

# Insight App Development

- [How to setup your own plugin](#)
  - [Maven](#)
  - [Plugin descriptor configuration](#)
  - [Java Example](#)

With the Insight JAVA API you may customize your organization to fit your special requirements. You may want to create your own plugin or post function to do that specific thing to help your organization. Read the Insight Javadoc to be able to create your own customizations and see the [Groovy script examples](#) to get more inspiration on what you can do and something to start with.

## How to setup your own plugin

When using the Insight JAVA-API from your own plugin, you need to specify the dependency to Insight so you are able to code against the Insight classes.

### Maven

pom.xml

```
<dependency>
  <groupId>com.riadalabs.jira.plugins</groupId>
  <artifactId>insight</artifactId>
  <version>${insight.version}</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>com.riadalabs</groupId>
  <artifactId>insight-core-model</artifactId>
  <version>0.2.2</version>
  <scope>provided</scope>
</dependency>
```

You need to install insight locally into your mvn repository. Read more about how to do that here: <https://maven.apache.org/guides/mini/guide-3rd-party-jars-local.html>

### Plugin descriptor configuration

You also need to specify the dependency in the plugin configuration (or as annotation).

In atlassian-plugin.xml it will look like this:

```
<component-import key="objectFacade"
  interface="com.riadalabs.jira.plugins.insight.channel.external.api.facade.ObjectFacade" />
```

### Java Example

Insight has exported all the necessary packages so you can just access the insight facade classes by doing something like this:

```
...

import com.riadalabs.jira.plugins.insight.channel.external.api.facade.ObjectFacade;
import com.riadalabs.jira.plugins.insight.services.model.ObjectBean;

public class MyPluginResource {

    ...
    private final ObjectFacade objectFacade;
    ...

    public MyPluginResource(final ObjectFacade objectFacade) {
        this.objectFacade = objectFacade;
    }

    public ObjectBean getInsightObject(String key) throws Exception {
        return objectFacade.loadObjectBean(key);
    }

}
```