

Artifactory REST API

Overview

Artifactory exposes its REST API through an auto-generated WADL file (courtesy of the [Jersey REST framework](#)).

This provides a convenient and up-to-date self-descriptive API and can be used by various tools/frameworks to automate the creation of REST calls.

The WADL file is available at the following URL:

`http://server:port/artifactory/api/application.wadl`

Usage

Artifactory REST API endpoints can be invoked in any of the standard ways to invoke a RESTful API. This section describes how to use the Artifactory REST API using cURL as an example.

Using and Configuring cURL

You can download cURL [here](#). Learn how to use and configure cURL [here](#).

Authentication

Artifactory's REST API supports four forms of authentication:

- Basic authentication using your username and password
- Basic authentication using your username and [API Key](#).
- Using a dedicated header (`X-JFrog-Art-Api`) with your API Key.
- Using an [access token](#) instead of a password for basic authentication.
- Using an [access token](#) as a bearer token in an authorization header (`Authorization: Bearer`) with your access token.

Artifactory version

To use your API key for **Basic Authentication**, it must be generated using Artifactory 4.4.3 or later. If generated on a previous version, you **must** regenerate your API key and use the new key as a password for basic authentication.

Using JFrog CLI

JFrog CLI is a compact and smart client that provides a simple interface to automate access to Artifactory. As a wrapper to the REST API, it offers a way to simplify automation scripts making them more readable and easier to maintain, features such as parallel uploads and downloads, checksum optimization and wildcards/regular expressions make your scripts more efficient and reliable. Please note that several of the functions available through the REST API are also available through JFrog CLI and you should consider which method best meets your needs.

For more details on download, installation and usage of JFrog CLI, please refer to the [JFrog CLI User Guide](#).

Example - Deploying an Artifact

The example below demonstrates how to invoke the [Deploy Artifact](#) REST API.

- You are using cURL from the unix command line, and are presently working from the **home (~) directory** of the user '**myUser**'
- You wish to deploy the file '*myNewFile.txt*', which is located in your Desktop directory, ('*~/Desktop/myNewFile.txt*')
- You have Artifactory running on your local system, on **port 8081**
- You wish to deploy the artifact into the '*my-repository*' repository, under the '*my/new/artifact/directory/*' folder structure, and wish to store the file there as '*file.txt*'
- You have configured a user in Artifactory named '*myUser*', with password '*myP455w0rd!*', and this user has permissions to deploy artifacts

- Your API Key is 'ABcdEF'
- Where possible, the same example is demonstrated using JFrog CLI

To deploy the file using your **username and password** for authentication, you would use the following command:

Using cURL with the REST API

```
curl -u myUser:myP455w0rd! -X PUT
"http://localhost:8081/artifactory/my-repository/my/new/artifact/directory/file.txt" -T
Desktop/myNewFile.txt
```

Using JFrog CLI

```
jfrog rt u file.txt
my-repository/my/new/artifact/directory/
--user=myUser --password=myP455w0rd!
```

To deploy the file using your **API Key** for basic authentication, you would use the following command:

Using cURL with the REST API

```
curl -u myUser:ABcdEF -X PUT
"http://localhost:8081/artifactory/my-repository/my/new/artifact/directory/file.txt" -T
Desktop/myNewFile.txt
```

Using JFrog CLI

```
jfrog rt u file.txt
my-repository/my/new/artifact/directory/
--apiKey=ABcdEF
```

To deploy the file using your **API Key** in a header, you would use the following command:

Using cURL with the REST API

```
curl -H "X-JFrog-Art-API:ABcdEF" -X PUT
"http://localhost:8081/artifactory/my-repository/my/new/artifact/directory/file.txt" -T
Desktop/myNewFile.txt
```

To deploy the file using your **access token** for basic authentication, you would use the following command:

Using cURL with an access token

```
curl -u myUser:<Token> -X PUT  
"http://localhost:8081/artifactory/my-repository/my/new/artifact/directory/file.txt" -T  
Desktop/myNewFile.txt
```

To deploy the file using your **access token** in a header, you would use the following command:

Using cURL with an access token

```
curl -H "Authorization: Bearer <Token>" -X PUT  
"http://localhost:8081/artifactory/my-repository/my/new/artifact/directory/file.txt" -T  
Desktop/myNewFile.txt
```

REST Resources

The sections below provide a comprehensive listing of the REST resources exposed by Artifactory.

For details on handling errors please refer to [ERROR RESPONSES](#) below.

Usage of REST resources is subject to security restrictions applicable to each individual resource.

BUILDS

All Builds

Description: Provides information on all builds

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/build

Produces: application/vnd.org.jfrog.build.Builds+json

Sample Output:

```
GET /api/build
{
  "uri":
"http://localhost:8081/artifactory/api/build"
  "builds" : [
    {
      "uri" : "/my-build",
      "lastStarted" : ISO8601
      (yyyy-MM-dd'T'HH:mm:ss.SSSZ)
    },
    {
      "uri" : "/jackrabbit",
      "lastStarted" : ISO8601
      (yyyy-MM-dd'T'HH:mm:ss.SSSZ)
    }
  ]
}
```

Build Runs

Description: Build Runs

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/build/{buildName}

Produces: application/vnd.org.jfrog.build.BuildsByName+json

Sample Output:

```
GET /api/build/my-build
{
  "uri":
"http://localhost:8081/artifactory/api/build/my-build"
  "buildsNumbers" : [
    {
      "uri" : "/51",
      "started" : ISO8601
      (yyyy-MM-dd'T'HH:mm:ss.SSSZ)
    },
    {
      "uri" : "/52",
      "started" : ISO8601
      (yyyy-MM-dd'T'HH:mm:ss.SSSZ)
    }
  ]
}
```

Build Upload

Description: Upload Build

Security: Requires a privileged user (can be anonymous)

Notes: All build modules must have the `build.name` and `build.number` properties set as well as the correct SHA1 and MD5 in order to be properly linked in the build info.

Usage: `PUT /api/build/ -H "Content-Type: application/json" --upload-file build.json`

Consumes: `application/vnd.org.jfrog.build.BuildsByName+json`

Example: `curl -X PUT "http://localhost:8081/artifactory/api/build" -H "Content-Type: application/json" --upload-file build.json`

Sample format:

▼ [Click to view sample build.json](#)

```
{
  "properties" : {
    /* Environment variables and properties
    collected from the CI server.

    The "buildInfo.env." prefix is added to
    environment variables and build related properties.
    For system variables there's no prefix. */
    "buildInfo.env.JAVA_HOME" : "",
    ...
  },
  "version" : "1.0.1", // Build Info schema version
  "name" : "My-build-name", // Build name
  "number" : "28", // Build number
  "type" : "MAVEN", // Build type (values currently
supported: MAVEN, GRADLE, ANT, IVY and GENERIC)
  "buildAgent" : { // Build tool information
    "name" : "Maven", // Build tool type
    "version" : "3.0.5" // Build tool version
  },
  "agent" : { // CI Server information
    "name" : "Jenkins", // CI server type
    "version" : "1.554.2" // CI server version
  },
  "started" : "2014-09-30T12:00:19.893+0300", //
Build start time in the format of
yyyy-MM-dd'T'HH:mm:ss.SSSZ
  "artifactoryPluginVersion" : "2.3.1",
  "durationMillis" : 9762, // Build duration in
milliseconds
  "artifactoryPrincipal" : "james", // Artifactory
principal (the Artifactory user used for
deployment)
  "url" :
"http://my-ci-server/jenkins/job/My-project-name/2
8/", // CI server URL
  "vcsRevision" :
"e4ab2e493afd369ae7bdc90d69c912e8346a3463", // VCS
revision
  "vcsUrl" :
"https://github.com/github-user/my-project.git", //
VCS URL
}
```

```

    "licenseControl" : { // Artifactory License
Control information
        "runChecks" : true, // Artifactory will run
automatic license scanning after the build is
complete (true/false)
        "includePublishedArtifacts" : true, // Should
Artifactory run license checks on the build
artifacts, in addition to the build dependencies
(true/false)
        "autoDiscover" : true, // Should Artifactory
auto discover licenses (true/false)
        "scopesList" : "", // A space-separated list of
dependency scopes/configurations to run license
violation checks on. If left empty all dependencies
from all scopes will be checked.
        "licenseViolationsRecipientsList" : "" //
Emails of recipients that should be notified of
license violations in the build info
(space-separated list)
    },
    "buildRetention" : { // Build retention
information
        "deleteBuildArtifacts" : true, // Automatically
remove build artifacts stored in Artifactory
(true/false)
        "count" : 100, // The maximum number of builds
to store in Artifactory.
        "minimumBuildDate" : 1407345768020, // Earliest
build date to store in Artifactory
        "buildNumbersNotToBeDiscarded" : [ ] // List of
build numbers that should not be removed from
Artifactory
    },
    /* List of build modules */
    "modules" : [ { // The build's first module
        "properties" : { // Module properties
            "project.build.sourceEncoding" : "UTF-8"
        },
        "id" : "org.jfrog.test:multi2:4.2-SNAPSHOT", //
Module ID
        /* List of module artifacts */
        "artifacts" : [ {
            "type" : "jar",
            "sha1" :
"2ed52ad1d864aec00d7a2394e99b3abca6d16639",
            "md5" : "844920070d81271b69579e17ddc6715c",
            "name" : "multi2-4.2-SNAPSHOT.jar"
        }, {
            "type" : "pom",
            "sha1" :
"e8e9c7d790191f4a3df3a82314ac590f45c86e31",
            "md5" : "1f027d857ff522286a82107be9e807cd",
            "name" : "multi2-4.2-SNAPSHOT.pom"
        }
    ]
}

```

```

    } ],
    /* List of module dependencies */
    "dependencies" : [ {
        "type" : "jar",
        "sha1" :
"99129f16442844f6a4a11ae22fbbbee40b14d774f",
        "md5" : "1f40fb782a4f2cf78f161d32670f7a3a",
        "id" : "junit:junit:3.8.1",
        "scopes" : [ "test" ]
    } ]
}, { // The build's second module
    "properties" : { // Module properties
        "project.build.sourceEncoding" : "UTF-8"
    },
    "id" : "org.jfrog.test:multi3:4.2-SNAPSHOT", //
Module ID
    /* List of module artifacts */
    "artifacts" : [ { // Module artifacts
        "type" : "war",
        "sha1" :
"df8e7d7b94d5ec9db3bfc92e945c7ff4e2391c7c",
        "md5" : "423c24f4c6e259f2774921b9d874a649",
        "name" : "multi3-4.2-SNAPSHOT.war"
    }, {
        "type" : "pom",
        "sha1" :
"656330c5045130f214f954643fdc4b677f4cf7aa",
        "md5" : "b0afa67a9089b6f71b3c39043b18b2fc",
        "name" : "multi3-4.2-SNAPSHOT.pom"
    } ],
    /* List of module dependencies */
    "dependencies" : [ {
        "type" : "jar",
        "sha1" :
"a8762d07e76cfde2395257a5da47ba7c1dbd3dce",
        "md5" : "b6a50c8a15ece8753e37cbe5700bf84f",
        "id" : "commons-io:commons-io:1.4",
        "scopes" : [ "compile" ]
    }, {
        "type" : "jar",
        "sha1" :
"342d1eb41a2bc7b52fa2e54e9872463fc86e2650",
        "md5" : "2a666534a425add50d017d4aa06a6fca",
        "id" :
"org.codehaus.plexus:plexus-utils:1.5.1",
        "scopes" : [ "compile" ]
    }, {
        "type" : "jar",
        "sha1" :
"449ea46b27426eb846611a90b2fb8b4dcf271191",
        "md5" : "25c0752852205167af8f31aleb019975",
        "id" :
"org.springframework:spring-beans:2.5.6",

```

```

        "scopes" : [ "compile" ]
    } ]
} ],
/* List of issues related to the build */
"issues" : {
    "tracker" : {
        "name" : "JIRA",
        "version" : "6.0.1"
    },
    "aggregateBuildIssues" : true, //whether or
not there are issues that already appeared in
previous builds
    "aggregationBuildStatus" : "Released",
    "affectedIssues" : [ {
        "key" : "RTFACT-1234",
        "url" :
"https://www.jfrog.com/jira/browse/RTFACT-1234",
        "summary" : "Description of the relevant
issue",
        "aggregated" : false //whether or not this
specific issue already appeared in previous builds
    }, {
        "key" : "RTFACT-5469",
        "url" :
"https://www.jfrog.com/jira/browse/RTFACT-5678",
        "summary" : "Description of the relevant
issue",
        "aggregated" : true
    } ]
} ]

```



```
    },  
  }  
}
```

Build Info

Description: Build Info

Since: 2.2.0

Security: Requires a privileged user with deploy permissions (can be anonymous)

Usage: GET /api/build/{buildName}/{buildNumber}

Produces: application/vnd.org.jfrog.build.BuildInfo+json

Sample Output:

```
GET /api/build/my-build/51  
{  
  "uri":  
    "http://localhost:8081/artifactory/api/build/my-build  
    /51"  
  "buildInfo" : {  
    ...  
  }  
}
```

Builds Diff

Description: Compare a build artifacts/dependencies/environment with an older build to see what has changed (new artifacts added, old dependencies deleted etc).

Since: 2.6.6

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/build/{buildName}/{buildNumber}?diff={OlderbuildNumber}

Produces: application/vnd.org.jfrog.build.BuildsDiff+json

Sample Output:

```
GET /api/build/my-build/51?diff=50
{
  "artifacts": {
    "updated": [],
    "unchanged": [],
    "removed": [],
    "new": []
  }, "dependencies": {
    "updated": [],
    "unchanged": [],
    "removed": [],
    "new": []
  }, "properties": {
    "updated": [],
    "unchanged": [],
    "removed": [],
    "new": []
  }
}
```

Page contents

- Overview
- Usage
 - Authentication
 - Using JFrog CLI
 - Example - Deploying an Artifact
- BUILDS
 - All Builds
 - Build Runs
 - Build Upload
 - Build Info
 - Builds Diff
 - Build Promotion
 - Promote Docker Image
 - Delete Builds
 - Build Rename
 - Push Build to Bintray
 - Distribute Build
 - Control Build Retention
- ARTIFACTS & STORAGE
 - Folder Info
 - File Info
 - Get Storage Summary Info
 - Item Last Modified
 - File Statistics
 - Item Properties
 - Set Item Properties
 - Delete Item Properties
 - Set Item SHA256 Checksum
 - Retrieve Artifact
 - Retrieve Latest Artifact
 - Retrieve Build Artifacts Archive
 - Retrieve Folder or Repository Archive
 - Trace Artifact Retrieval
 - Archive Entry Download
 - Create Directory
 - Deploy Artifact
 - Deploy Artifact by Checksum
 - Deploy Artifacts from Archive
 - Push a Set of Artifacts to Bintray

- Push Docker Tag to Bintray
- Distribute Artifact
- File Compliance Info
- Delete Item
- Copy Item
- Move Item
- Get Repository Replication Configuration
- Set Repository Replication Configuration
- Update Repository Replication Configuration
- Delete Repository Replication Configuration
- Scheduled Replication Status
- Pull/Push Replication
- Pull/Push Replication (Deprecated)
- Create or Replace Local Multi-push Replication
- Update Local Multi-push Replication
- Delete Local Multi-push Replication
- Enable or Disable Multiple Replications
- Get Global System Replication Configuration
- Block System Replication
- Unblock System Replication
- Artifact Sync Download
- Folder Sync (Deprecated)
- File List
- Get Background Tasks
- Empty Trash Can
- Delete Item From Trash Can
- Restore Item from Trash Can
- Optimize System Storage
- SEARCHES
 - Artifactory Query Language (AQL)
 - Artifact Search (Quick Search)
 - Archive Entries Search (Class Search)
 - GAVC Search
 - Property Search
 - Checksum Search
 - Bad Checksum Search
 - Artifacts Not Downloaded Since
 - Artifacts With Date in Date Range
 - Artifacts Created in Date Range
 - Pattern Search
 - Builds for Dependency
 - License Search
 - Artifact Version Search
 - Artifact Latest Version Search Based on Layout
 - Artifact Latest Version Search Based on Properties
 - Build Artifacts Search
 - List Docker Repositories
- SECURITY
 - Get Users
 - Get User Details
 - Get User Encrypted Password
 - Create or Replace User
 - Update User
 - Delete User
 - Expire Password for a Single User
 - Expire Password for Multiple Users
 - Expire Password for All Users
 - Unexpire Password for a Single User
 - Change Password
 - Get Password Expiration Policy
 - Set Password Expiration Policy
 - Configure User Lock Policy
 - Retrieve User Lock Policy
 - Get Locked Out Users
 - Unlock Locked Out User
 - Unlock Locked Out Users
 - Unlock All Locked Out Users
 - Create API Key
 - Regenerate API Key
 - Get API Key
 - Revoke API Key
 - Revoke User API Key

- Revoke All API Keys
- Get Groups
- Get Group Details
- Create or Replace Group
- Update Group
- Delete Group
- Get Permission Targets
- Get Permission Target Details
- Create or Replace Permission Target
- Delete Permission Target
- Effective Item Permissions
- Security Configuration
- Save Security Configuration (Deprecated)
- Activate Master Key Encryption
- Deactivate Master Key Encryption
- Set GPG Public Key
- Get GPG Public Key
- Set GPG Private Key
- Set GPG Pass Phrase
- Create Token
- Refresh Token
- Revoke Token
- Get Service ID
- Get Certificates
- Add Certificate
- Delete Certificate
- REPOSITORIES
 - Get Repositories
 - Repository Configuration
 - Create Repository
 - Update Repository Configuration
 - Delete Repository
 - Remote Repository Configuration (Deprecated)
 - Calculate YUM Repository Metadata
 - Calculate NuGet Repository Metadata
 - Calculate Npm Repository Metadata
 - Calculate Maven Index
 - Calculate Maven Metadata
 - Calculate Debian Repository Metadata
 - Calculate Opkg Repository Metadata
 - Calculate Bower Index
- SYSTEM & CONFIGURATION
 - System Info
 - System Health Ping
 - Verify Connection
 - General Configuration
 - Save General Configuration
 - Update Custom URL Base
 - License Information
 - Install License
 - HA License Information
 - Install HA Cluster Licenses
 - Delete HA Cluster License
 - Version and Add-ons information
 - Get Reverse Proxy Configuration
 - Update Reverse Proxy Configuration
 - Get Reverse Proxy Snippet
 - Create Bootstrap Bundle
- PLUGINS
 - Execute Plugin Code
 - Retrieve Plugin Info
 - Retrieve Plugin Info Of A Certain Type
 - Retrieve Build Staging Strategy
 - Execute Build Promotion
 - Reload Plugins
- IMPORT & EXPORT
 - Import Repository Content
 - Import System Settings Example
 - Full System Import
 - Export System Settings Example
 - Export System
- SUPPORT

- [Create Bundle](#)
- [List Bundles](#)
- [Get Bundle](#)
- [Delete Bundle](#)
- [ERROR RESPONSES](#)

Read More

- [Repository Configuration JSON](#)
- [Security Configuration JSON](#)
- [System Settings JSON](#)

Build Promotion

Description: Change the status of a build, optionally moving or copying the build's artifacts and its dependencies to a target repository and setting properties on promoted artifacts.

