

# Software Studio

## 軟體設計與實驗

# Cocos Creator & Firebase

Hung-Kuo Chu

Department of Computer Science  
National Tsing Hua University

**CS2410**



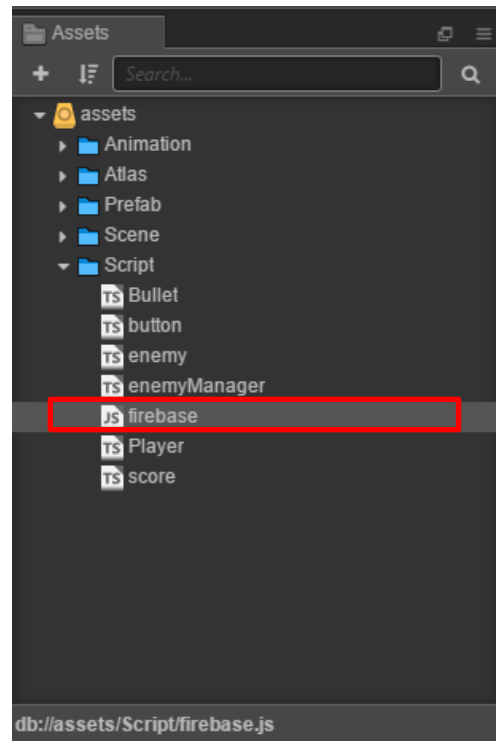
# Firebase API

- Follow the steps and then you can use Firebase API in your Cocos Creator project.
  1. Download **firebase.js** which TA provides
  2. Add file to project
  3. Set file as plugin
  4. Set project config



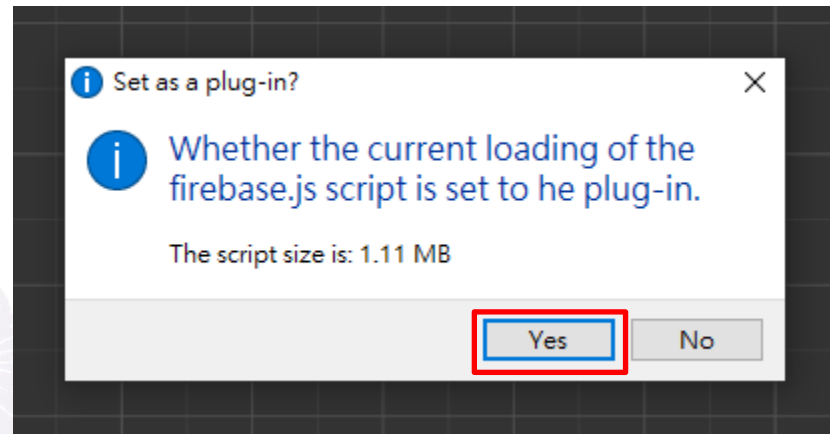
# Import firebase.js

- Download **firebase.js** file provided by TA and load it to your project.



