



Firebase Cloud Functions

[Firebase Cloud Functions](#) provide cloud functions inside the Firebase ecosystem. This means that we no longer need to provision servers and pay for the servers to remain idle.

The following example written in Javascript and ES6/ES2015 modules is as simple a function as we can make. the function acts upon a HTTPS request and take the subject parameter and use it to generate the response.

```
const functions = require('firebase-functions')

module.exports = {
  hello: functions.https.onRequest((req, res) => {
    const subject = req.query.subject || 'World'
    res.send('World' + `Hello ${subject}!`)
  }),
}
```

Once you've installed the function, you can call it using the following URL:

```
https://url-to-functions-server.net/hello
```

If you call it without parameters it will display **hello world!**. If you pass a parameter in the URL, like this:

```
https://url-to-functions-server.net/hello?subject=mom
```

Will display **hello mom!**

So now that we know what the function itself looks like.

Configuring and installing functions

The function itself is simple but the configuration takes some significant work.

For this example we'll start from scratch. It's always better to start from scratch than having to retrofit an existing set of functions.

The requirements for the project are:

- Have an existing Firebase account
- Enable the Blaze billing plan
- Have Node installed

The first step is to install the Firebase Tools package as a global install. This will make the `firebase` command available for later steps in the process

```
npm install -g firebase-tools
```

To initialize the process use the following command:

```
firebase init
```

It will produce output similar to the one below.

We first

```
You're about to initialize a Firebase project in this directory:
```

```
/Users/carlos/code/temp-functions
```

```
? Which Firebase CLI features do you want to set up for this folder? Press
```

```
: Configure and deploy Cloud Functions
```

```
=== Project Setup
```

```
First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.
```

```
? Please select an option: Use an existing project
```

```
? Select a default Firebase project for this directory: layout-experiments
```

i Using project layout-experiments (layout-experiments)

=== Functions Setup

A functions directory will be created in your project with a Node.js package pre-configured. Functions can be deployed with `firebase deploy`.

? What language would you like to use to write Cloud Functions? JavaScript
? Do you want to use ESLint to catch probable bugs and enforce style? No

- ✓ Wrote functions/package.json
- ✓ Wrote functions/index.js
- ✓ Wrote functions/.gitignore

? Do you want to install dependencies with npm now? Yes

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
i Writing gitignore file to .gitignore...

✓ Firebase initialization complete!