

# Mission Control REST API v2

## Overview

Mission Control exposes a rich REST API to allow fully automated management of Artifactory instances under your control.

## Version

### Deprecated

This version of Mission Control REST API v2 has been deprecated. We strongly recommend updating your scripts to the latest version. For details, please refer to [Mission Control REST API](#).

We strongly recommend that you upgrade your scripts to the latest API version, and to facilitate the required modifications, please refer to [V1 to V2 Mapping](#).

## Authentication

All Mission Control REST API endpoints require basic authentication using your username and password.

### Exception

The [System Health Check](#) endpoint does not require authentication.

## Error Handling

Mission Control REST API returns error responses in different formats depending on where the error occurred.

### Top Level Errors

Top level errors are returned if the request cannot be executed, and have the following format:

```
{
  "errors" : [
    {
      message string          "message" : <message>, // a descriptive error
      error category         "type" : <type>           // The
    }
  ]
}
```

For example:

```
{
  "errors": [
    {
      "message": "selected script has wrong type: REPOSITORY",
      "type": "Template processing"
    }
  ]
}
```

## Page Contents

- [Overview](#)
  - [Version](#)
  - [Authentication](#)
  - [Error Handling](#)
    - [Top Level Errors](#)
    - [Instance Errors](#)
    - [Repository Errors](#)
  - [Working with User Inputs](#)
  - [Mandatory Fields](#)
  - [Failure Policies](#)
    - [Supported Endpoints](#)

- REST Resources
  - SCRIPT MAPPINGS
    - Get Script List
    - List User Inputs
  - INSTANCES
    - Get Instances
    - Add Instance
    - Update Instance
    - Delete Instance
    - Execute Scripts on Instance
    - Get All Instances Status
    - Get Instance Status
  - REPOSITORIES
    - Get Repositories
    - Create Repository
    - Update Repository
  - SECURITY
    - Create User
    - Update User
    - Create User Group
    - Update User Group
    - Create Permission Target
    - Update Permission Target
  - LICENSE BUCKETS
    - Bucket Status
    - Attach License
  - DISASTER RECOVERY
    - Create a DR Pair
  - SYSTEM
    - System Health Check
- V1 to V2 Mapping

## Instance Errors

Instance errors are returned for API endpoints that act on instances, such as [Create User](#), [Create User Group](#) or [Update Instance](#) and others.

```

{
  "data": [
    {
      "success": "false",
      "message": <message>,
      "instanceName": <instance name>
    }
  ]
}
// "false" indicates an error occurred
// A descriptive error message string
// The instance on which the error occurred

```

For example:

```

{
  "data": [
    {
      "success": true,
      "instanceName": "localhost:8091/artifactory"
    },
    {
      "success": false,
      "message": "Connection refused",
      "instanceName": "localhost:8081/artifactory"
    }
  ]
}
//The action
on this instance succeeded
//The action
on this instance failed

```

## Repository Errors

Repository errors are returned for API endpoints that act on repositories, such as [Update Repository](#) and others.

```

{
  "data": [
    {
      "success": "false",
      "message": <message>,
      "instanceName": <instance name>,
      "repositoryKey": "maven-local"
    }
  ]
}
// "false" indicates an error occurred
// A descriptive error message string
// The instance on which the error occurred
// The repository on which the error occurred

```

For example:

```

{
  "data": [
    {
      "success": true,
      "instanceName": "localhost:8091/artifactory",
      "repositoryKey": "maven-local"
    },
    {
      "success": false,
      "message": "Connection refused",
      "instanceName": "localhost:8081/artifactory",
      "repositoryKey": "maven-local"
    }
  ]
}
// This
operation succeeded
// This
operation failed

```

## Working with User Inputs

Mission control configuration scripts offer the flexibility of letting you provide input just before the script is applied. When working with the Mission Control UI, the User Input screen allows you to enter all user input values required for the scripts you are about to apply.

When working with the REST API, another mechanism is provided that allows you to fully automate your management of Artifactory instances using [Script Mappings](#).

For a details on how to work with script mappings and user input with the Mission Control REST API, please refer to [Working with User Input](#).

## Mandatory Fields

For all endpoints, mandatory input fields are indicated by a plus sign (+).