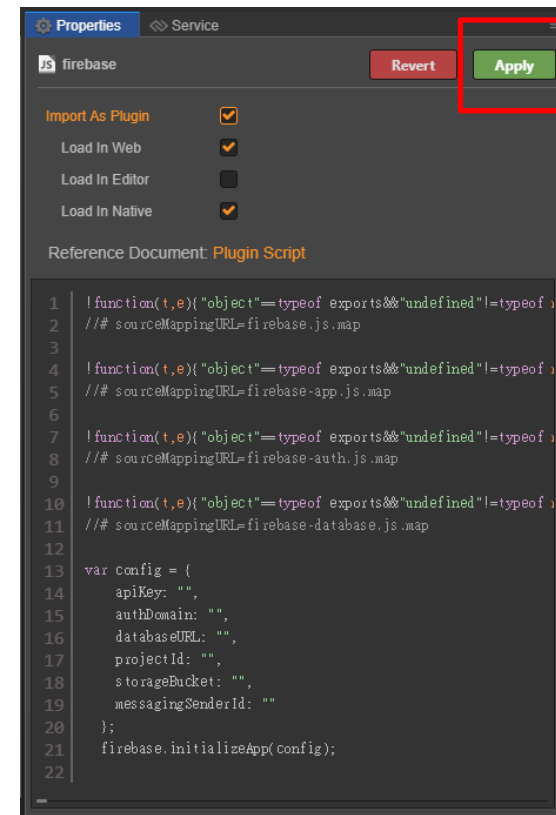
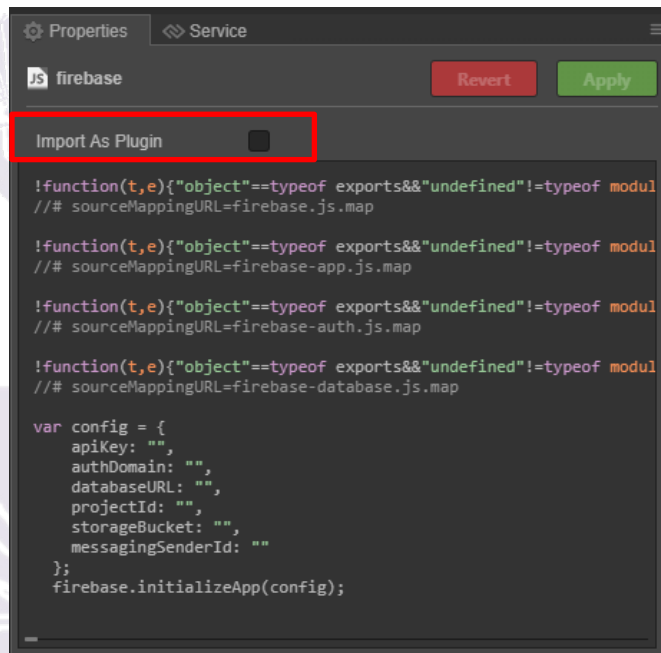
# Set firebase.js as Plugin

- After you drag the file to assets folder, Cocos Creator will ask you whether the file is set to plugin.
- Choose “**Yes**”.



# Set firebase.js as Plugin

- You can also set the file to plugin manually.
- Choose firebase.js, check **Import As Plugin**, and apply it.



# Change Config

- Modify the config in the firebase.js according to your firebase config.

```
o=Dn.prototype.put;return Dn.prototype.put=function(t,e,n,r){void 0!==r&&(r=i()),o.call(this,queryIdentifier()),listens:function(t){return t.repo.persistentConnection_.listens_},forceRegisterService("database",function(t,e,n){return Bn.getInstance().databaseFromApp(t,n)},{RTEST_ACCESS:Jn},null,!0),d())&&(module.exports=Gn)).apply(this,arguments)}catch(t){throw co}});  
//# sourceMappingURL=firebase-database.js.map  
  
var config = {  
  apiKey: "",  
  authDomain: "",  
  databaseURL: "",  
  projectId: "",  
  storageBucket: "",  
  messagingSenderId: ""  
};  
firebase.initializeApp(config);
```

Change to your firebase project config!



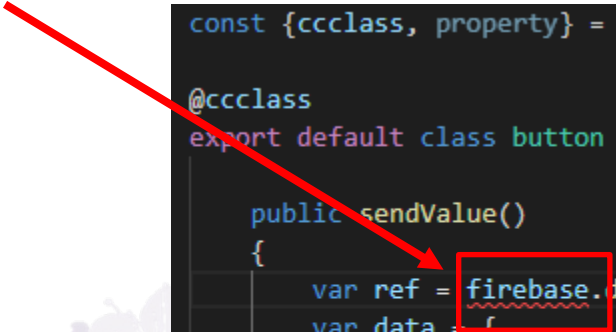
# Use Firebase API

- Now you can use Firebase API which we taught in class in your TypeScript scripts.
- Note that the “firebase not found” error can be ignored.

```
const {ccclass, property} = cc._decorator;

@ccclass
export default class button extends cc.Component {

    public sendValue()
    {
        var ref = firebase.database().ref('test');
        var data = {
            value: 0
        };
        ref.push(data);
    }
}
```



