Java +1 54

App Engine Home
- Java
  - Runtime Environment
  - Managed VMs Beta
  - Handling Requests
  - Java Tutorial
    1. Introduction
    2. Tutorial Setup
    3. Creating the Project
    4. Adding Code and UI
    5. Storing Data
    6. Uploading Your Application
  - Modules
  - Storing Data
  - Services
  - Configuration
  - YAML Configuration
  - Data Processing
  - Tools
  - Java API Reference
  - JRE Class White List
  - Release Notes

Write Feedback

Creating the Project

> **Important:** The instructions in this tutorial assume that you are using a terminal on your local machine and running a browser on that same machine to test against the local development server.

To create a new project, you'll need to use an App Engine-provided Maven App Engine artifact called `appengine-skeleton-archetype`, as described in Creating the project. The App Engine Maven artifact is what creates the project layout and files required to deploy and run on App Engine.

For more on using Maven with App Engine, see the Using Apache Maven Guide. As an alternative to Maven, you could instead use Eclipse via the Google Plugin For Eclipse or you could use Apache Ant.

Creating the project

To create the project:

1. You'll need an application ID (project ID) in order to deploy your app to production App Engine. Create a new project as follows:

   a. Visit Google Developers Console in your web browser, and click **Create Project**.
   b. Supply the project name *Guestbook* and accept the project ID that is auto-generated for you.
   c. Click **Create**.

   Make a note of the project ID, since you'll need to supply it as the application ID in the next step.

> **Note:** Alternatively, you can use the ID from an existing project in the Google Developers Console if you want to use it for this project.

2. Change to a directory where you want to build your project and invoke Maven as follows:

   ```
   mvn archetype:generate -Dappengine-ve
   ```

   replacing `your-app-id` with the project ID value you obtained in the previous step. The Maven `archetype:generate` command creates a new project with directories and files from the archetype you specify in this procedure. Note that the values you supply here and elsewhere in the remaining steps in this procedure are used in the `pom.xml` and `appengine-web.xml` files generated in the directories shown in the screencap located at the end of this procedure.

3. Select from the artifact list by supplying the number corresponding to `com.google.appengine.archetypes:appengine-skeleton-archetype`.

4. Select the most recent version from the displayed list of available archetype versions by accepting the default.

5. When prompted to `Define value for property 'groupId'`, supply the desired namespace for your app; to keep this tutorial in sync with the source files at GitHub, specify `com.example.guestbook`.

6. When prompted to `Define value for property 'artifactId'`, supply the desired project name; to keep this tutorial in sync with the source files at GitHub, use `guestbook`.

7. When prompted to `Define value for property 'version'`, accept the default value.

8. When prompted to `Define value for property 'package'`, supply your preferred package name (or accept the default). The

generated Java files will have the package name you specify here. Note that the source files at GitHub use the package name `com.example.guestbook`, so you should use that as the package name if you want to keep in sync with the sample source.

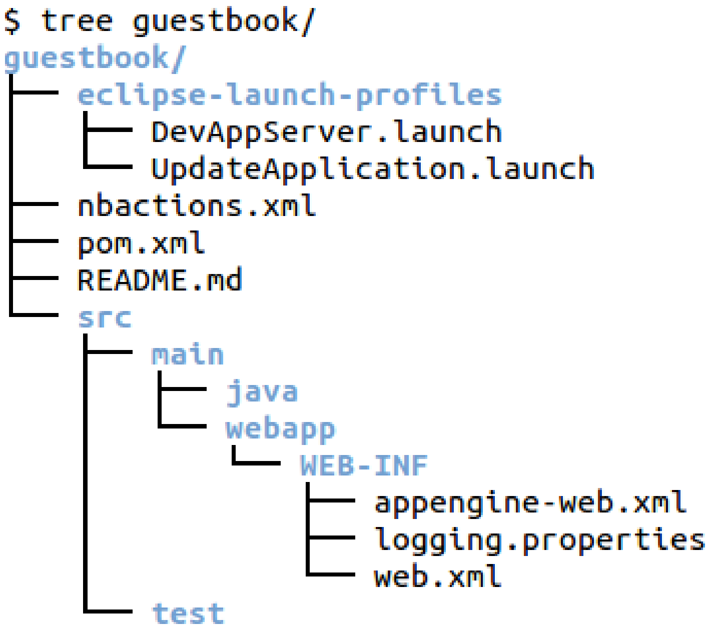
9. When prompted to confirm your choices, accept the default value (`Y`), and wait for the project to finish generating.
10. Change directory to the new `guestbook` that you just created.
11. At this point you won't have any source code. But build the project anyway to download any required libraries by invoking

    ```
    mvn clean install
    ```

12. Wait for the project to build. When the project successfully finishes you will see a message similar to this:

    ```
    [INFO] ---------------------------
    [INFO] BUILD SUCCESS
    [INFO] ---------------------------
    [INFO] Total time: 1:16.656s
    [INFO] Finished at: Mon Sep 15 11
    [INFO] Final Memory: 16M/228M
    [INFO] ---------------------------
    ```

13. Examine the project layout:



```
$ tree guestbook/
guestbook/
├── eclipse-launch-profiles
│   ├── DevAppServer.launch
│   └── UpdateApplication.launch
├── nbactions.xml
├── pom.xml
├── README.md
└── src
    ├── main
    │   ├── java
    │   └── webapp
    │       └── WEB-INF
    │           ├── appengine-web.xml
    │           ├── logging.properties
    │           └── web.xml
    └── test
```

- You'll add your own application Java

classes to `src/main/java/...`
  - You'll configure your application using the file `src/main/webapp/WEB-INF/appengine.web.xml`
  - You'll configure your application deployment using the file `src/main/webapp/WEB-INF/web.xml`

We'll describe what to do inside these subdirectories later.

You are now ready to add your own application code and UI.

Adding Application Code and UI >>

Last updated January 22, 2015.