All artifacts from all scopes are included by default while dependencies are not. Scopes are additive (or).

Since: 2.3.3

Notes: Requires Artifactory Pro

Security: Requires a privileged user (can be anonymous)

Usage: POST /api/build/promote/{buildName}/{buildNumber}

Consumes: application/vnd.org.jfrog.artifactory.build.PromotionRequest+json

```

POST /api/build/promote/my-build/51
{
  "status": "staged",          // new build status (any string)
  "comment" : "Tested on all target platforms.", // An optional comment
  describing the reason for promotion. Default: ""
  "ciUser": "builder",        // The user that invoked promotion from the CI
  server
  "timestamp" : ISO8601,      // the time the promotion command was received
  by Artifactory (It needs to be unique),
                                // the format is: 'yyyy-MM-dd'T'HH:mm:ss.SSSZ'. Example:
  '2016-02-11T18:30:24.825+0200'.
  "dryRun" : false,           // run without executing any operation in
  Artifactory, but get the results to check if the operation can succeed.
  Default: false
  "sourceRepo" : "libs-snapshot-local", // optional repository from which
  the build's artifacts will be copied/moved
  "targetRepo" : "libs-release-local",  // optional repository to move or
  copy the build's artifacts and/or dependencies
  "copy": false,              // whether to copy instead of move, when a target
  repository is specified. Default: false
  "artifacts" : true,         // whether to move/copy the build's artifacts.
  Default: true
  "dependencies" : false,     // whether to move/copy the build's
  dependencies. Default: false.
  "scopes" : [ "compile", "runtime" ], // an array of dependency scopes
  to include when "dependencies" is true
  "properties": {             // a list of properties to attach to the build's
  artifacts (regardless if "targetRepo" is used).
    "components": ["c1","c3","c14"],
    "release-name": ["fb3-ga"]
  },
  "failFast": true           // fail and abort the operation upon receiving an
  error. Default: true
}

```

Produces: application/vnd.org.jfrog.artifactory.build.PromotionResult+json

Sample Output:

```

{
  "messages" : [
    {
      "level": "error",
      "message": "The repository has denied...."
    },...
  ]
}

```

Promote Docker Image

Description: Promotes a Docker image from one repository to another

Since: 3.7

Notes: Requires Artifactory Pro

Security: Requires a privileged user

Usage: POST api/docker/<repoKey>/v2/promote

Consumes: application/json

```
{
  "targetRepo" : "<targetRepo>",          // The target repository for the
move or copy
  "dockerRepository" : "<dockerRepository>",    // The docker repository
name to promote
  "targetDockerRepository" : "<targetDockerRepository>" // An optional
docker repository name, if null, will use the same name as
'dockerRepository'
  "tag" : "<tag>",                // An optional tag name to promote, if null -
the entire docker repository will be promoted. Available from v4.10.
  "targetTag" : "<tag>",                // An optional target
tag to assign the image after promotion, if null - will use the same tag
  "copy": false                // An optional value to set whether to copy
instead of move. Default: false
}
```

Produces: application/text

Sample Usage:

```
POST api/docker/docker-local/v2/promote
{
  "targetRepo": "docker-prod",
  "dockerRepository": "jfrog/ubuntu"
}
```

Delete Builds

Description: Removes builds stored in Artifactory. Useful for cleaning up old build info data.

If the `artifacts` parameter is evaluated as 1 (0/false by default), build artifacts are also removed provided they have the corresponding `build.name` and `build.number` properties attached to them.

If the `deleteAll` parameter is evaluated as 1 (0/false by default), the whole build is removed.

Since: 2.3.0; artifact removal since 2.3.3;

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: DELETE /api/build/{buildName}[?buildNumbers=n1[,n2]][&artifacts=0/1][&deleteAll=0/1]

Produces: application/text

Sample Usage:

```
DELETE /api/build/my-build?buildNumbers=51,52,55&artifacts=1
```

The following builds has been deleted successfully: 'my-build#51', 'my-build#52', 'my-build#55'.

```
DELETE /api/build/my-build?deleteAll=1
```

All 'my-build' builds have been deleted successfully.

Build Rename

Description: Renames a build stored in Artifactory. Typically used to keep the build info in sync with a renamed build on the CI server.

Since: 2.2.5

Notes: Requires Artifactory Pro

Security: Requires a valid user with deploy permissions

Usage: POST /api/build/rename/{buildName}?to=newBuildName

Produces: application/text

Sample Usage:

```
POST /api/build/rename/myJobName?to=myNewJobName
```

Build renaming of 'myJobName' to 'myNewJobName' was successfully started.

Push Build to Bintray

Deprecated: This endpoint is deprecated and is replaced with [Distribute Build](#)

Description: Push a build to Bintray as a version.

Uses a descriptor file (that must have 'bintray-info' in it's filename and a .json extension) that is included with the build artifacts. For more details, please refer to [Pushing a Build](#).

Signing a version is controlled by the *gpgSign* parameter in the descriptor file, and the *gpgSign* paramater passed to this command. **The value passed to this command always takes precedence over the value in the descriptor file.**

If you also want a passphrase to be applied to your signature, specify *gpgPassphrase=<passphrase>*.

You may omit the descriptor file by passing 6 override parameters (see below). If you wish to use the descriptor file you should pass an empty json string instead.

Since: 3.5.0

Security: Requires a valid user with deploy permissions and Bintray credentials defined (for more details, please refer to [Entering your Bintray credentials](#)).

Usage: POST /api/build/pushToBintray/{build.name}/{build.number}?gpgPassphrase=<passphrase>[&gpgSign=true\false]

Consumes: application/vnd.org.jfrog.artifactory.build.BintrayDescriptorOverrideParams+json

Sample Input:


```

POST
/api/build/pushToBintray/testBuild/1?gpgPassphrase=password&gpgSign=true
{
  "subject": "myUser",
  "repoName": "test",
  "packageName": "overridePkg",
  "versionName": "overrideVer",
  "licenses": ["MIT"],
  "vcs_url": "https://github.com/bintray/bintray-client-java"
}

```

Produces: application/vnd.org.jfrog.artifactory.bintray.BintrayPushResponse+json

Sample Output:

```

{"Message": "Pushing build to Bintray finished successfully."}

```

Distribute Build

Description: Deploys builds from Artifactory to Bintray, and creates an entry in the corresponding Artifactory distribution repository specified.

Notes: Requires Artifactory Pro

Since: 4.8

Security: Requires an authenticated user.

Usage: POST /api/build/distribute/{buildName}/{buildNumber}

Consumes: application/json

```

{
  "publish" : "<true | false>"      // Default: true. If true, builds are
published when deployed to Bintray
  "overrideExistingFiles" : "<true | false>"  // Default: false. If true,
Artifactory overwrites builds already existing in the target path in
Bintray.
      // Existing version attributes are also overridden if defined
in the distribution repository Advanced Configuration
  "gpgPassphrase" : "<passphrase>"      // If specified, Artifactory will GPG
sign the build deployed to Bintray and apply the specified passphrase
  "async" : "<true | false>"      // Default: false. If true, the build will
be distributed asynchronously. Errors and warnings may be viewed in the log
file
  "targetRepo" : "<targetDistributionRepo>",  // The Distribution Repository
into which artifacts should be deployed
  "sourceRepos" : ["<repoKey>"]      // An array of local repositories from
which build artifacts should be deployed
  "dryRun" : "<true | false>"      // Default: false. If true, distribution
is only simulated. No files are actually moved.
}

```

Sample input:

```
POST /api/build/distribute/my-build/1
{
  "targetRepo" : "dist-repo-jfrog-artifactory",
  "sourceRepos" : ["yum-local"]
}
```

Control Build Retention

Description: Specifies retention parameters for build info

Since: 5.2.1

Security: Requires a privileged user with deploy permissions (can be anonymous)

Usage: POST /api/build/retention/{buildName}?async=<true | false>

Consumes: application/json

```
{
  "deleteBuildArtifacts" : <true | false>, // When true, automatically
removes build artifacts stored in Artifactory
  "count" : <count>, // The maximum number of builds to store in
Artifactory.
  "minimumBuildDate" : <date>, // Earliest build date to store in
Artifactory - ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ)
  "buildNumbersNotToBeDiscarded" : [ ] // List of build numbers that
should not be removed from Artifactory
}
```

Sample Usage:

```
POST /api/build/retention/myBuild?async=true

{
  "deleteBuildArtifacts" : true,
  "count" : 100, //
  "minimumBuildDate" : 1407345768020,
  "buildNumbersNotToBeDiscarded" : [ 5, 9]
}
```

ARTIFACTS & STORAGE

Folder Info

Description: Folder Info

For virtual use, the virtual repository returns the unified children. Supported by local, local-cached and virtual repositories.

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/storage/{repoKey}/{folder-path}

Produces: application/vnd.org.jfrog.artifactory.storage.FolderInfo+json

Sample Output:

```
GET /api/storage/libs-release-local/org/acme
{
  "uri":
  "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
  ",
  "repo": "libs-release-local",
  "path": "/org/acme",
  "created": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "createdBy": "userY",
  "lastModified": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "modifiedBy": "userX",
  "lastUpdated": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "children": [
    {
      "uri" : "/child1",
      "folder" : "true"
    }, {
      "uri" : "/child2",
      "folder" : "false"
    }
  ]
}
```

File Info

Description: File Info

For virtual use the virtual repository returns the resolved file. Supported by local, local-cached and virtual repositories.

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/storage/{repoKey}/{filePath}

Produces: application/vnd.org.jfrog.artifactory.storage.FileInfo+json

Sample Output:

```
GET /api/storage/libs-release-local/org/acme/lib/ver/lib-ver.pom
{
  "uri":
  "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
  /lib/ver/lib-ver.pom",
  "downloadUri":
  "http://localhost:8081/artifactory/libs-release-local/org/acme/lib/ver/lib
  -ver.pom",
  "repo": "libs-release-local",
  "path": "/org/acme/lib/ver/lib-ver.pom",
  "remoteUrl": "http://some-remote-repo/mvn/org/acme/lib/ver/lib-ver.pom",
  "created": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "createdBy": "userY",
  "lastModified": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "modifiedBy": "userX",
  "lastUpdated": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "size": "1024", //bytes
  "mimeType": "application/pom+xml",
  "checksums":
  {
    "md5" : string,
    "sha1" : string,
    "sha256" : string
  },
  "originalChecksums":{
    "md5" : string,
    "sha1" : string,
    "sha256" : string
  }
}
```

Get Storage Summary Info

Description: Returns storage summary information regarding binaries, file store and repositories.

Since: 4.2.0

Security: Requires a privileged user (Admin only)

Usage: GET /api/storageinfo

Produces: application/json

Sample Output:

```

GET /api/storageinfo
{
  "binariesSummary": {
    "binariesCount": "125,726",
    "binariesSize": "3.48 GB",
    "artifactsSize": "59.77 GB",
    "optimization": "5.82%",
    "itemsCount": "2,176,580",
    "artifactsCount": "2,084,408"
  },
  "fileStoreSummary": {
    "storageType": "filesystem",
    "storageDirectory": "/home/.../artifactory/devenv/.artifactory/data/filestore",
    "totalSpace": "204.28 GB",
    "usedSpace": "32.22 GB (15.77%)",
    "freeSpace": "172.06 GB (84.23%)"
  },
  "repositoriesSummaryList": [
    {
      "repoKey": "plugins-release",
      "repoType": "VIRTUAL",
      "foldersCount": 0,
      "filesCount": 0,
      "usedSpace": "0 bytes",
      "itemsCount": 0,
      "packageType": "Maven",
      "percentage": "0%"
    },
    {
      "repoKey": "repo",
      "repoType": "VIRTUAL",
      "foldersCount": 0,
      "filesCount": 0,
      "usedSpace": "0 bytes",
      "itemsCount": 0,
      "packageType": "Generic",
      "percentage": "0%"
    },
    ...
    {
      "repoKey": "TOTAL",
      "repoType": "NA",
      "foldersCount": 92172,
      "filesCount": 2084408,
      "usedSpace": "59.77 GB",
      "itemsCount": 2176580
    }
  ]
}

```

Item Last Modified

Description: Retrieve the last modified item at the given path. If the given path is a folder, the latest last modified item is searched for recursively. Supported by local and local-cached repositories.

Since: 2.2.5

Notes: Requires Artifactory Pro

Security: Requires a valid user with deploy permissions

Usage: GET /api/storage/{repoKey}/{item-path}?lastModified

Produces: application/vnd.org.jfrog.artifactory.storage.ItemLastModified+json

Sample Output:

```
GET /api/storage/libs-release-local/org/acme?lastModified
{
  "uri":
  "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
  /foo/1.0-SNAPSHOT/foo-1.0-SNAPSHOT.pom",
  "lastModified": ISO8601
}
```

File Statistics

Description: Item statistics record the number of times an item was downloaded, last download date and last downloader. Supported by local and local-cached repositories.

Since: 3.1.0

Security: Requires read privileges

Usage: GET /api/storage/{repoKey}/{item-path}?stats

Produces: application/vnd.org.jfrog.storage.StatsInfo+json

Sample Output:

```
GET /api/storage/libs-release-local/org/acme/foo/1.0/foo-1.0.jar?stats
{
  "uri":
  "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
  /foo/1.0/foo-1.0.jar",
  "lastDownloaded": Timestamp (ms),
  "downloadCount": 1337,
  "lastDownloadedBy": "user1"
}
```

Item Properties

Description: Item Properties. Optionally return only the properties requested.

Since: 2.2.1

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/storage/{repoKey}/{itemPath}?properties[=x[,y]]

Produces: application/vnd.org.jfrog.artifactory.storage.ItemProperties+json

Sample Output:

```
GET /api/storage/libs-release-local/org/acme?properties\[=x[,y]\]
{
  "uri":
  "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme"
  "properties":{
    "p1": ["v1", "v2", "v3"],
    "p2": ["v4", "v5", "v6"]
  }
}
```

Set Item Properties

Description: Attach properties to an item (file or folder). When a folder is used property attachment is recursive by default. In order to supply special characters (comma (,), backslash(\), pipe(|), equals(=)) as key/value you must add a backslash (\) before them. For example: `..?properties=a=1\=1` will attach key a with 1=1 as value.

To specify multiple properties, you can separate the items in one of the following ways:

- Use a semicolon - ; (recommended)
- Use the encoding for the pipe ("|") character - %7C
Alternatively, you may configure your NGINX to encode URLs so that if an unencoded pipe is used in the URL, NGINX will encode it to %7C. We recommend that you verify that this configuration does not break any other systems served by NGINX

Supported by local and local-cached repositories.

Notes: Requires Artifactory Pro

Since: 2.3.0

Security: Requires a privileged user (can be anonymous)

Usage: `PUT /api/storage/{repoKey}/{itemPath}?properties=p1=v1[,v2][|p2=v3][&recursive=1]`

Sample Usage:

```
PUT
/api/storage/libs-release-local/ch/qos/logback/logback-classic/0.9.9?properties=os=win,linux;qa=done&recursive=1
```

Delete Item Properties

Description: Deletes the specified properties from an item (file or folder). When a folder is used property removal is recursive by default. Supported by local and local-cached repositories.

Notes: Requires Artifactory Pro

Since: 2.3.2

Security: Requires a privileged user (can be anonymous)

Usage: `DELETE /api/storage/{repoKey}/{itemPath}?properties=p1[,p2][&recursive=1]`

Sample Usage:

```
DELETE
/api/storage/libs-release-local/ch/qos/logback/logback-classic/0.9.9?properties=os,qa&recursive=1
```

Set Item SHA256 Checksum

Description: Calculates an artifact's SHA256 checksum and attaches it as a property (with key "sha256"). If the artifact is a folder, then recursively calculates the SHA256 of each item in the folder and attaches the property to each item.

Since: 4.2.1

Security: Requires an admin user

Consumes: application/json

Usage: `POST /api/checksum/sha256 -H "Content-Type: application/json"`

Sample Usage:

```
POST /api/checksum/sha256 -H "Content-Type: application/json"
{
  "repoKey": "ext-snapshot-local",
  "path": "artifactory-powerpack-3.9.3/bin/"
}
```

Retrieve Artifact

Description: Retrieves an artifact from the specified destination.

You can also use [Property-based Resolution](#) as part of retrieving artifacts.

Security: Requires a user with 'read' permission (can be anonymous)

Usage: GET /repo-key/path/to/artifact.ext

Sample Usage:

```
GET
http://localhost:8081/artifactory/libs-release-local/ch/qos/logback/logback-classic/0.9.9/logback-classic-0.9.9.jar
```

Retrieve Latest Artifact

Description: Retrieves the latest artifact version from the specified destination.

Latest Maven Release/Integration: Specify `SNAPSHOT` or `[RELEASE]` for the version in the requested path to get the latest Maven integration or release artifact.

Latest Non-Maven Release/Integration: Specify `[INTEGRATION]` or `[RELEASE]` for the version in the requested path (replacing the `[folderItegRev]` and `[fileItegRev]` as defined by the repository's [layout](#)) to get the latest integration version or latest release version artifact accordingly based on alphabetical sorting.

Integration and release tokens cannot be mixed together.

You can also use [property-based resolution](#) as part of retrieving artifacts to restrict resolution of artifacts assigned with specific properties.