## Failure Policies

Some of the endpoints in the Mission Control REST API perform multiple operations. For example [Create Repository](#) may create several repositories on several Artifactory instances for a single API call. For [supported endpoints](#), you may specify how the system should behave if a one of the operations in a call fails by adding a "failurePolicy" tag to the JSON payload.

The options are:

Failure Policy	Description
ignore	[default] Ignore the failure and continue with the remaining operations
rollback	Roll back completed operations and cancel remaining ones
fail	Cancel all remaining operations immediately upon any failure
retry	Any failed operation should be retried once. If failure persists, all remaining operations should be cancelled

For example,

```
PUT /api/v2/security/permission_targets/{permission_target_name}
{
  "instancesIds":["art1","art2"],
  ...
  "failurePolicy" : "rollback" | "retry" | "fail" | "ignore"
}
```

## Supported Endpoints

- [Create Repository](#)
- [Update Repository](#)
- [Execute Scripts on Instance](#)
- [Create User](#)
- [Update User](#)
- [Create User Group](#)
- [Update User Group](#)
- [Create Permission Target](#)
- [Update Permission Target](#)

## REST Resources

JFrog Mission Control is currently in its second version which is significantly different from version 1. The following sections describe the endpoints for REST API v2

If you are still the REST API v1, please refer to [Mission Control REST API v1](#).

If you are ready to upgrade to the latest REST API (v2), please refer to [V1 to V2 Mapping](#).

## SCRIPT MAPPINGS

### Get Script List

**Description:** Gets the list of instance and repository configuration scripts available on Mission Control

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** GET /api/v2/scripts

**Consumes:** None

**Produces:** application/json

```
{
  "data": [
    {
      "name" : "string", // The script name
      "description" : "string", // The script description
      "target" : "INSTANCE" | "REPOSITORY" // Specifies whether this is an instance or
repository scripts
    }
  ]
}
```

#### Example:

```
GET /api/v2/scripts
{
  "data": [
    {
      "name": "local-default",
      "target": "REPOSITORY"
    },
    {
      "name": "remote-default",
      "target": "REPOSITORY"
    },
    {
      "name": "propertySet_prodready",
      "target": "INSTANCE"
    }
  ]
}
```

## List User Inputs

**Description:** Gets the list of Artifactory user inputs needed for scripts that are being applied

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** POST /api/v2/scripts/user\_inputs

**Consumes:** application/json

```
{
+   "scriptMappings" : [ { //An array of scriptMapping objects
+     "instanceName" : <instance name>, // The instance on which you want
to apply a script
    "repositoryKey" : <string>, // The repository on which you want to apply the script
// Mandatory if the operationType is UPDATE_REPOSITORY.
// *** Not applicable and should be omitted *** if the operationType is
CREATE_REPOSITORY or UPDATE_INSTANCE
    "scriptNames" : [<script name>, <script name>, ...] //The names of the scripts you
want to apply
  }
]
  "operationType": "CREATE_REPOSITORY" | "UPDATE_REPOSITORY" | "UPDATE_INSTANCE" //The type of
operation you want to perform in the subsequent call with the user inputs returned
}
```

**Produces:** application/json

```

{
  "data":
  [
    {
      "success" : true,
      "instanceName" : "localhost:8091/artifactory",
      "repositoryKey" : "maven-local",          // The repository on which the
script was applied
      // Only present if operationType was
      UPDATE_REPOSITORY.
      "scriptUserInputs" : [
        {
          "multiple" : "boolean",          // Whether this user input
item can take multiple values
          "type" : "STRING" | "BOOLEAN" | "INTEGER" | "INSTANCE" |
"REPOSITORY", // The type of this user input item
          "value" : "object",              // A default value for
this user input item
          "description" : "string", // A description for this user
input item
          "name" : "string",              // A logical name for this
user input item
          "id" : "string"                  // The identifier of this
user input item. This is the identifier that must be used in any subsequent API call
        } ]
      ]
    }
  ]
}

```

**Sample usage:**

Assume the following scripts have been defined:

Script Name	Script Body
local-string-userinput	<pre> localRepository('local-userInput') {   description = userInput (     name : "User Friendly Name", // Optional     type : "STRING", // "BOOLEAN", "INTEGER", "INSTANCE", "REPOSITORY"     value : "default value",     description : "please provide a value"   ) } </pre>
local-instance-userInput	<pre> test = userInput (   type : "INSTANCE" )  localRepository('repo-variables') {   description test.name } </pre>
local-repo-userInput	<pre> test = userInput (   type : "REPOSITORY" )  localRepository('repo-variables') {   description test.name } </pre>

```

POST /api/v2/scripts/user_inputs
{
  "scriptMappings": [{
    "instanceName": "Master",
    "scriptNames": ["local-string-userInput", "local-instance-userInput", "local-repo-userInput"]
  }],
  "operationType" : "CREATE_REPOSITORY"
}

Response:
{
  "data": [
    {
      "success": true,
      "instanceName": "Master",
      "scriptUserInputs": [
        {
          "id": "RepositoryMapper#0#description#0",
          "name": "User Friendly Name",
          "description": "please provide a value",
          "value": "default value",
          "type": "STRING",
          "multiple": false
        },
        {
          "id": "TemplateExecutor#0#test#0",
          "name": "test",
          "type": "INSTANCE",
          "multiple": false
        },
        {
          "id": "TemplateExecutor#0#test#1",
          "name": "test",
          "type": "REPOSITORY",
          "multiple": false
        }
      ]
    }
  ]
}

```

## INSTANCES

### Get Instances

**Description:** Gets the list of Artifactory instances managed by Mission Control

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** GET /api/v2/instances

**Consumes:** None

**Produces:** application/json

```

{
  "data": [
    {
      "name" : <string>           //The Artifactory instance name
      "url"  : <string>,         //The Artifactory instance URL
    }
  ]
}

```

**Example:**

```
GET /api/v2/instances
{
  "data": [
    {
      "name": "DRTarget",
      "url": "http://hostname-target/artifactory"
    },
    {
      "name": "Master",
      "url": "http://hostname-master/artifactory"
    },
    {
      "name": "AOL-ClusterNNN-100",
      "url": "https://mycompany.artifactoryonline.org/mycompany"
    }
  ]
}
```

## Add Instance

**Description:** Adds an Artifactory instance

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** POST /api/v2/instances

**Consumes:** application/json

```
{
  "name" : "<instance name>",
  "description": "<description text>",
  "url" : "<instance URL>",
  "username" : "<admin username>",
  "password" : "<admin password>",
  "location" : "<location text>"
}
```

**Produces:** application/json

**Example:** Add a node to an Artifactory HA cluster

```
POST /api/v2/instances
{
  "name" : "Master",
  "description": "description text",
  "url" : "http://artifactory-hostname/artifactory",
  "username" : "admin",
  "password" : "password",
  "location" : "node location"
}
```

## Update Instance

**Description:** Updates an Artifactory instance

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** PUT /api/v2/instances/{instance}

**Consumes:** application/json



```
{
  "description": "<description text>",
  "url" : "<instance URL>",
  "username" : "<admin username>",
  "password" : "<admin password>",
  "location" : "<location text>"
}
```

**Produces:** application/json

**Example:** Update Artifactory instance called "Master"

```
POST /api/v2/instances/Master
{
  "description": "new description text",
  "url" : "http://new-artifactory-hostname/artifactory",
  "username" : "new-admin",
  "password" : "new-password",
  "location" : "new node location"
}
```

## Delete Instance

**Description:** Removes an Artifactory instance from Mission Control (doesn't do anything to the instance itself)

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** DELETE /api/v2/instances/{instance}

**Example:** Delete Artifactory instance called "AOL-ClusterNNN-100"

```
DELETE /api/v2/instances/AOL-ClusterNNN-100

204
```

## Execute Scripts on Instance

**Description:** Executes a set of scripts on a set of Artifactory instances

**Since:** 1.3

**Security:** Requires an admin user

**Usage:** PUT /api/v2/execute\_scripts/instances

**Consumes:** application/json

```
{
  "scriptMappings": [
    {
      "instanceName": <Artifactory instance>,
      "scriptNames": [<script names>]
    }
  ]
}
```

**Example:** Apply scripts ldapSettings1 and propertySets1 to Artifactory instance called "Master".

```
{
  "scriptMappings": [
    {
      "instanceName": "Master",
      "scriptNames": [
        "ldapSettings1", "propertySets1"
      ]
    }
  ]
}
```

