

Package Management

Overview

The JFrog Platform brings the universal nature of Artifactory to full force with advanced package management for all major packaging formats in use today. As the only repository with a unique architecture that includes a filestore layer and a separate database layer, Artifactory is the only repository manager that can natively support current package formats as well as any new format that may arise from time to time.

With a paradigm of single-type-repositories, all repositories are assigned a type upon creation allowing efficient indexing to allow any client or dependency manager to work directly with Artifactory transparently as its natural repository.

The Packages view in the Application module provides easy access to information about all the packages in your repositories and supports:

- [Inspecting Packages](#)
- [Viewing Package Information](#)
- [Viewing Package Version Information](#)
- [Downloading Package Versions](#)
- [Adding Packages to Project](#)

Page Contents

- [Overview](#)
- [Supported Package Types](#)
- [Inspecting Packages](#)
 - [Filtering the Package List](#)
- [Viewing Package Information](#)
 - [Viewing Xray Data on Packages](#)
 - [Xray scanning requires Pro X, Enterprise with Xray, or an Enterprise+ license.](#)
- [Viewing Package Version Information](#)
 - [Viewing Xray Data on Package Versions](#)
- [Downloading Package Versions](#)
- [Adding Packages to Projects](#)

Supported Package Types

The JFrog Platform supports the following package formats with new formats added regularly as the need arises.

Alpine Linux	Use Artifactory to gain full control of your deployment and resolution process of Alpine Linux (* . apk) packages.
Bower	Boost your front end development by hosting your own Bower components and proxying the Bower registry in Artifactory.
Cargo	Enhance your capabilities for configuration management with Cargo using all the benefits of a repository manager.
Chef	Enhance your capabilities for configuration management with Chef using all the benefits of a repository manager.
CocoaPods	Speed up development with Xcode and CocoaPods with fully fledged CocoaPods repositories.
Conan	Artifactory is the only secure, private repository for C/C++ packages with fine-grained access control.
Conda	Artifactory natively supports Conda repositories for Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN.
CRAN	Deploy and resolve CRAN packages for the R language using dedicated CRAN repositories.
Debian	Host and provision Debian packages complete with GPG signatures.
Docker	Host your own secure private Docker registries and proxy external Docker registries such as Docker Hub.
Git LFS	Optimize your workflow when working with large media files and other binary resources.
Go Registry	Build Go projects while resolving dependencies through Artifactory, and then publish the resulting Go packages into a secure, private Go registry
Gradle	Resolve dependencies from and deploy build output to Gradle repositories when running Gradle builds.
Helm	Manage your Helm Charts in Artifactory and gain control over deployments to your Kubernetes cluster.
Maven	Artifactory is both a source for Maven artifacts needed for a build, and a target to deploy artifacts generated in the build process.
npm	Host your own node.js packages, and proxy remote npm repositories like npmjs.org through Artifactory.

NuGet	Host and proxy NuGet packages in Artifactory, and pull libraries from Artifactory into your various Visual Studio .NET applications.
Opkg	Optimize your work with OpenWrt using Opkg repositories. Proxy the official OpenWrt repository and cache remote <i>.ipk</i> files.
P2	Proxy and host all your Eclipse plugins via an Artifactory P2 repository, allowing users to have a single-access-point for all Eclipse updates.
PHP Composer	Provision Composer packages from Artifactory to the Composer command line tool, and access Packagist and other remote Composer metadata repositories.
Pub Repositories	Artifactory natively supports Dart packages, giving you full control of your deployment and resolution process of Flutter , Angular Dart , and general Dart programs.
Puppet	Configuration management meets repository management with Puppet repositories in Artifactory.
PyPI	Host and proxy PyPI distributions with full support for pip.
RPM	Distribute RPMs directly from your Artifactory server, acting as fully-featured YUM repository.
RubyGems	Use Artifactory to host your own gems and proxy remote gem repositories like rubygems.org .
SBT	Resolve dependencies from and deploy build output to SBT repositories when running SBT builds.
Swift	Artifactory natively supports a dedicated Swift registry, giving you full control of your deployment and resolution process of your Swift packages and the dependencies.
Terraform	A fully fledged Terraform repository solution giving your full control of your deployment and resolve process of Terraform Modules, Providers, and Backend packages.
Vagrant	Securely host your Vagrant boxes in local repositories.
VCS	Consume source files packaged as binaries.