NOTE:

1. Only local, cache and virtual repositories will be used.
2. To change the retrieve latest behavior to retrieve the latest version based on the created date you can add the following flag to `$ARTIFACTORY_HOME/etc/artifactory.system.properties` and add the following flag ***artifactory.request.searchLatestReleaseByDateCreated=true*** and restart Artifactory service.

Notes: Requires Artifactory Pro.

Since: Latest Maven: 2.6.0; Latest non-Maven: 2.6.2

Security: Requires a user with 'read' permission (can be anonymous)

Usage: GET /repo-key/path/to/artifact.ext

Sample Usage:

Download the latest Maven unique snapshot artifact:

```
GET
http://localhost:8081/artifactory/libs-release-local/ch/qos/logback/logback-classic/0.9.9-SNAPSHOT/logback-classic-0.9.9-SNAPSHOT.jar
```

Download the latest release artifact:

```
GET
http://localhost:8081/artifactory/ivy-local/org/acme/[RELEASE]/acme-[RELEASE].jar
```

Download the latest integration artifact:

```
GET
http://localhost:8081/artifactory/ivy-local/org/acme/1.0-[INTEGRATION]/acme-1.0-[INTEGRATION].jar
```


Retrieve Build Artifacts Archive

Description: Retrieves an archive file (supports zip/tar/tar.gz/tgz) containing all the artifacts related to a specific build, you can optionally provide mappings to filter the results,

the mappings support [regex capturing groups](#) which enables you to dynamically construct the target path inside the result archive file.

Notes: Requires Artifactory Pro

Since: 2.6.5

Security: Requires a privileged user (can be anonymous)

Usage: POST /api/archive/buildArtifacts -H "Content-Type: application/json"

Consumes: application/vnd.org.jfrog.artifactory.build.BuildArtifactsRequest+json

Produces: application/zip (for zip archive type), application/x-tar (for tar archive type), application/x-gzip (for [tar.gz/tgz](#) archive type)

Sample Usage:

```
POST /api/archive/buildArtifacts -H "Content-Type: application/json"
{
  +"buildName": "build-name" // The build name for search by
  +"buildNumber": "15" // The build number to search by, can be LATEST to
search for the latest build number
  -"buildStatus": "Released" // Optionally search by latest build status
(e.g: "Released")
  -"repos": ["libs-release-local,ext-release-local"] // Optionally refine
search for specific repos, omit to search within all repositories
  +"archiveType": "tar/zip/tar.gz/tgz" // The archive file type to return
  -"mappings": [ // Optionally refine the search by providing a list of
regex patterns to search by
    {
      "input": "(.+)/(.)-sources.jar",
      "output": "$1/sources/$2.jar" // Optionally provide different path of the
found artifacts inside the result archive, supports regex groups tokens
    },
    {
      "input": "(.)-release.zip"
    }
  ]
}
```

Retrieve Folder or Repository Archive

Description: Retrieves an archive file (supports zip/tar/tar.gz/tgz) containing all the artifacts that reside under the specified path (folder or repository root). Requires [Enable Folder Download](#) to be set.

Notes: Requires Artifactory Pro

Since: 4.1.0

Security: Requires a privileged user with read permissions on the path.

Usage: GET /api/archive/download/{repoKey}/{path}?archiveType={archiveType}&includeChecksumFiles=true

Produces: */*

Sample Usage:

```
GET /api/archive/download/my-local-repo/path/to/folder?archiveType=zip
{Stream containing contents of path my-local-repo/path/to/folder}

GET /api/archive/download/my-local-repo?archiveType=zip
{Stream containing contents of repo my-local-repo}
```

Trace Artifact Retrieval

Description: Simulates an artifact retrieval request from the specified location and returns verbose output about the resolution process. This API is useful for debugging artifact retrieval issues.

Security: As applied to standard artifact retrieval by the requesting user.

Since: 2.6.0

Usage: GET /repo-key/path/to/artifact.ext?trace

Produces: text/plain

Sample Output:

```
GET
http://localhost:8081/artifactory/libs-release-local/jmock/jmock/1.0.1/jmock-1.0.1.jar?trace

Request ID: 51c808f6
Repo Path ID: libs-release-local:jmock/jmock/1.0.1/jmock-1.0.1.jar
Method Name: GET
User: resolver
Time: 2012-05-06T10:49:09.528+03:00
Thread: pool-1-thread-31
Steps:
2012-05-06T10:49:09.587+03:00 Received request
2012-05-06T10:49:09.588+03:00 Request source = 0:0:0:0:0:0:0:1, Last
modified = 01-01-70 01:59:59 IST, If modified since = -1, Thread name =
pool-1-thread-31
2012-05-06T10:49:09.697+03:00 Retrieving info
2012-05-06T10:49:09.723+03:00 Identified resource as a file
...
2012-05-06T10:49:09.788+03:00 Responding with selected content handle
2012-05-06T10:49:09.807+03:00 Request succeeded
```

Archive Entry Download

Description: Retrieves an archived resource from the specified archive destination.

Security: Requires a user with 'read' permission (can be anonymous)

Usage: GET /repo-key/path/to/artifact.jar!*/path/to/archived/resource (**NOTE!** the '!' between the archive file name and the archive entry path)

Sample Output:

```
GET
http://localhost:8081/artifactory/repo1-cache/commons-lang/commons-lang/2.6/commons-lang-2.6.jar!/META-INF/LICENSE.txt
```

Create Directory

Description: Create new directory at the specified destination.

Notes: You can also [attach properties](#) as part of creating directories.

Security: Requires a user with 'deploy' permissions (can be anonymous)

Usage: PUT /repo-key/path/to/directory/

Produces: application/vnd.org.jfrog.artifactory.storage.ItemCreated+json

Sample Usage:

```
PUT /libs-release-local/path/to/directory/
{
  "uri":
  "http://localhost:8081/artifactory/libs-release-local/path/to/directory",
  "repo": "libs-release-local",
  "path": "/path/to/directory",
  "created": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "createdBy": "userY",
  "children" : [ ]
}
```

Deploy Artifact

Description: Deploy an artifact to the specified destination.

Notes: You can also [attach properties](#) as part of deploying artifacts.

Security: Requires a user with 'deploy' permissions (can be anonymous)

Usage: PUT /repo-key/path/to/artifact.ext

Produces: application/vnd.org.jfrog.artifactory.storage.ItemCreated+json

Sample Usage:

```
PUT /libs-release-local/my/jar/1.0/jar-1.0.jar
{
  "uri":
  "http://localhost:8081/artifactory/libs-release-local/my/jar/1.0/jar-1.0.jar",
  "downloadUri":
  "http://localhost:8081/artifactory/libs-release-local/my/jar/1.0/jar-1.0.jar",
  "repo": "libs-release-local",
  "path": "/my/jar/1.0/jar-1.0.jar",
  "created": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "createdBy": "userY",
  "size": "1024", //bytes
  "mimeType": "application/java-archive",
  "checksums":
  {
    "md5" : string,
    "sha1" : string
  },
  "originalChecksums":{
    "md5" : string,
    "sha1" : string
  }
}
```

Deploy Artifact by Checksum

Description: Deploy an artifact to the specified destination by checking if the artifact content already exists in Artifactory.

If Artifactory already contains a user readable artifact with the same checksum the artifact content is copied over to the new location and return a

response without requiring content transfer.

Otherwise, a 404 error is returned to indicate that content upload is expected in order to deploy the artifact.

Notes: You can also [attach properties](#) as part of deploying artifacts.

Security: Requires a user with 'deploy' permissions (can be anonymous)

Usage: PUT /repo-key/path/to/artifact.ext

Headers: X-Checksum-Deploy: true, X-Checksum-Sha1: sha1Value, X-Checksum-Sha256: sha256Value

Produces: application/vnd.org.jfrog.artifactory.storage.ItemCreated+json

Since: 2.5.1

Sample Output:

```
PUT /libs-release-local/my/jar/1.0/jar-1.0.jar
{
  "uri":
  "http://localhost:8081/artifactory/libs-release-local/my/jar/1.0/jar-1.0.j
ar",
  "downloadUri":
  "http://localhost:8081/artifactory/libs-release-local/my/jar/1.0/jar-1.0.j
ar",
  "repo": "libs-release-local",
  "path": "/my/jar/1.0/jar-1.0.jar",
  "created": ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ),
  "createdBy": "userY",
  "size": "1024", //bytes
  "mimeType": "application/java-archive",
  "checksums":
  {
    "md5" : string,
    "sha1" : string
  },
  "originalChecksums":{
    "md5" : string,
    "sha1" : string
  }
}
```

Deploy Artifacts from Archive

Description: Deploys an archive containing multiple artifacts and extracts it at the specified destination maintaining the archive's file structure. Deployment is performed in a single HTTP request and only the extracted content is deployed, not the archive file itself.

Supported archive types are: zip; tar; tar.gz; and tgz. **NOTE!** that deployment of compressed archives (unlike tar) may incur considerable CPU overhead.

Notes: Requires Artifactory Pro

Security: Requires a user with 'deploy' permissions (can be anonymous)

Usage: PUT path1/to/repo-key/ /path2/to/archive.zip

Headers: *X-Explode-Archive: true* - archive will be exploded upon deployment, *X-Explode-Archive-Atomic: true* - archive will be exploded in an atomic operation upon deployment

Produces: text/plain

Since: 2.6.3

Sample Usage:

```
PUT /libs-release-local/ /Users/user/Desktop/archive.zip
```

Push a Set of Artifacts to Bintray

Deprecated: This endpoint is deprecated and is replaced with [Distribute Artifact](#)

Description: Push a set of artifacts to Bintray as a version.

Uses a descriptor file (that must have 'bintray-info' in it's filename and a .json extension) that was deployed to artifactory, the call accepts the full path to the descriptor as a parameter.

For more details, please refer to [Pushing a Set of Files](#).

Signing a version is controlled by the *gpgSign* parameter in the descriptor file, and the *gpgSign* parameter passed to this command. **The value passed to this command always takes precedence over the value in the descriptor file.**

If you also want a passphrase to be applied to your signature, specify *gpgPassphrase=<passphrase>*.

Security: Requires a valid user with deploy permissions and Bintray credentials defined (for more details, please refer to [Entering your Bintray credentials](#)).

Usage: POST /api/bintray/push?descriptor=pathToDescriptorFile[&gpgPassphrase=passphrase][&gpgSign=true/false]

Since: 3.5.0

Produces: application/vnd.org.jfrog.artifactory.bintray.BintrayPushResponse+json

Sample Output:

```
{"Message": "Pushing build to Bintray finished successfully."}
```

Push Docker Tag to Bintray

Description: Push Docker tag to Bintray

Calculation can be synchronous (the default) or asynchronous.

Security: Requires a valid user with deploy permissions and Bintray credentials defined (for more details, please refer to [Entering your Bintray credentials](#)).

Usage: POST /api/bintray/docker/push/{repoKey}

Since: 3.6.0

Produces: text/plain

Sample Output:

```
POST api/bintray/docker/push/docker-local
{
  "dockerImage": "jfrog/ubuntu:latest", // The docker image to push, use
  ':' for specific tag or leave blank for 'latest'
  "bintraySubject": "shayy", // The Bintray Subject
  "bintrayRepo": "containers", // The Bintray Subject's repository
  "async": false // Optionally execute the push asynchronously. Default:
false
}
```

Distribute Artifact

Description: Deploys artifacts from Artifactory to Bintray, and creates an entry in the corresponding Artifactory distribution repository specified

Notes: Requires Artifactory Pro

Since: 4.8

Security: Requires an authenticated user.

Usage: POST /api/distribute

Consumes: application/json

```
{
  "publish" : "<true | false>"      // Default: true. If true, artifacts are
published when deployed to Bintray
  "overrideExistingFiles" : "<true | false>" // Default: false. If true,
Artifactory overwrites files already existing in the target path in
Bintray.
      // Existing version attributes are also overridden if defined
in the distribution repository Advanced Configuration
  "gpgPassphrase" : "<passphrase>"    // If specified, Artifactory will GPG
sign the version deployed to Bintray and apply the specified passphrase
  "async" : "<true | false>"          // Default: false. If true, the artifact
will be distributed asynchronously. Errors and warnings may be viewed in
the log file
  "targetRepo" : "<targetDistributionRepo>", // The Distribution Repository
into which artifacts should be deployed
  "packagesRepoPaths" : ["<localRepo/path/to/distribute>",
"<distRepo/path/to/distribute>"] // An array of local or distribution
repositories and corresponding paths to artifacts that should be deployed
to the specified target repository in Bintray
  "dryRun" : "<true | false>"        // Default: false. If true, distribution
is only simulated. No files are actually moved.
}
```

Sample input:

```
POST /api/distribute
{
  "targetRepo" : "dist-repo-jfrog-artifactory",
  "packagesRepoPaths" : ["yum-local/jfrog-artifactory-pro-4.7.6.rpm"]
}
```

File Compliance Info

Description: Get compliance info for a given artifact path. The result includes license and vulnerabilities, if any. Supported by local and local-cached repositories.

Notes: Requires Artifactory Pro, requires Black Duck addon enabled.

Since: 3.0.0

Security: Requires an authenticated user.

Usage: GET: /api/compliance/{repoKey}/{item-path}

Produces: application/json

Sample output:

```
GET:
/api/compliance/libs-release-local/ch/qos/logback/logback-classic/0.9.9/logback-classic-0.9.9.jar
{
  "licenses" : [ { "name": "LGPL v3", "url": "http://" }, { "name": "APL v2", "url": "http://" } ... ],
  "vulnerabilities" : [ { "name": "CVE-13427", "url": "http://" }, { "name": "CVE-1041", "url": "http://" } ... ]
}
```

Delete Item

Description: Deletes a file or a folder from the specified destination.

Security: Requires a user with 'delete' permission (can be anonymous)

Usage: DELETE /repo-key/path/to/file-or-folder

Sample Usage:

```
DELETE
http://localhost:8081/artifactory/libs-release-local/ch/qos/logback/logback-classic/0.9.9
```

Copy Item

Description: Copy an artifact or a folder to the specified destination. Supported by local repositories only.

Optionally suppress cross-layout module path translation during copy.

You can test the copy using a dry run.

Copy item behaves similarly to a standard file system and supports renames. If the target path does not exist, the source item is copied and optionally renamed. Otherwise, if the target exists and it is a directory, the source is copied and placed under the target directory.

Notes: Requires Artifactory Pro

Security: Requires a privileged user (can be anonymous)

Usage: POST /api/copy/{srcRepoKey}/{srcFilePath}?to={targetRepoKey}/{targetFilePath}&dry=1[&suppressLayouts=0/1(default)][&failFast=0/1]

Produces: application/vnd.org.jfrog.artifactory.storage.CopyOrMoveResult+json

Since: 2.2.2

Sample Output:

```
POST
/api/copy/libs-release-local/org/acme?to=/ext-releases-local/org/acme-new&dry=1
{
  "messages" : [
    {
      "level": "error",
      "message": "The repository has denied...."
    }, ...
  ]
}
```

Move Item

Description: Moves an artifact or a folder to the specified destination. Supported by local repositories only.

Optionally suppress cross-layout module path translation during move.

You can test the move using dry run.

Move item behaves similarly to a standard file system and supports renames. If the target path does not exist, the source item is moved and optionally renamed. Otherwise, if the target exists and it is a directory, the source is moved and placed under the target directory.

Notes: Requires Artifactory Pro

Security: Requires a privileged user (can be anonymous)

Usage: POST /api/move/{srcRepoKey}/{srcFilePath}?to={targetRepoKey}/{targetFilePath}&dry=1[&suppressLayouts=0/1(default)][&failFast=0/1]

Produces: application/vnd.org.jfrog.artifactory.storage.CopyOrMoveResult+json

Since: 2.2.2

Sample Output:

```
POST
/api/move/libs-release-local/org/acme?to=/ext-releases-local/org/acme-new&
dry=1
{
  "messages" : [
    {
      "level": "error",
      "message": "The repository has denied...."
    },...
  ]
}
```

Get Repository Replication Configuration

Description: Returns the replication configuration for the given repository key, if found. Supported by local and remote repositories.

Notes: Requires Artifactory Pro

Security: Requires a privileged user

Usage: GET /api/replications/{repoKey}

Produces: application/vnd.org.jfrog.artifactory.replications.ReplicationConfigRequest+json

Since: 3.1.1

Sample Usage:

```
GET /api/replications/libs-release-local
{
  "url" : "http://localhost:8081/artifactory/remote-repo",
  "socketTimeoutMillis" : 15000,
  "username" : "admin",
  "password" : "password",
  "enableEventReplication" : false,
  "enabled" : true,
  "cronExp" : "0 0 12 * * ?",
  "syncDeletes" : true,
  "syncProperties" : true,
  "syncStatistics" : false,
  "repoKey" : "libs-release-local",
  "pathPrefix" : "/path/to/repo"
}
```


Set Repository Replication Configuration

Description: Add or replace replication configuration for given repository key. Supported by local and remote repositories. Accepts the JSON payload returned from [Get Repository Replication Configuration](#) for a single and an array of configurations. If the payload is an array of replication configurations, then values for `cronExp` and `enableEventReplication` in the first element in the array will determine the corresponding values when setting the repository replication configuration.

Notes: Requires Artifactory Pro

Security: Requires a privileged user

Usage: PUT /api/replications/{repoKey}

Consumes: application/vnd.org.jfrog.artifactory.replications.ReplicationConfigRequest+json

Since: 3.1.1

Sample Usage:

```
PUT /api/replications/libs-release-local
```

Update Repository Replication Configuration

Description: Update existing replication configuration for given repository key, if found. Supported by local and remote repositories.

Notes: Requires Artifactory Pro

Security: Requires a privileged user

Usage: POST /api/replications/{repoKey}

Consumes: full or partial application/vnd.org.jfrog.artifactory.replications.ReplicationConfigRequest+json

Since: 3.1.1

Sample Usage:

```
POST /api/replications/libs-release-local
```

Delete Repository Replication Configuration

Description: Delete existing replication configuration for given repository key. Supported by local and local-cached repositories.

Notes: Requires Artifactory Pro

Security: Requires a privileged user

Usage: DELETE /api/replications/{repoKey}

Since: 3.1.1

Sample Usage:

```
DELETE /api/replications/libs-release-local
```

Scheduled Replication Status

Description: Returns the status of scheduled [cron-based](#) replication jobs define via the Artifactory UI on repositories. Supported by local, local-cached and remote repositories.

