

App Dev-Adding User Authentication to your Application-Java

GSP169



Overview

This lab shows you how to add authentication to your application using Firebase. This authorization identifies who you are, and determines what you can do. For more information, see [Authentication Overview](#).

Firebase is a comprehensive framework that allows you to create web and mobile applications. It integrates with Google Cloud so that you can import a Google Cloud project into a Firebase project. For more information about Firebase, see [Cloud Functions and Firebase](#).

The application used in this lab is an online Quiz application. You add Firebase authentication and then configure authentication to use a simple email address and password credential, and ensure that users must register and log in before taking a quiz.

Objectives

In this lab, you will perform the following tasks:

- Register a Google Cloud project with Firebase.
- Add Firebase configuration to a client-side web application.
- Write Java code to integrate Firebase Authentication into a client-side web application.

Setup and requirements

Qwiklabs setup

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.

Open Google Console

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)

Username
google2727032_student@qwiklabs.n

Password
k68CZxsxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

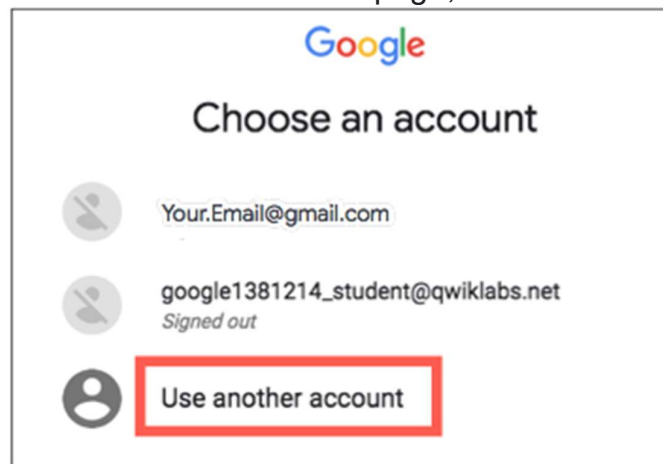
[New to labs? View our introductory video!](#)

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another**



Account.

3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

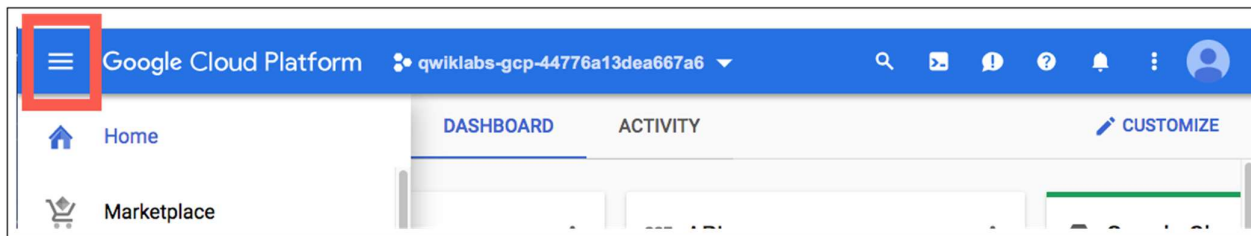
4. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-

left.

