Google App Engine

Inhaltsverzeichnis

Introduction

Overview

- Eclipse for Java
- GitHub for code hosting
- App deployment on GoogleAppEngine

Technologie

We used eclipse Kepler as IDE. The project is also hosted on github (https://github.com/lightstalker89/MIT_SAD/tree/master/3nd_sem/cc/grill_mu/Google%20App%20Engine/Example/).

The application was deployed on google app engine and can be accessed by following url: http://grillmurrent.appspot.com/ . For testing is is necessary that you have a google account otherwise you are not able to log in.

References

Google App Engine Tutorial

http://www.vogella.com/tutorials/GoogleAppEngineJava/article.html

Google App Engine Documentation:

https://cloud.google.com/appengine/docs/java/tools/devserver

JPA:

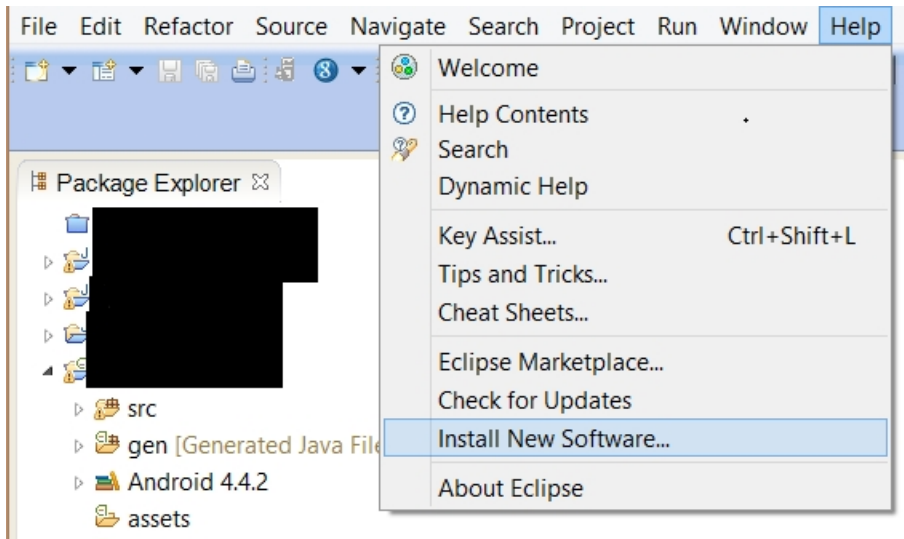http://de.wikipedia.org/wiki/Java_Persistence_API

Installation

Important for developing with GAE (Google App Engine) is step 2 from the tutorial: "Installing of the Google Tools for Eclipse".
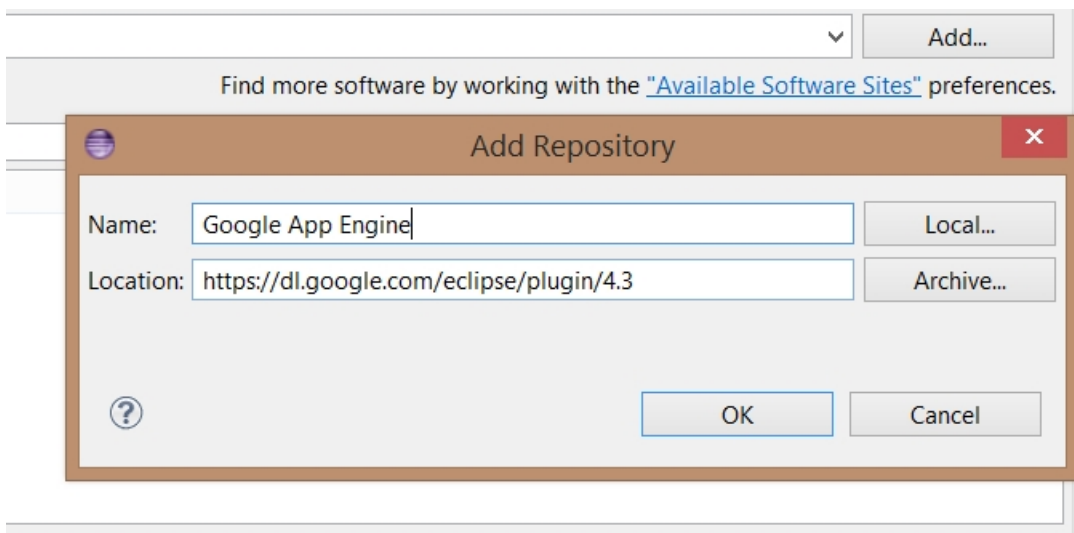
It is important that you install the right Google app engine plug-in for your eclipse version. The image below shows the specific URLs for the last 3 eclipse versions.

