REST APIS AND DEPLOYING TO FIREBASE FUNCTIONS

Overview

- □ Introduction to REST APIs
- □ Configuring Firebase functions on our apps
- □ Examining our first NodeJS code
- □ Modules in JavaScript
- □ Deploying our NodeJS functions to Firebase

REST API’s

https://www.youtube.com/watch?v=7YcW25PHnAA&t=82s

Application Programming Interface (API)

- An API expresses a software component in terms of its operations, inputs, outputs and underlying types

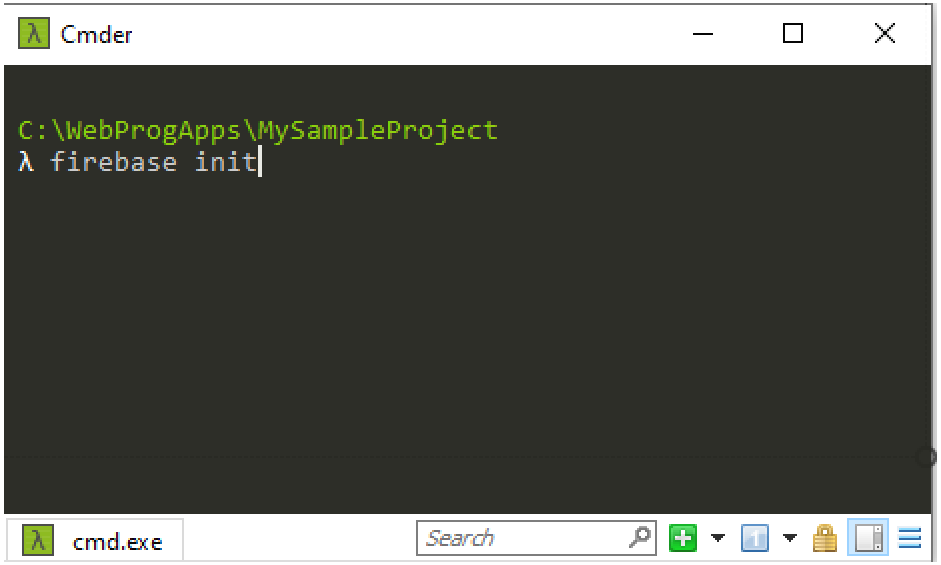
- RESTful APIs
  - **RE**presentational **S**tate **T**ransfer
- Use HTTP methods (verbs), GET, POST …

- Interface is exposed and can be invoked without caring about the underlying implementation

