Android Studio

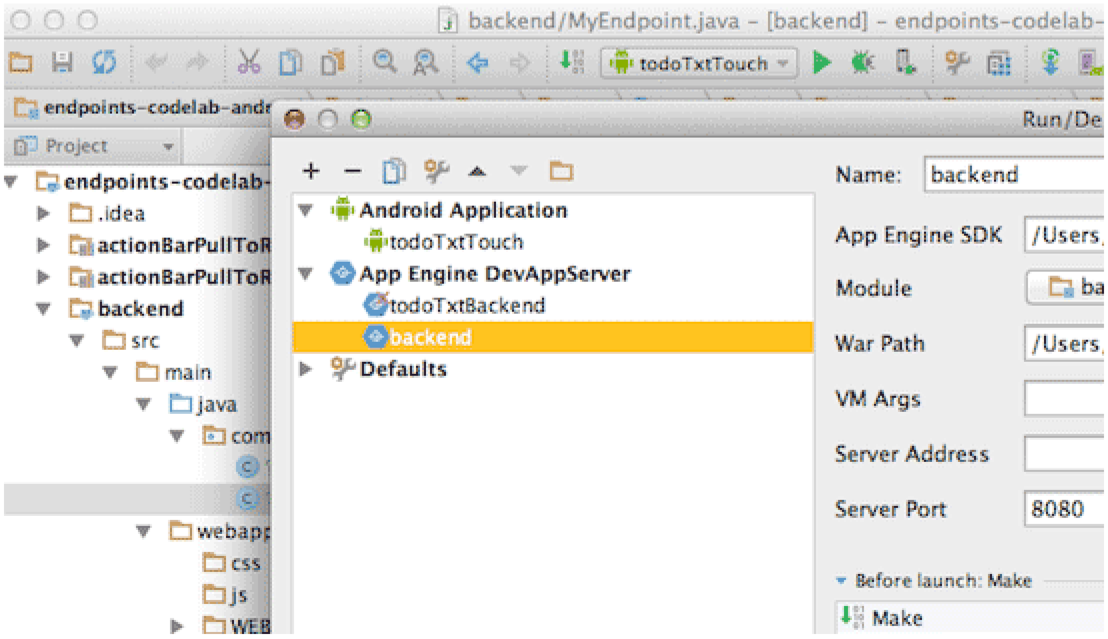
Running, Testing, and Deploying the Backend

You can run and test your backend locally, and deploy the backend using Android Studio.

Running and testing the backend locally

After adding the Endpoints backend template, you should test it locally to make sure your environment is functioning properly. To run the backend template locally:

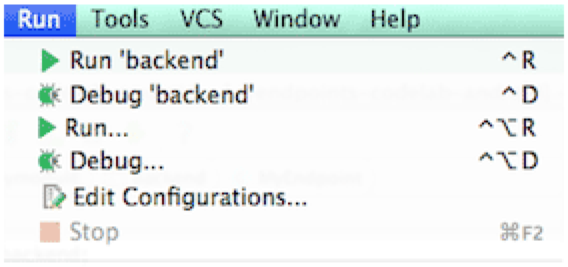
1. Click **Run** > **Edit Configurations** to open the *Run/Debug Configurations* form:



Name: backend
App Engine SDK /Users
Module ba
War Path /Users
VM Args
Server Address
Server Port 8080
Before launch: Make
Make

Note that when you add the backend module template to your project a new run configuration is created for the backend; it has the module name you assigned during backend creation.

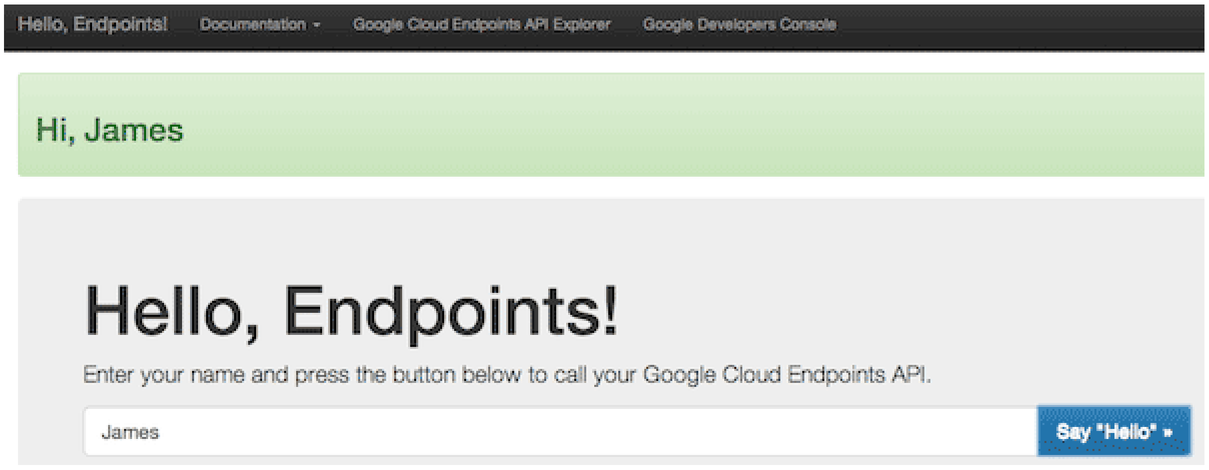
2. In the left pane, under App Engine DevAppServer, locate the backend module you just added, select it, and click **OK**.
3. Click **Build** > **Rebuild Project** and wait for the build to finish.
4. Select **Run** > **Run** `<your-backend-module-name>` to launch the backend in the local App Engine development server in Android Studio:



5. Wait for the backend to start up in the development server; when it finishes loading, a message similar to the following is displayed in the console:

```
Jun 18, 2014 4:07:30 PM com.google.appengine.tools.development.AbstractModu
INFO: The admin console is running at http://localhost:8080/_ah/admin
Jun 18, 2014 4:07:30 PM com.google.appengine.tools.development.DevAppServer
INFO: Dev App Server is now running
```

6. Navigate to http://localhost:8080 (http://localhost:8080) on the machine running your Android Studio project. If everything went well, you should see the following page:


Hello, Endpoints!
Documentation
Google Cloud Endpoints API Explorer
Google Developers Console
Hi, James
Hello, Endpoints!
Enter your name and press the button below to call your Google Cloud Endpoints API.
James
Say "Hello" »

(The page shown is for the Endpoints backend. The page for the other backend types will be slightly different.)

Notice the links at the top of the page, especially the link that takes you to the Google Cloud Console, where you can create or configure a project. You'll need a project to deploy your backend after your development and testing work is complete.

At this point, the default backend from the template is successfully running. However, your Android app is not yet connected to the backend. To access the backend, you need to add some code your Android app as described in the appropriate README file:

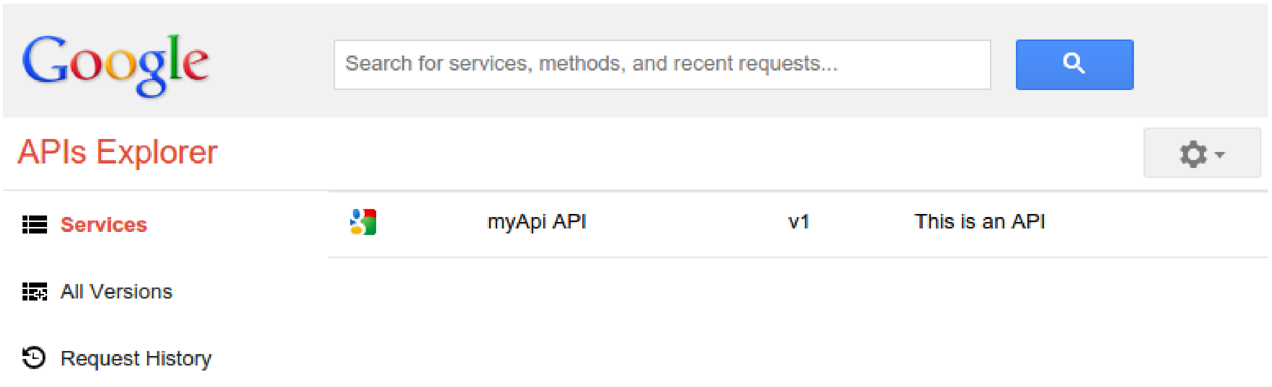
- App Engine Java Servlet Module (https://github.com/GoogleCloudPlatform/gradle-appengine-templates/tree/master/HelloWorld)
- App Engine Java Endpoints Module (https://github.com/GoogleCloudPlatform/gradle-appengine-templates/tree/master/HelloEndpoints)
- App Engine Backend with Google Cloud Messaging (https://github.com/GoogleCloudPlatform/gradle-appengine-templates/tree/master/GcmEndpoints)

You'll also need to add whatever customized behavior you want your backend to have.

Testing the Endpoints backend using API Explorer

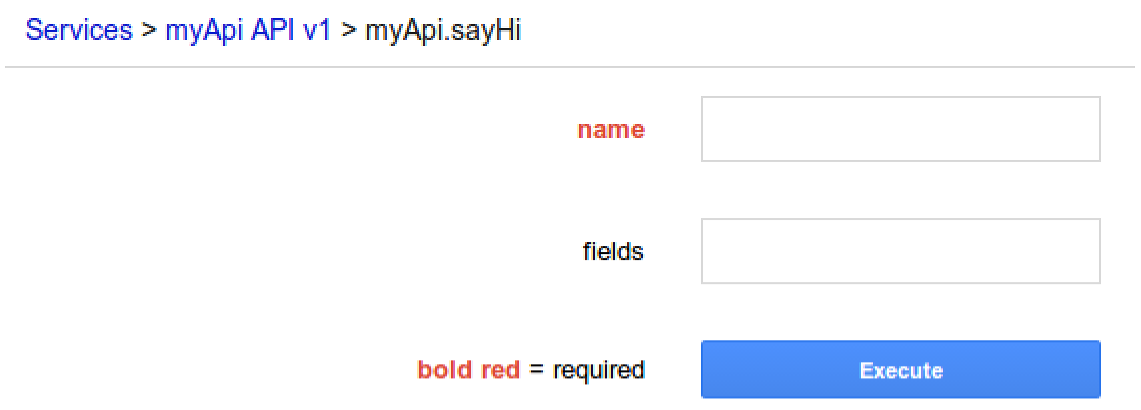
For backends that have Endpoints, you can test the functioning of the API directly using the built-in Google API Explorer:

1. Start and run your backend locally as described previously.
2. Navigate to http://localhost:8080/_ah/api/explorer (http://localhost:8080/_ah/api/explorer) on the machine running your Android Studio project. If everything went well, you should see a page similar to the following:



myApi API v1 This is an API

3. Click the API name, for example **myApi API**, to display the methods available from this API.
4. Click a method, for example, **myApi.sayHi** to display the Explorer form for this method:



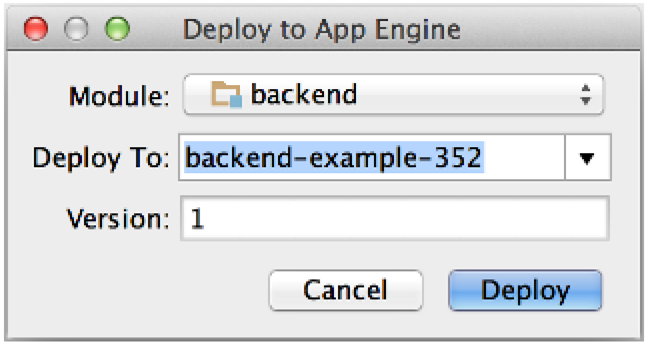
5. Supply a value for the input field if it has one; using the default backend as an example, supply your name in the **name** field. Click **Execute** and check the Response for the expected response.

Deploying backends

If your backend is working locally, you can deploy it to Google App Engine.

Before deploying your backend, you need to create a Cloud project for it and add the project ID to your backend, as described in the README for your backend template (Servlet Module (https://github.com/GoogleCloudPlatform/gradle-appengine-templates/tree/master/HelloWorld)/ Endpoints Module (https://github.com/GoogleCloudPlatform/gradle-appengine-templates/tree/master/HelloEndpoints)/ Backend with Google Cloud Messaging (https://github.com/GoogleCloudPlatform/gradle-appengine-templates/tree/master/GcmEndpoints)) To deploy your backend to App Engine from Android Studio:

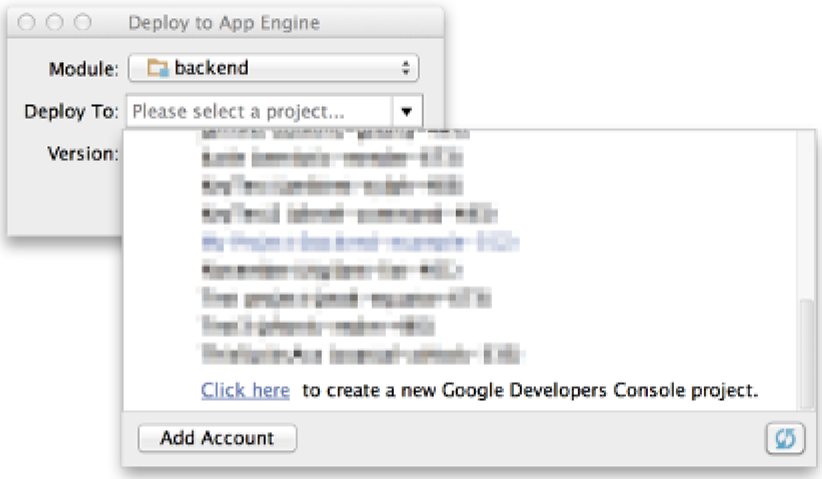
1. Stop the backend, if it is running locally, by selecting **Run** > **Stop**.
2. Run **Build** > **Deploy Module to App Engine**.
   - If you are running this task for the first time, you will be prompted to sign-in with your Google Account. Choose an account and sign in.
3. In the **Deploy to App Engine** dialog, select your module. From the **Deploy To:** dropdown list, choose a Cloud Console project (in this example, `backend-example-352`).



Module: backend
Deploy To: backend-example-352
Version: 1

Scroll to the bottom of the dropdown list if you need to create a new project or sign in with a different account:

- To create a new project, scroll to the bottom of the dropdown and click **Click Here**. This takes you to the Cloud Console to create the project. Back in the dialog in Android Studio, click the Refresh button to include your new project in the dropdown. Select the new project.
- If you want to create a new account or specify a different existing account, scroll to the bottom of the dropdown and click **Add Account**.



Module: backend
Deploy To: Please select a project...

Click here to create a new Google Developers Console project.

4. Back in the **Deploy to App Engine** dialog, click **Deploy**. You can monitor the status of your deployment in the Android Studio console.

Last updated December 4, 2019.