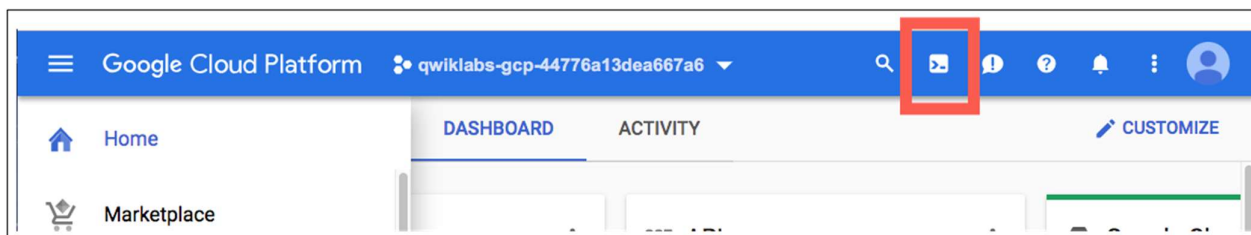


The Google Cloud Shell

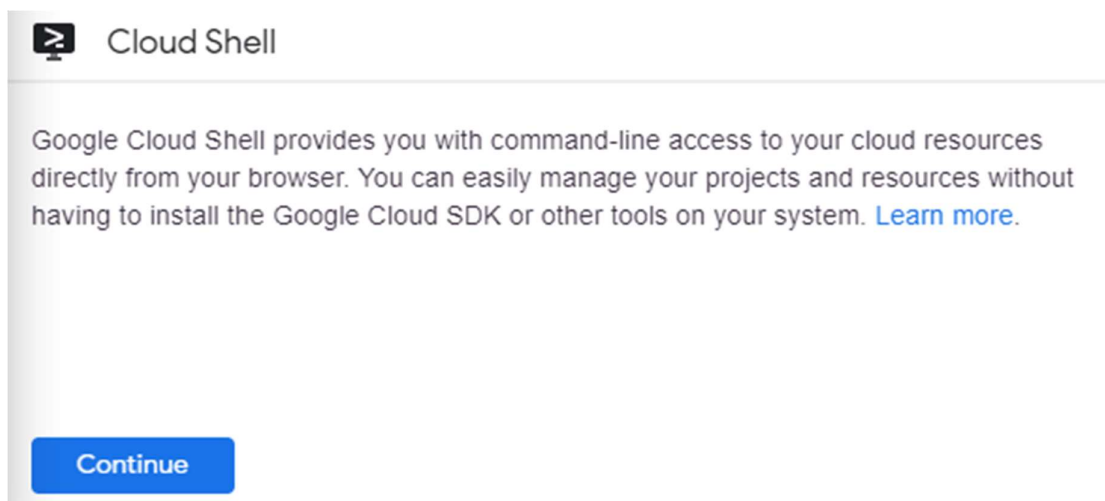
Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

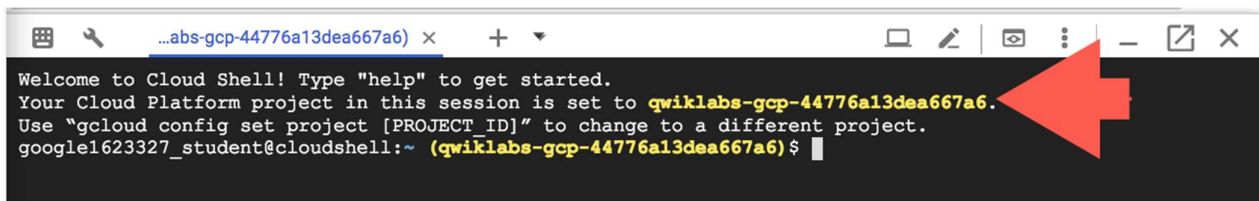
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.



It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



```
...abs-gcp-44776a13dea667a6) x + ▾
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6) $
```

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

```
gcloud auth list
```

(Output)

```
Credentialed accounts:
- <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:
- google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)

```
[core]
project = <project_ID>
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

For full documentation of `gcloud` see the [gcloud command-line tool overview](#).

Prepare the Quiz application

In this section, you access Cloud Shell, clone the git repository containing the Quiz application, configure environment variables, and run the application.

Clone source code in Cloud Shell

To clone the repository for this lab enter the following command:

```
git clone https://github.com/GoogleCloudPlatform/training-data-analyst
```

Configure and run the case study application

1. Change to the working directory::

```
cd ~/training-data-analyst/courses/developingapps/java/firebase/start
```

2. Configure the application dependencies:

```
3. . prepare_environment.sh
```

This script file:

- Creates an App Engine application.
- Creates a Cloud Storage bucket named `gs://[Project-ID]-media`.
- Exports two environment variables: `G_CLOUD_PROJECT` and `G_CLOUD_BUCKET`.
- Runs `mvn clean install`.
- Creates entities in Cloud Datastore.
- Prints out the Project ID.

When the application dependencies are configured, you will see the following output:

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 8.225 s  
[INFO] Finished at: 2018-05-16T13:01:20-04:00  
[INFO] Final Memory: 21M/51M  
[INFO] -----
```

Click *Check my progress* to verify the objective.

4. To run the application, enter the following command:

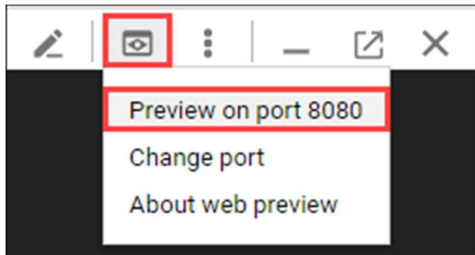
```
5. mvn spring-boot:run
```

The application has started when you see the last line of the output, which looks similar to the following example output:

```
08:01:40.245 [restartedMain] INFO c.g.training.appdev.QuizApplication - Started QuizApplication in 12.428 seconds (JVM running for 13.485)
```

Start the Quiz application

In Cloud Shell, click **Web preview** > **Preview on port 8080** to preview the Quiz application.



Leave this window open for now, because you will need the Cloud Shell Web preview domain (which is in the form: 8080-dot-2958229-dot-devshell.appspot.com) later in this lab.

Examine the Quiz application code

In this section and throughout the lab you'll review the Quiz application code in a code editor. You can use the shell editors that are installed on Cloud Shell, such as nano or vim, or use the Cloud Shell code editor. This lab uses the Cloud Shell code editor.

Launch the Cloud Shell text editor

From Cloud Shell, click on the **Open Editor** icon in the top ribbon. Click **Open in a new window**.

Navigate to the `/training-data-analyst/courses/developingapps/java/firebase/start` folder using the file browser panel on the left side of the editor.

In the `firebase` folder, notice the `end` folder. The `end` folder contains the same files as the `start` folder, but each file in the `end` folder contains the complete code required to perform this lab.

Review the client application

1. Select the `index.html` in the `.../src/main/resources/static/client` folder. Do not confuse with `index.html` in the `static` folder.

This file is the single page in the AngularJS SPA. It contains `<script></script>` tags for the application libraries and code and markup where the SPA will render dynamic output.