- Accessed via a URL

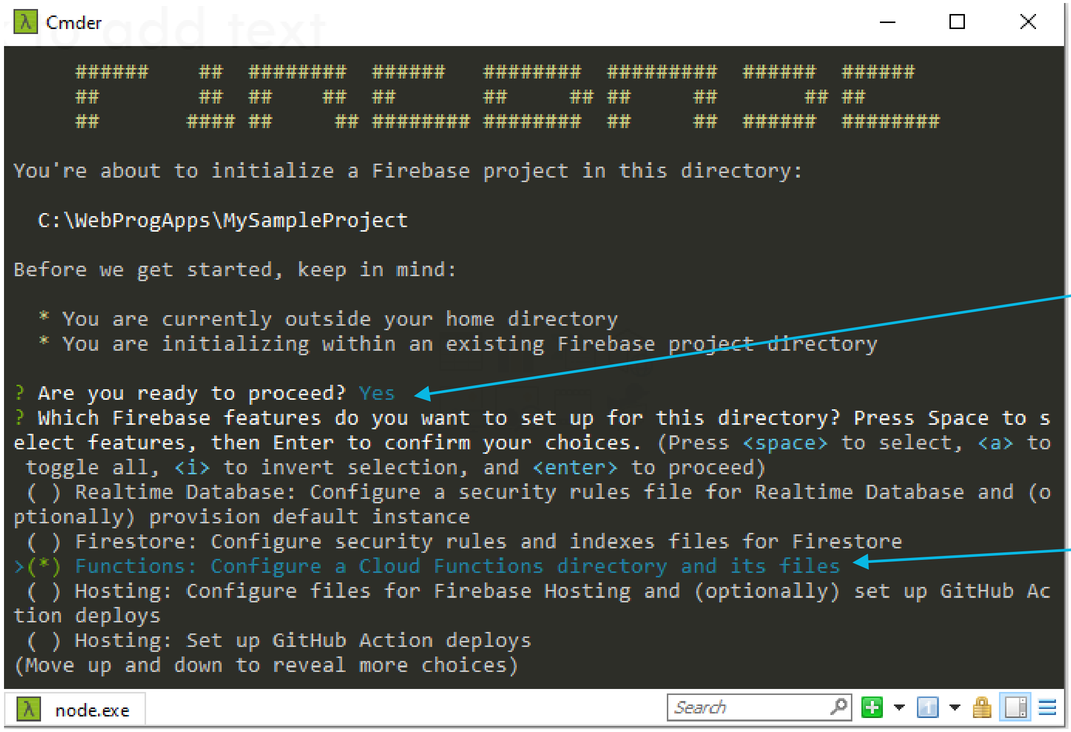
Adding Firebase Functions

- We are using Firebase functions to host our REST APIS
- Thus as when we setup hosting we must now initialise functions within our projects
- Using the command line, navigate to the directory containing your project
- firebase init



```
C:\WebProgApps\MySampleProject
λ firebase init
```

Adding Firebase Functions cont.

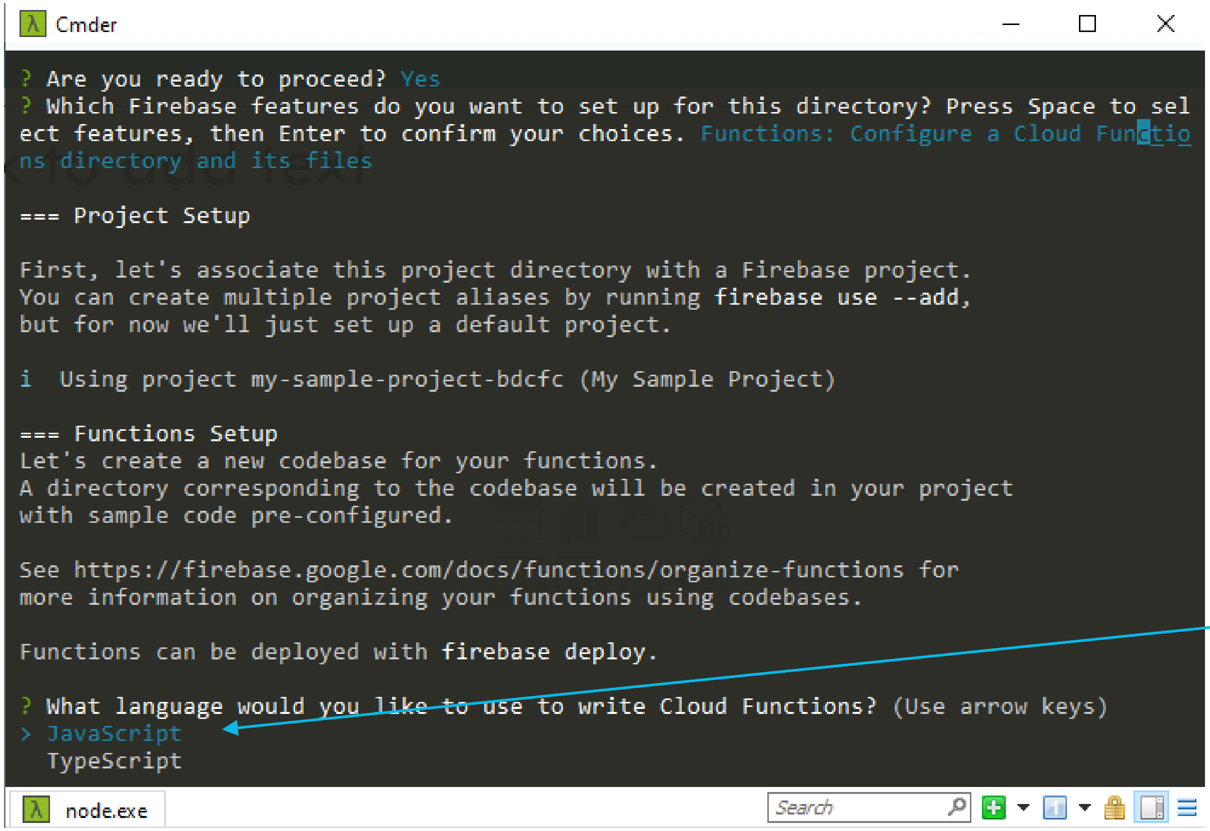


```
     ######    ##  ########  ######   ########  #########  ######  ######
     ##        ##  ##    ##  ##       ##     ## ##     ##       ## ##
     ##       #### ##     ## ######## ########  ##     ##  ######  ########

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

  C:\WebProgApps\MySampleProject

Before we get started, keep in mind:

  * You are currently outside your home directory
  * You are initializing within an existing Firebase project directory

? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to s
elect features, then Enter to confirm your choices. (Press <space> to select, <a> to
 toggle all, <i> to invert selection, and <enter> to proceed)
 ( ) Realtime Database: Configure a security rules file for Realtime Database and (o
ptionally) provision default instance
 ( ) Firestore: Configure security rules and indexes files for Firestore
>(*) Functions: Configure a Cloud Functions directory and its files
 ( ) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Ac
tion deploys
 ( ) Hosting: Set up GitHub Action deploys
(Move up and down to reveal more choices)
```