## Get All Instances Status

**Description:** Gets the status of all managed Artifactory instances

**Since:** 1.6

**Security:** Requires an admin user

**Usage:** GET /api/v2/instances/monitoring/status

**Consumes:** None

**Produces:** application/json

```
{
  "data": [
    {
      "instanceName": <instance name>, //String
      "upTimeInSec": <Number of seconds instance has been up>, //Integer (when instanceState is ONLINE)
      "instanceState": <instance state> //String: ONLINE | OFFLINE | UNAUTHORIZED | ERROR
    }
  ]
}
```

### Example

```
{
  "data": [
    {
      "instanceName": "test-instance-2",
      "instanceState": "OFFLINE"
    },
    {
      "instanceName": "test-instance-1",
      "upTimeInSec": 1744204,
      "instanceState": "ONLINE"
    }
  ]
}
```

## Get Instance Status

**Description:** Gets the status of a specific managed Artifactory instance

**Since:** 1.6

**Security:** Requires an admin user

**Usage:** GET /api/v2/instances/[instance-name]/monitoring/status

**Consumes:** None

**Produces:** application/json

```
{
  "data":
  {
    "instanceName": <instance name>, //String
    "upTimeInSec": <Number of seconds instance has been up>, //Integer (when instanceState is ONLINE)
    "instanceState": <instance state> //String: ONLINE | OFFLINE | UNAUTHORIZED | ERROR
  }
}
```

### Example

```
{
  "data": {
    "instanceName": "test-instance-1",
    "upTimeInSec": 1744204,
    "instanceState": "ONLINE"
  }
}
```

## REPOSITORIES

## Get Repositories

**Description:** Gets a list of repositories in an Artifactory instance

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** GET /api/v2/instances/{instance name}/repositories

**Consumes:** None

**Produces:** application/json

**Example:** Get a list of repositories in instance AOL-ClusterNNN-100.

```
GET /api/v2/instances/cluster-126-10100/repositories
{
  "data": [
    {
      "repositoryKey": "libs-release-local",
      "description": "",
      "type": "local",
      "packageType": "maven"
    },
    {
      "repositoryKey": "libs-snapshot-local",
      "description": "",
      "type": "local",
      "packageType": "maven"
    },
    {
      "repositoryKey": "plugins-release-local",
      "description": "Local repository for plugins",
      "type": "local",
      "packageType": "maven"
    }
  ]
}
```

## Create Repository

**Description:** Creates a repository in a set of Artifactory instances. For more information, please refer to [Configuring Repositories](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** POST /api/v2/execute\_scripts/repositories

**Consumes:** application/json

```
{
  "scriptMappings":[
+
create the repository
repository
applied. One of these should refer to the repositoryKey field
previous call to userInputs
    {
      "instanceName" : <string>,           //Artifactory instance on which to
      "scriptNames" : [<string>],         //scripts to apply when creating the
      "scriptUserInputs":                 //user inputs required for the scripts
      {
        <userInputId> : <user input value> // userInputId obtained from
      }
    }
  ]
}
```

**Produces:** application/json

**Example:** Use the following script, to create a local repository with default values.

```

localRepository('local-default') {
    description "Public description"
    notes "Some internal notes"
    includesPattern "**/*" // default
    excludesPattern "" // default
    repoLayoutRef "maven-2-default"
    packageType "generic" // "maven" | "gradle" | "ivy" | "sbt" | ... | "generic"
    debianTrivialLayout false
    checksumPolicyType "client-checksums" // default | "server-generated-checksums"
    handleReleases true // default
    handleSnapshots true // default
    maxUniqueSnapshots 0 // default
    snapshotVersionBehavior "unique" // "non-unique" default | "deployer"
    suppressPomConsistencyChecks false // default
    blackedOut false // default
    propertySets // (["ps1", "ps2"])
    archiveBrowsingEnabled false
    calculateYumMetadata false
    yumRootDepth 0
}

```

```

POST /api/v2/execute_scripts/repositories
{
    "scriptMappings":[
        {
            "instanceName": "DRTarget",
            "scriptNames":[
                "local-default"
            ]
        }
    ]
}

```

## Update Repository

**Description:** Updates a repository in a set of Artifactory instances. For more information, please refer to [Configuring Repositories](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** PUT /api/v2/execute\_scripts/repositories