# Use Firebase API

- You can put a declaration of the **firebase** object to make IntelliSense happy. **This is completely optional.**
- Place the following line anywhere in your scripts or put it in “firebase.d.ts”, located at the top level of your project.

```
declare const firebase: any;
```



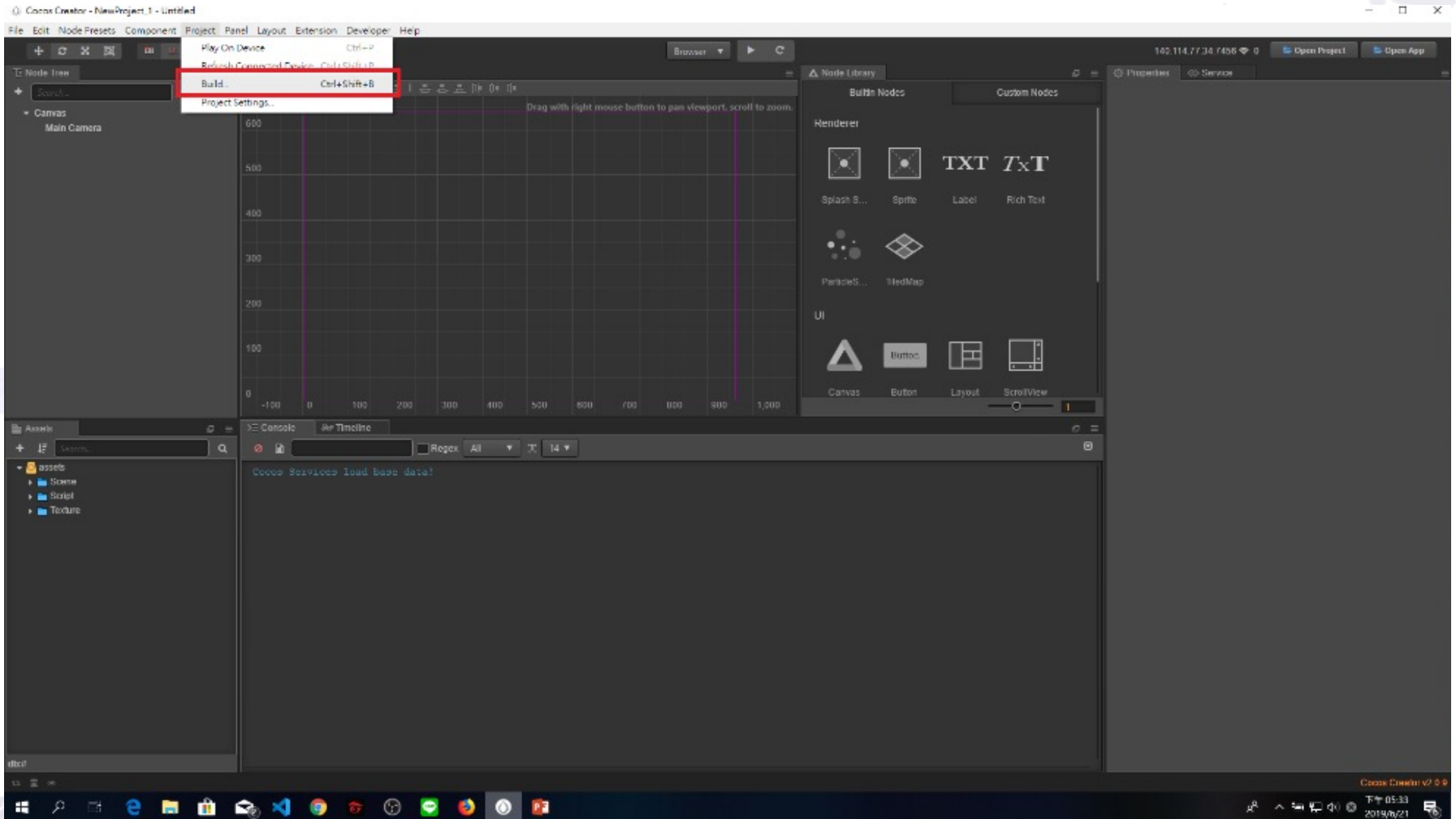


# Deployment

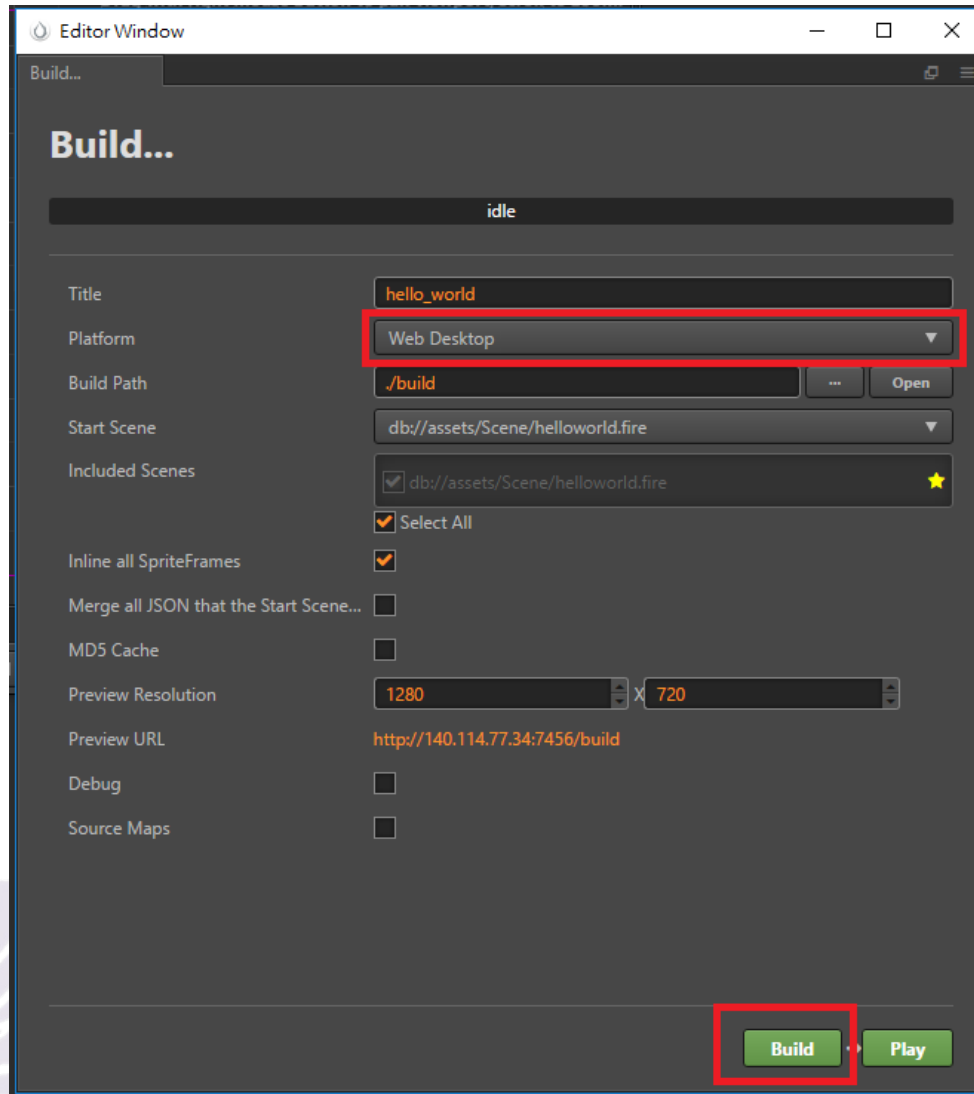
- Follow the steps then you can use firebase deploy to run your Cocos Creator project.
  1. Build in Cocos Creator
  2. Firebase initialization
  3. Set public folder to build folder
  4. Firebase deployment