Notes: Requires Artifactory Pro

Security: Requires a user with 'read' permission (can be anonymous)

Usage: GET /api/replication/{repoKey}

Produces: application/vnd.org.jfrog.artifactory.replication.ReplicationStatus+json

```
GET /api/replication/remote-libs
{
  "status": {status}
  "lastCompleted": {time}
  [
    { "repository Name" : "target-repol", "status" : {status},
    "lastCompleted" : {time} },
    ...
    { "repository Name" : "target-repon", "status" : {status},
    "lastCompleted" : {time} }
  ]
}
where:
{status} = never_run|incomplete(running or
interrupted)|error|warn|ok|inconsistent
{time} = time in ISO8601 format (yyyy-MM-dd'T'HH:mm:ss.SSSZ), or null if
never completed
```

Since: 2.4.2

Sample Usage:

```
GET /api/replication/remote-libs
{
  "status" : "ok",
  "lastCompleted" : 2015-12-27T15:08:49.050+02:00"
  "repositories" :
  [
    { "repository Name" : "remote-libs1", "status" : "ok", "lastCompleted" :
    "2015-12-27T15:07:49.050+02:00" },
    { "repository Name" : "remote-libs2", "status" : "ok", "lastCompleted" :
    "2015-12-27T15:08:49.050+02:00" }
  ]
}
```

Pull/Push Replication

Description: Schedules immediate content replication between two Artifactory instances.

Replication can optionally include properties and delete items if they do not exist in the source repository.

This API completes the existing [cron-based](#) replication exposed via the Artifactory UI and allows for pre-scheduled execution.

Pull Replication - pulls content from a remote Artifactory repository to a local cache of the remote repository.

Push Replication - pushes content from a local repository into a local repository of another Artifactory instance.

Multi-push Replication - pushes content from a local repository into a local repository of several Artifactory instances. This feature is only available with Artifactory Enterprise license.

The type of replication initiated depends on the type of repository specified in the `repoPath` parameter.

If `repoPath` is a local repository, a push replication will be triggered. You may specify multiple target repositories in the payload for multi-push replication, but all must be local to their respective instances.

If `repoPath` is a remote repository cache, a pull replication will be triggered. Note that in this case you may only specify a single repository in the payload.

Important note - If no repositories are provided in the payload, Artifactory will trigger all existing replication configurations.

Security: Requires a privileged user (can be anonymous) For non-admin users, the maximum number of files that will be replicated is as defined by the `artifactory.search.userQueryLimit` system property.

Usage: POST /api/replication/execute/{repoPath}

Consumes: application/json

```
[
  {
+   "url" : "<URL of the repository at the remote Artifactory instance>",
+   "username" : "<username at the remote Artifactory instance>",
+   "password" : "<password at the remote Artifactory instance>",
-   "proxy" : "<name of the proxy (if used) at the remote Artifactory instance>"
-   "properties" : "<true | false>", // When true, properties of replicated artifacts will be synchronized also
-   "delete" : "<true | false>" // When true, items that were deleted remotely will also be deleted locally (including properties metadata)
  }
]
```

+=mandatory; **-=**optional

Since: 4.7.5

Sample Usage:

```
// Single push replication
POST /api/replication/execute/libs-release-local
{
  [
    {
      "url":"http://localhost:8082/artifactory/libs-release-local",
      "username":"admin",
      "password":"password",
      "proxy":"localProxy"
    }
  ]
}

// Pull replication
POST /api/replication/execute/libs-remote
{
  [
    {
      "url":"http://localhost:8082/artifactory/libs-remote",
      "username":"admin",
      "password":"password",
      "properties" : "true",
      "delete" : "true"
    }
  ]
}

// Multi-push replication
POST /api/replication/execute/libs-release-local
{
```

```
[
  {
    "url": "http://localhost:8082/artifactory/libs-release-local",
    "username": "admin",
    "password": "password",
    "proxy": "localProxy",
    "properties" : "true",
    "delete" : "true"
  },
  {
    "url": "http://localhost:8082/artifactory/ext-release-local",
    "username": "admin",
    "password": "password",
    "properties" : "true",
    "delete" : "true"
  },
  {
    "url": "http://localhost:8082/artifactory/plugins-release-local",
    "username": "admin",
    "password": "password",
    "properties" : "true",
    "delete" : "true"
  }
]
```

```
// Trigger configured push replication
POST /api/replication/execute/libs-release-local
```

```
// Trigger configured pull replication
POST /api/replication/execute/libs-remote
```

Pull/Push Replication (Deprecated)

Description: Schedules immediate content replication between two Artifactory instances. Replication can include properties and can optionally delete local items if they do not exist in the source repository.

This API completes the existing [cron-based](#) replication exposed via the Artifactory UI and allows for on-demand execution.

Pull Replication - pulls content from a remote Artifactory repository to a local cache of the remote repository.

Push Replication - pushes content from a local repository into a remote Artifactory local repository.

Supported by local, local-cached and remote repositories.

Notes: Requires Artifactory Pro

Security: Requires a privileged user (can be anonymous) For non-admin users will replicate at max the number of files as defined by the `artifactory.search.userQueryLimit` system property.

Usage: POST /api/replication/{srcRepoKey}/{srcPath}

Consumes: application/vnd.org.jfrog.artifactory.replication.ReplicationRequest+json

Since: 2.4.0

Sample Usage:

```
POST /api/replication/libs-release-local/com/acme
{
  //The following is only applicable for push replication
  + "url" : "https://repo.nmiy.org/repo-key", // The remote repository URL
  + "username": "replicator", //The name of a user with deploy permissions
  on the remote repository
  + "password": "****", //The remote repository password
  - "properties": true, //Sync item properties (true by default)
  - "delete": true, //Sync deletions (false by default)
  - "proxy": "org-prox", //A name of an Artifactory-configured proxy to use
  for remote requests
}
```

`+=mandatory; -=optional`

Create or Replace Local Multi-push Replication

Description: Creates or replaces a local multi-push replication configuration. Supported by local and local-cached repositories.

Notes: Requires an enterprise license

Security: Requires an admin user.

Usage: PUT /api/replications/multiple/{repo-key}

Consumes: application/vnd.org.jfrog.artifactory.replications.MultipleReplicationConfigRequest+json

Since: 3.7

Sample Usage:

```

PUT /api/replications/multiple/libs-release-local
{
  "cronExp": "0 0/9 14 * * ?",
  "enableEventReplication": true,
  "replications": [
    {
      + "url": "http://localhost:8081/artifactory/repo-k",
      + "socketTimeoutMillis": 15000,
      + "username": "admin",
      + "password": "password",
      - "enableEventReplication": true,
      - "enabled": true,
      - "syncDeletes": false,
      - "syncProperties": true,
      - "syncStatistics" : false,
      - "repoKey": "libs-release-local"
    }
  ,
    {
      + "url": "http://localhost:8081/artifactory/repo-v",
      + "socketTimeoutMillis": 15000,
      + "username": "admin",
      + "password": "password",
      - "enableEventReplication": true,
      - "enabled": true,
      - "syncDeletes": false,
      - "syncProperties": true,
      - "syncStatistics" : false,
      - "repoKey": "libs-release-local"
    }
  ]
}

```

+ =mandatory; -=optional

Update Local Multi-push Replication

Description: Updates a local multi-push replication configuration. Supported by local and local-cached repositories.

Notes: Requires an enterprise license

Security: Requires an admin user.

Usage: POST /api/replications/multiple/{repo-key}

Consumes: application/vnd.org.jfrog.artifactory.replications.MultipleReplicationConfigRequest+json

Since: 3.7

Sample Usage:

```

POST /api/replications/multiple/libs-release-local
{
  "cronExp": "0 0/9 14 * * ?",
  "enableEventReplication": true,
  "replications": [
    {
+   "url": "http://localhost:8081/artifactory/repo-k",
+   "socketTimeoutMillis": 15000,
+   "username": "admin",
+   "password": "password",
-   "enableEventReplication": true,
-   "enabled": true,
-   "syncDeletes": false,
-   "syncProperties": true,
-   "syncStatistics": false,
-   "repoKey": "libs-release-local"
    }
  ,
    {
+   "url": "http://localhost:8081/artifactory/repo-v",
+   "socketTimeoutMillis": 15000,
+   "username": "admin",
+   "password": "password",
-   "enableEventReplication": true,
-   "enabled": true,
-   "syncDeletes": false,
-   "syncProperties": true,
-   "syncStatistics": false,
-   "repoKey": "libs-release-local"
    }
  ]
}

```

+ =mandatory; -=optional

Delete Local Multi-push Replication

Description: Deletes a local multi-push replication configuration. Supported by local and local-cached repositories.

Notes: Requires an enterprise license

Security: Requires an admin user.

Usage: DELETE /api/replications/{repoKey}?url={replicatedURL}

If the url parameter is omitted, all multi-push replication configurations for the source repository are deleted.

Produces: application/vnd.org.jfrog.artifactory.replications.ReplicationConfigRequest+json, application/vnd.org.jfrog.artifactory.replications.MultipleReplicationConfigRequest+json

Since: 3.7

Sample Usage:

```
DELETE
/api/replications/libs-release-local?url=http://10.0.0.1/artifactory/libs-
release-local

//Delete all multi-push replication configurations for libs-release-local
DELETE /api/replications/libs-release-local
```

Enable or Disable Multiple Replications

Description: Enables/disables multiple replication tasks by repository or Artifactory server based in include and exclude patterns.

Notes: Requires Artifactory Pro

Security: Requires a privileged user

Usage: POST /api/replications/{enable | disable}

Consumes: application/json

Since: 4.4.3

Sample Usage:

```
//Enable/disable all push replications except those going out to
http://artimaster:port/artifactory and
https://somearti:port/artifactory/local-repo.
POST /api/replications/{enable | disable}
{
  "include" : [ "*" ],
  "exclude" : [ "http://artimaster:port/artifactory/**",
    "https://somearti:port/artifactory/local-repo" ]
}

//Enable/disable all push replications except those going out to
http://artidr:port/artifactory
POST /api/replications/{enable | disable}
{
  "include" : [ "*" ],
  "exclude" : [ "http://artidr:port/artifactory/**" ]
}
```

Get Global System Replication Configuration

Description: Returns the global system replication configuration status, i.e. if push and pull replications are blocked or unblocked.

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/system/replications

Produces: application/json

Since: 4.7.2

Sample Usage:


```
GET /api/system/replications
{
  "blockPullReplications": false,
  "blockPushReplications": false
}
```

Block System Replication

Description: Blocks replications globally. Push and pull are true by default. If false, replication for the corresponding type is not blocked.

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: POST api/system/replications/block?push=[true|false]&pull=[true|false]

Produces: text/plain

Since: 4.7.2

Sample Usage:

```
POST /api/system/replications/block
Successfully blocked all replications, no replication will be triggered.
```

Unblock System Replication

Description: Unblocks replications globally. Push and pull are true by default. If false, replication for the corresponding type is not unblocked.

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: POST api/system/replications/unblock?push=[true|false]&pull=[true|false]

Produces: text/plain

Since: 4.7.2

Sample Usage:

```
POST /api/system/replications/unblock
Successfully unblocked all replications
```

Artifact Sync Download

Description: Downloads an artifact with or without returning the actual content to the client. When tracking the progress marks are printed (by default every 1024 bytes). This is extremely useful if you want to trigger downloads on a remote Artifactory server, for example to force eager cache population of large artifacts, but want to avoid the bandwidth consumption involved in transferring the artifacts to the triggering client. If no content parameter is specified the file content is downloaded to the client.

Notes: This API requires Artifactory Pro.

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/download/{repoKey}/{filePath}[?content=none/progress]&mark=numOfBytesToPrintANewProgressMark]

Produces: application/octet-stream, text/plain (depending on content type)

Since: 2.2.2

Sample Output:

```
GET
/api/download/my-remote/org/acme/1.0/acme-1.0.jar?content=progress&mark=512
.....
.....
.....

Completed: 150/340 bytes
```

Folder Sync (Deprecated)

Description: Triggers a no-content download of artifacts from a remote Artifactory repository for all artifacts under the specified remote folder.

Can optionally delete local files if they do not exist in the remote folder, overwrite local files only if they are older than remote files or never overwrite local files.

The default is not to delete any local files and to overwrite older local files with remote ones. By default progress marks of the sync are displayed.

The default timeout for the remote file list is 15000 milliseconds (15 seconds).

Notes: This API is **deprecated**. Requires Artifactory Pro

Security: Requires a privileged user (can be anonymous) For non-admin users will replicate at max the number of files as defined by the `artifactory.search.userQueryLimit` system property.

Usage: GET

`/api/sync/{remoteRepositoryKey}/{folderPath}[?progress=showProgress][&mark=numOfBytesToPrintANewProgressMark][&delete=deleteExistingFiles][&overwrite=never/force][&timeout=fileListTimeoutInMillis]`

Produces: text/plain

Since: 2.2.4

Sample Output:

```
GET /api/sync/my-remote/org/acme/1.0?progress=1&delete=1
.....
.....
.....
.....

Completed: 970/1702 bytes
.....
.....
Completed: 1702/1702 bytes
Completed with 0 errors and 2 warnings (please check the server log for
more details).
```

File List

Description: Get a flat (the default) or deep listing of the files and folders (not included by default) within a folder.

For deep listing you can specify an optional depth to limit the results.

Optionally include a map of metadata timestamp values as part of the result (only properties are displayed in since 3.0.0).

folder inclusion since 2.3.2; checksum inclusion since: 2.3.3; include folder root path since: 2.5.2. Supported by all types of repositories.

Since: 2.2.4

Notes: Requires Artifactory Pro

Security: Requires a non-anonymous privileged user.

Usage: GET `/api/storage/{repoKey}/{folder-path}?list[&deep=0/1][&depth=n][&listFolders=0/1][&mdTimestamps=0/1][&includeRootPath=0/1]`

Produces: application/vnd.org.jfrog.artifactory.storage.FileList+json

Sample Output:

```

GET
/api/storage/libs-release-local/org/acme?list&deep=1&listFolders=1&mdTimes
tamps=1
{
  "uri":
"http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
",
  "created": ISO8601,
  "files" : [
    {
      "uri": "/archived",
      "size": "-1",
      "lastModified": ISO8601,
      "folder": "true"
    },
    {
      "uri": "/doc.txt",
      "size": "253207", //bytes
      "lastModified": ISO8601,
      "folder": "false",
      "sha1": sha1Checksum,
      "mdTimestamps": { "properties" : lastModified (ISO8601) }
    },
    {
      "uri": "/archived/doc1.txt",
      "size": "253100", //bytes
      "lastModified": ISO8601,
      "folder": "false",
      "sha1": sha1Checksum,
      "mdTimestamps": { "properties" : lastModified (ISO8601) }
    },...
  ]
}

```

Get Background Tasks

Description: Retrieves list of background tasks currently scheduled or running in Artifactory. In HA, the nodeId is added to each task. Task can be in one of few states: scheduled, running, stopped, canceled. Running task also shows the task start time.

Since: 4.4.0

Security: Requires a valid admin user

Usage: GET /api/tasks

Produces: application/json

Sample Output:

```
{
  "tasks" : [ {
    "id" :
    "artifactory.UpdateIndicesJob#d7321feb-6fd9-4e27-8f0e-954137be855b",
    "type" :
    "org.artifactory.addon.gems.index.GemsVirtualIndexHandler$UpdateIndicesJob",
    "state" : "scheduled",
    "description" : "Gems Virtual Repositories Index Calculator",
    "nodeId" : "artifactory-primary"
  }, {
    "id" :
    "artifactory.VirtualCacheCleanupJob#82bb1514-ea34-4a71-940d-78a61887981e",
    "type" : "org.artifactory.repo.cleanup.VirtualCacheCleanupJob",
    "state" : "scheduled",
    "description" : "",
    "nodeId" : "artifactory-primary"
  }, {
    "id" :
    "artifactory.BinaryStoreGarbageCollectorJob#039664ac-990d-4a32-85e1-decd0b508142",
    "type" :
    "org.artifactory.storage.binstore.service.BinaryStoreGarbageCollectorJob",
    "state" : "running",
    "started" : "2015-05-15T15:39:37.566+02:00",
    "description" : "Binaries Garbage Collector",
    "nodeId" : "artifactory-primary"
  } ]
}
```

Empty Trash Can

Description: Empties the trash can permanently deleting all its current contents.

Notes: Requires Artifactory Pro

Security: Requires a valid admin user

Usage: POST /api/trash/empty

Since: 4.4.3

Delete Item From Trash Can

Description: Permanently deletes an item from the trash can.

Notes: Requires Artifactory Pro

Security: Requires a valid admin user

Usage: DELETE /api/trash/clean/{repoName/path}

Since: 4.4.3

Sample usage:

```
DELETE /api/trash/clean/npm-local
```

Restore Item from Trash Can

Description: Restore an item from the trash can.

Notes: Requires Artifactory Pro

Security: Requires a valid admin user

Usage: POST /api/trash/restore/{from path}?to={to path}

Since: 4.4.3

Sample usage:

```
POST /api/trash/restore/npm-local?to=npm-local2
```

```
Successfully restored trash items
```

Optimize System Storage

Description: Raises a flag to invoke balancing between redundant storage units of a sharded filestore following the next garbage collection.

Since: 4.6.0

Notes: This is an advanced feature intended for administrators.

Security: Requires a valid admin user.

Usage: POST /api/system/storage/optimize

Produces: text/plain

Sample Usage:

```
POST /api/system/storage/optimize
```

```
200 OK
```

SEARCHES

All searches return limited results for anonymous users (same limits as in the user interface).

Artifactory Query Language (AQL)

Description: Flexible and high performance search using [Artifactory Query Language \(AQL\)](#).

Since: 3.5.0

Security: Requires an authenticated user. Certain domains/queries may require Admin access.