**Consumes:** application/json

```

{
    "scriptMappings" :[
+
+
repository
repository. The user inputs must
on the user input type
        {
            "instanceName" : <string>, //Artifactory instance on which to update the
            "repositoryKey" : <string>, //Name of the repository to update
            "scriptNames" :[<string>], //scripts to apply when updating the
            "scriptUserInputs" : //user inputs required for the scripts applied
            {
                <userId> : <user input value> // string or object, depending
            }
        }
    ]
}

```

**Produces:** application/json

**Example:** Use the following script called local-repo-userinput to update the description field on local repository ext-release-local on instance Master.

```

test = userInput (
  type : "REPOSITORY"
)
localRepository('repo-variables') {
  description test.name
}

```

```

PUT /api/v2/execute_scripts/repositories
{
  "scriptMappings":[
    {
      "instanceName": "Master",
      "repositoryKey": "ext-release-local",
      "scriptNames":[
        "local-repo-userInput"
      ],
      "scriptUserInputs" : {
        "TemplateExecutor#0#test#0": {"instanceName": "Master", "repositoryKey": "ext-
release-local" }
      }
    }
  ]
}

Response:
{
  "data": [
    {
      "success": true,
      "instanceName": "Master",
      "repositoryKey": "ext-release-local"
    }
  ]
}

```

## SECURITY

### Create User

**Description:** Creates a user on a set of Artifactory instances. For more information, please refer to [Managing Users](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** POST /api/v2/security/users

**Consumes:** application/json

```

{
+   "instanceNames": [<string>], // The Artifactory instances on
which to create this user
+   "user" :
    {
+       "name" : <string>, // The user's name
+       "email" : <string>, // The
user's email
+       "password" : <string>, // The user's
password in clear-text
+       "admin" : <boolean>, // If
true, this is an admin user
+       "profileUpdatable" : <boolean>, // If true, this user can update
their profile
+       "internalPasswordDisabled" : <boolean> // If true, this user cannot use internal
password when external authentication (such as LDAP) is enabled.
    }
}

```

**Produces:** application/json

**Example:** Create user "johns" with the below parameters on Artifactory instances "cluster-121-10100" and "cluster-126-10100"

```
POST /api/v2/security/users
{
  "instanceNames": ["cluster-121-10100","cluster-126-10100"],
  "user": {
    "name": "johns",
    "email": "johns@somewhere.com",
    "password": "12345678",
    "admin": false,
    "profileUpdatable": false,
    "internalPasswordDisabled": false
  }
}
```

Response body:

```
{
  "data": [
    {
      "success": true,
      "instanceName": "cluster-121-10100"
    },
    {
      "success": true,
      "instanceName": "cluster-126-10100"
    }
  ]
}
```

## Update User

**Description:** Updates a user on a set of Artifactory instances. For more information, please refer to [Managing Users](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** PUT /api/v2/security/users/{username}

**Consumes:** application/json

```
{
+   "instanceNames": [<string>],           // The Artifactory instances on which to create
this user
+   user:
  {
+     "email" : <string>,                 // The user's email
    "password" : <string>,               // The user's password in clear-text
    "admin" : <boolean>, // If true, this is an admin user
    "profileUpdatable" : <boolean>, // If true, this user can update their profile
    "internalPasswordDisabled" : <boolean> // If true, this user cannot use internal password
when external authentication (such as LDAP) is enabled.
  }
}
```

**Produces:** application/json

**Example:** Update the parameters of user "johns" in Artifactory instances "cluster-121-10100" and "cluster-126-10100"

```
PUT /api/v2/security/users/johns
{
  "instanceNames": ["cluster-121-10100","cluster-126-10100"],
  "user": {
    "name" : "johns"
    "email": "johns@newdomain.com"
    "password": "changed",
    "admin": false,
    "profileUpdatable": true,
    "internalPasswordDisabled": false
  }
}
```

```
Response:
{
  "data": [
    {
      "success": true,
      "instanceName": "cluster-121-10100"
    },
    {
      "success": true,
      "instanceName": "cluster-126-10100"
    }
  ]
}
```

## Create User Group

**Description:** Creates a user group on a set of Artifactory instances. For more information please refer to [Creating and Editing Groups](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** POST /api/v2/security/user\_groups