Select Yes

Select functions and hit enter

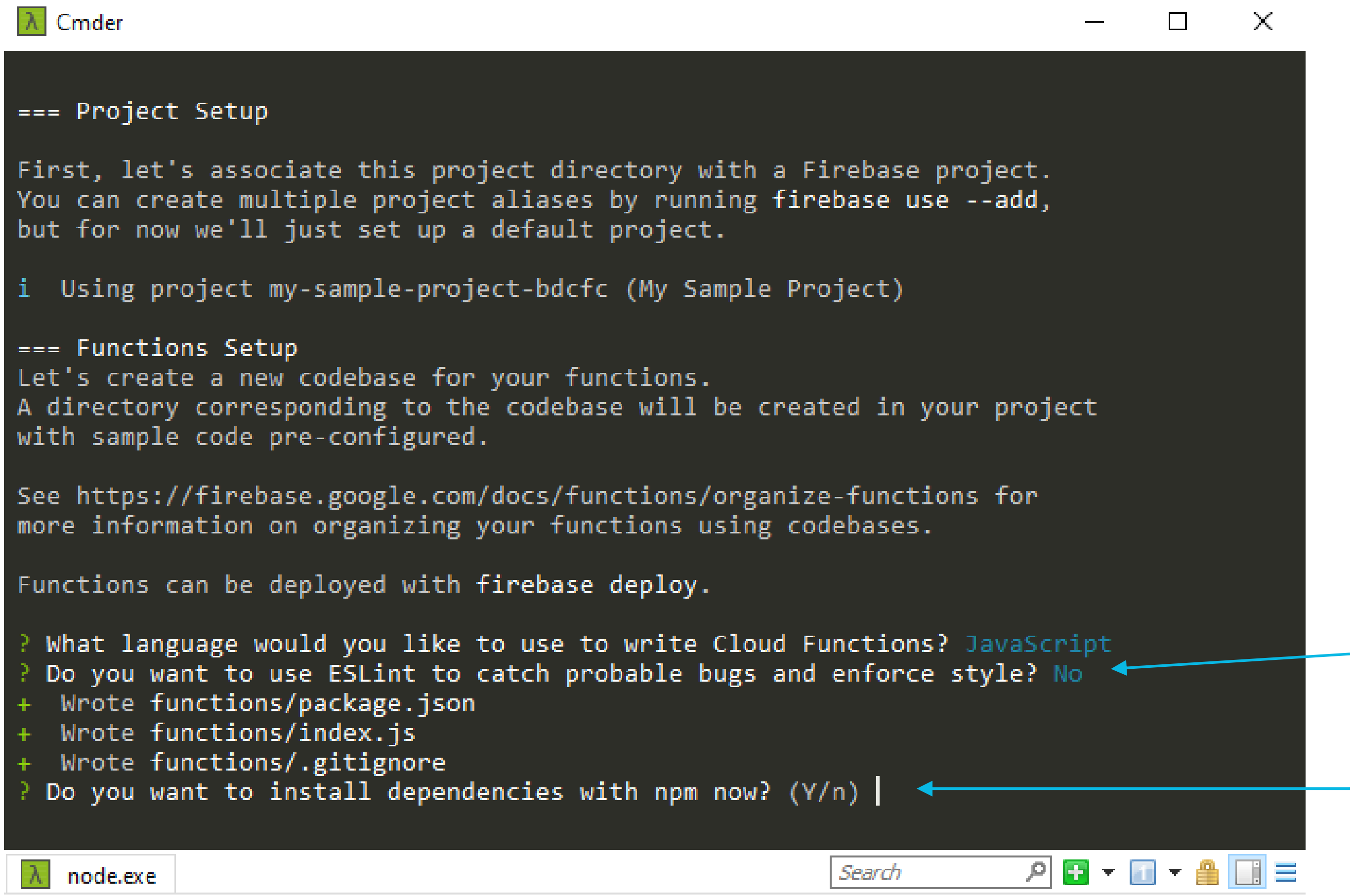
Adding Firebase Functions cont.



```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to sel
ect features, then Enter to confirm your choices. Functions: Configure a Cloud Functio
ns directory and its files

=== 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.

i  Using project my-sample-project-bdcfc (My Sample Project)

=== Functions Setup
Let's create a new codebase for your functions.
A directory corresponding to the codebase will be created in your project
with sample code pre-configured.

See https://firebase.google.com/docs/functions/organize-functions for
more information on organizing your functions using codebases.