2. Select `qiq-login-template.html` in the `.../src/main/resources/static/client/app/auth/` folder.

This file contains the AngularJS template for the Login component.

Notice how it contains a couple of textboxes and a button. The button has an event handler that runs code when it is clicked.

3. Select `qiq-login.js`.

This file contains an AngularJS component. It allows the user to log in to the application or to navigate to a registration page.

Work with Firebase

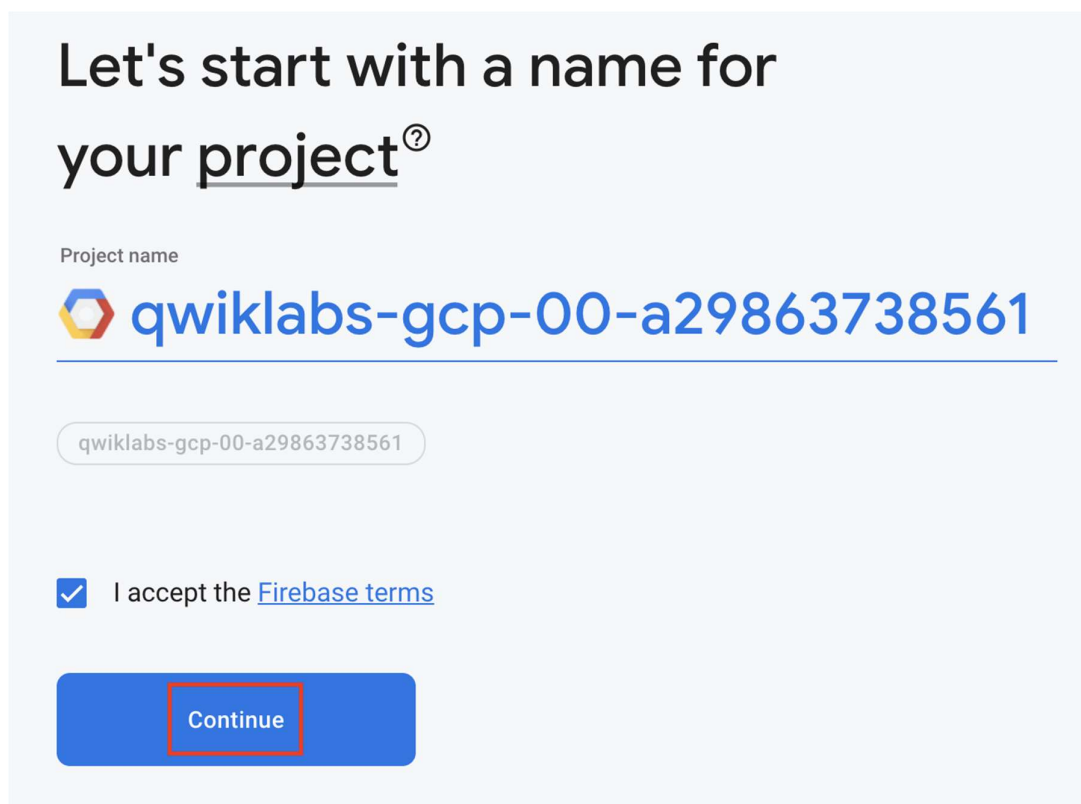
In this section, you will create a Firebase project for your Google Cloud project and enable the Authentication feature.

Create a Firebase project

1. From the Google Cloud console, open a new browser tab and navigate to <https://console.firebase.google.com/>.
Be sure to open the new tab from the Google Cloud console window to stay in the lab environment. You may need to sign in again using your Qwiklabs username and password.


Add your project to Firebase:

1. On the **Welcome to Firebase** page, click **Add project**.
2. In the **Add a project** dialog, select your Qwiklabs Google Cloud Project from the **Project name** dropdown.
3. Accept the Firebase terms and conditions and click **Continue**.



Let's start with a name for
your project[?]

Project name

 **qwiklabs-gcp-00-a29863738561**

qwiklabs-gcp-00-a29863738561

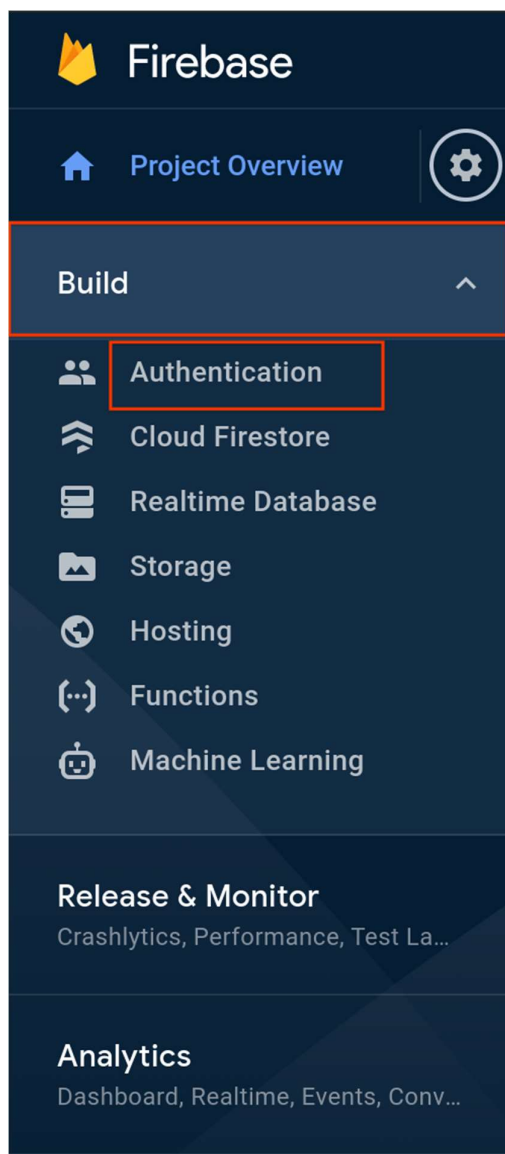
☒ I accept the [Firebase terms](#)

Continue

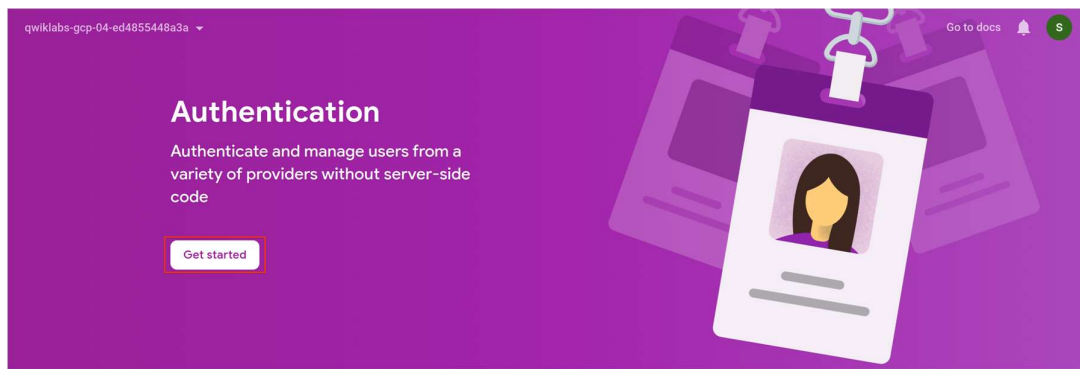
4. In the **Confirm Firebase billing plan** dialog, click **CONFIRM PLAN**.
5. Click **Continue**.
6. Click **Continue**.
