# Repository Management with Nexus

## Configure maven to use nexus

### Introduction

To use Nexus, you will configure Maven to check Nexus instead of the public repositories. To do this, you’ll need to edit your mirror settings in your ~/.m2/settings.xml file. First, we’re going to demonstrate how to configure Maven to consult your Nexus installation instead of retrieving artifacts directly from the Maven Central repository. After we override the central repository and demonstrate that Nexus is working, we’ll circle back to provide a more sensible set of settings that will cover both releases and snapshots.

### Configuring Maven to Use a Single Nexus Group

If you are adopting Nexus for internal development you should configure a single Nexus group which contains both releases and snapshots. To do this, add snapshot repositories to your public group, and add the following mirror configuration to your Maven settings in ~/.m2/settings.xml.

**Configuring Maven to Use a Single Nexus Group.**

<settings>

<mirrors>

<mirror>

<!--This sends everything else to /public -->

<id>nexus</id>

<mirrorOf>\*</mirrorOf>

<url>http://localhost:8081/nexus/content/groups/public</url>

</mirror>

</mirrors>

<profiles>

<profile>

<id>nexus</id>

<!--Enable snapshots for the built in central repo to direct -->

<!--all requests to nexus via the mirror -->

<repositories>

<repository>

<id>central</id>

<url>http://central</url>

<releases><enabled>true</enabled></releases>

<snapshots><enabled>true</enabled></snapshots>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>central</id>

<url>http://central</url>

<releases><enabled>true</enabled></releases>

<snapshots><enabled>true</enabled></snapshots>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

<activeProfiles>

<!--make the profile active all the time -->

<activeProfile>nexus</activeProfile>

</activeProfiles>

</settings>

In [Configuring Maven to Use a Single Nexus Group](http://books.sonatype.com/nexus-book/reference/maven-sect-single-group.html#ex-maven-nexus-simple) we have defined a single profile: nexus. It configures a repository and a pluginRepository with the id "central" that overrides the same repositories in the super pom. The super pom is internal to every Apache Maven install and establishes default values. These overrides are important since they change the repositories by enabling snapshots and replacing the URL with a bogus URL. This URL is overridden by the mirror setting in the same settings.xml file to point to the URL of your single Nexus group. This Nexus group can therefore contain release as well as snapshot artifacts and Maven will pick them up.

The mirrorOf pattern of \* causes any repository request to be redirected to this mirror and therefore to your single repository group, which in the example is the public group.

It is possible to use other patterns in the mirrorOf field. A possible valuable setting is to use "external:\*". This matches all repositories expect those using localhost or file based repositories. This is used in conjunction with a repository manager when you want to exclude redirecting repositories that are defined for integration testing. The integration test runs for Apache Maven itself require this setting.

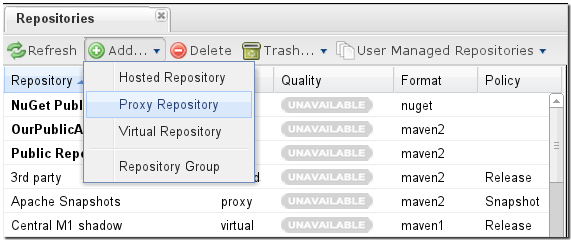
### Adding Custom Repositories for Missing Dependencies

If you’ve configured your Maven settings.xml to list the Nexus public group as a mirror for all repositories, you might encounter projects which are unable to retrieve artifacts from your local Nexus installation. This usually happens because you are trying to build a project which has defined a custom set of repositories and snapshotRepositories in a pom.xml. When you encounter a project which contains a custom repository element in a pom.xml add this repository to Nexus as a new proxy repository and then add the new proxy repository to the public group.

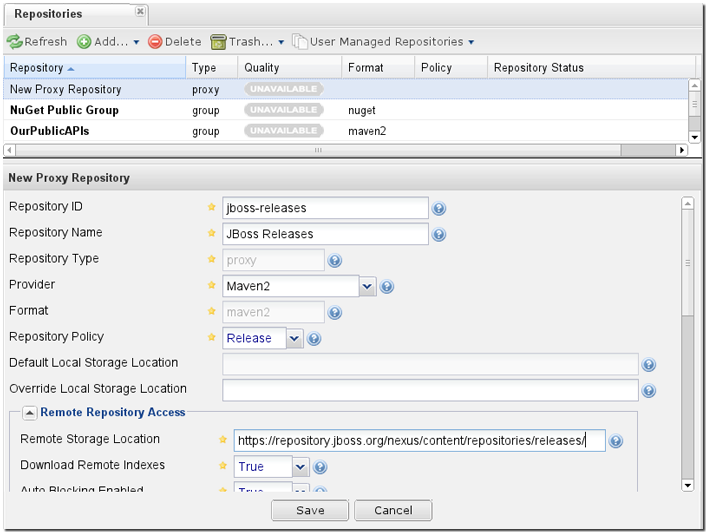
### Adding a New Repository

To add a repository, log into Nexus as an Administrator, and click on the Repositories link in the left-hand navigation menu in the Views/Repositories section as displayed in