Usage: POST /api/search/aql

Consumes: text/plain

Sample Usage:

```
POST /api/search/aql
items.find(
  {
    "repo": {"$eq": "libs-release-local"}
  }
)
```

Produces: application/json

Sample Output:

```
{
  "results" : [
    {
      "repo" : "libs-release-local",
      "path" : "org/jfrog/artifactory",
      "name" : "artifactory.war",
      "type" : "item type",
      "size" : "75500000",
      "created" : "2015-01-01T10:10:10",
      "created_by" : "Jfrog",
      "modified" : "2015-01-01T10:10:10",
      "modified_by" : "Jfrog",
      "updated" : "2015-01-01T10:10:10"
    }
  ],
  "range" : {
    "start_pos" : 0,
    "end_pos" : 1,
    "total" : 1
  }
}
```

Artifact Search (Quick Search)

Description: Artifact search by part of file name.

Searches return file info URIs. Can limit search to specific repositories (local or caches).

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/artifact?name=name[&repos=x[,y]]

Headers (Optionally): X-Result-Detail: info (To add all extra information of the found artifact), X-Result-Detail: properties (to get the properties of the found artifact), X-Result-Detail: info, properties (for both).

Produces: application/vnd.org.jfrog.artifactory.search.ArtifactSearchResult+json

Sample Output:

```
GET /api/search/artifact?name=lib&repos=libs-release-local
{
  "results": [
    {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver/lib-ver.pom"
    }, {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver2/lib-ver2.pom"
    }
  ]
}
```

Archive Entries Search (Class Search)

Description: Search archive for classes or any other resources within an archive.

Can limit search to specific repositories (local or caches).

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/archive?name=[archiveEntryName][&repos=x[,y]]

Produces: application/vnd.org.jfrog.artifactory.search.ArchiveEntrySearchResult+json

Sample Output:

```
GET
/api/search/archive?name=*Logger.class&repos=third-party-releases-local,repol-cache
{
  "results" : [
    {
      "entry":
      "org/apache/jackrabbit/core/query/lucene/AbstractIndex.LoggingPrintStream.class",
      "archiveUris": [

        "http://localhost:8081/artifactory/api/storage/third-party-releases-local/org/apache/jackrabbit/jackrabbit-core/1.2.3/jackrabbit-core-1.2.3.jar",

        "http://localhost:8081/artifactory/api/storage/third-party-releases-local/org/apache/jackrabbit/jackrabbit-core/1.3.1/jackrabbit-core-1.3.1.jar"
      ]
    }, {
      "entry": "org/codehaus/plexus/logging/AbstractLogger.class",
      "archiveUris": [

        "http://localhost:8081/artifactory/api/storage/repol-cache/org/codehaus/plexus/plexus-container-default/1.0-alpha-9-stable-1/plexus-container-default-1.0-alpha-9-stable-1.jar"
      ]
    }
  ]
}
```

GAVC Search

Description: Search by Maven coordinates: GroupId, ArtifactId, Version & Classifier.

Search must contain at least one argument. Can limit search to specific repositories (local and remote-cache).

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/gavc?[g=groupId][&a=artifactId][&v=version][&c=classifier][&repos=x[,y]]

Headers (Optionally): X-Result-Detail: info (To add all extra information of the found artifact), X-Result-Detail: properties (to get the properties of the found artifact), X-Result-Detail: info, properties (for both).

Produces: application/vnd.org.jfrog.artifactory.search.GavcSearchResult+json

Sample Output:

```

GET
/api/search/gavc?g=org.acme&a=artifact&v=1.0&c=sources&repos=libs-release-
local
{
  "results": [
    {
      "uri":
"http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
/artifact/1.0/artifact-1.0-sources.jar"
    }, {
      "uri":
"http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
/artifactB/1.0/artifactB-1.0-sources.jar"
    }
  ]
}

```

Property Search

Description: Search by properties.

If no value is specified for a property - assume "". Can limit search to specific repositories (local, remote-cache or virtual).

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/prop?p1=v1,v2[&p2=v3]&repos=x[,y]

Headers (Optionally): X-Result-Detail: info (To add all extra information of the found artifact), X-Result-Detail: properties (to get the properties of the found artifact), X-Result-Detail: info, properties (for both).

Produces: application/vnd.org.jfrog.artifactory.search.MetadataSearchResult+json

Sample Output:

```

GET /api/search/prop?p1=v1,v2&p2=v3&repos=libs-release-local
{
  "results" : [
    {
      "uri":
"http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
/lib/ver/lib-ver.pom"
    }, {
      "uri":
"http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
/lib/ver2/lib-ver2.pom"
    }
  ]
}

```

Checksum Search

Description: Artifact search by checksum (md5, sha1, or sha256)

Searches return file info URIs. Can limit search to specific repositories (local, remote-cache or virtual).

Notes: Requires Artifactory Pro

Since: 2.3.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/checksum?md5=md5sum?sha1=sha1sum?sha256=sha256sum[&repos=x[,y]]

Headers (Optionally): X-Result-Detail: info (To add all extra information of the found artifact), X-Result-Detail: properties (to get the properties of the found artifact), X-Result-Detail: info, properties (for both).

Produces: application/vnd.org.jfrog.artifactory.search.ChecksumSearchResult+json

Sample Output:

```
GET
/api/search/checksum?sha256=9a7fb65f15e00aa2a22c1917d0dafd4374fee8daf0966a
4d94cd37a0b9acafb9&repos=libs-release-local
{
  "results": [
    {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/jfro
      g/build-info-api/1.3.1/build-info-api-1.3.1.jar"
    }
  ]
}
```

Bad Checksum Search

Description: Find all artifacts that have a bad or missing client checksum values (md5 or sha1)

Searches return file info uris. Can limit search to specific repositories (local, remote-cache or virtual).

Notes: Requires Artifactory Pro

Since: 2.3.4

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/badChecksum?type=md5|sha1[&repos=x[,y]]

Produces: application/vnd.org.jfrog.artifactory.search.BadChecksumSearchResult+json

Sample Output:

```
GET /api/search/badChecksum?type=md5&repos=libs-release-local
{
  "results": [
    {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/jfro
      g/build-info-api/1.3.1/build-info-api-1.3.1.jar"
      "serverMd5": "4040c7c184620af0a0a8a3682a75eb7"
      "clientMd5": "4040c7c184620af0a0a8a3682a75e44" //On missing
      checksum this element will be an empty string
    }
  ]
}
```

Artifacts Not Downloaded Since

Description: Retrieve all artifacts not downloaded since the specified Java epoch in **milliseconds**.

Optionally include only artifacts created before the specified `createdBefore` date, otherwise only artifacts created before `notUsedSince` are returned.

Can limit search to specific repositories (local or caches).

Since: 2.2.4

Security: Requires a privileged non-anonymous user.

Usage: GET /api/search/usage?notUsedSince=javaEpochMillis[&createdBefore=javaEpochMillis][&repos=x[,y]]

Produces: application/vnd.org.jfrog.artifactory.search.ArtifactUsageResult+json

Sample Output:

```
GET
/api/search/usage?notUsedSince=long&createdBefore=long&repos=libs-release-
local
{
  "results" : [
    {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver/lib-ver.jar",
      "lastDownloaded": ISO8601
    }, {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver2/lib-ver2.jar",
      lastDownloaded: ISO8601
    }
  ]
}
```

Artifacts With Date in Date Range

Description: Get all artifacts with specified dates within the given range. Search can be limited to specific repositories (local or caches).

Since: 3.2.1

Security: Requires a privileged non-anonymous user.

Usage: GET /api/search/dates?[from=fromVal][&to=toVal][&repos=x[,y]][&dateFields=c[,d]]

Parameters: The `from` and `to` parameters can be either a long value for the java epoch (**milliseconds** since the epoch), or an ISO8601 string value. `from` is mandatory. If `to` is not provided, `now()` will be used instead, and if either are omitted, `400 bad request` is returned.

The `dateFields` parameter is a comma separated list of date fields that specify which fields the `from` and `to` values should be applied to. The date fields supported are: `created`, `lastModified`, `lastDownloaded`.

It is a mandatory field and it also dictates which fields will be added to the JSON returned.

If ANY of the specified date fields of an artifact is within the specified range, the artifact will be returned.

Produces: application/vnd.org.jfrog.artifactory.search.ArtifactResult+json

Sample Output:

```
GET
/api/search/dates?dateFields=created,lastModified,lastDownloaded&from=long
&to=long&repos=libs-release-local
{
  "results" : [
    {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver/lib-ver.jar",
      "created": ISO8601,
      "lastModified": ISO8601,
      "lastDownloaded": ISO8601
    }, {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver2/lib-ver2.jar",
      "created": ISO8601.
      "lastModified": ISO8601,
      "lastDownloaded": ISO8601
    }
  ]
}
```

Artifacts Created in Date Range

Description: Get All Artifacts Created in Date Range

If 'to' is not specified use now(). Can limit search to specific repositories (local or remote-cache).

Since: 2.2.0

Security: Requires a privileged non-anonymous user.

Usage: GET /api/search/creation?from=javaEpochMillis[&to=javaEpochMillis][&repos=x[,y]]

Produces: application/vnd.org.jfrog.artifactory.search.ArtifactCreationResult+json

Sample Output:

```
GET /api/search/creation?from=long&to=long&repos=libs-release-local
{
  "results" : [
    {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver/lib-ver.jar",
      "created": ISO8601
    }, {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme
      /lib/ver2/lib-ver2.jar",
      "created": ISO8601
    }
  ]
}
```

Pattern Search

Description: Get all artifacts matching the given Ant path pattern

Since: 2.2.4

Notes: Requires Artifactory Pro. Pattern "*" is not supported to avoid overloading search results.

Security: Requires a privileged non-anonymous user.

Usage: GET /api/search/pattern?pattern=[repo-key:this/is/a/*pattern*.war](#)

Produces: application/vnd.org.jfrog.artifactory.search.PatternResultFileSet+json

Sample Output:

```
GET /api/search/pattern?pattern=libs-release-local:killer/*/ninja/*/*.jar
{
  "repositoryUri" :
  "http://localhost:8081/artifactory/libs-release-local",
  "sourcePattern" : "libs-release-local:killer/*/ninja/*/*.jar",
  "files" : [
    "killer/coding/ninja/1.0/monkey-1.0.jar",
    "killer/salty/ninja/1.5-SNAPSHOT/pickle-1.5-SNAPSHOT.jar"
  ]
}
```

Builds for Dependency

Description: Find all the builds an artifact is a dependency of (where the artifact is included in the build-info dependencies)

Notes: Requires Artifactory Pro

Since: 2.3.4

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/dependency?sha1=sha1Checksum

Produces: application/vnd.org.jfrog.artifactory.search.DependencyBuilds+json

Sample Output:

```
GET /api/search/dependency?sha1=451a3c5f8cfa44c5d805379e760b5c512c7d250b
{
  "results" : [
    {
      "uri": "http://localhost:8081/artifactory/api/build/my-build/50"
    }, {
      "uri": "http://localhost:8081/artifactory/api/build/my-build/51"
    }
  ]
}
```

License Search

Description: Search for artifacts that were already tagged with license information and their respective licenses.

To search by specific license values use Property Search with the 'artifactory.licenses' property.

When the autofind parameter is specified Artifactory will try to automatically find new license information and return it as part of the result in the found field.

Please note that this can affect the speed of the search quite dramatically, and will still search only on already-tagged artifacts.

Default parameter values when unspecified: unapproved=1, unknown=1, notfound=0, neutral=0, approved=0, autofind=0.

Can limit search to specific repositories (local, remote-cache or virtual).

Since: 2.3.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/search/license?unapproved=1[&unknown=1][¬found=0][&neutral=0][&approved=0][&autofind=0][&repos=x[,y]]

Produces: application/vnd.org.jfrog.artifactory.search.LicenseResult+json

Sample Output:

```
GET
/api/search/license?approved=1&unknown=1&autofind=1&repos=libs-release-local, staging
{
  "results" : [
    {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme/lib/ver/lib-ver.jar",
      "license": "lgplv2",
      "found": "lgplv2",
      "status": "approved"
    }, {
      "uri":
      "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme/lib/ver/lib-ver.jar",
      "license": "cddl v1",
      "found": "gplv3",
      "status": "neutral"
    }, {
      "uri":
      "http://localhost:8081/artifactory/api/storage/staging/org/acme/lib/ver2/lib-ver2.jar",
      "license": "gplv3",
      "found": "gplv3",
      "status": "unapproved"
    }
  ]
}
```

Artifact Version Search

Description: Search for all available artifact versions by GroupId and ArtifactId in local, remote or virtual repositories.

Search can be limited to specific repositories (local, remote and virtual) by settings the `repos` parameter.

Release/integration versions: Unless the `version` parameter is specified, both release and integration versions are returned. When `version` is specified, e.g. `1.0-SNAPSHOT`, result includes only integration versions.

Integration versions are determined by the [repository layout](#) of the repositories searched. For integration search to work the repository layout requires an 'Artifact Path Pattern' that contains the `baseRev` token and then the `fileIntegRev` token with only literals between them.

Remote searches: By default only local and cache repositories are used. When specifying `remote=1`, Artifactory searches for versions on remote repositories. **NOTE!** that this can dramatically slow down the search.

For Maven repositories the remote `maven-metadata.xml` is consulted. For non-maven layouts, remote file listing runs for all remote repositories that have the 'List Remote Folder Items' checkbox enabled.

Filtering results (Artifactory 3.0.2+): The `version` parameter can accept the `*` and/or `?` wildcards which will then filter the final result to match only those who match the given version pattern.

Since: 2.6.0

Notes: Requires Artifactory Pro

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/versions?[g=groupId]&a=artifactId[&v=version][&remote=0/1][&repos=x[,y]]

Produces: application/vnd.org.jfrog.artifactory.search.ArtifactVersionsResult+json

Sample Output:

```
GET /api/search/versions?g=org.acme&a=artifact&repos=libs-release-local
{
  "results": [
    {
      "version": "1.2",
      "integration": false
    }, {
      "version": "1.0-SNAPSHOT",
      "integration": true
    }, {
      "version": "1.0",
      "integration": false
    }
  ]
}
```

Artifact Latest Version Search Based on Layout

Description: Search for the latest artifact version by groupId and artifactId, based on the layout defined in the repository

Search can be limited to specific repositories (local, remote-cache or virtual) by settings the `repos` parameter. When searching in a virtual repository, each child-repository layout will be consulted accordingly.

Latest release vs. latest integration: Unless the `version` parameter is specified, the search returns the latest artifact release version. When `version` is specified, e.g. `1.0-SNAPSHOT`, the result is the latest integration version. Integration versions are determined by the [repository layout](#) of the repositories searched. For integration search to work the repository layout requires an "Artifact Path Pattern" that contains the `baseRev` token and then the `fileIntegRev` token with only literals between them.

Remote searches: By default only local and cache repositories will be used. When specifying `remote=1`, Artifactory searches for versions on remote repositories. **NOTE!** that this can dramatically slow down the search.

For Maven repositories the remote `maven-metadata.xml` will be consulted. For non-Maven layouts, remote file listing runs for all remote repositories that have the 'List Remote Folder Items' checkbox enabled.

Filtering results (Artifactory 3.0.2+): The `version` parameter can accept the `*` and/or `?` wildcards which will then filter the final result to match only those who match the given version pattern.

Artifact path pattern: The `[org]` and `[module]` fields must be specified in the [artifact path pattern](#) of the repository layout for this call to work.

Since: 2.6.0

Notes: Requires Artifactory Pro

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/search/latestVersion?[g=groupId]&a=artifactId[&v=version][&remote=1][&repos=x[,y]]

Produces: text/plain

Sample Output:

```
GET
/api/search/latestVersion?g=org.acme&a=artifact&v=1.0-SNAPSHOT&repos=libs-
snapshot-local

1.0-201203131455-2
```

Artifact Latest Version Search Based on Properties

Description: Search for artifacts with the latest value in the "version" property. **Only artifacts with a "version" property expressly defined in**

lower case will be returned. Results can be filtered by specifying additional properties.

{repo}: Specify a repository to search through or replace with "_any" to search through all repositories

{path}: Specify a path to search through or replace with "_any" to search through all paths

listFiles=0 (default): Artifactory will only retrieve the latest version

listFiles=1: Artifactory will retrieve the latest version and the corresponding files

You may specify filters to restrict the set of artifacts that are searched by adding any properties to your search URL

Notes: Requires Artifactory Pro

Since: 3.1.1

Security: Requires an authenticated user (not anonymous) to use the api and read permission to the repository of each artifact.

Usage: GET /api/versions/{repo}/{path}?[listFiles=0/1]&[<property key>=<property value>]&[<property key>=<property value>]

Consumes: json

Examples:

```
Return the latest version and corresponding artifacts by searching for
through all repositories whose path starts with a/b and are annotated with
the properties os=win and license=GPL.
```

```
GET /api/versions/_any/a/b?os=win&license=GPL&listFiles=1
```

```
{
  "version" : "1.1.2",
  "artifacts" : [ {
    "uri" : "http://...."
  } ]
}
```

```
Return the latest version (without the corresponding artifacts) by
searching through all repositories whose path starts with a/b and are
annotated with the properties os=win and license=GPL.
```

```
Return only the version.
```

```
GET /api/versions/_any/a/b?os=win&license=GPL
```

```
{
  "version" : "1.1.2",
  "artifacts" : []
}
```

Build Artifacts Search

Description: Find all the artifacts related to a specific build.

Notes: Requires Artifactory Pro

Since: 2.6.5

Security: Requires a privileged user (can be anonymous)

Usage: POST /api/search/buildArtifacts

Consumes: application/vnd.org.jfrog.artifactory.search.BuildArtifactsRequest+json

Sample Usage:

```

POST /api/search/buildArtifacts
{
  +"buildName": "build-name" // The build name for search by
  +"buildNumber": "15" // The build number to search by, can be LATEST to
search for the latest build number
  -"buildStatus": "Released" // Optionally search by latest build status
(e.g: "Released")
  -"repos": ["libs-release-local,ext-release-local"] // Optionally refine
search for specific repos, omit to search within all repositories
  -"mappings" [ // Optionally refine the search by providing a list of
regex patterns to search by
    {
      "input": "(.+)-sources.jar"
    },
    {
      "input": "(.+)-javadoc.jar"
    }
  ]
}

```

Produces: application/vnd.org.jfrog.artifactory.search.BuildArtifactsSearchResult+json

Sample Output:

```

POST /api/search/buildArtifacts
{
  "results" : [
    {
      "downloadUri":
      "http://localhost:8081/artifactory/libs-release-local/org/acme/lib/ver/lib-
sources.jar"
    }, {
      "downloadUri":
      "http://localhost:8081/artifactory/ext-release-local/org/acme/lib/ver/lib-
ver-javadoc.jar"
    }
  ]
}

```

List Docker Repositories

Description: Lists all Docker repositories hosted in under an Artifactory Docker repository.