Functions can be deployed with firebase deploy.

? What language would you like to write Cloud Functions? (Use arrow keys)
> JavaScript
  TypeScript
```

Select JavaScript

Adding Firebase Functions cont.

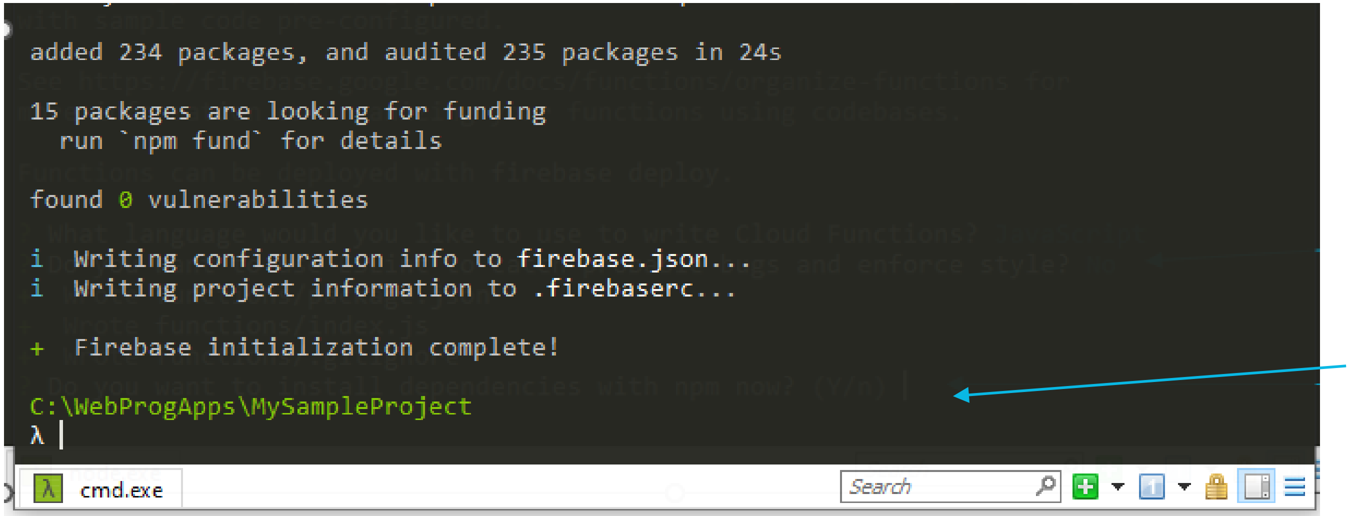


```
=== Project Setup

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.

i  Using project my-sample-project-bdcfc (My Sample Project)

=== Functions Setup
Let's create a new codebase for your functions.
A directory corresponding to the codebase will be created in your project
with sample code pre-configured.

See https://firebase.google.com/docs/functions/organize-functions for
more information on organizing your functions using codebases.

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? (Y/n)
```

