



# Developers Guide

# IWA

## Instant Web Application

Date: August 12th 2011  
Version: 1.0  
Autor: Mischa Christen, m@ood.ch

### 1 Change Log

| Version | Change        | Date       | Autor           |
|---------|---------------|------------|-----------------|
| 1.0     | First Version | 12.08.2011 | Mischa Christen |

## 2 Content

|   |   |   |
|---|---|---|
| 1 | Change Log .....  | 1 |
| 3 | Create an IWA based application that runs in Google App Engine .....        | 3 |
| 4 | Create an IWA based application that runs with Apache Tomcat and MySql..... | 8 |

This guide has been tested with the following configuration:

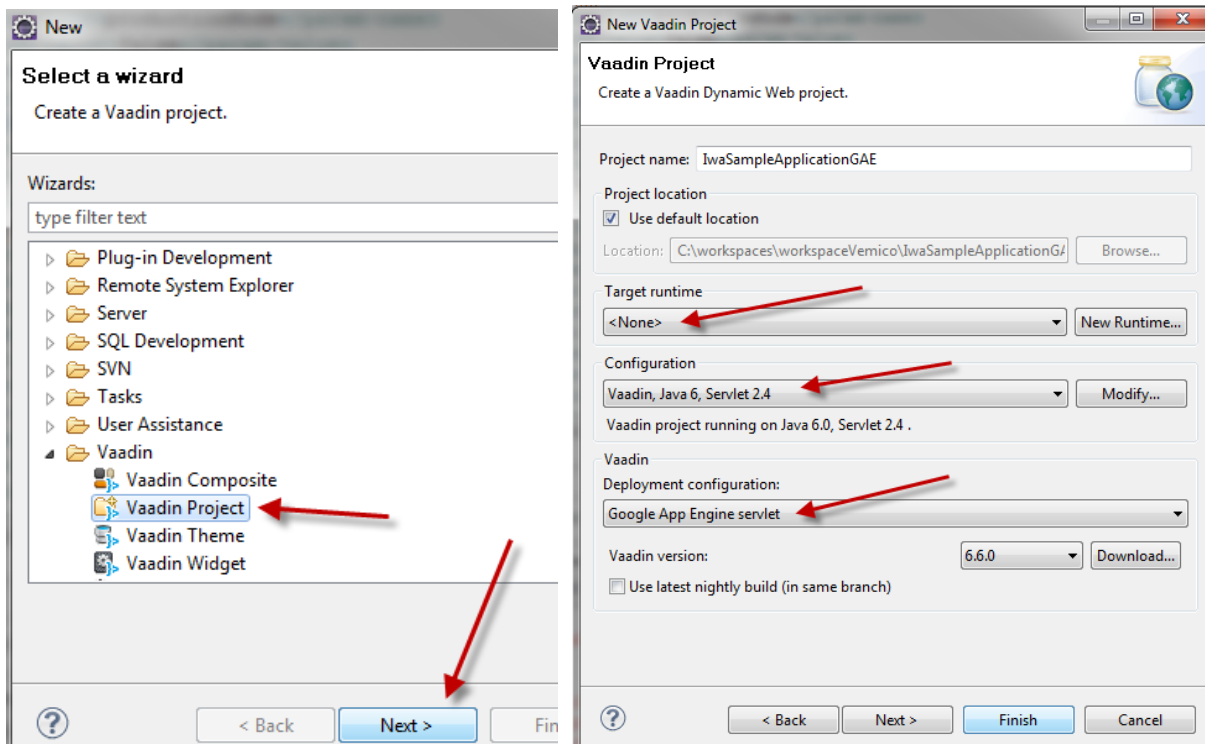
- Java 1.6.0\_25
- Eclipse 3.6 Helios
- AppEngine SDK 1.5.2
- GWT 2.3
- Vaadin 6.6
- Tomcat 6
- MySQL 5.5.13
- Datanucleus (JPA Provider) 1.1.5