Eclipse version	Installation instructions	Direct plugin link
Eclipse 4.4 (Luna)	Plugin for Eclipse 4.4 (Luna)	https://dl.google.com/eclipse/plugin/4.4
Eclipse 4.3 (Kepler)	Plugin for Eclipse 4.3 (Kepler)	https://dl.google.com/eclipse/plugin/4.3
Eclipse 3.8/4.2 (Juno)	Plugin for Eclipse 3.8/4.2 (Juno)	https://dl.google.com/eclipse/plugin/4.2

For our project we used eclipse 4.3. So we used the following URL https://dl.google.com/eclipse/plugin/4.3 . All you have to do is to navigate to the "Install New Software" window. See image below.

In the window click "Add.." and enter your specif plug-in URL.

After a successful connection, the necessary packages are shown in the data grid. It is recommended to install every package, so click "Select all" and then "Next".

Name	Version
▷ ☑ ▫▫▫ Developer Tools	
▷ ☑ ▫▫▫ Google App Engine Tools for Android (requires ADT)	
▷ ☑ ▫▫▫ Google App Engine Tools for Maven (requires m2e-wtp 1.5+)	
▷ ☑ ▫▫▫ Google Plugin for Eclipse (required)	
▷ ☑ ▫▫▫ GWT Designer for GPE	
▷ ☑ ▫▫▫ SDKs	

[Select All] [Deselect All]   1 item selected

Details

☑ Show only the latest versions of available software   ☑ Hide items that are already installed
☑ Group items by category   What is already installed?
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software

(?)   < Back   Next >   Finish

After successful installation you are ready to deploy on the Google app engine.

Setup JPA and DAO

To use JPA in your project you must create the JPA configuration file "persistence.xml" in the "META-INF" directory. This file contains the configuration that tells JPA to use the App Engine datastore.

```
▲ 🗂 com.grillmurrent.gae.java.todo
   ▲ 📂 src
      ▲ ⊞ com.grillmurrent.gae.java.todo
         ▷ 🗋 ServletCreateTodo.java
         ▷ 🗋 ServletRemoveTodo.java
      ▲ ⊞ com.grillmurrent.gae.java.todo.dao
         ▷ 🗋 Dao.java
         ▷ 🗋 EMFService.java
      ▷ ⊞ com.grillmurrent.gae.java.todo.model
      ▲ 📂 META-INF
            🗋 persistence.xml
         📄 log4j.properties
   ▷ 📚 App Engine SDK [App Engine - 1.9.17]
   ▷ 📚 JRE System Library [jre]
   ▷ 📂 war
```

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
```

```
          http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd" version="1.0">

    <persistence-unit name="transactions-optional">
        <provider>org.datanucleus.api.jpa.PersistenceProviderImpl</provider>
        <properties>
            <property name="datanucleus.NontransactionalRead" value="true"/>
            <property name="datanucleus.NontransactionalWrite" value="true"/>
            <property name="datanucleus.ConnectionURL" value="appengine"/>
            <property name="datanucleus.singletonEMFForName" value="true"/>
        </properties>
    </persistence-unit>
</persistence>
```

It is important that you set up the right provider. The provider which is used in the tutorial (http://www.vogella.com/tutorials/GoogleAppEngineJava/article.html#todo_jpa) is wrong because in the new version of the google app engine you have to use "`org.datanucleus.api.jpa.PersistenceProviderImpl`".

The DAO (Data Access Object) class provides the possibility to query for all todos and to add and delete todos. Because of this project the class has been extended by the functionality to add and get a ToDo history object.

Client

Below you can see an image of the developed Google app engine application. It is possible to create ToDos an to arrange them in order of importance. Furthermore all completed ToDos were saved in a history table. If you want to try this application by yourself please visit: http://grillmurrent.appspot.com/

☑ Todos

You have a total number of 1 Todos.

Short description	Long Description	URL	Importance ⬇	Done
Test	Test ToDo	http://grillmurrent.appspot.com/TodoApplication.jsp	1	Done

Your finished ToDos 3

Short description	Long Description	URL	Importance ⬇
Test	Tes	http://1-dot-grillmurrent.appspot.com/	5
dasdsa	dasdsa	http://1-dot-grillmurrent.appspot.com/TodoApplication.jsp	5
dasdas	dasdsad	http://1-dot-grillmurrent.appspot.com/TodoApplication.jsp	1

New todo

Summary

Description

URL

Importance

Create