**Consumes:** application/json

```
{
+   "instanceNames": [<string>],           // The Artifactory instances on which to create
this user
+   "userGroup" : {
+     "name" : <string>,                   // The group's name
      "description" : <string>,           // A description for this group
      "autoJoin" : <boolean>,           // If true, new users created in the target Artifactory instance
will automatically be added to this group
      "users" : [<string>]               // The list of users (by user name) to include in this group
    }
}
```

**Produces:** application/json

**Example:** Creates a user group called "developers" along with the specified parameters on Artifactory instances "cluster-121-10100" and "cluster-126-10100"

```

POST /api/v2/security/user_groups
{
  "instanceNames": ["cluster-121-10100", "cluster-126-10100"],
  "userGroup": {
    "name": "developers",
    "description": "The developer group",
    "autoJoin": false,
    "users": ["johns", "ronaldm"]
  }
}

Response:
{
  "data": [
    {
      "success": true,
      "instanceName": "cluster-121-10100"
    },
    {
      "success": true,
      "instanceName": "cluster-126-10100"
    }
  ]
}

```

## Update User Group

**Description:** Updates a user group on a set of Artifactory instances. For more information please refer to [Creating and Editing Groups](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** PUT /api/v2/security/user\_groups/{group name}

**Consumes:** application/json

```

{
+   "instanceNames": [<string>],           // The Artifactory instances on which to update
this group
+   "userGroup": {
      "name" : <string>
      "autoJoin" : <boolean>,           // If true, new users created in the target Artifactory instance
will automatically be added to this group
      "description" : <string> // A description for this group
      "users" : [<string>],           // The new list of users (by user name) to include in this group.
This list replaces the current set of users in the group
    }
}

```

**Produces:** application/json

**Example:** Update the "developers" user group with the specified parameters on Artifactory instances "cluster-121-10100" and "cluster-126-10100"



```

PUT /api/v2/security/user_groups/developers
{
  "instanceNames": ["cluster-121-10100","cluster-126-10100"],
  "userGroup": {
    "name" : "developers,"
    "description": "The changed developer group",
    "autoJoin": false,
    "users": []
  }
}

Response:
{
  "data": [
    {
      "success": true,
      "instanceName": "cluster-121-10100"
    },
    {
      "success": true,
      "instanceName": "cluster-126-10100"
    }
  ]
}

```

## Create Permission Target

**Description:** Creates a permission target on a set of Artifactory instances. For more information, please refer to [Managing Permissions](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** POST /api/v2/security/permission\_targets

**Consumes:** application/json

```

{
+   "instanceNames" : [<string>],           // The Artifactory instances to which this
permission target should be applied
+   "permissionTarget" : {
+       "name" : <string>,                 // A name for this permission
target
+       "repositories" : [<string>],       // The repositories to which this permission
target applies
+       "anyRemote" : <boolean>,           // If true, applies to any remote repository
+       "anyLocal" : <boolean>,           // If true, applies to any local repository
+       "excludesPattern" : <string>,     // Excludes pattern to filter out certain
repositories
+       "includesPattern" : <string>,     // Includes pattern to filter in certain
repositories
+       "principals":                     // The principles to which this permission target should be applied
{
+         "users" :
+         {
+           <userName> : [{permission}] // The users and corresponding permissions they
are given where m=admin; d=delete; w=deploy; n=annotate; r=read
+         },
+         "groups" :
+         {
+           <groupName> : [{permission}] // The groups and corresponding permissions
they are given where m=admin; d=delete; w=deploy; n=annotate; r=read
+         }
+       }
+     }
}

```

**Produces:** application/json

**Example:** Creates a permission target called "releasers" on repository "ext-release-local" for users "johns" and "colonels" and groups "developers" and "itmanagers" (each with corresponding permissions) on Artifactory instances "cluster-121-10100" and "cluster-126-10100".

```

POST /api/v2/security/permission_targets
{
  "instanceNames": ["cluster-121-10100", "cluster-126-10100"],
  "permissionTarget" : {
    "name" : "releasers",
    "includesPattern" : "***",
    "excludesPattern" : "",
    "anyLocal" : true,
    "anyRemote" : false,
    "repositories" : ["ext-release-local"],
    "principals": {
      "users" : {
        "johns": ["r", "w", "m"],
        "colonels" : ["d", "w", "n", "r"]
      },
      "groups" : {
        "developers" : ["m", "r", "n"],
        "itmanagers" : ["r"]
      }
    }
  }
}

```

```

Response:
{
  "data": [
    {
      "success": true,
      "instanceName": "cluster-121-10100"
    },
    {
      "success": true,
      "instanceName": "cluster-126-10100"
    }
  ]
}

```

## Update Permission Target

**Description:** Updates a permission target on a set of Artifactory instances. Any permissions previously set for users or groups are replaced with the specified permissions. For more information, please refer to [Managing Permissions](#) in the Artifactory documentation.

