

Firestore Application Development – Task Management System

Aravindhana, Venkiteswamy Manimagalai

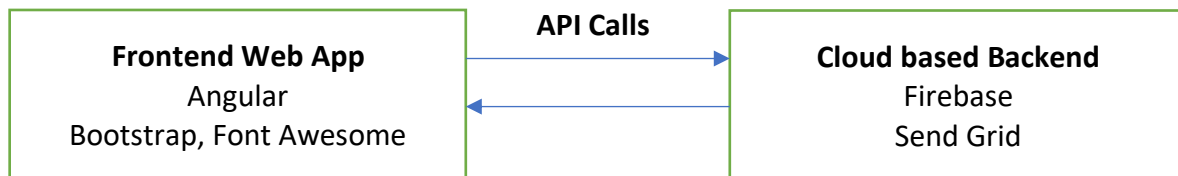
avenkit2@binghamton.edu

GitHub Link

Abstract:

Our main goal is to build a server less application with firebase products. In this app I have used firebase products like authentication, cloud firestore, cloud storage, hosting and cloud functions. The application is to build a task management system named Task it to demonstrate and explore above mentioned various firebase products.

System Design:



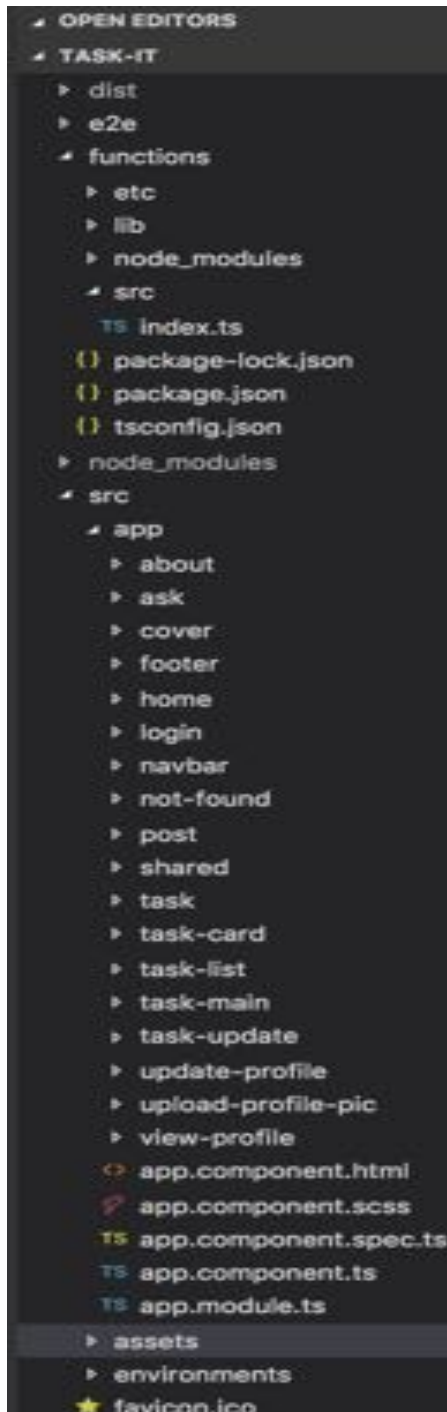
How Task It works?

- Sign-up/set-up your account/profile using any given social media.
- Link friends, family, colleague, employee, and etc. accounts.
- You can give or ask for a task to - or from your friend.
- If you give the task, then your friend should accept the task.
- If you received the task you can accept the same and start to work on it.
- You can also ask for a task from your friend.
- You can add relevant documents to the task.
- You can give the regular update on the progress of the task.
- Receive regular emails for requests and updates of the task.

Web front end:

Developed in HTML 5, CSS3, Typescript(ES6), Angular 5, Bootstrap 4

[Angular](#) (commonly referred to as "Angular 2+" or "Angular v2 and above") is a Typescript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations.



```
Angular CLI: 1.7.3
Node: 8.1.2
OS: darwin x64
Angular: 5.2.9
... animations, common, compiler, compiler-cli, core, forms
... http, language-service, platform-browser
... platform-browser-dynamic, router

@angular/cdk: 5.2.4
@angular/cli: 1.7.3
@angular/material: 5.2.4
@angular-devkit/build-optimizer: 0.3.2
@angular-devkit/core: 0.3.2
@angular-devkit/schematics: 0.3.2
@ngtools/json-schema: 1.2.0
@ngtools/webpack: 1.10.2
@schematics/angular: 0.3.2
@schematics/package-update: 0.3.2
typescript: 2.5.3
webpack: 3.11.0
Aravinds-MacBook-Pro:task-it aravind$
```