# Build in Cocos Creator



# Build in Cocos Creator



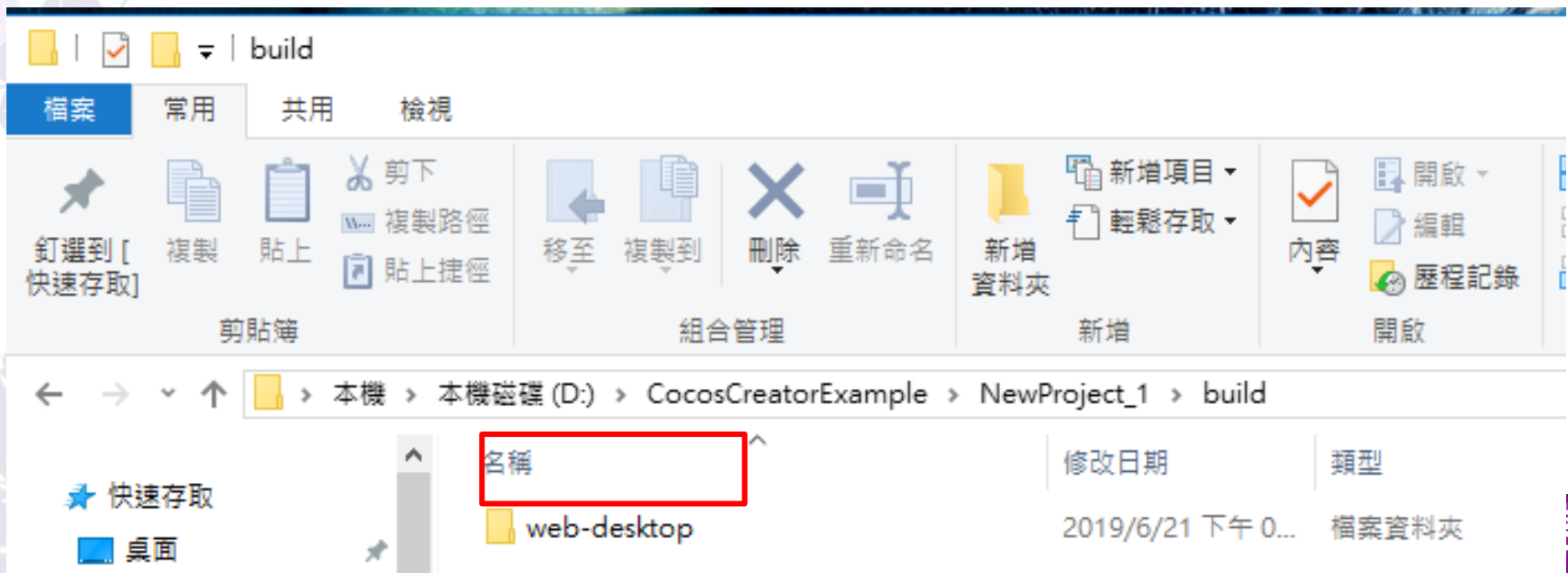
1. Choose **Web Desktop**  
or **Web Mobile**