7. Accept the final two terms of conditions and then click **Add Firebase**.
8. Then click **Continue**.

Configure Firebase Authentication

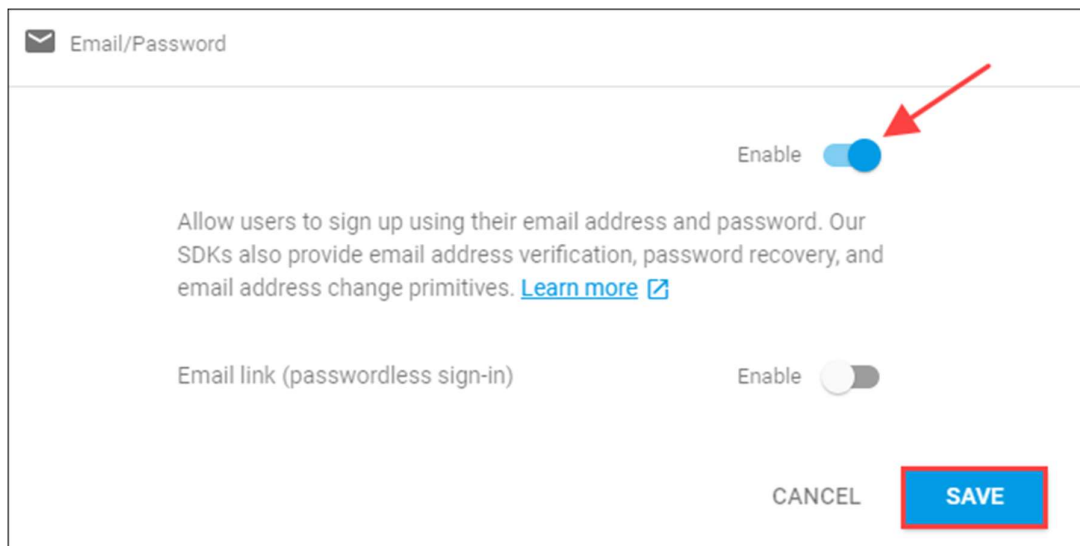
1. In the left pane of the Firebase page, click **Build > Authentication**.



2. On the **Authentication** page, click **Get started**.



3. On the **Sign-in providers** page, hover over the Email/Password provider, and then click the edit icon (a pencil).
4. Click **Enable** to allow users to sign up using their email address and password. Click **Save**.



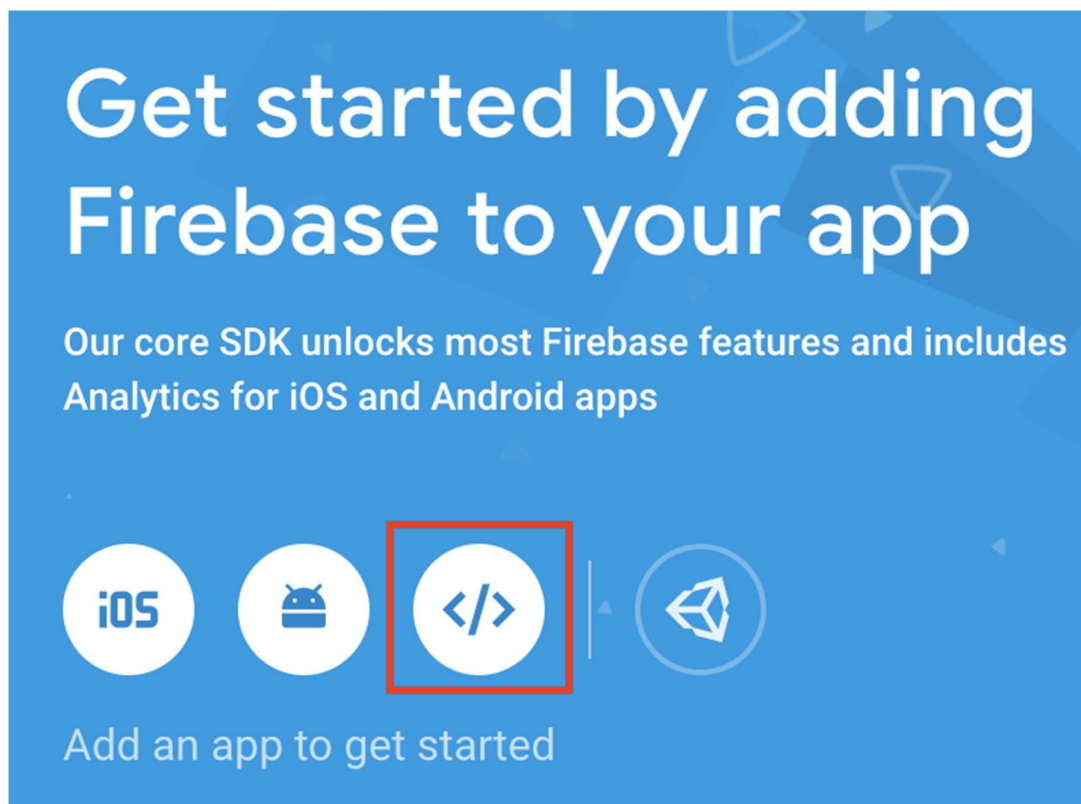
5. Under **Authorized Domains**, click **ADD DOMAIN**.
6. Copy the domain format `8080-dot-2999129-dot-devshell.appspot.com` and paste into the **Domain** text box, then click **Add**.

Integrate a Client-Side Web Application with Firebase

In this section, you will apply your Firebase configuration to your client-side web application.

Apply Firebase configuration to a web application

1. Click **Project Overview** from the left-hand navigation menu. From the "Get started by adding Firebase to your app" console, select the web icon:



2. This will bring you to a "Add Firebase to your web app" page. For the **App nickname** field, enter in **firebase_app**. Then click **Register app**.

Click *Check my progress* to verify the objective.

3. You will now be brought to the "Add Firebase SDK" page. To copy the Firebase configuration, click the copy button in the bottom right corner of the code block. A sample configuration is shown below.

Firebase configuration example (DO NOT COPY)