- As shown above, this App is developed using Angular version 5.2.9 and Angular CLI version 1.7.3.
- Various components used in the angular are shown in the left image
- Used typescript as the working JS language along with sass CSS (.scss) for CSS files.
- Angular uses web API call to communicate with the firebase.
- Angular firestore 4 library is used to access and implement the API's.
- Shared service is used for firebase authentication along with various validation and pipe services.
- Production and development environment are configured under environments folder.
- Firebase functions is also implemented in the same folder as of Angular CLI.
- Separate config files for Angular CLI and firebase functions are maintained.
- Augmented with Bootstrap and Font Awesome for Improved UX-UI.

[Bootstrap/ Font Awesome:](#)

For good UI-UX, I have used bootstrap and font awesome,

Bootstrap: Version – 4.0.0 used for style.

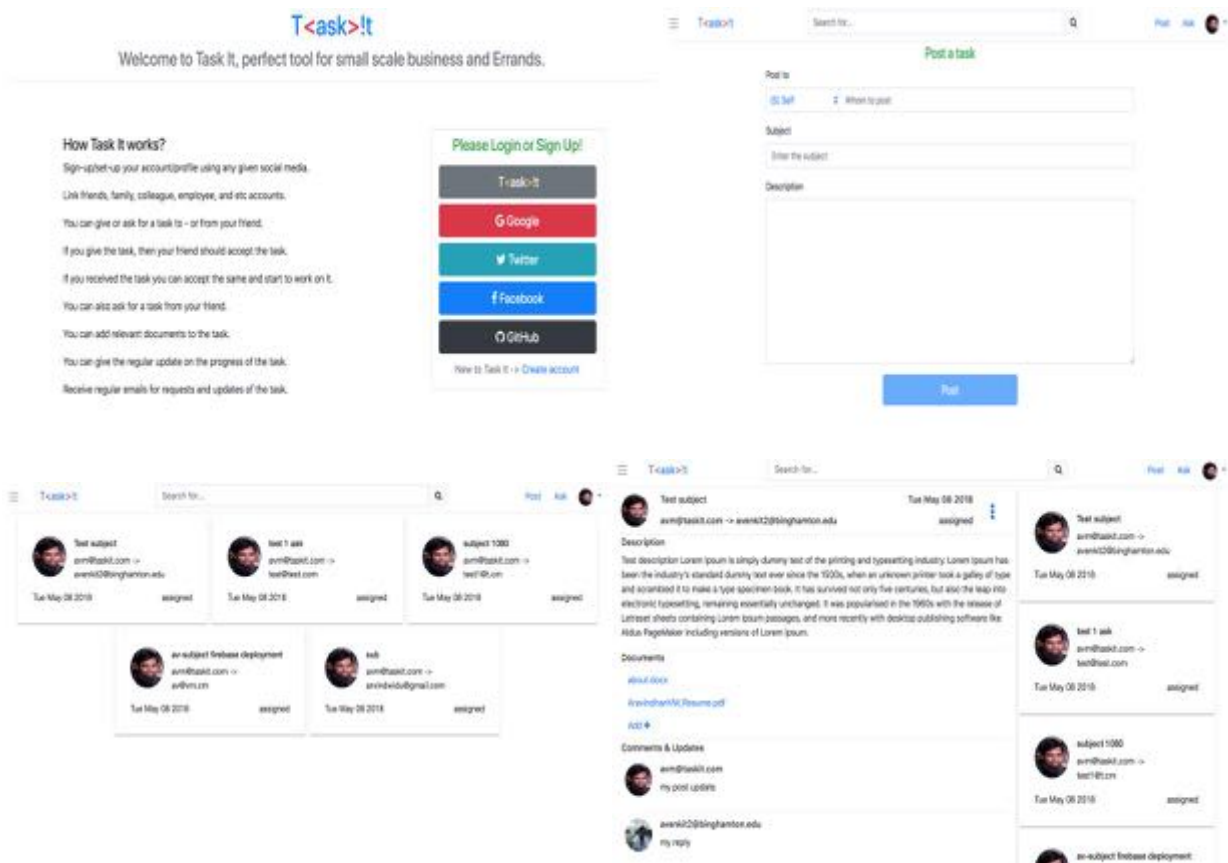
```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI"
crossorigin="anonymous"></script>
```

