text.
Application version unique identifier.

ApplicationKey
Type: Text.
Application unique identifier.

Version
Type: Text (50).
Version of the application.

ChangeLog
Type: Text.
ChangeLog of the Application Version. It is the log message created by the user for that Application Version.

CreatedAt
Type: Date Time.
When was the Application Version created.
**InUse**
Type: Boolean.
Defines if the Application Version is being used, i.e. a running version or a version in a Deployment.

**MobileVersions**
Type: MobileVersion List.
List of mobile versions.

**PrimaryColor**
Type: Text (50).
The primary color of the application interface.

**NativeHash**
Type: Text (50).
The native hash relative to the mobile platform.

**ModuleVersions**
Type: ModuleVersion List.
List of module versions.

---

**ApplicationVersionCreate**

A structure holding the new version name for the application and for its native applications, if applicable.

**Attributes**

**ChangeLog**
Type: Text.
Change log of the version to be created.

**Version**
Type: Text (50).
Version of the application.

**MobileVersions**
Type: MobileVersion List.
List of mobile versions.

**ModuleVersionKeys**
Type: optional, Text List.
(DEPRECATED) List of module version keys to validate if the current state of the application is still the expected one. If you provide module version keys, they will still be validated.
AppStatusInEnv

Status of application in a given environment.

Attributes

**EnvironmentKey**
Type: Text.
Environment unique identifier.

**BaseApplicationVersionKey**
Type: Text.
Base application version unique identifier. If app is not modified in environment, this is the application version deployed.

**IsModified**
Type: Boolean.
True if the application has been changed since the last tag, false otherwise.

**IsModifiedReason**
Type: Text.
Indicates the application status.

**IsModifiedMessage**
Type: Text.
Indicates the application status.

**ConsistencyStatus**
Type: Text (50).
Indicates the application consistency status.

**ConsistencyStatusMessages**
Type: Text (2000).
Messages regarding the consistency status of the application.

**MobileAppsStatus**
Type: [MobileAppStatusInEnv List](#).
Status of mobile apps in environment.

**ModuleStatusInEnvs**
Type: [ModuleStatusInEnv List](#).
Status of modules in environment.

**DeploymentZoneKey**
Type: Text.
Deployment Zone unique identifier
### Deployment

Deployment information with the operations executed.

#### Attributes

<table>
<thead>
<tr>
<th>Key</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment unique identifier.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SourceEnvironmentKey</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source environment unique identifier.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TargetEnvironmentKey</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target environment unique identifier.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notes</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment notes.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CreatedOn</th>
<th>Type: Date Time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date and time when the deployment plan was created.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CreatedBy</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the user who created the deployment plan.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CreatedByUsername</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username of the user who created the deployment plan.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SavedOn</th>
<th>Type: Date Time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The date and time when the deployment plan was saved.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SavedBy</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the user who last saved the deployment plan.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SavedByUsername</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username of the user who last saved the deployment plan.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StartedOn</th>
<th>Type: Date Time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The date and time when the deployment started.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StartedBy</th>
<th>Type: Text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the user who started the deployment.</td>
<td></td>
</tr>
</tbody>
</table>
**StartedByUsername**
Type: Text.
Username of the user who started the deployment.

**AbortedOn**
Type: Date Time.
The date and time when the deployment was aborted.

**AbortedBy**
Type: Text.
Name of the user who aborted the deployment.

**AbortedByUsername**
Type: Text.
Username of the user who aborted the deployment.

**ApplicationOperations**
Type: ApplicationOperation List.
List of Application Operations included in the deployment.

---

**DeploymentData**

Deployment information with the operations executed.

**Attributes**

**Key**
Type: Text.
Deployment unique identifier.

**SourceEnvironmentKey**
Type: Text.
Source environment unique identifier.

**TargetEnvironmentKey**
Type: Text.
Target environment unique identifier.

**Notes**
Type: Text.
Deployment notes.

**CreatedOn**
Type: Date Time.
Date and time when the deployment plan was created.

**CreatedBy**
Type: Text.
Name of the user who created the deployment plan.
CreatedByUsername
Type: Text.
Username of the user who created the deployment plan.

SavedOn
Type: Date Time.
The date and time when the deployment plan was saved.

SavedBy
Type: Text.
Name of the user who last saved the deployment plan.

SavedByUsername
Type: Text.
Username of the user who last saved the deployment plan.

StartedOn
Type: Date Time.
The date and time when the deployment started.

StartedBy
Type: Text.
Name of the user who started the deployment.

StartedByUsername
Type: Text.
Username of the user who started the deployment.

AbortedOn
Type: Date Time.
The date and time when the deployment was aborted.

AbortedBy
Type: Text.
Name of the user who aborted the deployment.

AbortedByUsername
Type: Text.
Username of the user who aborted the deployment.

ApplicationOperations
Type: ApplicationOperation List.
List of Application Operations included in the deployment.
Note: Only the ApplicationVersionKey and DeploymentZoneKey attributes are used.

DeploymentMessage
Message from a deployment operation log.
Attributes

**Instant**
Type: Date Time.
Date and time when the message was logged.

**Message**
Type: Text.
Details of the message.

---

DeploymentTechnology

Deployment Hosting technology of the Deployment Zone.

**Attributes**

**Key**
Type: Text.
Deployment Technology unique identifier.

**Name**
Type: Text.
Name of deployment technology

---

DeploymentZone

Deployment Zone of an environment.