**Since:** 1.0

**Security:** Requires an admin user

**Usage:** PUT /api/v2/security/permission\_targets/{permission target name}

**Consumes:** application/json

```

{
+   "instanceNames" : [<string>],           // The Artifactory instances to which this
permission target should be applied
+   "permissionTarget" :
    {
+       "repositories" : [<string>],       // The repositories to which this permission
target applies
        "anyRemote" : <boolean>,         // If true, applies to any remote repository
        "anyLocal" : <boolean>,         // If true, applies to any local repository
        "excludesPattern" : <string>,    // Excludes pattern to filter out certain
repositories
        "includesPattern" : <string>,    // Includes pattern to filter in certain
repositories
        "principals" :                  // The principles to which this permission target should be applied
        {
            "users" :
            {
                <userName> : [{permission}] // The users and corresponding permissions they
are given where m=admin; d=delete; w=deploy; n=annotate; r=read
            },
            "groups" :
            {
                <groupName> : [{permission}] // The groups and corresponding permissions
they are given where m=admin; d=delete; w=deploy; n=annotate; r=read
            }
        }
    }
}

```

**Produces:** application/json

**Example:** Updates the permission target called "releasers" on repository "ext-release-local" for "itmanagers" awarding them full permissions on any remote repository on Artifactory instances "cluster-121-10100" and "cluster-126-10100".

```

PUT /api/v2/permissionTargets/releasers
{
  "instanceNames": ["cluster-121-10100","cluster-126-10100"],
  "permissionTarget" : {
    "anyRemote" : true,
    "repositories" : ["ext-release-local"],
    "principals": {
      "users" : {
        "colonels" : ["d","w","n","r"]
      },
      "groups" : {
        "itmanagers" : ["d","w","n","r"]
      }
    }
  }
}

```

## LICENSE BUCKETS

### Bucket Status

**Description:** Gets the status of the specified license bucket .

**Since:** 1.3

**Security:** None

**Usage:** GET /api/v2/buckets/{bucket-name}/report

**Consumes:** None

**Produces:** application/json

```

{
  "data": {
    "id": <bucket-id>,
    "size": <number of licenses in the bucket>,
    "licenses": {
      "used": <number of licenses currently being used>,
      "available": <number of licenses currently available>,
      "maxUsed": <max number of licenses that were ever used concurrently>,
    }
  }
}

```

**Sample usage:** Getthestatusofbucket with IDabcdefg

```

GET /api/v2/buckets/abcdefg/report
{
  "data": {
    "id": "abcdefg",
    "size": 14,
    "licenses": {
      "used": 1,
      "available": 13,
      "maxUsed": 1
    }
  }
}

```

## Attach License

**Description:** Attaches a license from the specified bucket, or a number of licenses to an Artifactory 5.x HA cluster.

**Since:** 1.3

**Security:** None

**Usage:** POST /api/v2/attach\_lic/buckets/{bucket-name}

**Consumes:** application/json

```

{
  "instanceName" : <Artifactory instance to which the license should be attached>,
  "deploy" : true <If true, the license is actually deployed to the instance>,
  "numberOfLicenses" : < number of licenses to deploy to an Artifactory 5.x HA cluster. Optional,
  default value is 1>,
}

```

**Sample usage:**

```

POST /api/v2/attach_lic/buckets/abcdefg
{
  "instanceName" : "Master",
  "deploy" : true,
  "numberOfLicenses" : 7
}

Response:
{
  "data": {
    "success": true,
    "message": "License deployed to instance Master"
  }
}