Font awesome: used for icons, buttons and fonts.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
```

Both Bootstrap and Font Awesome are loaded from CDN (content delivery network) for quick load of the required files.

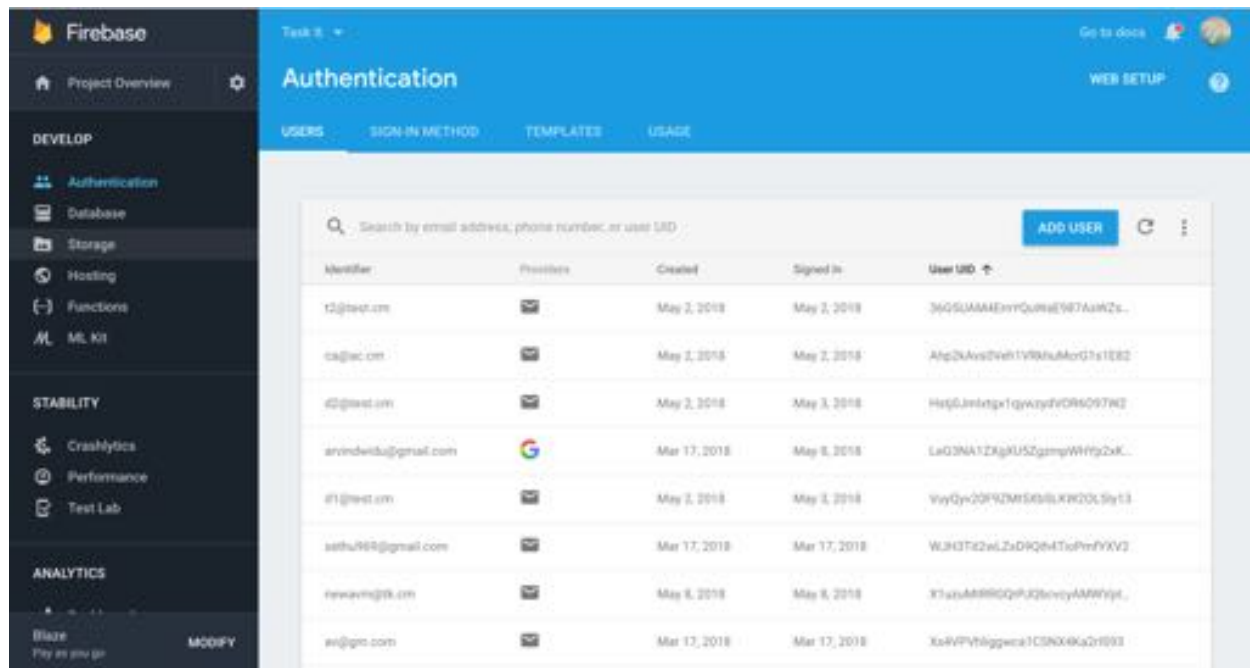
Overall UI look: <https://task-it-ae210.firebaseio.com>



Firebase

[Firebase Authentication:](#)

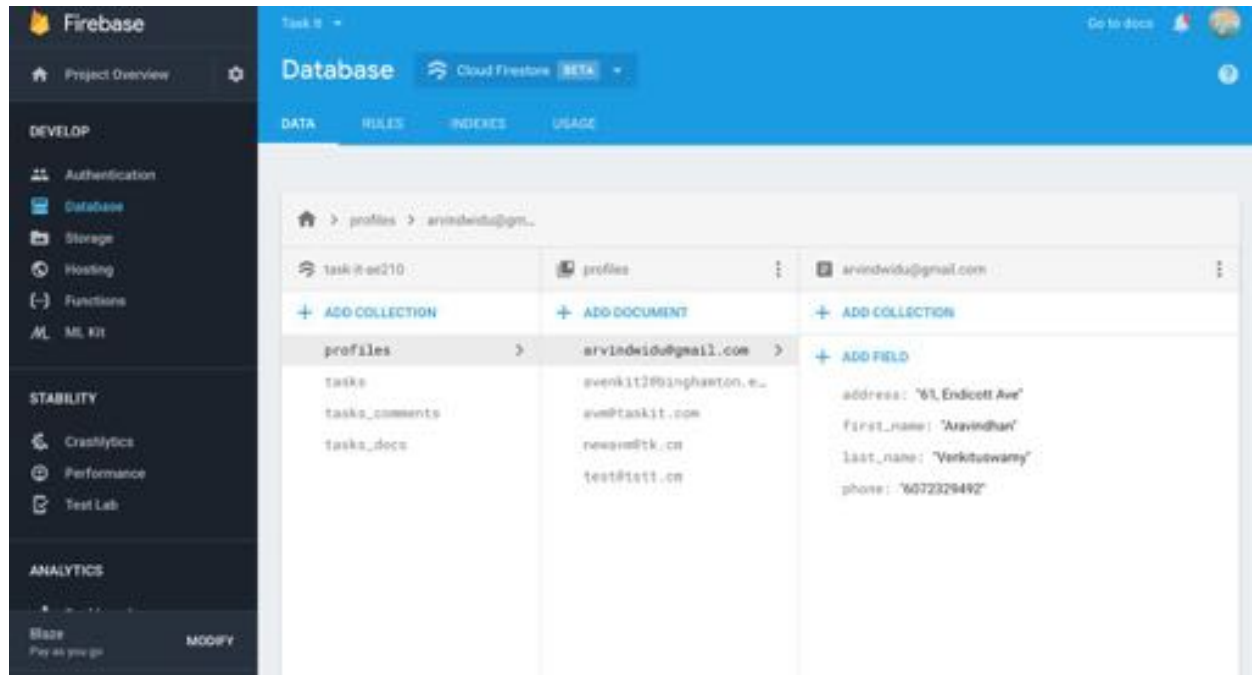
Firebase Auth is a service that can authenticate users using only client-side code. It supports social login providers Facebook, GitHub, Twitter and Google (and Google Play Games). Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.



- Firebase authentication is used here for global authentication, various mode of auth can be enabled.
- I have implemented authentication using email and google account, likewise we can also use Facebook, twitter, GitHub, and etc.
- For email authentication one has to create account using firebase sign-up API, here I have used Angular firestore 2 library to access the same.
- As per the goggle account login is concerned, the google API can take care of utilization of the existing account or you can also create new google account to proceed further.