Clicking on this link should bring up a window that lists all of the repositories which Nexus knows about. You’ll then want to create a new proxy repository. To do this, click on the Add link that is directly above the list of repositories. When you click the Add button, click the down arrow directly to the right of the word Add, this will show a drop-down which has the options: Hosted Repository, Proxy Repository, Virtual Repository, and Repository Group. Since you are creating a proxy repository, click on Proxy Repository.



Once you do this, you will see a screen resembling [Figure 4.2, “Configuring a Proxy Repository”](http://books.sonatype.com/nexus-book/reference/config-sect-new-repo.html#fig-add-repo). Populate the required fields Repository ID and the Repository Name. The Repository ID will be part of the URL used to access the repository, so it is recommended to avoid characters that could cause problems. Set the Repository Policy to "Release", and the Remote Storage Location to the public URL of the repository you want to proxy.

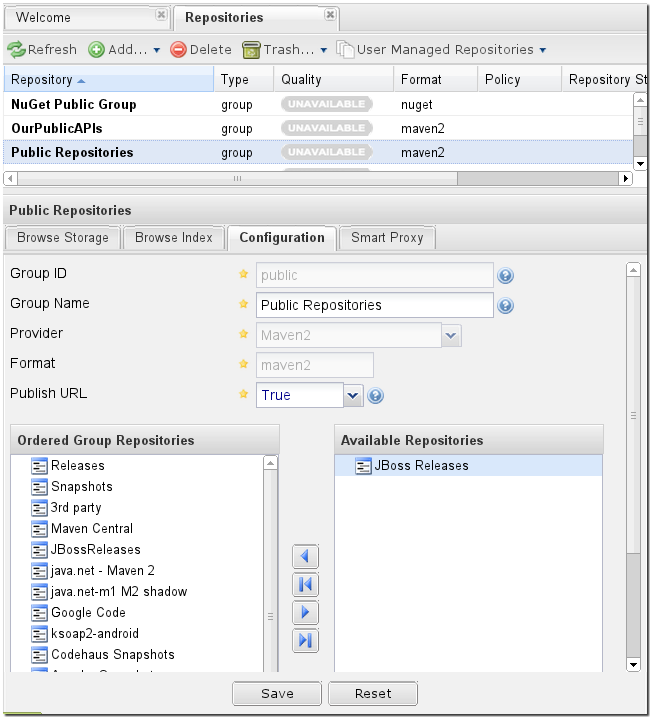


Once you’ve filled out this screen, click on the Save button. Nexus will then be configured to proxy the repository.

### Adding a Repository to a Group

Next you will need to add the new repository to the Public Repositories Nexus repository group. To do this, click on the Repositories link in the left-hand Nexus menu in the Views/Repositories section. Nexus lists Groups and Repositories in the same list so click on the public group. After clicking on the Public Repositories group, you should see the Browse and Configuration tabs in the lower half of the Nexus window.

If you click on a repository or a group in the Repositories list and you do not see the Configuration tab, this is because your Nexus user does not have administrative privileges. To perform the configuration tasks outlined in this chapter, you will need to be logged in as a user with administrative privileges. f Clicking on the Configuration tab will bring up a screen which looks like



To add the new repository to the public group, find the repository in the Available Repositories list on the right, click on the repository you want to add and drag it to the left to the Ordered Group Repositories list. Once the repository is in the Ordered Group Repositories list you can click and drag the repository within that list to alter the order in which a repository will be searched for a matching artifact.

If you were not using a repository manager, you would have added these repositories to the repository element of your project’s POM, or you would have asked all of your developers to modify ~/.m2/settings.xml to reference two new repositories. Instead, you used the Nexus repository manager to add the two repositories to the public group. If all of the developers are configured to point to the public group in Nexus, you can freely swap in new repositories without asking your developers to change local configuration, and you’ve gained a certain amount of control over which repositories are made available to your development team.