Inspecting Packages

The Packages page provides easy access to information about all the packages in your repositories.

You have quick access to the most important summary information about the latest package versions and you can easily drill down for more details about previous versions. Filters and sorting features are available for your convenience, as well as cross-reference links to the Builds and Artifacts pages.

For some package types, you can download packages and copy installation commands when drilling down into a package.

To view information about packages, from the **Application** module, go to **Artifactory | Packages**.

Packages		Modified	↓	
 com.vladsch.flexmark:flexmark-ext-gfm-tables 15-12-19 14:45:06 +0200 Last version: 0.18.4	 BSD 2-Clause License	 1 Versions	 Scanned Xray	 0 Downloads
 cybertec/pgwatch2 15-12-19 06:29:47 +0200 Last version: latest		 1 Versions	 Scanned Xray	 0 Downloads
 commons-lang:commons-lang 15-12-19 06:29:44 +0200 Last version: 2.6	 Apache-2.0 License	 4 Versions	 Scanned Xray	 0 Downloads

Filtering the Package List

Initially, by default, each panel contains information about the last version of the package. In the initial view, the list includes all the available package types, sorted by lexical order according to the package name by default, in descending order. The user can sort and filter the list. The user's new sort and filter setting becomes the new default.

To change the sort criteria, click the drop-down arrow and select one of the following sort options:

- **Name:** Name of package
- **Downloads:** Number times package was downloaded

To toggle the sort order, click the arrow to the right of the sort option list.

Packages		Modified	↑	↓
 org.webjars:webjars-locator 17-12-19 05:30:33 +0200 Last version: 0.28	 MIT License	 6 Versions	 Scanned Xray	 0 Downloads
		<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <ul style="list-style-type: none"> Modified Name Downloads Created </div>		

Viewing Package Information

In the Packages list, the package summary information is displayed, with the package name and logo in the left top corner, and the creation date of the latest version and its version number. The following information is displayed in the upper right of the panel.

License	Name of license covering the package
----------------	--------------------------------------

Versions	Number of versions of the package
Xray	Indicates the status of the Xray scan. For more information, see Xray Security and Compliance . <i>Xray scanning requires Pro X, Enterprise with Xray, or an Enterprise+ license.</i>
Downloads	Total number of times the package (in its various versions) has been downloaded
Tags	Metadata tags (available only for npm and NuGet)

Click on a Package to view the Package versions.

In the **Versions** section, use the **View By** toggle to select one of the following views:

- **List:** Displays information about the package versions.
- **Graph:** Displays security and license violations informations from JFrog Xray with the number of downloads per version.
For more information, see [Xray Security and Compliance](#).
Xray scanning requires Pro X, Enterprise with Xray, or an Enterprise+ license.

Packages > org.slf4j:jcl-over-slf4j

org.slf4j:jcl-over-slf4j

15-12-19 14:47:22 +0200 Last version: 1.7.29

Apache-2.0
License

3
Versions

Scanned
Xray

0
Downloads

Versions

Filter

View By: List Graph

Version	Repositories	Created	Downloads	Xray Status
1.7.29	1 jcenter-cache	12-12-19 15:1...	0	No Vulnerabilities
1.7.25	1 jcenter-cache	12-12-19 15:1...	0	Not Scanned
1.5.6	1 jcenter-cache	12-12-19 14:4...	0	No Vulnerabilities

The **List** option displays the following information about the package versions:

Version	Package version numbers
Repositories	Name of repositories that contain the package version
Digest	The package's SHA 256 digest (available only for Docker)
Last Modified	Date when the package version was last modified
Downloads	Number of times package version was downloaded

Xray Status

The following Xray status indicators are displayed:

- Severity of the package vulnerability (Low / Medium / High)
- Not Scanned
- No Vulnerabilities
- Pending Scan

For more information, see [Xray Security and Compliance](#).

Xray scanning requires Pro X, Enterprise with Xray, or an Enterprise+ license.

NPM Packages Only

For npm package types, the `> npm i angular` appears to the right of the package name. For details, see [Adding Packages to Projects](#).

