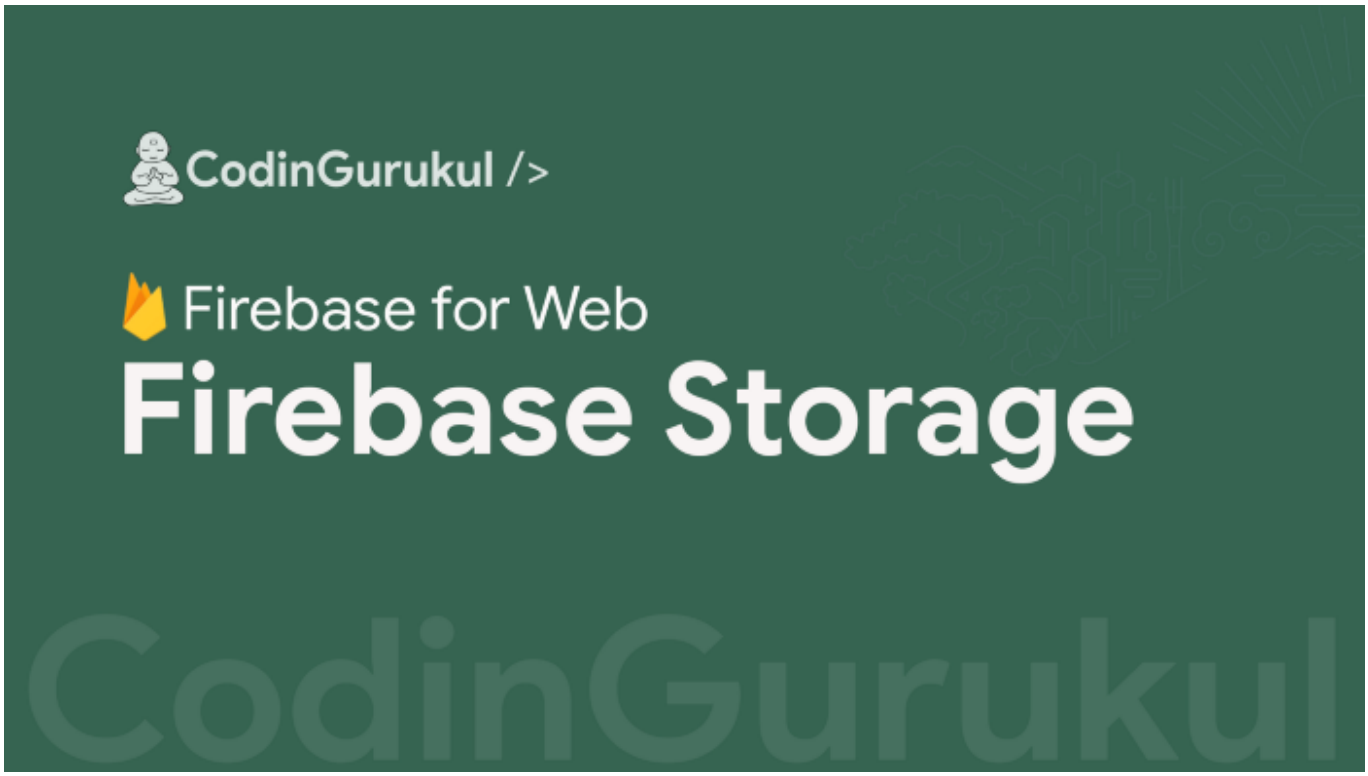

Firebase for Web: Firebase Storage

Firebase for Web: Firebase Storage

Vrijraj Singh Follow
Jan 4, 2019 · 3 min read

Cloud Storage for Firebase helps you to upload and share user-generated content, such as images and video, which allows you to build rich media content into your apps.

It is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps. You can use Firebase SDKs to store images, audio, video, or other user-generated content. Your data is stored in a Google Cloud Storage bucket, an exabyte scale object storage solution with high availability and global redundancy. Cloud Storage lets you securely upload these files directly from mobile devices and web browsers, handling spotty networks with ease.

Lets Start

Step 1:

Create a Firebase Project in your console and setup your project with your web app.

Add Security rule for Firebase Storage

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
       // Only Authenticated user can upload the file
       allow read, write: if request.auth != null;

       // Anyone can upload the file/ No Auth Require
       // allow read, write: true;
    }
  }
}
```

Step 2:

In this example, I have created a simple file input tag and add a JavaScript event `onchange()` with id `files`.

```
<input type="file" onchange="uploadFile()" id="files" name="files[]"
multiple />
```

Step 3:

Create a reference for Firebase Storage, In order to upload or download files, delete files, or get or update metadata, you must create a reference to the file you want to operate on. A reference can be thought of as a pointer to a file in the cloud. References are lightweight, so you can create as many as you need, and they are also reusable for multiple operations.

Create references from the `storage()` service in your Firebase app. This reference points to the root of your Cloud Storage bucket.

```
// Created a Storage Reference with root dir
var storageRef = firebase.storage().ref();
```

Get the file from the DOM and Create a reference to file

```
// Get the file from DOM
var file = document.getElementById("files").files[0];
console.log(file);

//dynamically set reference to the file name
var thisRef = storageRef.child(file.name);
```

Upload from a `Blob` or `File` Once you've created an appropriate reference, you then call the `put()` method. `put()` takes files via the JavaScript File and Blob APIs and uploads them to Cloud Storage.

```
//put request upload file to firebase storage

thisRef.put(file).then(function(snapshot) {
    console.log('Uploaded a blob or file!');
});
```

and Complete code:

```
<input type="file" onchange="uploadFile()" id="files" name="files[]"
multiple />

<script>

    //function to save file
    function uploadFile(){

      // Created a Storage Reference with root dir
      var storageRef = firebase.storage().ref();

      // Get the file from DOM
      var file = document.getElementById("files").files[0];
      console.log(file);
```

```
        //dynamically set reference to the file name
        var thisRef = storageRef.child(file.name);

        //put request upload file to firebase storage
        thisRef.put(file).then(function(snapshot) {
           alert("File Uploaded")
           console.log('Uploaded a blob or file!');
        });
      }

</script>
```

I would suggest you go through the documentation for more clarity on this.
https://firebase.google.com/docs/storage/

Well, that was it for now.
Next time, I'll tell you how to retrieve the files back from the storage and add user's details to Firebase Database.

Thanks

Happy Coding!

JavaScript Firebase Firebase Storage Storage Cloud Stroage