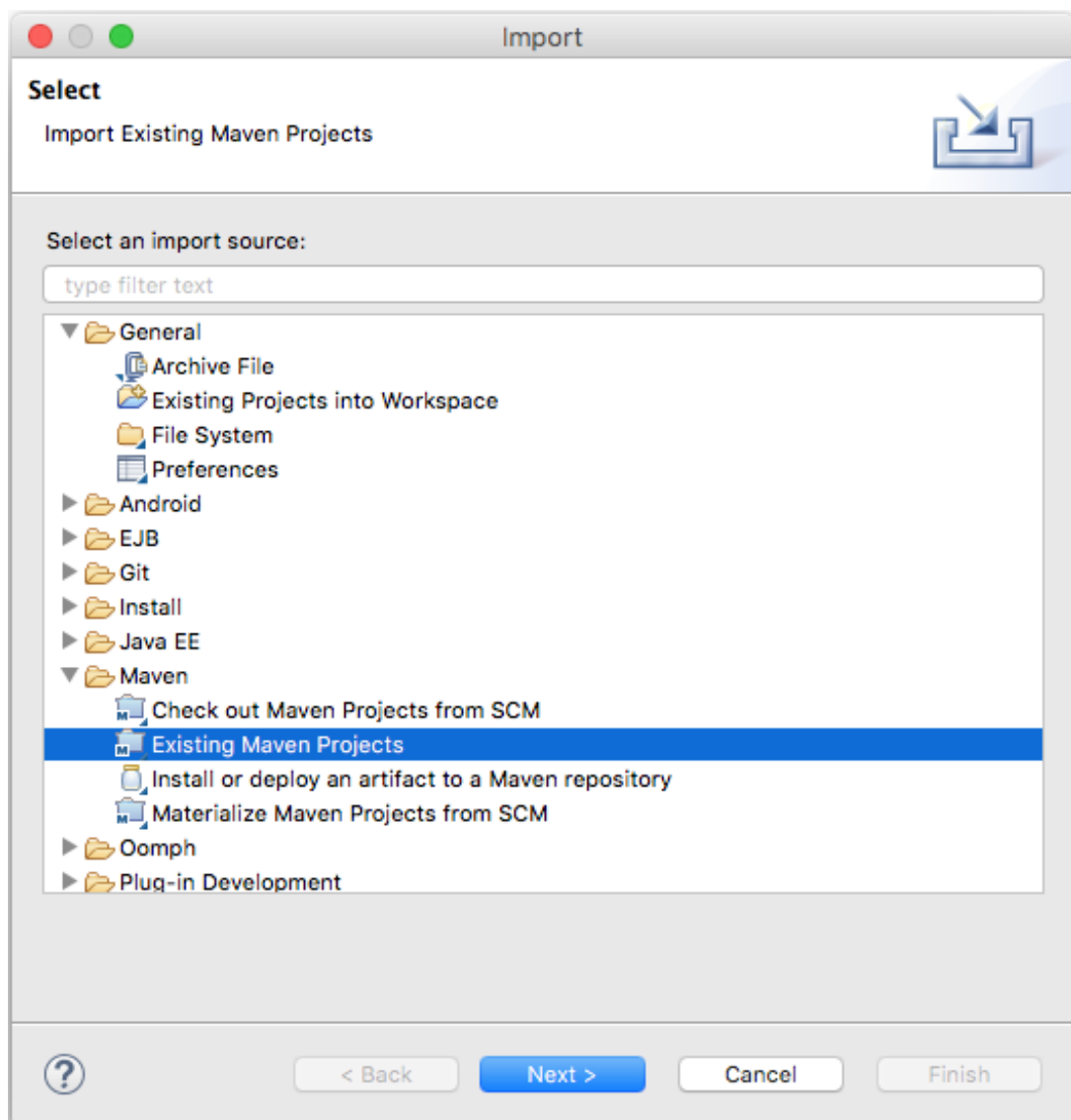# Debugging Adapters

Overview

You can debug Java code implemented for use in **Java** or **JavaScript adapters** via IDEs such as Eclipse, IntelliJ and alike.
This tutorial demonstrates debugging a Java adapter using the Eclipse IDE.