### 3 Create an IWA based application that runs in Google App Engine

#### 1. Prerequisites:

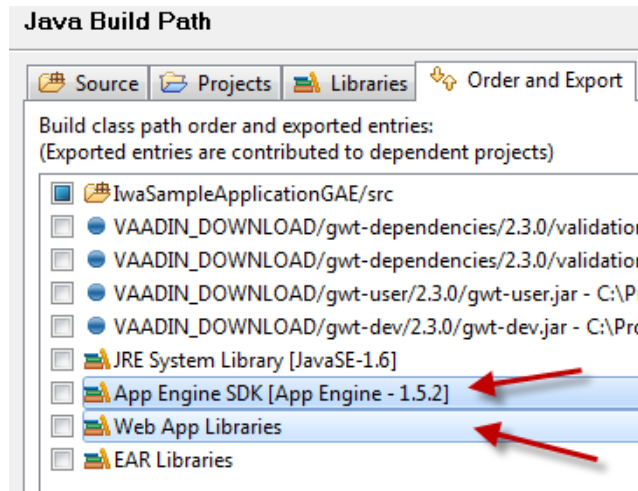
- Install Eclipse (Java EE Version)
- Install Google Plugin (Eclipse Helios Update Site: <http://dl.google.com/eclipse/plugin/3.6>)
- Install Vaadin Plugin (Eclipse Update Site: <http://vaadin.com/eclipse>)

#### 2. Create a new Vaadin Project

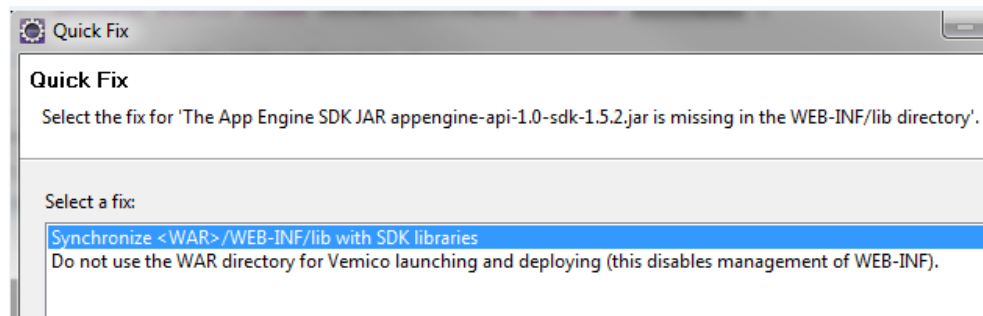


- Target Runtime: None
  - Configuration: Vaadin, Java6, Servlet 2.4
  - Deployment Configuration: Google App Engine Servlet (this will also create appengine-web.xml, logging.properties and make sure the directory structure meets the App Engine requirements (output folder is "war" instead of "WebContent")
3. Check that your project uses UTF-8 encoding
  4. Right Click project, Google | App Engine Settings: "Use Google App Engine"

5. Move the AppEngine SDK above the web libraries in the build path order:



6. Synchronize the missing jars (will become a problem once you deploy your App to the GAE):



**Caution:** Do not add them to the build path, as they are only needed for deployment to GAE, for local development these jars are already contained in the GAE SDK. Adding them may interfere with the DataNucleus Enhancer!

7. Also add the following jars (from the [Iwa Download Website](#), the IwaSampleApplicationGAE-x.y.z.zip file) to your war/WEB-INF/lib directory:
  - xom-1.2.6.jar
  - confirmdialog-1.1.1.jar
  - iwa-x.y.z.jar

8. Add the folder VAADIN (from the IWA Download page) to war/WEB-INF

- To serve these files statically, add the following lines to appengine-web.xml:

```
<static-files>
  <include path="/VAADIN/**" />
</static-files>
```

- Complete appengine-web.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application>MyFirstVaadinAppForAppEngine</application>
  <version>1</version>