Viewing Xray Data on Packages

Required JFrog Subscriptions



Xray scanning requires Pro X, Enterprise with Xray, or an Enterprise+ license.

In the Package list view, you can quickly and periodically review the status of your security and compliance for all your scanned packages on your indexed resources to gain information about the Xray scan status and assigned licences on the latest version of the package.

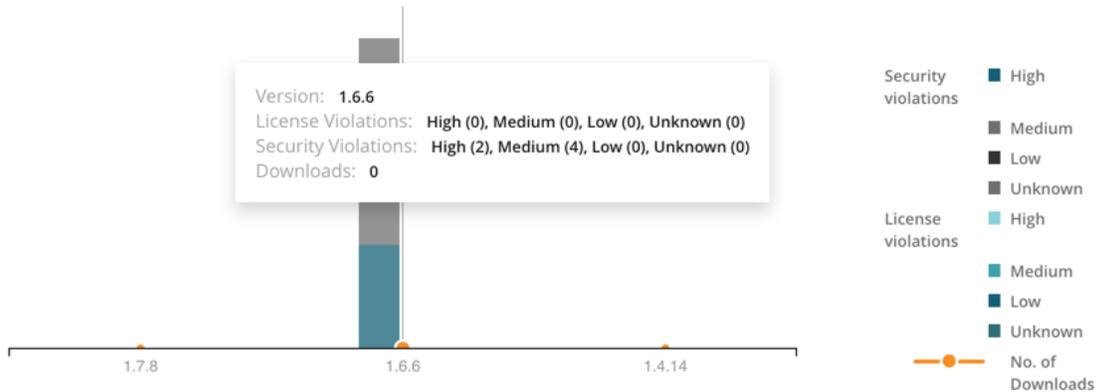
Package Name	Version	License	Scan Status	Downloads
angular	1.7.8	MIT License	Scanned Xray	0
jenkins	latest		Scanned Xray	0

From the list view, you can toggle to the Graph tab to view a graph displaying a breakdown according of security or license violations according to severity.

angular browser framework client-side

Versions

View By:



Viewing Package Version Information

Click on the version number to view details about a particular package version, in the detailed table.

The information in the summary section, in the top panel, now displays summary information about the selected package version.

To download the package version to your computer, click **Download**, located on the right, below the summary information. For more information, see [Downloading Package Versions](#).

The detailed table now appears with the following tabs and information:

Readme

Applies to npm packages. Contains readme documentation.

Packages > angular > 1.7.8

npm angular / 1.7.8 `> npm i angular@1.7.8`
 04-11-19 14:18:00 +0200 'PM4th18 o11/4/20191'

html enhanced for web apps

No Vulnerabilities Xray Severity MIT License 0 Downloads

Download

Readme Builds Xray Data >

packaged angular

This repo is for distribution on `npm` and `bower`. The source for this module is in the [main AngularJS repo](#). Please file issues and pull requests against that repo.

Install

You can install this package either with `npm` or with `bower`.

Builds

In the **Build** section, use the **View** toggle to select one of the following views:

- **Produced By:** Displays information about the builds that produced the package versions.
- **Used By:** Displays information about the builds that used the package versions as dependencies.

The information includes the name, number, and creation date of each build. Click on the build name to open the Build page with the full information about the build.

Packages > angular > 1.7.8

npm angular / 1.7.8 `> npm i angular@1.7.8`
 04-11-19 14:18:00 +0200 'PM4th18 o11/4/20191'

html enhanced for web apps

No Vulnerabilities Xray Severity MIT License 0 Downloads

Download

Readme **Builds** Xray Data >

View: **Produced By** (12) | Used By

Build Name ^	Build Number	Creation Date
testim-io-build	132	17-08-19 11:29:13 +0300 'AM17th29 o8...
testim-io-build	139	21-08-19 11:29:09 +0300 'AM21st29 o8/...
testim-io-build	144	26-08-19 11:29:08 +0300 'AM26th29 o8...

Xray Data

Xray scanning requires Pro X, Enterprise with Xray, or an Enterprise+ license.

For more information, see [Viewing Xray data on Package Versions](#).

Docker Layers

Applies to Docker packages. Lists the layer related information.