Notes: Requires Artifactory Pro

Security: Requires a privileged user

Usage: GET /api/docker/{repo-key}/v2/_catalog

Produces: application/json


```
{
  "repositories": [
    <name>,
    ...
  ]
}
```

Since: 4.4.3

Sample Usage:

```
GET /api/docker/docker-local/v2/_catalog
{
  "repositories": [
    "busybox",
    "centos",
    "hello-world"
  ]
}
```

SECURITY

Get Users

Description: Get the users list

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/security/users

Produces: [application/vnd.org.jfrog.artifactory.security.Users+json](#)

Sample Output:

```
GET /api/security/users
[
  {
    "name": "davids"
    "uri" : "http://localhost:8081/artifactory/api/security/users/davids"
  }, {
    "name": "danl"
    "uri" : "http://localhost:8081/artifactory/api/security/users/danl"
  }
]
```

Get User Details

Description: Get the details of an Artifactory user

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/security/users/{userName}

Produces: `application/vnd.org.jfrog.artifactory.security.User+json`

Sample Output:

```
GET /api/security/users/davids
{
  user.json
}
```

Get User Encrypted Password

Description: Get the encrypted password of the authenticated requestor

Since: 3.3.0

Security: Requires a privileged user

Usage: GET /api/security/encryptedPassword

Produces: plain/text

Sample Output:

```
GET /api/security/encryptedPassword

AP5v2zs9ga7CJNZb74u3arAKE5B
```

Create or Replace User

Description: Creates a new user in Artifactory or replaces an existing user

Since: 2.4.0

Notes: Requires Artifactory Pro

Missing values will be set to the default values as defined by the consumed type.

Security: Requires an admin user

Usage: PUT /api/security/users/{userName}

Consumes: `application/vnd.org.jfrog.artifactory.security.User+json`

Sample Usage:

```
PUT /api/security/users/davids
{
  user.json
}
```

Update User

Description: Updates an exiting user in Artifactory with the provided user details.

Since: 2.4.0

Notes: Requires Artifactory Pro

Missing values will be set to the default values as defined by the consumed type

Security: Requires an admin user

Usage: POST /api/security/users/{userName}

Consumes: `application/vnd.org.jfrog.artifactory.security.User+json`

Sample Usage:

```
POST /api/security/users/davids
{
  user.json
}
```

Delete User

Description: Removes an Artifactory user.

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: DELETE /api/security/users/{userName}

Produces: application/text

Sample Usage:

```
DELETE /api/security/users/davids

User 'davids' has been removed successfully.
```

Expire Password for a Single User

Description: Expires a user's password

Since: 4.4.2

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: POST /api/security/users/authorization/expirePassword/{userName}

Sample Usage:

```
POST /api/security/users/authorization/expirePassword/davids
```

Expire Password for Multiple Users

Description: Expires password for a list of users

Since: 4.4.2

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: POST /api/security/users/authorization/expirePassword -H "Content-type: application/json" -d '[{userA}, {userB}]'

Sample Usage:

```
POST /api/security/users/authorization/expirePassword -H "Content-type:
application/json" -d ' [{davids}, {johnb}]'
```

Expire Password for All Users

Description: Expires password for all users

Since: 4.4.2

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: POST /api/security/users/authorization/expirePasswordForAllUsers

Sample Usage:

```
POST /api/security/users/authorization/expirePasswordForAllUsers
```

Unexpire Password for a Single User

Description: Unexpires a user's password

Since: 4.4.2

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: POST /api/security/users/authorization/unexpirePassword/{userName}

Produces: application/text

Sample Usage:

```
POST /api/security/users/authorization/unexpirePassword/davids
```

Change Password

Description: Changes a user's password

Since: 4.4.2

Notes: Requires Artifactory Pro

Security: Admin can apply this method to all users, and each (non-anonymous) user can use this method to change his own password.

Usage: POST /api/security/users/authorization/changePassword -H "Content-type: application/json" -d '{ "userName": "{user}", "oldPassword": "{old password}", "newPassword1": "{new password}", "newPassword2": "{verify new password}" }

Produces: application/text

Sample Usage:

```
POST /api/security/users/authorization/changePassword -H "Content-type: application/json" -d ' { "userName" : "davids", "oldPassword" : "op", "newPassword1" : "np", "newPassword2" : "np" }
```

Get Password Expiration Policy

Description: Retrieves the password expiration policy

Since: 4.4.2

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/security/users/authorization/passwordExpirationPolicy

Produces: application/json

Sample Usage:

```
GET /api/security/users/authorization/passwordExpirationPolicy
{
  "enabled": "true"
  "passwordMaxAge": "60"
  "notifyByEmail": "true"
}
```

Set Password Expiration Policy

Description: Sets the password expiration policy

Since: 4.4.2

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: PUT /api/security/users/authorization/passwordExpirationPolicy -H "Content-type: application/json" -d ' { "enabled" : "true|false", "passwordMaxAge" : "1-999", "notifyByEmail": "true|false" }

Produces: application/json

Sample Usage:

```
POST /api/security/users/authorization/passwordExpirationPolicy -H
"Content-type: application/json" -d ' { "enabled" : "true",
"passwordMaxAge" : "60", "notifyByEmail": "true" }
```

Configure User Lock Policy

Description: Configures the user lock policy that locks users out of their account if the number of repeated incorrect login attempts exceeds the configured maximum allowed.

Security: Requires a valid admin user

Usage: PUT /api/security/userLockPolicy

Produces: application/text

Since: 4.4

Sample usage:

```
PUT http://{host}:{port}/artifactory/api/security/userLockPolicy -H
'Content-Type: application/json'-d '
{
  "enabled" : true|false,
  "loginAttempts" : {value}
}'
```

Retrieve User Lock Policy

Description: Retrieves the currently configured user lock policy.

Security: Requires a valid admin user

Usage: GET /api/security/userLockPolicy

Produces: application/json

Since: 4.4

Sample usage:

```
GET http://{host}:{port}/artifactory/api/security/userLockPolicy
'{
  "enabled" : true|false,
  "loginAttempts" : {value}
}'
```

Get Locked Out Users

Description: If locking out users is enabled, lists all users that were locked out due to recurrent incorrect login attempts.

Security: Requires a valid admin user

Usage: GET </api/security/lockedUsers>

Produces: application/json

Since: 4.4

Sample Usage:

```
GET /api/security/lockedUsers

[ "usera", "userb", ... ]
```

Unlock Locked Out User

Description: Unlocks a single user that was locked out due to recurrent incorrect login attempts.

Security: Requires a valid admin user

Usage: POST </api/security/unlockUsers/{userName}>

Produces: application/text

Since: 4.4

Sample Usage:

```
POST /api/security/unlockUsers/{userName}
```

Unlock Locked Out Users

Description: Unlocks a list of users that were locked out due to recurrent incorrect login attempts.

Security: Requires a valid admin user

Usage: POST </api/security/unlockUsers>

Produces: application/text

Since: 4.4

Sample Usage:

```
POST /api/security/unlockUsers -H 'Content-Type: application/json' -d '[
{userA}, {userB} ]'
```

Unlock All Locked Out Users

Description: Unlocks all users that were locked out due to recurrent incorrect login attempts.

Security: Requires a valid admin user

Usage: POST </api/security/unlockAllUsers>

Produces: application/text

Since: 4.4

Sample Usage:

```
POST /api/security/unlockAllUsers
```

Create API Key

Description:Create an API key for the current user. Returns an error if API key already exists - use regenerate API key instead.

Since: 4.3.0

Usage: POST /api/security/apiKey

Produces: application/json

Sample input:

```
POST /api/security/apiKey
```

Sample output:

```
{
  "apiKey": "30loposOtVFyCmrT+cXmCASCmVMPrSYXkWIjIyDCXsY="
}
```

Regenerate API Key

Description:Regenerate an API key for the current user

Since: 4.3.0

Usage: PUT /api/security/apiKey

Produces: application/json

Sample input:

```
PUT /api/security/apiKey
```

Sample output:

```
{
  "apiKey": "30loposOtVFyCmrT+cXmCASCmVMPrSYXkWIjIyDCXsY="
}
```

Get API Key

Description:Get the current user's own API key

Since: 4.3.0

Usage: GET /api/security/apiKey

Produces: application/json

Sample usage:

```
GET /api/security/apiKey
```

Sample output:

```
{
  "apiKey": "30l0posOtVFyCMrT+cXmCASCmVMPrSYXkWIjjiyDCXsY="
}
```

Revoke API Key

Description: Revokes the current user's API key

Since: 4.3.0

Usage: DELETE /api/security/apiKey

Produces: application/json

Revoke User API Key

Description: Revokes the API key of another user

Since: 4.3.0

Security: Requires a privileged user (Admin only)

Usage: DELETE /api/security/apiKey/{username}

Produces: application/json

Revoke All API Keys

Description: Revokes all API keys currently defined in the system

Since: 4.3.0

Security: Requires a privileged user (Admin only)

Usage: DELETE /api/security/apiKey?deleteAll={0/1}

Produces: application/json

Get Groups

Description: Get the groups list

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/security/groups

Produces: [application/vnd.org.jfrog.artifactory.security.Users+json](#), [application/vnd.org.jfrog.artifactory.security.Groups+json](#), [application/vnd.org.jfrog.artifactory.security.PermissionTargets+json](#)

Sample Output:

```
GET /api/security/groups
[
  {
    "name": "readers"
    "uri" : "http://localhost:8081/artifactory/api/security/groups/readers"
  }, {
    "name": "tech-leads"
    "uri" :
"http://localhost:8081/artifactory/api/security/groups/tech-leads"
  }
]
```

Get Group Details

Description: Get the details of an Artifactory Group

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/security/groups/{groupName}

Produces: application/vnd.org.jfrog.artifactory.security.Group+json

Sample Output:

```
GET /api/security/groups/dev-leads
{
  group.json
}
```

Create or Replace Group

Description: Creates a new group in Artifactory or replaces an existing group

Since: 2.4.0

Notes: Requires Artifactory Pro

Missing values will be set to the default values as defined by the consumed type.

Security: Requires an admin user

Usage: PUT /api/security/groups/{groupName}

Consumes: application/vnd.org.jfrog.artifactory.security.Group+json

Sample Usage:

```
PUT /api/security/groups/dev-leads
{
  group.json
}
```

Update Group

Description: Updates an exiting group in Artifactory with the provided group details.

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: POST /api/security/groups/{groupName}

Consumes: application/vnd.org.jfrog.artifactory.security.Group+json

Sample Usage:

```
POST /api/security/groups/dev-leads
{
  group.json
}
```

Delete Group

Description: Removes an Artifactory group.

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: DELETE /api/security/groups/{groupName}

Produces: application/text

Sample Usage:

```
DELETE /api/security/groups/dev-leads

Group 'dev-leads' has been removed successfully.
```

Get Permission Targets

Description: Get the permission targets list

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/security/permissions

Produces: [application/vnd.org.jfrog.artifactory.security.Users+json](#), [application/vnd.org.jfrog.artifactory.security.Groups+json](#), [application/vnd.org.jfrog.artifactory.security.PermissionTargets+json](#)

Sample Output:

```
GET /api/security/permissions
[
  {
    "name": "readSourceArtifacts"
    "uri" :
    "http://localhost:8081/artifactory/api/security/permissions/readSourceArtifacts"
  }, {
    "name": "populateCaches"
    "uri" :
    "http://localhost:8081/artifactory/api/security/permissions/populateCaches"
  }
]
```

Get Permission Target Details

Description: Get the details of an Artifactory Permission Target

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: GET /api/security/permissions/{permissionTargetName}

Produces: [application/vnd.org.jfrog.artifactory.security.PermissionTarget+json](#)

Sample Output:

```
GET /api/security/permissions/populateCaches
{
  permission-target.json
}
```

Create or Replace Permission Target

Description: Creates a new permission target in Artifactory or replaces an existing permission target

Since: 2.4.0

Notes: Requires Artifactory Pro

Missing values will be set to the default values as defined by the consumed type.

Security: Requires an admin user

Usage: PUT /api/security/permissions/{permissionTargetName}

Consumes: application/vnd.org.jfrog.artifactory.security.PermissionTarget+json

Sample Usage:

```
PUT /api/security/permissions/populateCaches
{
  permission-target.json
}
```

Delete Permission Target

Description: Deletes an Artifactory permission target.

Since: 2.4.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: DELETE /api/security/permissions/{permissionTargetName}

Produces: application/text

Sample usage:

```
DELETE /api/security/permissions/populateCaches

Permission Target 'remoteCachePopulation' has been removed successfully.
```

Effective Item Permissions

Description: Returns a list of effective permissions for the specified item (file or folder).

Only users and groups with some permissions on the item are returned. Supported by local and local-cached repositories.

Permissions are returned according to the following conventions:

m=admin; d=delete; w=deploy; n=annotate; r=read

Notes: Requires Artifactory Pro

Since: 2.3.4

Security: Requires a valid admin or local admin user.

Usage: GET /api/storage/{repoKey}/{itemPath}?permissions

Produces: application/vnd.org.jfrog.artifactory.storage.ItemPermissions+json

Sample Output:

```
GET /api/storage/libs-release-local/org/acme?permissions
{
  "uri":
  "http://localhost:8081/artifactory/api/storage/libs-release-local/org/acme"
  "principals": {
    "users" : {
      "bob": ["r","w","m"],
      "alice" : ["d","w","n", "r"]
    },
    "groups" : {
      "dev-leads" : ["m","r","n"],
      "readers" : ["r"]
    }
  }
}
```

Security Configuration

Description: Retrieve the security configuration (security.xml).

Since: 2.2.0

Notes: This is an advanced feature - make sure the new configuration is really what you wanted before saving.

Security: Requires a valid admin user

Usage: GET /api/system/security

Produces: application/xml

Sample Output:

```
GET /api/system/security

<security.xml/>
```

Save Security Configuration (Deprecated)

Description: Save the security configuration (security.xml). Requires the security.xml file from the same version.

Since: 2.2.0

Notes: This API is **deprecated**.

Security: Requires a valid admin user

Usage: POST /api/system/security

Consumes: application/xml

Sample Usage:

```
POST /api/system/security

<security.xml/>
```

Activate Master Key Encryption

Description: Creates a new master key and activates master key encryption.

Since: 3.2.2

Notes: This is an advanced feature intended for administrators

Security: Requires a valid admin user

Usage: POST /api/system/encrypt

Produces: text/plain

Sample Usage:

```
POST /api/system/encrypt
```

```
DONE
```

Deactivate Master Key Encryption

Description: Removes the current master key and deactivates master key encryption.

Since: 3.2.2

Notes: This is an advanced feature intended for administrators

Security: Requires a valid admin user

Usage: POST /api/system/decrypt

Produces: text/plain

Sample Usage:

```
POST /api/system/decrypt
```

```
DONE
```

Set GPG Public Key

Description: Sets the public key that Artifactory provides to Debian and Opkg clients to verify packages

Security: Requires a valid admin user

Usage: PUT /api/gpg/key/public

Produces: text/plain

Since: 3.3

Sample Usage:

```
PUT /api/gpg/key/public
```

Get GPG Public Key

Description: Gets the public key that Artifactory provides to Debian and Opkg clients to verify packages

Security: Requires an authenticated user, or anonymous (if "Anonymous Access" is globally enabled)

Usage: GET /api/gpg/key/public

Produces: text/plain

Since: 3.3

Sample Usage:

```
GET /api/gpg/key/public
```

Set GPG Private Key

Description: Sets the private key that Artifactory will use to sign Debian and ipk packages

Security: Requires a valid admin user

Usage: PUT /api/gpg/key/private

Produces: text/plain

Since: 3.3

Sample Usage:

```
PUT /api/gpg/key/private
```

Set GPG Pass Phrase

Description: Sets the pass phrase required signing Debian and ipk packages using the private key

Security: Requires a valid admin user

Usage: PUT /api/gpg/key/passphrase

Headers: -H X-GPG-PASSPHRASE:passphrase

Produces: text/plain

Since: 3.3

Sample Usage:

```
PUT /api/gpg/key/passphrase
```

Create Token

Description: Creates an access token

Since: 5.0.0

Security: Requires a valid user

Usage: POST /api/security/token

Content-Type: application/x-www-form-urlencoded

Produces: application/json

```
{
  "access_token": "<the access token>",
  "expires_in":   "<Validity period in seconds>",
  "scope":        "<access scope>",
  "token_type":   "Bearer",
  "refresh_token": "<the refresh token if access_token is refreshable>"
}
```

Sample Usage:

```
curl -uadmin:password -XPOST
"http://localhost:8081/artifactory/api/security/token" -d "username=johnq"
-d "scope=member-of-groups:readers"

200
{
  "access_token": "adsdgbtybbeeyh...",
  "expires_in": 3600,
  "scope": "api:* member-of-groups:readers",
  "token_type": "Bearer",
  "refresh_token": "fgsfgsdugh8dgu9s8gy9hsg..."
}
```

This endpoint takes the following parameters:

| | |
|--------------------|--|
| grant_type | <p>[Optional, default: "client_credentials"]</p> <p>The grant type used to authenticate the request. In this case, the only value supported is "client_credentials" which is also the default value if this parameter is not specified.</p> |
| username | <p>The user name for which this token is created. If the user does not exist, a transient user is created. Non-admin users can only create tokens for themselves so they must specify their own username.</p> <p>If the user does not exist, the <code>member-of-groups</code> scope token must be provided (e.g. <code>member-of-groups: g1, g2, g3...</code>)</p> |
| scope | <p>[Optional if the user specified in username exists]</p> <p>The scope to assign to the token provided as a space-separated list of scope tokens. Currently there are three possible scope tokens:</p> <ul style="list-style-type: none"> "api:*" - indicates that the token grants access to REST API calls. This is always granted by default whether specified in the call or not. member-of-groups:[<group-name>] - indicates the groups that the token is associated with. The token grants access according to the permission targets specified for the groups listed. Specify "*" for group-name to indicate that the token should provide the same access privileges that are given to the group of which the logged in user is a member. A non-admin user can only provide a scope that is a subset of the groups to which he belongs "jfrt@<instance-id>:admin" - provides admin privileges on the specified Artifactory instance. This is only available for administrators. <p>If omitted and the username specified exists, the token is granted the scope of that user.</p> |
| expires_in | <p>[Optional, default: 3600]</p> <p>The time in seconds for which the token will be valid. To specify a token that never expires, set to zero. Non-admin can only set a value that is equal to or less than the default 3600.</p> |
| refreshable | <p>[Optional, default: false]</p> <p>If true, this token is refreshable and the refresh token can be used to replace it with a new token once it expires.</p> |
| audience | <p>[Optional, default: Only the service ID of the Artifactory instance that created the token]</p> <p>A space-separated list of the other Artifactory instances or services that should accept this token identified by their Artifactory Service IDs as obtained from the Get Service ID endpoint.</p> <p>In case you want the token to be accepted by all Artifactory instances you may use the following audience parameter "audience=jfrt@*".</p> |

Refresh Token

Description: Refresh an access token to extend its validity. If only the access token and the refresh token are provided (and no other

parameters), this pair is used for authentication. If username or any other parameter is provided, then the request must be authenticated by a token that grants admin permissions.