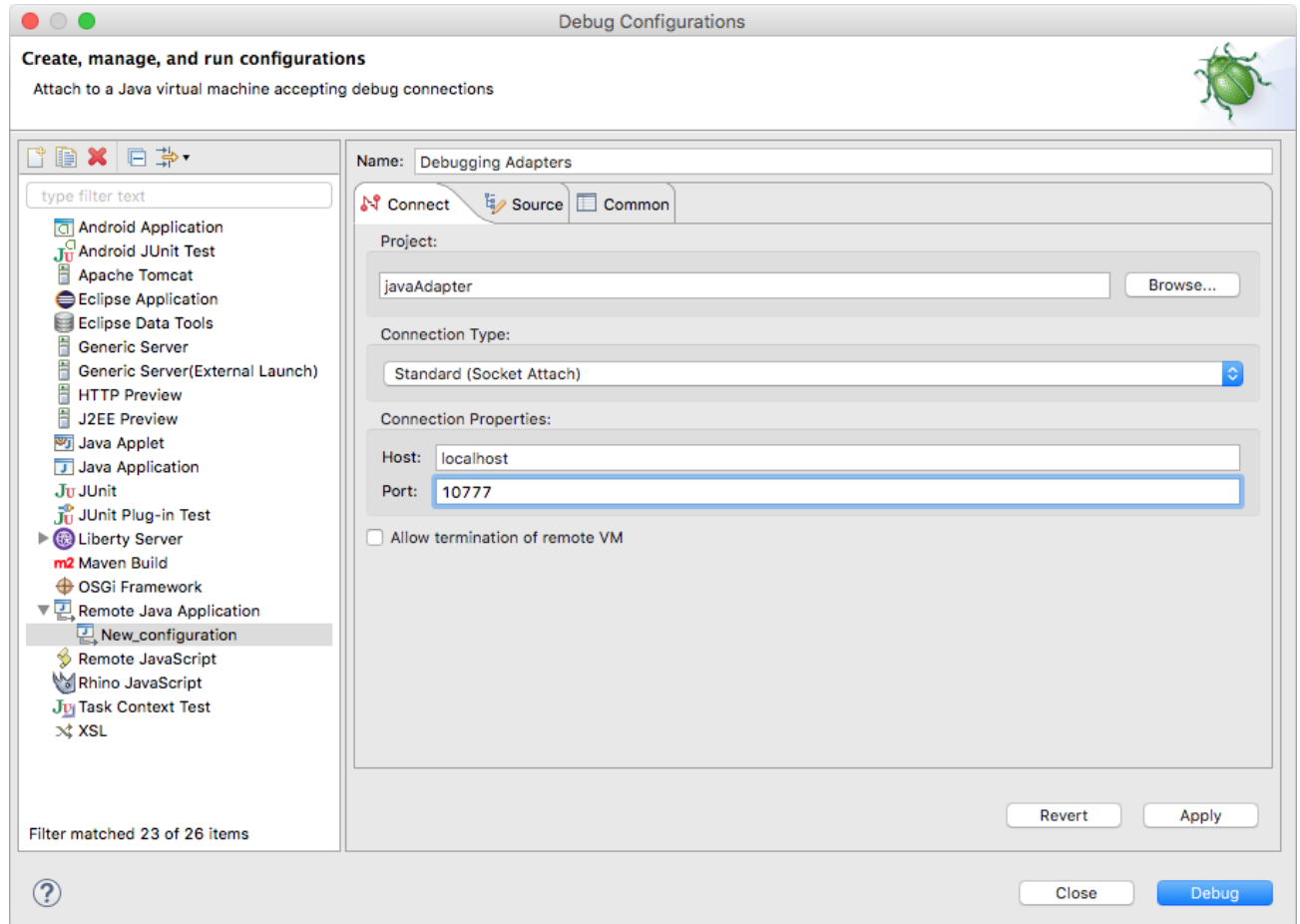### Debugging adapters in Eclipse

Before an adapter's Java code can be debugged, Eclipse needs to be configured as follows:

1. **Maven integration** - Starting Eclipse Kepler (v4.3), Maven support is built-in in Eclipse.
   If your Eclipse instance does not support Maven, follow the m2e instructions
   (http://www.eclipse.org/m2e/) to add Maven support.

2. Once Maven is available in Eclipse, import the adapter Maven project:

3. Provide debugging parameters:

- Click **Run → Debug Configurations**.
- Double-click on **Remote Java application**.
- Provide a **Name** for this configuration.
- Set the **Port** value to "10777".
- Click **Browse** and select the Maven project.
- Click **Debug**.

4. Click on **Window → Show View → Debug** to enter *debug mode*. You can now debug the Java code normally as you would do a standard Java application. You need to issue a request to the adapter to make its code run and hit any set breakpoints. This can be accomplished by following the instructions on how to call an adapter resource in the Testing adapters section (../creating-adapters/#testing-adapters).

Debug - javaAdapter/src/main/java/com/ibm/mfp/sample/JavaAdapterResource.java - Eclipse - /Users/idanadar/Desktop/workspace

Quick Access | Java EE | Debug

Debug | Project Explorer

- javaAdapter
  - src/main/java
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
    - main
      - adapter-resources
      - java
        - com
          - ibm
            - mfp

Variables | **Breakpoints**

☑ JavaAdapterResource

JavaAdapterResource.java

```
7
8  package com.ibm.mfp.sample;
9
10 import java.util.List;
25
26
27 @Path("/users")
28 Breakpoint:JavaAdapterResource.source {
29     /*
30      * For more info on JAX-RS see https://jsr311.java.net/nonav/releases/1.1/index.html
31      */
32
33     //Define logger (Standard java.util.Logger)
34     static Logger logger = Logger.getLogger(JavaAdapterResource.class.getName());
35
```

Outline

- com.ibm.mfp.sample
  - JavaAdapterResource
    - logger : Logger
    - hello() : String
    - helloUser(String) : String
    - helloUserQuery(String) : String
    - enterInfo(String, String, String, int, Stri...
    - newUsers(List<String>) : String

Console | Tasks

No consoles to display at this time.