2. Press Build



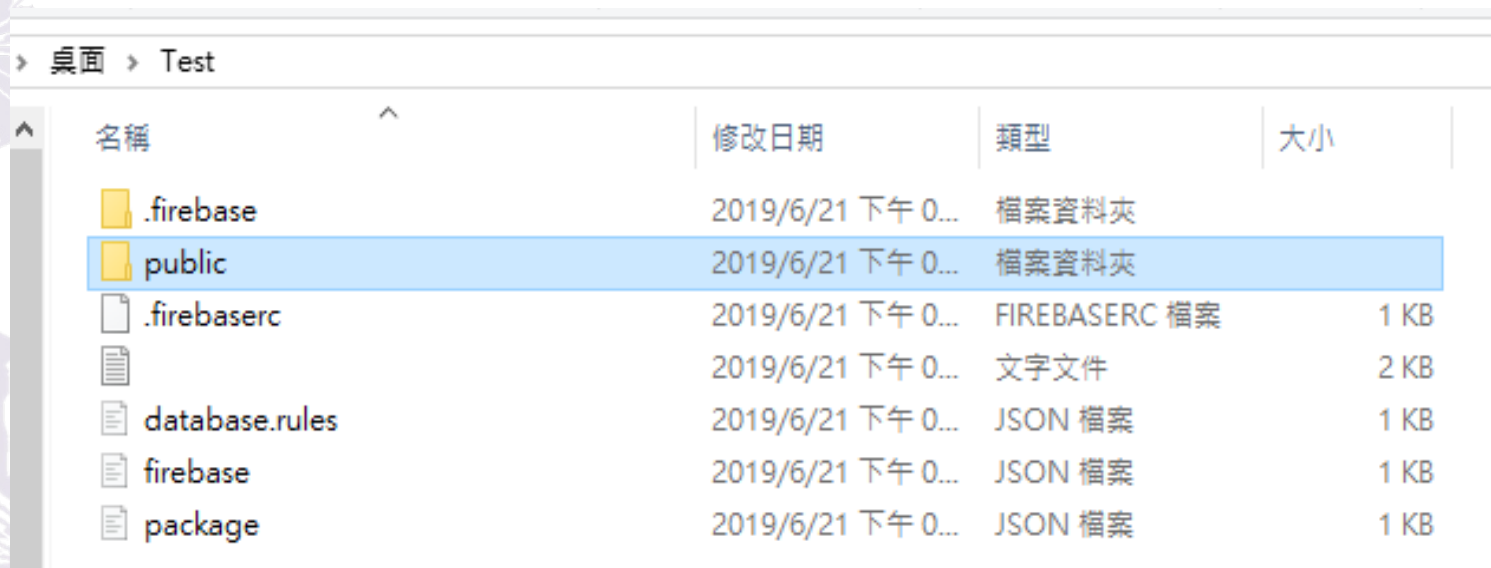
# Build in Cocos Creator

- If built successfully, you can find your source files in **[project\_path]/build/web-desktop** or **[project\_path]/build/web-mobile**



# Firebase initialization

- Run firebase initialization in your project's folder. You can Firebase Hosting Appendix to review how to initialize. Be careful not to overwrite your build folder!



| 桌面 > Test       |                   |               |      |  |
|-----------------|-------------------|---------------|------|--|
| 名稱              | 修改日期              | 類型            | 大小   |  |
| .firebase       | 2019/6/21 下午 0... | 檔案資料夾         |      |  |
| public          | 2019/6/21 下午 0... | 檔案資料夾         |      |  |
| .firebaseconfig | 2019/6/21 下午 0... | FIREBASERC 檔案 | 1 KB |  |
| firebase.json   | 2019/6/21 下午 0... | 文字文件          | 2 KB |  |
| database.rules  | 2019/6/21 下午 0... | JSON 檔案       | 1 KB |  |
| firebase        | 2019/6/21 下午 0... | JSON 檔案       | 1 KB |  |
| package         | 2019/6/21 下午 0... | JSON 檔案       | 1 KB |  |



# Set public directory

- You'll be asked to set the public directory at the **Hosting Setup** step. Use the path **“build/web-desktop”** (or **web-mobile**) as your public directory, so Firebase will deploy your game using your build's files.

## === Hosting Setup

Your **public** directory is the folder (relative to your project directory) that will contain Hosting assets to be uploaded with **firebase deploy**. If you have a build process for your assets, use your build's output directory.

```
? What do you want to use as your public directory? build/web-desktop
? Configure as a single-page app (rewrite all urls to /index.html)? No
? Set up automatic builds and deploys with GitHub? No
+ Wrote build/web-desktop/404.html
? File build/web-desktop/index.html already exists. Overwrite? No
i Skipping write of build/web-desktop/index.html
```



# Set public directory

- If you made a mistake here, you can change it by editing **firebase.json** after initialization. Note the “public” property.

```
firebase.json X
firebase.json > ...
1  {
2    "hosting": {
3      "public": "build/web-desktop",
4      "ignore": [
5        "firebase.json",
6        "**/.*",
7        "**/node_modules/**"
8      ],
9      "rewrites": [
10       {
11         "source": "**",
12         "destination": "/index.html"
13       }
14     ]
15   }
16 }
```



# Firestore Deployment

- Deploy your files to Firestore. You can check Firestore Hosting Appendix to review how to deploy.





# Finish

- Finish all the steps, then you can see your game run on Firebase.



thank  
you!

Question