Since: 5.0.0

Security: Requires a valid user (unless both access token and refresh token are provided)

Usage: POST /api/security/token

Content-Type: application/x-www-form-urlencoded

Produces: application/json (Please refer to [Create Tokens](#))

Sample Usage:

```
curl -XPOST "http://localhost:8081/artifactory/api/security/token" -d
"grant_type=refresh_token" -d "refresh_token=fgsg53t3g..." -d
"access_token=gsfdgw35gt..."

200 (Success) As in Create Token

400 (Error) If the token was created by a different Artifactory instance
(and hence cannot be refreshed)
```

This endpoint takes the following parameters:

| | |
|---------------|---|
| grant_type | Should be set to refresh_token. |
| refresh_token | The refresh token of the access token that needs to be refreshed. |
| access_token | [Required if username is omitted] The access token to refresh. |
| username | Please refer to Create Tokens . |
| scope | Note: access_token and username are mutually exclusive, so only one of these parameters should be specified. If access_token is provided, the new token is created with the same settings as that token. |
| expires_in | |
| refreshable | |
| audience | |

Revoke Token

Description: Revoke an access token

Since: 5.0.0

Security: Requires a valid user

Usage: POST /api/security/token/revoke

Content-Type: application/x-www-form-urlencoded

Produces: application/json

Sample Usage:


```
curl -uadmin:password -XPOST
"http://localhost:8081/artifactory/api/security/token/revoke" -d
"token=fasdt3..."

200 OK (Also returned if the token was already revoked or non-existent)

400 (Error) If the token was created by a different Artifactory instance
(and hence cannot be revoked)
```

This endpoint takes the following parameters:

| | |
|-------|-------------------------|
| token | The token to be revoked |
|-------|-------------------------|

Get Service ID

Description: Provides the service ID of an Artifactory instance or cluster

Since: 5.0.0

Security: Requires an admin user

Usage: GET /api/system/service_id

Produces: text/plain

Sample Usage:

```
curl -uadmin:password -XGET
"http://localhost:8081/artifactory/api/system/service_id"

200
jfrt@ee27b1d1-534d-4723-80b5-5bd893d19c43
```

Get Certificates

Description: Returns a list of installed SSL certificates.

Since: 5.4.0

Security: Requires an admin user

Usage: GET /api/system/security/certificates

Produces: application/json

```
[
  {
    "certificateAlias" : "<The Certificate Alias>",
    "issuedTo" : "<The entity to whom the certificate was issued>",
    "issuedBy" : "<The issuing entity>",
    "issuedOn" : "<ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ)>",
    "validUntil" : "<ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ)>",
    "fingerPrint" : "<The certificate's SHA256 fingerprint>"
  }
]
```

Sample Usage:

```
GET /api/system/security/certificates

[
  {
    "certificateAlias" : "example1",
    "issuedTo" : "JFrog",
    "issuedBy" : "Some_CA",
    "issuedOn" : "Sun May 01 2017 10:00:00 GMT +02:00 (UTC)",
    "validUntil" : "Sun May 01 2019 10:00:00 GMT +02:00 (UTC)",
    "fingerPrint" : "ab:cd:ef:gh"
  },
  {
    "certificateAlias" : "example2",
    "issuedTo" : "Cool-Company",
    "issuedBy" : "Some_Other_CA",
    "issuedOn" : "Sun May 01 2017 10:00:00 GMT +02:00 (UTC)",
    "validUntil" : "Sun May 01 2019 10:00:00 GMT +02:00 (UTC)",
    "fingerPrint" : "ab:cd:ef:gh"
  }
]
```

Add Certificate

Description: Adds an SSL certificate.

Since: 5.4.0

Security: Requires an admin user

Usage: POST /api/system/security/certificates/{Certificate_alias} -T {Certificate PEM file}

Consumes: application/text

Produces: application/json

```
{
  "status" : 200,
  "message" : ["The certificates were successfully installed"]
}
```

Delete Certificate

Description: Deletes an SSL certificate.

Since: 5.4.0

Security: Requires an admin user

Usage: DELETE /api/system/security/certificates/{Certificate_alias}

Produces: application/json

Sample Usage:

```
DELETE /api/security/certificates/cert1
```

Response:

```
{
  "status" : 200,
  "message" : "The certificates were successfully deleted"
}
```

REPOSITORIES

Get Repositories

Description: Returns a list of minimal repository details for all repositories of the specified type.

Since: 2.2.0

Security: Requires a privileged user (can be anonymous)

Usage: GET /api/repositories[?type=repositoryType (local|remote|virtual)]

Produces: application/vnd.org.jfrog.artifactory.repositories.RepositoryDetailsList+json

Sample Output:

```
GET /api/repositories
[
  {
    "key" : "libs-releases-local",
    "type" : "LOCAL",
    "description" : "Local repository for in-house libraries",
    "url" : "http://localhost:8081/artifactory/libs-releases-local"
  }, {
    "key" : "libs-snapshots-local",
    "type" : "LOCAL",
    "description" : "Local repository for in-house snapshots",
    "url" : "http://localhost:8081/artifactory/libs-snapshots-local"
  }
]
```

Repository Configuration

Description: Retrieves the current configuration of a repository. Supported by local, remote and virtual repositories.

Since: 2.3.0

Notes: Requires Artifactory Pro

Security: Requires a valid user for a shared remote repository and admin user for anything else. Shared remote repository data is sanitized for security when a non-admin user is used.

Usage: GET /api/repositories/{repoKey}

Produces: application/vnd.org.jfrog.artifactory.repositories.LocalRepositoryConfiguration+json, application/vnd.org.jfrog.artifactory.repositories.RemoteRepositoryConfiguration+json, application/vnd.org.jfrog.artifactory.repositories.VirtualRepositoryConfiguration+json

Sample Output:

```
GET /api/repositories/libs-release-local
{
  repository-config.json
}
```

Create Repository

Description: Creates a new repository in Artifactory with the provided configuration. Supported by local, remote and virtual repositories. A position may be specified using the `pos` parameter. If the map size is shorter than `pos` the repository is the last one (the default behavior).

Since: 2.3.0

Notes: Requires Artifactory Pro

An existing repository with the same key are removed from the configuration and its content is removed!

Missing values are set to the default values as defined by the consumed type spec.

Security: Requires an admin user

Usage: PUT /api/repositories/{repoKey}[?pos=position]

Consumes: [application/vnd.org.jfrog.artifactory.repositories.LocalRepositoryConfiguration+json](#), [application/vnd.org.jfrog.artifactory.repositories.RemoteRepositoryConfiguration+json](#),

[application/vnd.org.jfrog.artifactory.repositories.VirtualRepositoryConfiguration+json](#)

Sample Usage:

```
PUT /api/repositories/libs-release-local?pos=2
{
  repository-config.json
}
```

Update Repository Configuration

Description: Updates an exiting repository configuration in Artifactory with the provided configuration elements. Supported by local, remote and virtual repositories.

Since: 2.3.0

Notes: Requires Artifactory Pro

The class of a repository (the `rclass` attribute) cannot be updated.

Security: Requires an admin user

Usage: POST /api/repositories/{repoKey} -H "Content-Type: application/json"

Consumes: [application/vnd.org.jfrog.artifactory.repositories.LocalRepositoryConfiguration+json](#), [application/vnd.org.jfrog.artifactory.repositories.RemoteRepositoryConfiguration+json](#),

[application/vnd.org.jfrog.artifactory.repositories.VirtualRepositoryConfiguration+json](#)

Sample Usage:

```
POST /api/repositories/libs-release-local -H "Content-Type:
application/json"
{
  repository-config.json
}
```

Delete Repository

Description: Removes a repository configuration together with the whole repository content. Supported by local, remote and virtual repositories.

Since: 2.3.0

Notes: Requires Artifactory Pro

Security: Requires an admin user

Usage: DELETE /api/repositories/{repoKey}

Produces: application/text

Sample Usage:

```
DELETE /api/repositories/libs-release-local
```

Repository 'libs-release-local' and all its content have been removed successfully.

Remote Repository Configuration (Deprecated)

Description: Repository Configuration (Deprecated)

Gets the shared configuration of a remote repository.

Since: 2.2.0

Notes: This API is **deprecated**. Use the Get Repository Configuration API instead.

Security: Requires a valid user for a shared remote repository and admin user for anything else. Shared remote repository data will be sanitized for security when non-admin user is used.

Usage: GET /api/repositories/{remoteRepoName}/configuration

Produces: application/vnd.org.jfrog.artifactory.repositories.SharedRemoteRepositoryConfiguration+json

Sample Output:

```
GET /api/repositories/remote-repo/configuration
{
  repository-config.json
}
```

Calculate YUM Repository Metadata

Description: For Local repositories: calculates/recalculates the YUM metadata for this repository, based on the RPM package currently hosted in the repository. Supported by local and virtual repositories only.

Calculation can be synchronous (the default) or asynchronous.

For Virtual repositories, calculates the merged metadata from all aggregated repositories on the specified path. The **path** parameter must be passed for virtual calculation.

Please see the [YUM integration](#) documentation for more details.

Notes: Requires Artifactory Pro. Immediate calculation requests cannot be called on repositories with automatic asynchronous calculations enabled (applies to local repositories only).

Security: Up to version 4.8, requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST /api/yum/{repoKey}?path={path to repodata dir}[&async=0/1]

Produces: application/text

Since: 2.3.5

Sample Output:

```
POST /api/yum/yum-local?async=1
POST /api/yum/yum-virtual?path=7/os/x86_64&async=1

YUM metadata calculation for repository 'yum-local' accepted.
```

Calculate NuGet Repository Metadata

Description: Recalculates all the NuGet packages for this repository (local/cache/virtual), and re-annotate the NuGet properties for each NuGet package according to its internal nuspec file.

Please see the [NuGet integration](#) documentation for more details.

Supported by local, local-cache, remote and virtual repositories.

Notes: Requires Artifactory Pro.

Security: Up to version 4.8 , requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST /api/nuget/{repoKey}/reindex

Produces: application/text

Since: 3.0.3

Sample Output:

```
POST /api/nuget/nuget-local/reindex
```

```
NuGet reindex calculation for repository 'nuget-local' accepted.
```

Calculate Npm Repository Metadata

Description: Recalculates the npm search index for this repository (local/virtual). Please see the [Npm integration](#) documentation for more details. Supported by local and virtual repositories.

Notes: Requires Artifactory Pro.

Security: Up to version 4.8 , requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST /api/npm/{repoKey}/reindex

Produces: application/text

Since: 3.2.0

Sample Output:

```
POST /api/npm/npm-local/reindex
```

```
Recalculating index for npm repository npm-local scheduled to run
```

Calculate Maven Index

Description: Calculates/caches a Maven index for the specified repositories.

For a virtual repository specify all underlying repositories that you want the aggregated index to include.

Calculation can be forced, which for remote repositories will cause downloading of a remote index even if a locally cached index has not yet expired; and index recalculation based on the cache on any failure to download the remote index, including communication errors (the default behavior is to only use the cache when a remote index cannot be found and returns a 404). Forcing has no effect on local repositories index calculation.

Please see the [Exposing Maven Indexes](#) documentation for more details.

Notes: Requires Artifactory Pro.

Security: Up to version 4.8 , requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST /api/maven[?repos=x[,y]][&force=0/1]

Produces: application/text

Since: 2.5.0

Sample Output:

```
POST /api/maven?repos=libs-release-local,ext-release-local&force=1
```

```
Maven index refresh for repositories '[libs-release-local,
ext-release-local]' has been accepted.
```

Calculate Maven Metadata

Description: Calculates Maven metadata on the specified path (local repositories only).

Security: Up to version 4.8 , requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST /api/maven/calculateMetadata/{repoKey}/{folder-path}?{nonRecursive=true | false}

Produces: application/text

Since: 3.0.2

Sample Output:

```
POST /api/maven/calculateMetadata/libs-release-local/org/acme
OK
```

Calculate Debian Repository Metadata

Description: Calculates/recalculates the Packages and Release metadata for this repository, based on the Debian packages in it.

Calculation can be synchronous (the default) or asynchronous. Please refer to [Debian Repositories](#) for more details. Supported by local repositories only.

From version 4.4, by default, the recalculation process also writes several entries from the Debian package's metadata as properties on all of the artifacts (based on the control file's content).

This operation may not always be required (for example, if the Debian files are intact and were not modified, only the index needs to be recalculated. The operation is resource intensive and can be disabled by passing the *?writeProps=0* query param.

Notes: Requires Artifactory Pro.

Security: Up to version 4.8 , requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST api/deb/reindex/{repoKey} [?async=0/1][?writeProps=0/1]

Headers (Optionally): -H X-GPG-PASSPHRASE:passphrase

Produces: application/text

Since: 3.3

Sample Output:

```
POST /api/deb/reindex/debian-local

Recalculating index for Debian repository debian-local scheduled to run.
```

Calculate Opkg Repository Metadata

Description: Calculates/recalculates the Packages and Release metadata for this repository, based on the ipk packages in it (in each feed location).

Calculation can be synchronous (the default) or asynchronous. Please refer to [Opkg Repositories](#) for more details. Supported by local repositories only.

By default, the recalculation process also writes several entries from the ipk package's metadata as properties on all of the artifacts (based on the control file's content).

This operation may not always be required (for example, if the ipk files are intact and were not modified, only the index needs to be recalculated. The operation is resource intensive and can be disabled by passing the *?writeProps=0* query param.

Notes: Requires Artifactory Pro.

Security: Up to version 4.8 , requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST api/opkg/reindex/{repoKey} [?async=0/1][?writeProps=0/1]

Headers (Optionally): -H X-GPG-PASSPHRASE:passphrase

Produces: application/text

Since: 4.4

Sample Output:

```
POST /api/opkg/reindex/opkg-local

Recalculating index for Opkg repository opkg-local scheduled to run.
```

Calculate Bower Index

Description: Recalculates the index for a Bower repository.

Notes: Requires Artifactory Pro.

Security: Up to version 4.8 , requires a valid admin user. From version 4.8 only requires the set of permissions assumed by Manage (Manage + Delete/Overwrite + Deploy/Cache + Annotate + Read).

Usage: POST api/bower/{repoKey}/reindex

Produces: application/text

Since: 3.6.0

Sample Output:

```
POST /api/bower/bower-local/reindex
```

```
Bower index for refresh for bower-local has been accepted
```

SYSTEM & CONFIGURATION

System Info

Description: System Info

Get general system information.

Since: 2.2.0

Security: Requires a valid admin user

Usage: GET /api/system

Produces: text/plain

Sample Output:

```
GET /api/system
```

```
system info output text
```

System Health Ping

Description: Get a simple status response about the state of Artifactory

Returns 200 code with an 'OK' text if Artifactory is working properly, if not will return an HTTP error code with a reason.

Since: 2.3.0

Security: Requires a valid user (can be anonymous). If artifactory.ping.allowUnauthenticated=true is set in artifactory.system.properties, then no authentication is required even if anonymous access is disabled.

Usage: GET /api/system/ping

Produces: text/plain

Sample Output:

```
GET /api/system/ping
```

```
OK
```

Verify Connection

Description: Verifies a two-way connection between Artifactory and another product

Returns Success (200) if Artifactory receives a similar success code (200) from the provided endpoint. See possible error codes below.

Since: 4.15.0

Security: Requires an admin user.

Usage: POST /api/system/verifyconnection

Consumes: application/json

```
POST /api/system/verifyconnection
{
  + "endpoint" : "<The URL that Artifactory should connect to>",
  - "username" : "<Username to be used for the connection test>",
  - "password" : "<Password to be used for the connection test>"
}
```

Produces: application/json

Upon success (200):
200 Successfully connected to endpoint

Upon error, returns 400 along with a JSON object that contains the error returned from the other system.

Sample Output:

```
{
  "errors" : [ {
    "status" : 400,
    "message" : "Received error from endpoint url: HTTP 404: Page not
found"
  } ]
}
```

General Configuration

Description: Get the general configuration (artifactory.config.xml).

Since: 2.2.0

Security: Requires a valid admin user

Usage: GET /api/system/configuration

Produces: application/xml (http://www.jfrog.org/xsd/artifactory-v1_7_3.xsd)

Sample Output:

```
GET /api/system/configuration

<artifactory.config.xml/>
```

Save General Configuration

Description: Save the general configuration (artifactory.config.xml).

Since: 2.2.0

Notes: This is an advanced feature - make sure the new configuration is really what you wanted before saving.

Security: Requires a valid admin user

Usage: POST /api/system/configuration

Consumes: application/xml (http://www.jfrog.org/xsd/artifactory-v1_7_3.xsd)

Sample Usage:

```
POST /api/system/configuration

<artifactory.config.xml/>
```

Update Custom URL Base

Description: Changes the Custom URL base

Since: 3.9.0

Security: Requires a valid admin user

Usage: PUT /api/system/configuration/baseUrl

Example: curl -X PUT "http://localhost:8081/artifactory/api/system/configuration/baseUrl" -d 'https://mycompany.com:444/artifactory' -uadmin:password -H "Content-type: text/plain"

Sample Output:

```
URL base has been successfully updated to
"https://mycompany.com:444/artifactory".
```

License Information

Description: Retrieve information about the currently installed license.

