

Distributed System and Cloud Computing Lab

Google App Engine with the Eclipse IDE.

Aim: To develop applications using Google App Engine with the Eclipse IDE.

Theory:

Google App Engine (GAE) is a fully managed, serverless platform provided by Google Cloud for building and deploying applications. With Google App Engine, you can focus on writing code without worrying about managing infrastructure, scaling, or other operational concerns. GAE supports multiple programming languages, including **Java, Python, Go, PHP, Node.js**, and others.

Eclipse IDE, a popular open-source integrated development environment, is widely used by developers to build, test, and deploy applications. By integrating **Google App Engine** with **Eclipse**, developers can create cloud-based applications and seamlessly deploy them to the Google Cloud Platform (GCP) environment. In this guide, we will explore how to use **Google App Engine** with **Eclipse IDE** to build scalable applications.

1. Google App Engine Overview

- a. **Google App Engine** provides a platform to build and host applications on Google Cloud. It abstracts away the underlying infrastructure management, so developers can focus on writing business logic.
- b. **Features:**
 - i. **Automatic Scaling:** App Engine automatically scales your application based on incoming traffic.
 - ii. **Managed Services:** Provides features such as databases, storage, and security.
 - iii. **Integrated with GCP Services:** It integrates well with other GCP services like **BigQuery, Pub/Sub, Firestore**, and more.
 - iv. **Multi-language support:** It supports Java, Python, Go, Node.js, and more.

2. **Google Cloud SDK**

- a. The **Google Cloud SDK** is a set of tools that allows developers to interact with GCP resources from the command line. The SDK includes the **gcloud** CLI and other tools that help manage Google Cloud resources.
- b. To work with Google App Engine in Eclipse, you need the Google Cloud SDK installed, which includes the **App Engine SDK**.

3. **App Engine Standard Environment vs. Flexible Environment**

- a. **Standard Environment:** Ideal for applications with predictable traffic patterns, and it supports automatic scaling and limited language runtimes.
- b. **Flexible Environment:** Suitable for applications with more complex workloads or that require custom runtime environments.

4. **Eclipse IDE Setup for Google App Engine Development**

- a. **Eclipse IDE** can be extended with **Google Cloud Tools for Eclipse** to enable App Engine development. These tools offer features like:
 - i. Local development server for running applications.
 - ii. Easy deployment to Google Cloud.
 - iii. Integration with Google Cloud services (e.g., Cloud Datastore, Cloud Storage).

Steps:

Step 1: After creating a workspace, the following screen will appear:

1. Go to **HELP → Eclipse Marketplace**.
2. In the Eclipse Marketplace window, type "**google**" in the search box and press **Enter**.
3. Click **Confirm**.
4. When the next window pops up, click on **Trust Selected**.
5. Click **Restart Now** to apply the changes.
6. After restarting, close the Welcome page.

Step 2 (Optional):

If the tools don't install successfully:

1. Once the Eclipse window appears, navigate to **WINDOW → PREFERENCES**.
2. In the Preferences window, search for **Google**.
3. Select **App Engine Java Components** and click **Apply** and then **Close**.

Note: If the above steps don't show the screen, follow these instructions to install the Google Cloud SDK:

1. Visit the Google Cloud SDK Documentation.
2. Click on **Installing Google Cloud SDK**.
3. Run the **gcloud init** command in the Command Prompt.
4. Choose the option to **Log in with a new account**.
5. Sign in with your Google account and click **Allow**.
6. If you see the message "**This account has no projects**", create a new project.
7. To update the SDK to the latest version, run the command: **gcloud components update**.
8. Close the Command Prompt.

Step 3:

1. In the Eclipse workspace, click the dropdown menu and select **Create New Project → Google App Engine Standard Java Project**.
2. In the **New App Engine Standard Project** window, provide the following details:
 - **Project Name:** MySandboxProject
 - **Java Package:** com.gonevertical.server.sandbox
3. Click **Next**.
4. Under **App Engine Standard Libraries**, select:
 - **App Engine API**
 - **Objectify**
5. Click **Finish**. The **HelloAppEngine.java** file will appear.
6. Go to **Window → Show View → Other**.
7. In the **Show View** window, select **Server → Servers → Open**.
8. Then go to **Window → Editor → Console**.