  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
  </system-properties>

  <static-files>
    <include path="/VAADIN/**" />
  </static-files>

  <sessions-enabled>true</sessions-enabled>
</appengine-web-app>
```

- The following files can be taken from the IWA Download page:
  - Add the JPA config file persistence.xml to src/META-INF
  - Add the translations.xml file to war/WEB-INF
  - Add the iwa.properties to war/WEB-INF

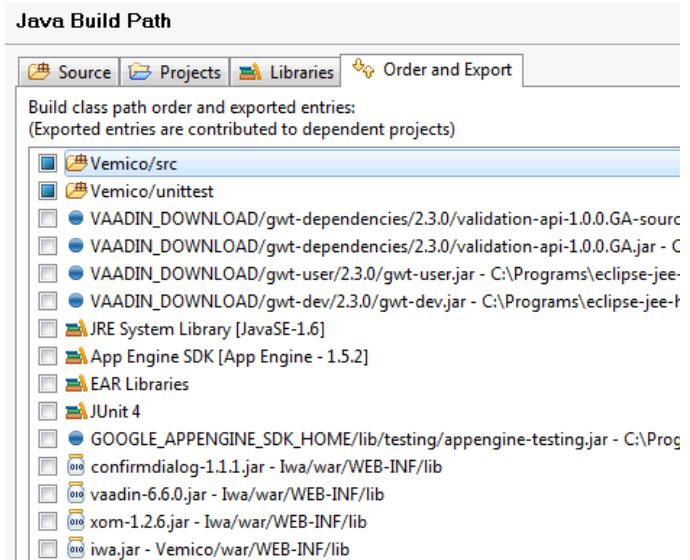
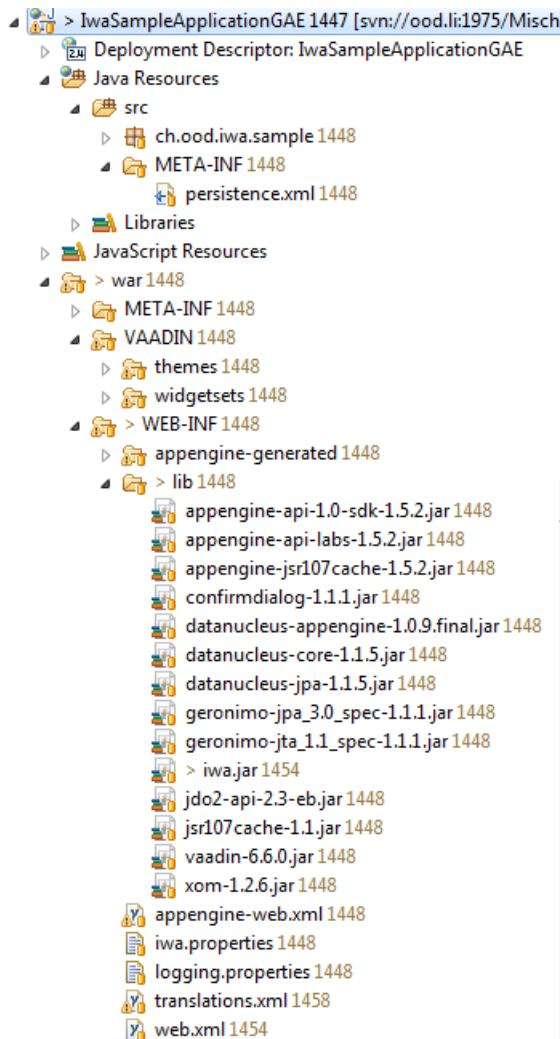
9. Add the IwaContextListener and the i18n servlet to the war/WEB-INF/web.xml:

```
<listener>
  <listener-class>ch.ood.iwa.IwaContextListener</listener-class>
</listener>