- Various user and respective mode of authentication example is shown above image, the list of user logged in.

Firestore Cloud Firestore:

Store and sync data between users and devices - at global scale - using a cloud-hosted, NoSQL database. Cloud Firestore gives us live synchronization and offline support along with efficient data queries. Its integration with other Firebase products enables us to build truly server less apps.

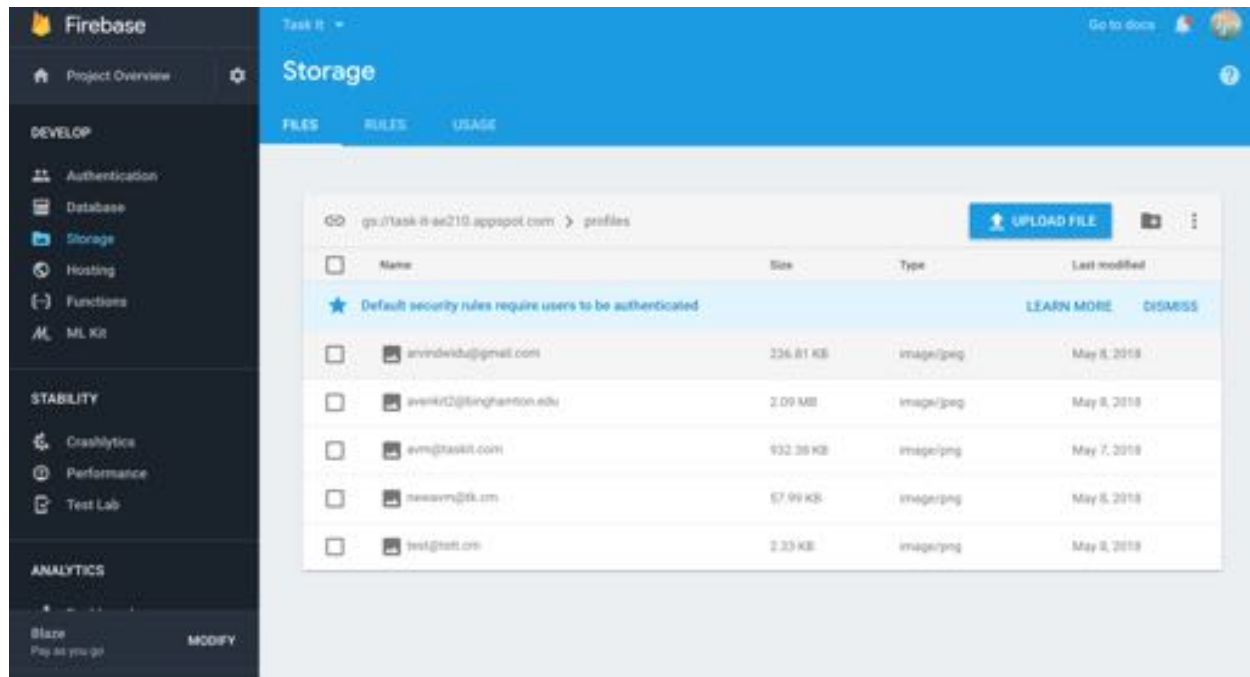


- In this application I have used firebase cloud store, which in turn is based on big data model with folders, collections, documents, and fields.
- Created, Updated, Read, Deleted various collections like profiles, tasks, tasks_docs, tasks_comments.
- Profiles is used to store user profile information such as name, address, phone number and email ID, here respective email ID is used as document key.
- Tasks collection is used to store the various tasks asked and posted for, here time stamp of the time of a tasks document created is used as document ID.
- tasks_docs and Tasks_comments store respective task's relevant document and comment updates, here the respective time stamp ID of the task is used as document ID again to relate each task document and updates.
