

# Write Your Own Connector from Scratch

The first step before writing your own connector is to visit [Connectors](#) and making sure there isn't a connector that already does what you need.

As with everything in the Aspire ecosystem, you'll need to develop the connector in Java, using Maven as the dependency and deploy manager, so make sure you already have everything set up.

## Create a Java Project Using the Connector Archetype

We have wrapped an example connector project into a Maven Archetype to make it easier for developers to start. It contains a basic FileSystem Connector implementation and lots of comments to guide you through your development.

Let's start!

### 1. Execute the archetype command

Open your command line terminal and execute:

```
> mvn archetype:generate -DarchetypeGroupId=com.searchtechnologies.aspire -DarchetypeArtifactId=aspire-connector-archetype -DarchetypeVersion=3.1 -DrepositoryId=stPublic
```

You will be prompted to enter:

- **artifactId:** This is the name for your connector. Typically we use the name of the repository you are trying to extract information from. All sub project names and classes will be based on this, and it is time consuming to change later, so you may wish to pause for a moment to think about it, the standard format is **aspire-{repository}-source**, so for example if you are writing a connector for Sharepoint, it would be "**aspire-sharepoint-source**"
- **groupId:** This is the name of the organizational unit to which your connector belongs. It defaults to **com.searchtechnologies.aspire**
- **version:** This will be the version number of Aspire that you wish this connector to compile against. All Aspire dependencies will be at this version and the connector will be created with this version.
- **repositoryType:** This will be the prefix the Archetype will use to name the Java classes

### 2. Import the project into Eclipse

From the previous step you now should be able to see a folder created with the "artifactId" you entered as its name. Now let's import that project into Eclipse:

- a. Open Eclipse
- b. Go to File > Import
- c. Select Maven > Existing Maven Projects
- d. In the popup window browse and select your project in the "Root Directory" field, then Click on Finish

### 3. Look at the example code

Now open the source in Eclipse and look through all of it. The principal class to look for is the RAP one (the implementation of the RepositoryAccessProvider interface) which is the entry point from Aspire in order to crawl a repository. The code implemented there is a simple FileSystem Scanner which scans recursively from a start root folder.

### 4. Compile and Test

Now it is time to compile and see the connector working in Aspire

- a. Compile the code and run the unit tests by opening the Command Line into your project folder and executing:

```
> mvn clean install
```

- b. Start Aspire
- c. Go to the Content Source Administration section in the UI
- d. Add a new Custom Content Source
- e. Enter the connector coordinates as **groupId:artifactId:version**
- f. Configure and save the content source
- g. Run a full and an incremental crawl

### 5. Modify as necessary

Now it is time for you to implement your connector logic.