**Attributes**

**Key**
Type: Text.
Deployment Zone unique identifier.

**Name**
Type: Text.
Name of deployment zone

**IsDefault**
Type: Boolean.
True if the deployment zone is the default one in the environment

**DeploymentTechnology**
Type: DeploymentTechnology.
The deployment technology of the current deployment zone.
DownloadLink

The link for the application binary file.

Attributes

url
Type: Text.
The link for the application binary file.

expires
Type: Date Time.
The expiration date and time of the returned link.

Environment

An environment and its information.

Attributes

Key
Type: Text.
Unique identifier of the environment.

Name
Type: Text (50).
Name of the environment.

OSVersion
Type: Text (50).
Platform Server version. [X.X.X.X]

Order
Type: Integer.
The order of the environment as registered in Lifetime.

HostName
Type: Text (50).
Hostname of the environment as registered.

UseHTTPS
Type: Boolean.
Indicates if connections to the environment are made using HTTPS.
EnvironmentType
Type: Text.
Indicates the type of the environment. [Development | Test | Production]

NumberOfFrontEnds
Type: Integer.
Number of front-end servers in the environment.

ApplicationServerType
Type: Text (50).
Stack of the application server. [.NET | JAVA]

ApplicationServer
Type: Text (50).
Application server in use. [IIS | JBoss | WebLogic]

DatabaseProvider
Type: Text (50).
Type of database provider. [SqlServer | Oracle]

IsCloudEnvironment
Type: Boolean.
Indicates if the environment is running on a cloud service.

MobileAppStatusInEnv

Status of mobile application in a given environment.

Attributes

EnvironmentKey
Type: Text.
Environment unique identifier.

NativePlatform
Type: Text.
Name of native platform. [Android | iOS]

VersionNumber
Type: Text.
The version number, like for example 1.5.4, of the native build. It is used to be able to map the
version to the version in the Android or iOS store.

HasBinaryAvailable
Type: Boolean.
True if the binary of the application is available for the current configuration.

IsConfigured
Type: Boolean.
True if the application is configured.
IsConfigurationChanged
Type: Boolean.
True if the configuration of the Mobile Application has changed in the environment.

IsModified
Type: Boolean.
True if the Native Hash of the Mobile Application does not match the one in the AppVersionNativeBuild baseline.

MobileVersion

A mobile version and its information.

Attributes
NativePlatform
Type: Text.
Name of native platform. [Android | iOS]

VersionNumber
Type: Text (50).
The version number, like for example 1.5.4, of the native build. It is used to be able to map the version to the version in the Android or iOS store.

VersionDescription
Type: Text.
The description of the mobile version.

Module

Module information and the status in the environments where the modules are running.

Attributes
Key
Type: Text.
Module unique identifier.

Name
Type: Text (50).
Name of the module.

Description
Type: Text (50).
Description of the module.

**Kind**
Type: Text (50).
Module type (eSpace or Extension).

**ModuleStatusInEnv**
Type: ModuleStatusInEnv List.
Status of the module in environments.

**ModuleConflict**
A module conflict.

**Attributes**

**ProducerModuleKey**
Type: Text.
Producer Module unique identifier.

**ConsumerModuleKey**
Type: Text.
Consumer Module unique identifier.

**TotalRequiredElements**
Type: Integer.
Total number of required elements.

**ConflictType**
Type: Text.
Type of conflict. [Producer Module Missing | Producer Element Missing | Producer Element Incompatible | Consumer Module Outdated | Newer Producer Module Available | IncompatiblePlatformServer | ConsumerModuleMoved | ProducerModuleMoved | NameCollision]

**ModuleElement**
Element version information, such as action, entity, structure, among others.

**Attributes**

**Key**
Type: Text.
Module element unique identifier.
Name
  Type: Text (50).
  Name of the element as specified by the developer.

ElementType
  Type: Text.
  Type of the element, such as action, entity, structure.

CompatibilitySignatureHash
  Type: Text (50).
  Hash of the element signature. Can be used to validate if the element version is compatible with
  another version, not producing a broken reference.

FullSignatureHash
  Type: Text (50).
  Hash of the element. Can be used to uniquely identify an element version.

ModuleKey
  Type: Text.
  Unique identifier of the module where the element is publicly supplied, among others.

ModuleStatusInEnv

Status of module in a given environment.

Attributes

ApplicationKey
  Type: Text.
  Application unique identifier.

EnvironmentKey
  Type: Text.
  Environment unique identifier.

ModuleVersionKey
  Type: Text.
  Module version unique identifier.

ConsistencyStatus
  Type: Text (50).
  Indicates the module consistency status.

ConsistencyStatusMessages
  Type: Text (2000).
  Messages regarding the consistency status of the module.
ModuleVersion

A module version and its information.

Attributes

Key
Type: Text.
Module version unique identifier.

ModuleKey
Type: Text.
Module unique identifier.

CreatedOn
Type: Date Time.
Date and time of the module version creation.

CreatedBy
Type: Text (50).
Name of the user that created the version.

CreatedByUsername
Type: Text (50).
Username of the user that created the version.

GeneralHash
Type: Text (100).
Non-unique hash of the module version. Can be used to validate if two module versions have semantic differences.

DirectUpgradeFromHash
Type: Text (100).
If this module version is the result of a direct upgrade of another version, then this field contains the key of that version

Articles in this Section

- [REST API Authentication](https://success.outsystems.com/Documentation/11/Reference/OutSystems_APIs/LifeTime_Deployment_API_v2)
  Using OutSystems service account tokens to authenticate REST API method calls.

- [LifeTime Deployment API Examples](https://success.outsystems.com/Documentation/11/Reference/OutSystems_APIs/LifeTime_Deployment_API_v2)
  Check how to perform common tasks using the LifeTime Deployment API.