

E-PORTFOLIO: GOOGLE WEB TOOLKIT

BY RUTH WEBER



GWT

STRUCTURE

Overview

Possibilities

Pro's and Con's

First Example

-> Your turn

OVERVIEW

Open source project

Web development

Programming language: Java

Compiles Java to JavaScript

Server-Client Connection

GWT development plugin in browser

POSSIBILITIES

Includes own widgets

CSS Styling

JUnit Test

PRO'S AND CON'S

Pro's:

- Overall browser compatibility
- Coding in Java
- Debugging
- Easy integration to Google API's

Con's:

- Compiling time takes time
- More Java than it would be in JavaScript

FIRST EXAMPLE

Windows:

```
cd myDirectory/gwt-2.7.0
```

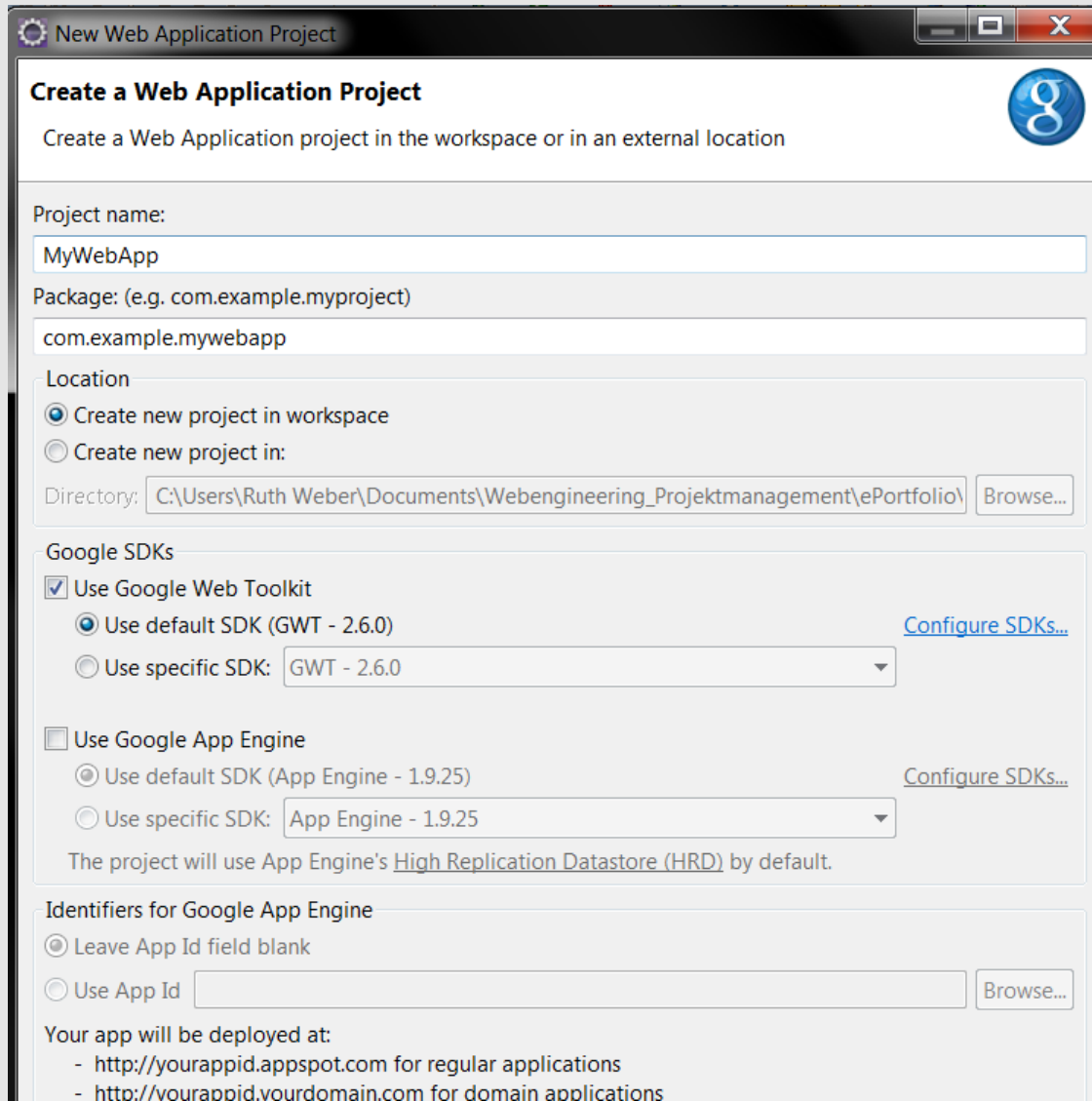
```
webAppCreator -out <nameOfYourApp>  
              -junit  
              "pathToEclipse/plugins/org.junit_4.x/junit.jar"  
              com.example.myapp.<nameOfYourApp>
```

Linux/Mac:

```
cd myDirectory/gwt-2.7.0
```

```
chmod u+x webAppCreator  
./webAppCreator -out <nameOfYourApp>  
               -junit  
               "pathToEclipse/plugins/org.junit_4.x/junit.jar"  
               com.example.myapp.<nameOfYourApp>
```

FIRST EXAMPLE



Create a Web Application Project

Create a Web Application project in the workspace or in an external location

Project name:

Package: (e.g. com.example.myproject)

Location

☒ Create new project in workspace

☐ Create new project in:

Directory:

Google SDKs

☒ Use Google Web Toolkit

☒ Use default SDK (GWT - 2.6.0) [Configure SDKs...](#)

☐ Use specific SDK: ▼

☐ Use Google App Engine

☒ Use default SDK (App Engine - 1.9.25) [Configure SDKs...](#)

☐ Use specific SDK: ▼

The project will use App Engine's [High Replication Datastore \(HRD\)](#) by default.

Identifiers for Google App Engine

☒ Leave App Id field blank

☐ Use App Id

Your app will be deployed at:

- <http://yourappid.appspot.com> for regular applications
- <http://yourappid.yourdomain.com> for domain applications

YOUR TURN

Try to create a stock watcher with Google Web Toolkit take the example as base. In the following there is a overview what you should do.

→ change the HTML file

→ initialize the needed GWT widgets (FlexTable, Button, Panel, TextBox)

- table header should be: „symbol“, „price“ and „change (in percent)“
- There exists a `RootPanel.get(„id“).add(...)`, this connects the widgets with the HTML page.

→ create **StockPrice** class with getter and setter and **getPriceChange** in percent

You can follow the tutorial:

→ <https://github.com/RuWe/e-Portfolio-GWT/blob/master/documents/TutorialGWT.pdf?raw=true>