Since: 3.3.0

Security: Requires a valid admin user

Usage: GET /api/system/license

Produces: application/json

Sample Output:

```
GET /api/system/license
{
  "type" : "Commercial",
  "validThrough" : "May 15, 2014",
  "licensedTo" : "JFrog inc."
}
```

Install License

Description: Install new license key or change the current one.

Since: 3.3.0

Security: Requires a valid admin user

Usage: POST /api/system/license

Produces: application/json

Consumes: application/json ({ "licenseKey": "your supplied license key ..." })

Sample Output:

```
POST /api/system/license
{
  "status" : 200,
  "message" : "The license has been successfully installed."
}
```

HA License Information

Description: Retrieve information about the currently installed licenses in an HA cluster.

Since: 5.0.0

Security: Requires a valid admin user

Usage: GET /api/system/licenses

Produces: application/json

```
[ {
  "type" : "Enterprise",
  "validThrough" : "<validity date formatted MMM DD, YYYY>",
  "licensedTo" : "<Company name>",
  "licenseHash" : "<license hash code>",
  "nodeId" : "<Node ID of the node activated with this license | Not in use>",
  "nodeUrl" : "<URL of the node activated with this license | Not in use>",
  "expired" : <true | false>
}]
```

Sample Output:

```
GET /api/system/licenses
```

```
[ {
  "type" : "Enterprise",
  "validThrough" : "May 15, 2018",
  "licensedTo" : "JFrog",
  "licenseHash" : "179b7ea384d0c4655a00dfac7285a21d986a17923",
  "nodeId" : "art1",
  "nodeUrl" : "http://10.1.16.83:8091/artifactory",
  "expired" : false
}, {
  "type" : "Enterprise",
  "validThrough" : "May 15, 2018",
  "licensedTo" : "JFrog",
  "licenseHash" : "e10b8aa1d1dc5107439ce43debc6e65dfef71afd3",
  "nodeId" : "Not in use",
  "nodeUrl" : "Not in use",
  "expired" : false
} ]
```

Install HA Cluster Licenses

Description: Install a new license key(s) on an HA cluster.

Since: 5.0.0

Security: Requires an admin user

Usage: POST /api/system/licenses

Consumes: application/json

```
[{
  "licenseKey": "<License key>"
}]
```

Produces: application/json

```
{
  "status" : 200,
  "messages" : {
    ["<License key>" : "<status message>"]
  }
}
```

Sample Usage:

```
POST /api/system/licenses
[
  {
    "licenseKey": "tL9r2Y...lDBiktbbt"
  },
  {
    "licenseKey": "DiYgVA...P7nvyNI7q"
  }
]
```

Response:

```
{
  "status" : 200,
  "messages" : {
    "tL9r2Y...lDBiktbbt" : "OK",
    "DiYgVA...P7nvyNI7q" : "OK",
  }
}
```

Delete HA Cluster License

Description: Deletes a license key from an HA cluster.

Since: 5.0.0

Security: Requires an admin user

Usage: DELETE /api/system/licenses?licenseHash=licenseHash1, licenseHash2...

Produces: application/json

```
{
  "status" : 200,
  "messages" : [{"<License hash code>" : "<status message>"}]
}
```

Sample Usage:

```
DELETE /api/system/licenses?licenseHash=tL9r2YlDBiktbbt, DiYgVAP7nvyNI7q
```

Response:

```
{
  "status" : 200,
  "messages" : {
    "tL9r2YlDBiktbbt" : "OK",
    "DiYgVAP7nvyNI7q" : "OK"
  }
}
```

Version and Add-ons information

Description: Retrieve information about the current Artifactory version, revision, and currently installed Add-ons

Since: 2.2.2

Security: Requires a valid user (can be anonymous)

Usage: GET /api/system/version

Produces: application/vnd.org.jfrog.artifactory.system.Version+json

Sample Output:

```
GET /api/system/version
{
  "version" : "2.2.2",
  "revision" : "10427",
  "addons" : [ "build", "ldap", "properties", "rest", "search", "sso",
    "watch", "webstart" ]
}
```

Get Reverse Proxy Configuration

Description: Retrieves the reverse proxy configuration

Since: 4.3.1

Security: Requires a valid admin user

Usage: GET /api/system/configuration/webServer

Produces: application/json

Sample Output:

```
GET /api/system/configuration/webServer

{
  "key" : "nginx",
  "webServerType" : "NGINX",
  "artifactoryAppContext" : "artifactory",
  "publicAppContext" : "artifactory",
  "serverName" : "jfrog.com",
  "serverNameExpression" : ".*jfrog.com",
  "artifactoryServerName" : "localhost",
  "artifactoryPort" : 8081,
  "sslCertificate" : "/etc/ssl/myKey.cert",
  "sslKey" : "/etc/ssl/myKey.key",
  "dockerReverseProxyMethod" : "SUBDOMAIN",
  "useHttps" : true,
  "useHttp" : true,
  "sslPort" : 443,
  "httpPort" : 76
}
```

Update Reverse Proxy Configuration

Description: Updates the reverse proxy configuration

Since: 4.3.1

Security: Requires an admin user

Usage: POST /api/system/configuration/webServer

Consumes: application/json

Sample Usage:

```
POST /api/system/configuration/webServer

{
  "key" : "nginx",
  "webServerType" : "NGINX",
  "artifactoryAppContext" : "artifactory",
  "publicAppContext" : "artifactory",
  "serverName" : "jfrog.com",
  "serverNameExpression" : ".*jfrog.com",
  "artifactoryServerName" : "localhost",
  "artifactoryPort" : 8081,
  "sslCertificate" : "/etc/ssl/myKey.cert",
  "sslKey" : "/etc/ssl/myKey.key",
  "dockerReverseProxyMethod" : "SUBDOMAIN",
  "useHttps" : true,
  "useHttp" : true,
  "sslPort" : 443,
  "httpPort" : 76
}
```

Get Reverse Proxy Snippet

Description: Gets the reverse proxy configuration snippet in text format

Since: 4.3.1

Security: Requires a valid user (not anonymous)

Usage: GET /api/system/configuration/reverseProxy/nginx

Produces: text/plain

Sample Usage:

```
GET /api/system/configuration/reverseProxy/nginx

## add ssl entries when https has been set in config
ssl_certificate      /etc/ssl/myKey.cert;
ssl_certificate_key  /etc/ssl/myKey.key;
ssl_session_cache   shared:SSL:1m;
ssl_prefer_server_ciphers   on;
## server configuration
server {
    listen 443 ssl;
    listen 76 ;
    server_name ~(?<repo>.+)\.jfrog.com jfrog.com;

    if ($http_x_forwarded_proto = '') {
        set $http_x_forwarded_proto $scheme;
    }
    ## Application specific logs
    ## access_log /var/log/nginx/jfrog.com-access.log timing;
    ## error_log /var/log/nginx/jfrog.com-error.log;
    rewrite ^/$ /artifactory/webapp/ redirect;
    rewrite ^/artifactory$ /artifactory/webapp/ redirect;
}
```

Create Bootstrap Bundle

Description: This rest is relevant for High Availability set up. It will create a bootstrap bundle on the primary node of an Artifactory HA installation that will include all the relevant keys so a new node can access the database and fetch all the relevant configuration files. The same bundle must be installed on all nodes during an installation of new nodes or if upgrading from a version older than 5.0. For more details, please refer to [Installing Artifactory HA](#).

Since: 5.0.0

Security: Requires an admin user

Usage: POST /api/system/bootstrap_bundle

Produces: application/json

```
{
  "file" : "<Location on primary node where bootstrap bundle was created>"
}
```

Sample usage:

```
POST /api/system/bootstrap_bundle

{
  "file" : "/opt/jfrog/artifactory/etc/bootstrap.bundle.tar.gz"
}
```

PLUGINS

Execute Plugin Code

Description: Executes a named execution closure found in the `executions` section of a **user plugin**.

Execution can take parameters and be synchronous (the default) or asynchronous.

When parameters can have multiple values, you can separate the items in one of the following ways:

- Use a semicolon - ; (recommended)
 - Use the encoding for the pipe ("|") character - %7C
- Alternatively, you may configure your NGINX to encode URLs so that if an unencoded pipe is used in the URL, NGINX will encode it to %7C. We recommend that you verify that this configuration does not break any other systems served by NGINX

Since: 2.3.1

Notes: Requires Artifactory Pro

Security: Requires an authenticated user (the plugin can control which users/groups are allowed to trigger it)

Usage: POST /api/plugins/execute/{executionName}?[params=p1=v1[,v2][[p2=v3][&async=1]]

Produces: text/plain

Sample Output:

```
POST
/api/plugins/execute/cleanup?params=suffix=SNAPSHOT;types=jar,war,zip&asyn
c=1
OK
```

Retrieve Plugin Info

Description: Retrieves **user plugin** information for **Executions** and **Staging** plugins (subject to the permissions of the provided credentials).

Since: 2.5.2

Notes: Requires Artifactory Pro

Security: Requires an authenticated user.

Usage: GET /api/plugins

Produces: application/json

Sample Output:


```
GET /api/plugins
{
  "executions": [
    {
      "name": "execution1",
      "version": "version",
      "description": "description",
      "users": ["user1"],
      "groups": ["group1", "group2"],
      "params": {}
    }
  ],
  "staging": [
    {
      "name": "strategy1",
      "version": "1.0",
      "description": "desc",
      "params": {"key1": "vall"}
    }
  ]
}
```

Retrieve Plugin Info Of A Certain Type

Description: Retrieves all available **user plugin** information (subject to the permissions of the provided credentials) of the specified type.

Since: 2.5.2

Notes: Requires Artifactory Pro

Security: Requires an authenticated user.

Usage: GET /api/plugins/{pluginType}

Produces: application/json

Sample Output:

```
GET /api/plugins/staging
{
  "staging": [
    {
      "name": "strategy1",
      "version": "1.0",
      "description": "desc",
      "params": {"key1": "vall"}
    }
  ]
}
```

Retrieve Build Staging Strategy

Description: Retrieves a build staging strategy defined by a **user plugin**.

When passing in parameters that may take multiple values, you can separate the items in one of the following ways:

- Use a semicolon - ; (recommended)
- Use the encoding for the pipe ("|") character - %7C
Alternatively, you may configure your NGINX to encode URLs so that if an unencoded pipe is used in the URL, NGINX will encode it to %7C. We recommend that you verify that this configuration does not break any other systems served by NGINX

Since: 2.5.2

Notes: Requires Artifactory Pro

Security: Requires an authenticated user.

Usage: GET /api/plugins/build/staging/{strategyName}?buildName={buildName}&[params=p1=v1[,v2][[p2=v3]]

Produces: application/vnd.org.jfrog.plugins.BuildStagingStrategy

Sample Output:

```
GET
/api/plugins/build/staging/strategy1?buildName=build1&params=types=jar,war,zip
{
  "defaultModuleVersion":
  {
    "moduleId": "moduleId",
    "nextRelease": "nextRelease",
    "nextDevelopment": "nextDevelopment"
  },
  "vcsConfig":
  {
    "useReleaseBranch": true,
    "releaseBranchName": "branchName",
    "createTag": true,
    "tagUrlOrName": "tagUrl",
    "tagComment": "comment",
    "nextDevelopmentVersionComment": "comment"
  },
  "promotionConfig":
  {
    "targetRepository": "repoKey",
    "comment": "comment",
    "status": "statusName"
  }
}
```

Execute Build Promotion

Description: Executes a named promotion closure found in the `promotions` section of a [user plugin](#).

Since: 2.5.2

Notes: Requires Artifactory Pro

Security: Requires an authenticated user.

Usage: POST /api/plugins/build/promote/{promotionName}/{buildName}/{buildNumber}?[params=p1=v1[,v2][[p2=v3]]

Produces: text/plain

Sample Output:

```
POST
/api/plugins/build/promote/promotion1/build1/3?params=types=jar,war,zip
OK
```

Reload Plugins

Description: Reloads user plugins if there are modifications since the last user plugins reload. Works regardless of the automatic user plugins refresh interval.

Since: 2.9.0

Notes: Requires Artifactory Pro

Security: Requires a valid admin user

Usage: POST /api/plugins/reload

Produces: text/plain

Sample Output:

```
POST /api/plugins/reload
Successfully loaded:  myplugin1.groovy, myplugin2.groovy
```

IMPORT & EXPORT

Import Repository Content

Description: Import one or more repositories.

Since: 2.2.2

Security: Requires a valid admin user

Usage: POST: /api/import/repositories

Requests Params:

path - The file system path to import from. This may point to a specific folder to import data for a single repository, or to the parent "repositories" folder to import data for all repositories.

repo - Empty/null repo -> all

metadata - Include metadata - default 1

verbose - Verbose - default 0

Produces: text/plain

Sample Output:

```
POST: /api/import/repositories?path=pathToRepos&verbose=1
```

Import System Settings Example

Description: Returned default Import Settings JSON.

Since: 2.4.0

Security: Requires a valid admin user

Usage: GET: /api/import/system

Produces: [application/vnd.org.jfrog.artifactory.system.ImportSettings+json](#)

Sample Usage:

```
GET /api/import/system
{
  "importPath" : "/import/path",
  "includeMetadata" : true,
  "verbose" : false,
  "failOnError" : true,
  "failIfEmpty" : true
}
```

Full System Import

Description: Import full system from a server local Artifactory export directory.

Since: 2.4.0

Security: Requires a valid admin user

Usage: POST: /api/import/system

Consumes : [application/vnd.org.jfrog.artifactory.system.ImportSettings+json](#)

Produces: text/plain

Sample Usage:

```
POST /api/import/system
{
  import-settings.json
}
```

Export System Settings Example

Description: Returned default Export Settings JSON.

Since: 2.4.0

Security: Requires a valid admin user

Usage: GET: /api/export/system

Produces: [application/vnd.org.jfrog.artifactory.system.ExportSettings+json](#)

Sample Usage:

```
GET /api/export/system
{
  "exportPath" : "/export/path",
  "includeMetadata" : true,
  "createArchive" : false,
  "bypassFiltering" : false,
  "verbose" : false,
  "failOnError" : true,
  "failIfEmpty" : true,
  "m2" : false,
  "incremental" : false,
  "excludeContent" : false
}
```

Export System

Description: Export full system to a server local directory.

Since: 2.4.0

Security: Requires a valid admin user

Usage: POST: /api/export/system

Consumes : [application/vnd.org.jfrog.artifactory.system.ExportSettings+json](#), [application/json](#)

Produces: text/plain

Sample Usage:

```
POST /api/export/system{ export-settings.json }
```

SUPPORT

Create Bundle

Description: Create a new support bundle.

Since: 4.3.0

Security: Requires an admin user

Notes: All bundle items are optional.

Usage: POST /api/support/bundles/

Sample Usage:

```

POST /api/support/bundles
{
  - "systemLogsConfiguration" : {
    "enabled" : true,(default)
    "startDate" : {date-in-millis},
    "endDate" : {date-in-millis}
  },
  - "systemInfoConfiguration" : {
    "enabled" : true (default)
  },
  - "securityInfoConfiguration" : {
    "enabled" : true, (default)
    "hideUserDetails" : true (default)
  },
  - "configDescriptorConfiguration" : {
    "enabled" : true, (default)
    "hideUserDetails" : true (default)
  },
  - "configFilesConfiguration" : {
    "enabled" : true (default),
    "hideUserDetails" : true (default)
  },
  - "storageSummaryConfiguration" : {
    "enabled" : true (default)
  },
  - "threadDumpConfiguration" : {
    "enabled" : true, (default)
    "count" : {amount-of-dumps},          (1 is default)
    "interval" : {interval in millis} (0 is default)
  }
}

```

NOTE: systemLogsConfiguration parameter can also be expressed as number of days as follows:

```

...
- "systemLogsConfiguration" : {
  "enabled" : true,(default)
  "daysCount" : {number-of-days}
},
...

{
  "bundles" : [
    "/api/support/bundles/support-bundle-20151118-1709272-1447859367247.zip"
  ]
}

```

List Bundles

Description: Lists previously created bundle currently stored in the system.

Since: 4.3.0

Security: Requires a privileged user (Admin only)

Usage: GET /api/support/bundles/

Produces: application/json

Sample Usage:

```
GET /api/support/bundles

{
  "bundles" : [
    "/api/support/bundles/support-bundle-20151118-1709272-1447859367247.zip",
    "/api/support/bundles/support-bundle-20151117-1035500-1447749350025.zip",
    "/api/support/bundles/support-bundle-20151117-1035147-1447749314704.zip"
  ]
}
```

Get Bundle

Description: Downloads a previously created bundle currently stored in the system.

Since: 4.3.0

Security: Requires a privileged user (Admin only)

Usage: GET /api/support/bundles/{bundle-name}

Produces: application/json

Sample Usage:

```
GET /api/support/bundles/support-bundle-20151122-1705472-1448211947280.zip
```

Delete Bundle

Description: Deletes a previously created bundle from the system.

Since: 4.3.0

Security: Requires a privileged user (Admin only)

Usage: DELETE /api/support/bundles/{bundle-name}

Produces: application/json

Sample Usage:

```
DELETE
/api/support/bundles/support-bundle-20151122-1705472-1448211947280.zip
```

ERROR RESPONSES

In case of an error, Artifactory will return an error response in JSON format. The response contains the HTTP status code and error message.

For example, a badly formatted API call would return the "404, File not found" response below:

```
{
  "errors" : [ {
    "status" : 404,
    "message" : "File not found."
  } ]
}
```

Sample input:

```
POST /api/security/apiKey
{
  "apiKey": "30loposOtVFyCMrT+cXmCAscmVMPrSYXkWIjIyDCXsY="
}
```

```
PUT /api/storage/libs-release-local/ch/qos/logback/logback-classic/0.9.9?properties=os=win,linux;qa=done&recursive=1
```

When parameters can have multiple values, you can separate the items in one of the following ways:

- Use a semicolon - ; (recommended)
 - Use the encoding for the pipe ("|") character - %7C
- Alternatively, you may configure your NGINX to encode URLs so that if an unencoded pipe is used in the URL, NGINX will encode it to %7C. We recommend that you verify that this configuration does not break any other systems served by NGINX