Select no to ESLint

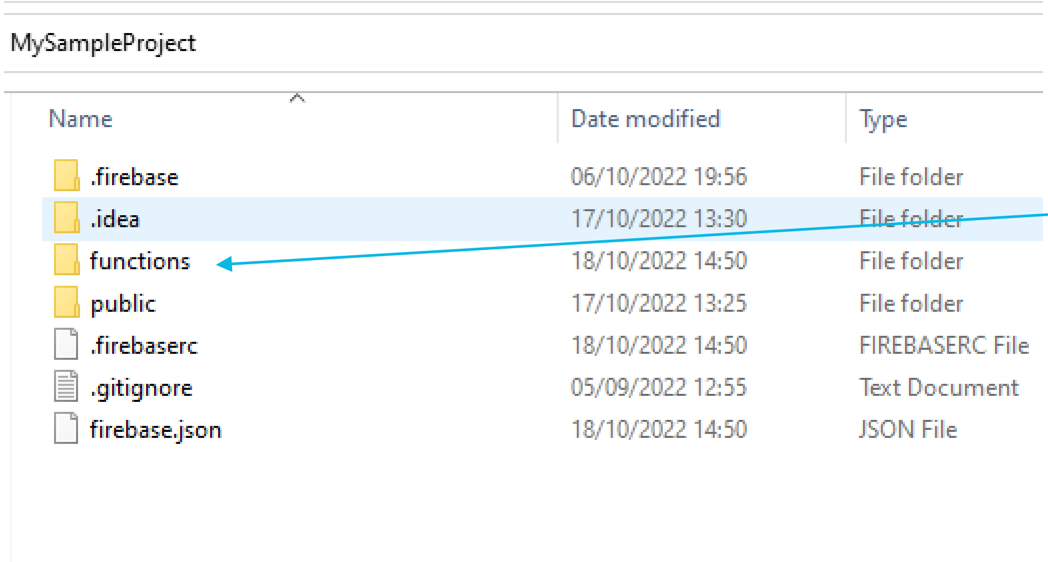
Select Yes to installing the dependencies

Adding Firebase Functions cont.



```
added 234 packages, and audited 235 packages in 24s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

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

+  Firebase initialization complete!

C:\WebProgApps\MySampleProject
λ |
```