<servlet>
  <servlet-name>I18NServlet</servlet-name>
  <servlet-class>org.vaadin.appfoundation.i18n.InternationalizationServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>I18NServlet</servlet-name>
  <url-pattern>/I18NServlet/*</url-pattern>
</servlet-mapping>
```

10. Your directory structure and the build path should now look something like this:





Links for additional hints:

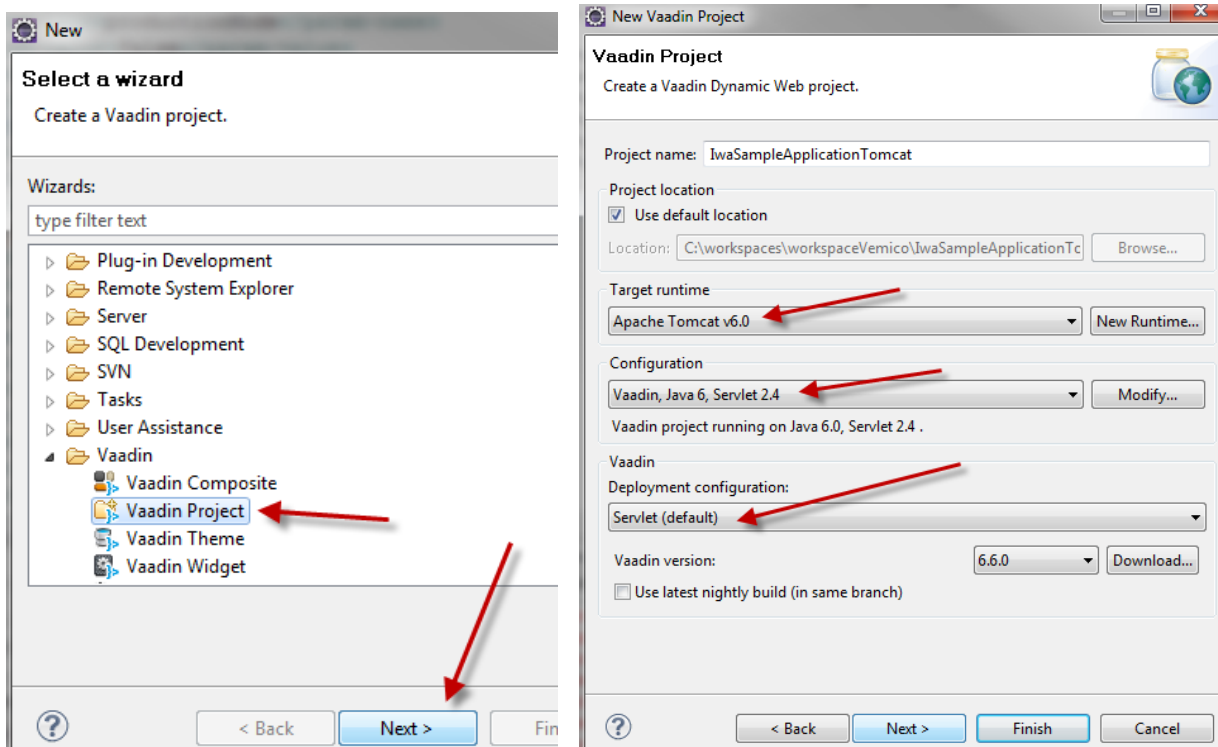
- Vaadin Quickstart with Eclipse:  
<http://vaadin.com/book/-/page/getting-started.first-project.html>
- Vaadin in App Engine:  
<http://vaadin.com/wiki/-/wiki/Main/Google%20AppEngine%20HOWTO>
- Local Unittesting:  
<http://code.google.com/intl/de-DE/appengine/docs/java/tools/localunittesting.html>

## 4 Create an IWA based application that runs with Apache Tomcat and MySql

### 1. Prerequisites:

- Install Eclipse (Java EE Version)
- Install Apache Tomcat 6 (<http://tomcat.apache.org/download-60.cgi>)
- Install MySql (<http://www.mysql.com/downloads/mysql/5.5.html#downloads>)
- Install Vaadin Plugin (Eclipse Update Site: <http://vaadin.com/eclipse>)
- Install Datanucleus Plugin (Eclipse Update Site: <http://www.datanucleus.org/downloads/eclipse-update>)

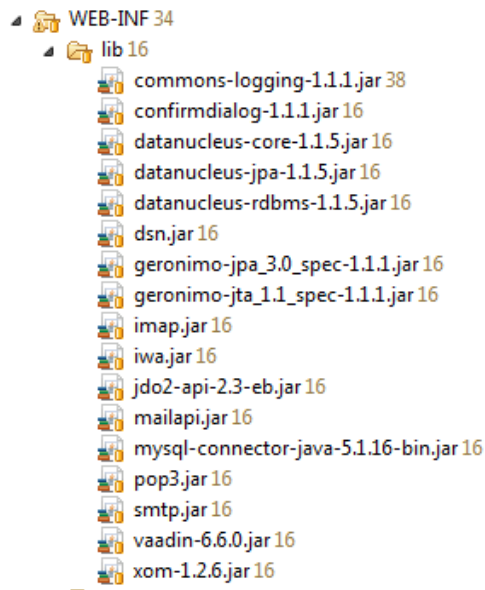
### 2. Create new Vaadin Project



- Target Runtime: Tomcat 6
- Configuration: Vaadin, Java6, Servlet 2.4
- Deployment Configuration: Servlet



3. Add the following jars (from the [Iwa Download Website](#), the IwaSampleApplicationTomcat-x.y.z.zip file) to your WebContent/WEB-INF/lib directory:



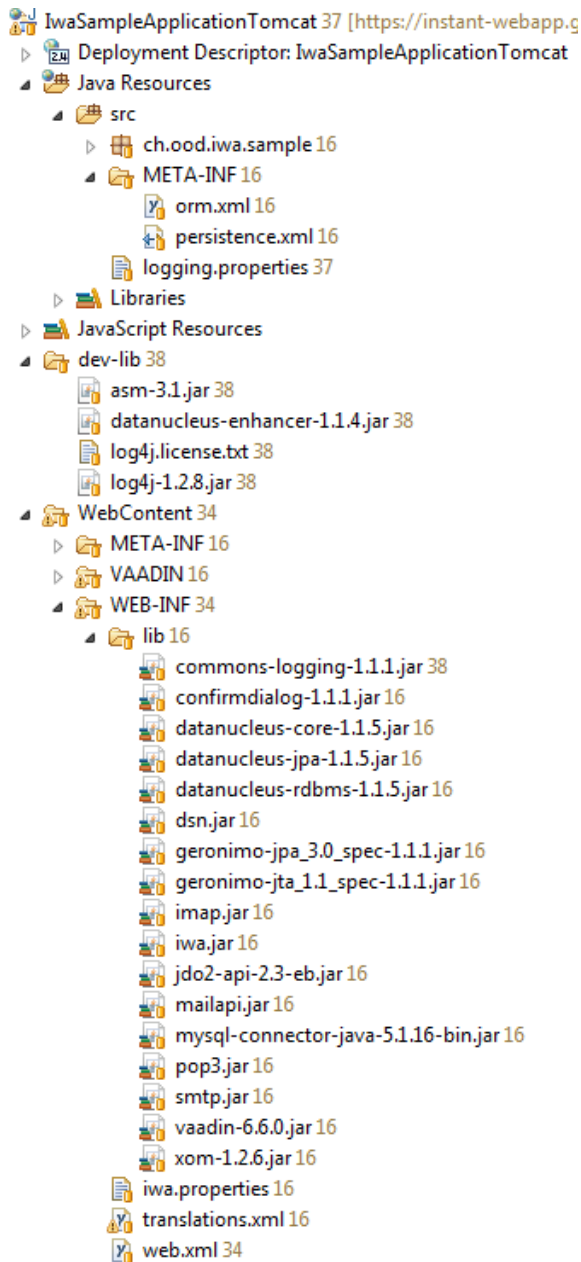
4. Add the folder VAADIN (from the IWA Download page) to WebContent/WEB-INF
5. The following files can be taken from the IWA Download page:
  - Add the JPA config file persistence.xml and orm.xml to src/META-INF
  - Add the logging.properties file to src
  - Add the translations.xml file to WebContent /WEB-INF
  - Add the iwa.properties to WebContent /WEB-INF
6. Add the IwaContextListener and the i18n servlet to the WebContent /WEB-INF/web.xml:

```
<listener>
  <listener-class>ch.ood.iwa.IwaContextListener</listener-class>
</listener>

<servlet>
  <servlet-name>I18NServlet</servlet-name>
  <servlet-class>org.vaadin.appfoundation.i18n.InternationalizationServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>I18NServlet</servlet-name>
  <url-pattern>/I18NServlet/*</url-pattern>
</servlet-mapping>
```

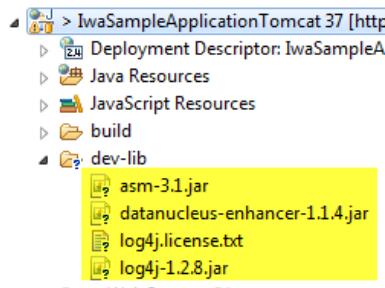
7. Your directory structure should now look something like this:



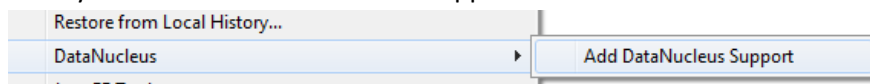
8. Create a schema “iwa” with a user “iwa” (Password “iwa”) in your MySQL instance, make shure the settings in src/META-INF/persistence.xml match your environment

## 9. DataNucleus Settings

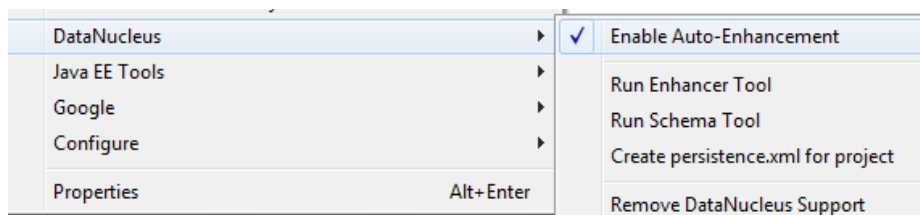
- DataNucleus uses Byte-code enhancement. If you wish to add your own entity classes, you might want to add the “DataNucleus Support” to enhance your classes.
- Add the following jars to your project classpath:



- Then you can turn on DataNucleus support:



- Be aware that the DataNucleus menu is only available in the “Package Explorer” view!



## 10. Create new new Tomcat Server Runtime in Eclipse (see

<http://www.eclipse.org/webtools/jst/components/ws/1.5/tutorials/InstallTomcat/InstallTomcat.html> for hints on this if you are not familiar with it)