9. Once the **Console** screen appears, click on **Servers**.
10. Right-click on the server → **New** → **Server**.
11. In the **Define a New Server** window, select **App Engine Standard**, then click **Next** and **Finish**.
12. From the **Available Projects** section, select **MySandboxProject** and click **Add**.
13. After adding **MySandboxProject** to the **Configured Projects**, click **Finish**.
14. Click on **App Engine Standard** at **localhost** → **MySandboxProject**.
15. Right-click on **App Engine Standard** at **localhost** and select **Debug**.

Program Code:

HelloAppEngine.java

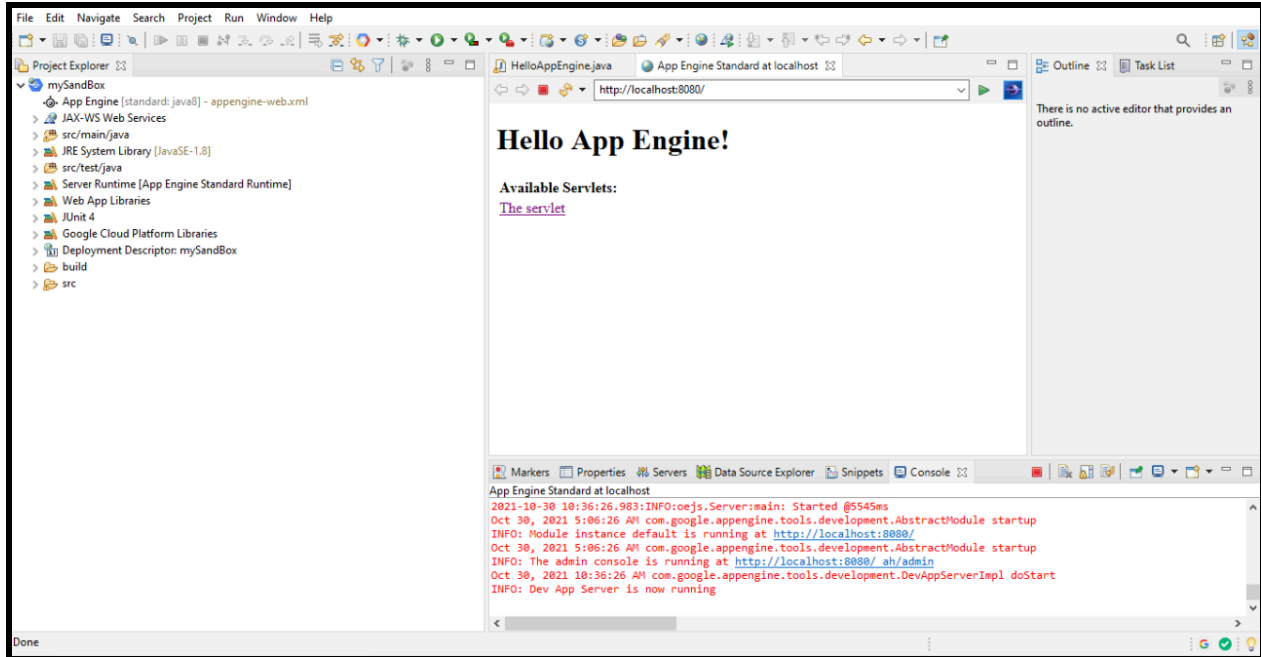
```
package com.gonevertical.server.sandbox;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(
    name = "HelloAppEngine",
    urlPatterns = {"/hello"}
)
public class HelloAppEngine extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
IOException {
        response.setContentType("text/plain");
        response.setCharacterEncoding("UTF-8");
        response.getWriter().println("Hello App Engine!");
    }
}
```

Output:



Conclusion:

By following the steps outlined, you can successfully set up Google App Engine with Eclipse IDE to develop and deploy Java-based applications on the Google Cloud Platform. The process involves installing necessary tools like the Google Cloud SDK and Google Cloud Tools for Eclipse, setting up a new App Engine Standard Java Project, and configuring your environment for local development and debugging.

With these configurations, developers can efficiently build scalable, cloud-based applications without worrying about the underlying infrastructure. The integration of App Engine with Eclipse provides a streamlined development experience, enabling seamless deployment and management of applications. By utilizing Google App Engine's automatic scaling and managed services, developers can focus on writing business logic while Google handles the operational aspects, ensuring better performance, reliability, and scalability for your applications.