Finished installing the dependencies

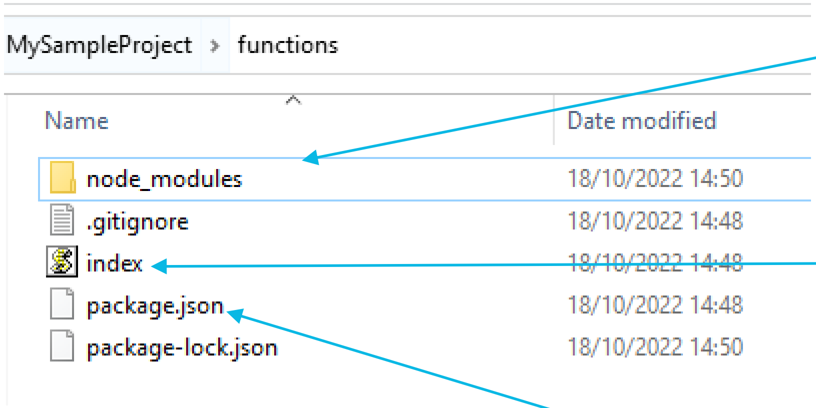


MySampleProject

Name	Date modified	Type
.firebase	06/10/2022 19:56	File folder
.idea	17/10/2022 13:30	File folder
functions	18/10/2022 14:50	File folder
public	17/10/2022 13:25	File folder
.firebaserc	18/10/2022 14:50	FIREBASERC File
.gitignore	05/09/2022 12:55	Text Document
firebase.json	18/10/2022 14:50	JSON File

Now be a new functions directory

Examining the functions folder



MySampleProject › functions

Name	Date modified
node_modules	18/10/2022 14:50
.gitignore	18/10/2022 14:48
index	18/10/2022 14:48
package.json	18/10/2022 14:48
package-lock.json	18/10/2022 14:50

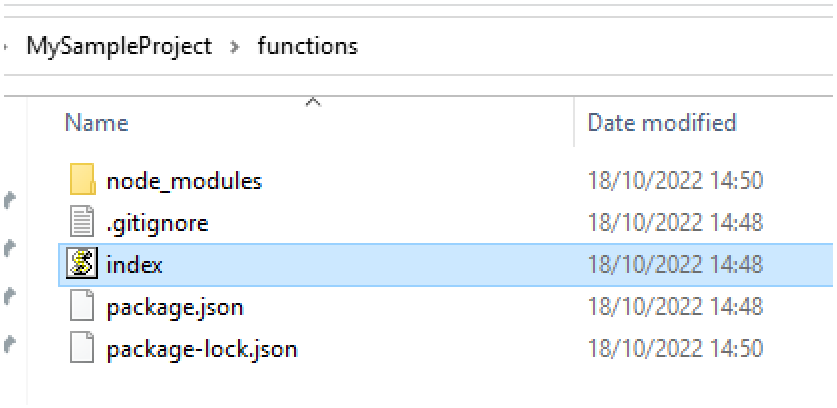
Contains dependencies

Where you will write your NodeJS code and functions

Allows you to define dependencies and other config settings

Examine the code

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

// // Create and Deploy Your First Cloud Functions
// // https://firebase.google.com/docs/functions/write-firebase-functions
//
// exports.helloWorld = functions.https.onRequest((request, response) => {
//   functions.logger.info("Hello logs!", {structuredData: true});
//   response.send("Hello from Firebase!");
// });
```



MySampleProject › functions

Name	Date modified
node_modules	18/10/2022 14:50
.gitignore	18/10/2022 14:48
index	18/10/2022 14:48
package.json	18/10/2022 14:48
package-lock.json	18/10/2022 14:50

What is a library

- Libraries in programming are just blocks of code that you import so your program can use them?