```

## Detach License

**Description:** Detaches a license from an Artifactory instance and returns it to the specified bucket

**Since:** 1.3

**Security:** None

**Usage:** DELETE /api/v2/detach\_lic/buckets/{bucket-name}

**Consumes:** application/json

```
{
  "InstanceName" : <Artifactory instance from which to detach the license>
}
```

#### Sample usage:

```
DELETE /api/v2/detach_lic/buckets/abcdefg
{
  "InstanceName" : "Master"
}

Response (success)
2014

Response (error - instance is online)
{
  "errors": [
    {
      "type": "Exception",
      "message": "Instance `Master` is still using the license."
    }
  ]
}
```

## DISASTER RECOVERY

### Create a DR Pair

**Description:** Matches up a Master and Target instance as a DR pair.

**Since:** 1.5

**Security:** None

**Usage:** POST /api/v2/dr-configs

**Consumes:** application/json

```
{
  "source": "master_instance_name",
  "target": "target_instance_name"
}
```

#### Sample usage:

```

POST /api/v2/dr-configs
{
  "source": "corp-west",
  "target": "corp-west-dr"
}

Response:
{
  "data": {
    "id": "0be405a8-2713-4ec6-a775-d34072e1b2d5",
    "sourceId": "276dd14f-8579-4f64-967e-46214fc7eafe",
    "targetId": "daclf570-096d-4104-9b06-881588e0adc0",
    "active": "NONE",
    "drReplicationsEnabled": false,
    "state": "NONE"
  }
}

```

## SYSTEM

### System Health Check

**Description:** Simple ping to Mission Control to see if it is running.

**Since:** 1.0

**Security:** None

**Usage:** GET /api/v2/ping

**Consumes:** None

**Produces:** application/json

```

{
  "data" : true
}

```

## V1 to V2 Mapping

To facilitate updating your scripts to use the latest API, the following table presents a mapping between endpoints in V1 and the corresponding endpoints in V2 of the REST API.

Category	Description	Method	V1 Endpoint	V2 Endpoint
<b>Instances</b>	Get list of artifactory instances	GET	/api/v1/instances	<a href="#">/api/v2/instances</a>
	Add instance	POST	/api/v1/instances	<a href="#">/api/v2/instances</a>
	Update Instance	PUT	Not available	<a href="#">/api/v2/instances/{name}</a>
	Get repositories for instance	GET	/api/v1/instances/{name}/repositories	<a href="#">/api/v2/instances/{name}/repositories</a>
	Delete instance by name	DELETE	/api/v1/instances/{name}	<a href="#">/api/v2/instances/{name}</a>
<b>Security</b>	Create user	POST	api/v1/users	<a href="#">/api/v2/security/users</a>
	Update user	PUT	/api/v1/users/{name}	<a href="#">/api/v2/security/users/{name}</a>
	Create user group	POST	/api/v1/userGroups	<a href="#">/api/v2/security/user_groups</a>
	Update user group	PUT	/api/v1/userGroups/{name}	<a href="#">/api/v2/security/user_groups/{name}</a>
	Create permission target	POST	/api/v1/permissionTargets	<a href="#">/api/v2/security/permission_targets</a>
	Update permission target by name	PUT	/api/v1/permissionTargets/{name}	<a href="#">/api/v2/security/permission_targets/{name}</a>
<b>License Buckets</b>	Get bucket status	GET	/api/v1/buckets/{id}/status	<a href="#">/api/v2/buckets/{id}/report</a>
	Attach a license	POST	/api/v1/buckets/{id}/licenses	<a href="#">/api/v2/attach_lic/buckets/{id}</a>
	Detach a license	DELETE	/api/v1/buckets/{id}/licenses	<a href="#">/api/v2/detach_lic/buckets/{id}</a>
<b>Execute Scripts</b>	Create repository	POST	/api/v1/repositories	<a href="#">/api/v2/execute_scripts/repositories</a>
	Update repository	PUT	/api/v1/repositories	<a href="#">/api/v2/execute_scripts/repositories</a>
	Execute scripts on instance	PUT	/api/v1/instances	<a href="#">/api/v2/execute_scripts/instances</a>

<b>Scripts</b>	Get scripts	GET	/api/v1/scripts	<a href="#">/api/v2/scripts</a>
	Get script user inputs	GET	/api/v1/userInputs	<a href="#">/api/v2/scripts/user_inputs</a>
<b>System</b>	System health check (ping)	GET	/api/v1/ping	<a href="#">/api/v2/ping</a>