```
<!-- DO NOT COPY THIS BLOCK - IT'S A SAMPLE -->
```

```

<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/6.2.0/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#config-web-app -->

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyAZNRXW1DylpbJu2fmfjIcyKYtz0nOz7Pc",
    authDomain: "qwiklabs-gcp-9a7cb292df849681.firebaseio.com",
    databaseURL: "https://qwiklabs-gcp-9a7cb292df849681.firebaseio.com",
    projectId: "qwiklabs-gcp-9a7cb292df849681",
    storageBucket: "qwiklabs-gcp-9a7cb292df849681.appspot.com",
    messagingSenderId: "776822443186",
    appId: "1:776822443186:web:5c5e92b0d93ae206"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>

```

4. In the **Cloud Shell code editor**, open and update .../src/main/resources/static/client/index.html:

Paste the Firebase configuration markup just before the other `<script></script>` tags at the bottom of the page.

5. Now replace this this line of code that you copied over:

```
<script src="https://www.gstatic.com/firebasejs/7.14.2/firebase-app.js"></script>
```

with the following:

```
<script src="https://www.gstatic.com/firebasejs/4.2.0/firebase.js"></script>
```

5. Click the **Open Terminal** icon.
6. In Cloud Shell, press Ctrl-C to stop the application, and then re-enter the following command to restart your application.

```
mvn spring-boot:run
```

Run the application

1. Return to the Quiz application and refresh your browser.
2. In the navigation bar, click **Take Test**.

You should not be able to take a test without being logged in.

3. Register for access. In the navigation bar, click **Register**.

Enter any email address and password into the form fields, and then click **Register**.

You will have to use a password with a certain level of complexity, like abcd1234\$.
You should be logged in and redirected to the Google Cloud quiz.

4. In the navigation bar, click **Logout**.

You should be logged out and redirected to the Quiz homepage.

Try logging in again, register another identity, and have fun taking the quiz.

Congratulations!

This concludes the self-paced lab, App Dev: Adding User Authentication to Your Application - Java! You added Firebase authentication to an online Quiz application, and then configured authentication to ensure that users must register and log in before taking a quiz.

Finish your Quest



This lab is part of the [Application Development - Java](#) and [Cloud Development](#) Quests. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in this Quest and get immediate completion credit if you've taken this lab. See other available [Qwiklabs Quests](#).

Take your next lab

Take another lab in this Qwiklabs Quest, [App Dev: Deploying the Application into Kubernetes Engine - Java](#), or take another lab:

- [Cloud Security Scanner: Qwik Start](#)
- [Rent-a-VM to Process Earthquake Data](#)

Next steps

- Learn more about [Java on the Google Cloud](#).
- Get up to speed with Firebase, take [Firebase SDK for Cloud Functions](#), or check out [Google Cloud Datastore Documentation](#).

Manual Last Updated: March 26, 2021

Lab Last Tested: March 26, 2021

Copyright 2021 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.