```
require('firebase-functions’);
```

- Who writes them?
  - Other Devs, yourself, could be anyone really…

- Why use them?
  - You can’t use Firebase without this library.
    - It’s like trying to control your TV without a remote
    - It’s like trying to drive a car without a steering wheel, pedals etc.

- It’s their platform so you must use their interface!

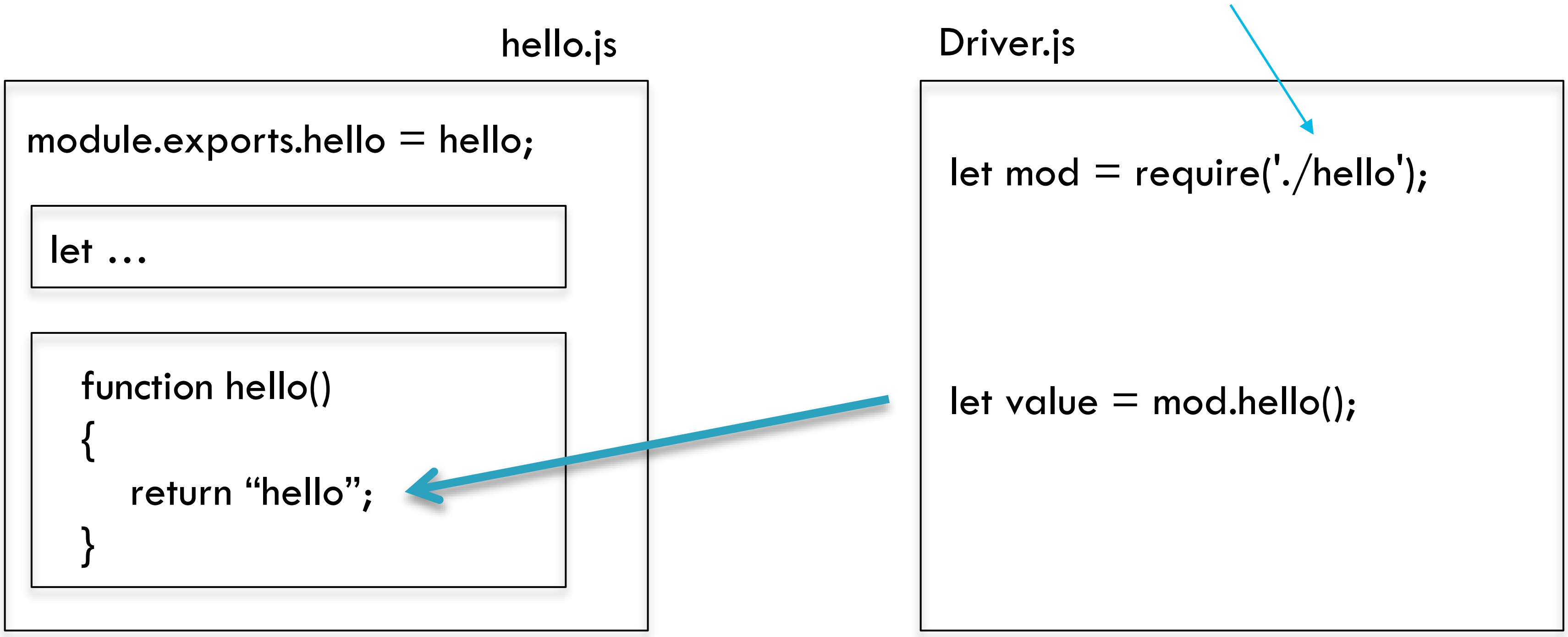
JS modules

- JavaScript allows the developer to package up code and functionality into modules

- Think of them as set of packaged functions that you wish to include in your application

JS modules

Both files in the same folder



hello.js

```
module.exports.hello = hello;

let …

function hello()
{
    return “hello”;
}
```

Driver.js

```
let mod = require('./hello');

let value = mod.hello();
```

Target

Source

Modules

- Examining our first REST API (cloud function)

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

// // Create and Deploy Your First Cloud Functions
// // https://firebase.google.com/docs/functions/write-firebase-functions
//
```

Exporting a module “helloWorld”

```
exports.helloWorld = functions.https.onRequest((request, response) => {
  functions.logger.info("Hello logs!", {structuredData: true});
  response.send("Hello from Firebase!");
});
```

Deploying our first cloud function

- □ The command is the same *firebase deploy*!
- □ You will see the function URL outputted to the console
- □ Paste it into the browser address bar to see if it works…Don’t Google it! ☺

Summary

- □ Introduction to REST APIs
- □ Configuring Firebase functions on our apps
- □ Examining our first NodeJS code
- □ Modules in JavaScript
- □ Deploying our NodeJS functions to Firebase