The screenshot shows a navigation menu with 'Builds', 'Xray Data', 'Distribution', and 'Docker Layers' (highlighted in green). Below the menu, there are two expandable sections: 'COPY' (with a 'latest' button) and 'ENTRYPOINT'. The 'COPY' section is expanded, showing a list of layers with their SHA hashes and file paths. The first layer has a SHA hash of sha256:12b47c68955c9d18c7e2058d3d349e70b600cee9fa38665e0fe2fdaacdd929a2 and a file path of file:9f0a7faf8951842e0f42c1a3f3bb54ff4ec5263064508077347c57376da68b46 in /usr/local/bin/plugins.sh. The second layer has a timestamp of 2018-07-17 16:20:21 and a size of 1.6 kB.

Distribution

Requires an Enterprise+ license.

Displays the Release Bundles containing the package version, the Release Bundle Distribution status and when they were last updated. Click the Release Bundle Name to view the Bundle in the **Distribution** page.

The screenshot shows the 'Distribution' section for the package 'angular / 1.7.8'. At the top, there is a terminal snippet showing the command 'npm i angular@1.7.8' and its output '04-11-19 14:18:00 +0200 'PM4th18 o11/4/20191''. To the right, there are three icons: 'No Vulnerabilities Xray Severity', 'MIT License', and '0 Downloads'. Below this, there is a 'Download' button. The main content is a table with columns: 'Name', 'Version', 'Distribution Status', and 'Last update'. The table has two rows of data.

Name ^	Version	Distribution Status	Last update
asd	asd	READY_FOR_DISTRIBUTION	03-10-19 18:28:08 +0300 'P...
test-bundle-xray	Copy_of_1.0	READY_FOR_DISTRIBUTION	23-10-19 14:50:08 +0300 'P...

Repositories

Displays where the package versions exist in Artifactory. The locations are indicated by the repository names and the full paths to the packages in Artifactory. Enter version numbers or repository names to filter the list.

Click on the path to open the Artifact Repository Browser, showing the location of the package in the Tree view.

Packages > angular-ui-bootstrap > 0.13.1

npm angular-ui-bootstrap / 0.13.1

26-05-18 19:48:09 +03:00 [Download](#) `npm i angular-ui-boots...`

bootstrap widgets for angular

No Vulnerabilities
Xray Severity

MIT
License

2.0
Downloads

Readme Builds Xray Data **Repositories** >

1 Item

Filter

Repository ^	Path
npm-remote-cache	npm-remote-cache/angular-ui-bootstrap/-/angular-ui-bootstrap-0.13.1.tgz

Viewing Xray Data on Package Versions

Selecting a package version displays detailed Xray data information.

In the top pane, you can view the Xray severity and license assigned to the version.

Under the **Xray Data** tab, you can view these dedicated Xray related tabs with the option to run a set of actions on the version. For detailed information on each tab, see [Analyzing Resource Scan Results](#).

Packages > angular > 1.6.6

npm angular / 1.6.6

13-08-19 19:06:27 +0300 'PM13th6 o8/13/20192'

html enhanced for web apps

`> npm i angular@1.6.6`

High
Xray Severity

MIT
License

0
Downloads

[Download](#)

Readme Builds **Xray Data** Distribution >

Violations (6) Security (3) Licenses (1) Descendants Ancestors Actions ~

Under the **Xray Data** tab, you can view these dedicated Xray related tabs with the option to run a set of actions on the version. For detailed information on each tab, see [Analyzing Resource Scan Results](#).

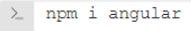
Downloading Package Versions

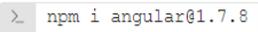
To download a package to your computer from the version-level information page, select the version and click **Download**, located on the right below the summary information.

Adding Packages to Projects

 Only available for npm packages.

It is usually more convenient to use the copy command button than using the **Download** button.

To add the latest version of package to a project, click  `> npm i angular`. The command displayed in the text box is copied to the clipboard. Paste the command into the command line on your terminal. Execute the command line to automatically add the latest version of the package to the package.json file.

When the version-level information is displayed, select a specific version and click  `> npm i angular@1.7.8` to copy the command for the selected version to the clipboard. Continue as described above to add the version of the package to the package.json file.