- The docs and comments are stored as in list format for respective task.

Firebase Cloud Storage:

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. We can use it to store images, audio, video, or other user-generated content.

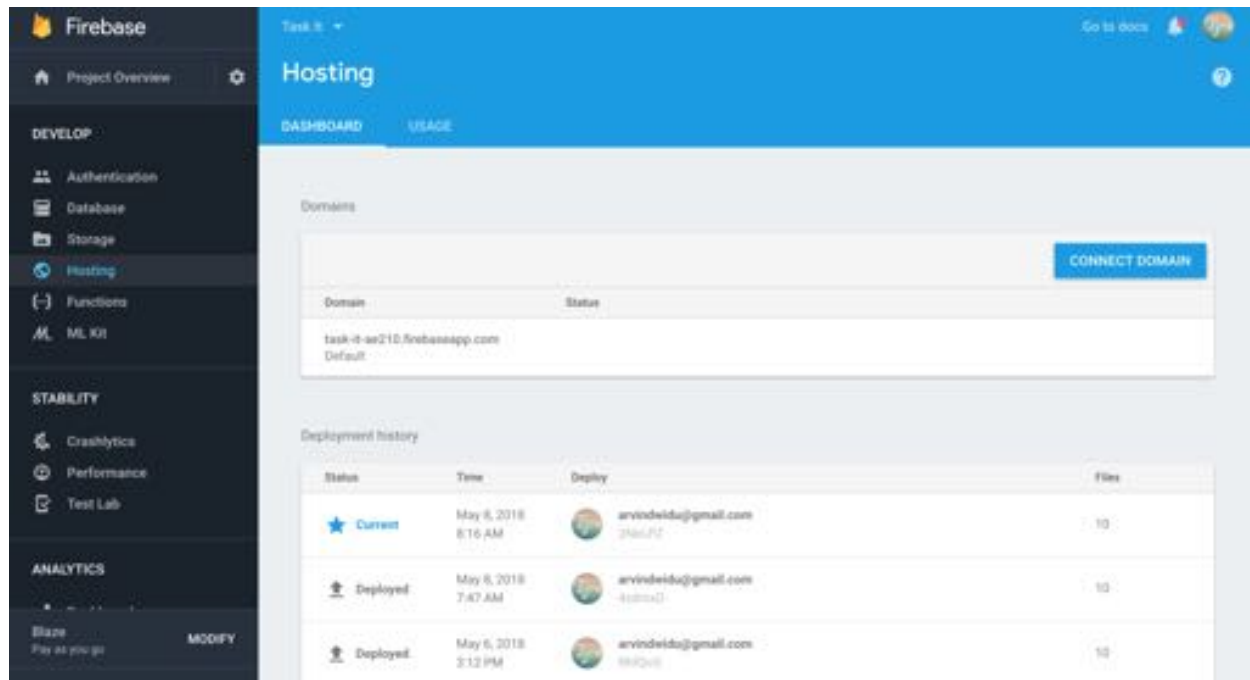
Firebase Storage is backed by Google Cloud Storage.



- Used Angular firestore based storage library to use API calls to communicate with firebase, file upload, retrieve, and delete operations where implemented.
- Utilized separate folder for profiles and tasks files, the profiles folder stores the profile images whereas the tasks folder stores the respective task files with in a folder named after respective time stamp ID.
- Once the respect file or image is upload a promise call back is accessed to retrieve the download URL for the respective file.
- Thus retrieved download URL is stored in firebase cloud store database, here in tasks_docs for task files.
- We can create various storage bucket location to store the file, bucket location depth can also be indefinite.
- Here the bucket ID are created based on respective document ID, for example, profile images are stored on bases of email Id and tasks files are based on time stamp ID.

Firestore Hosting:

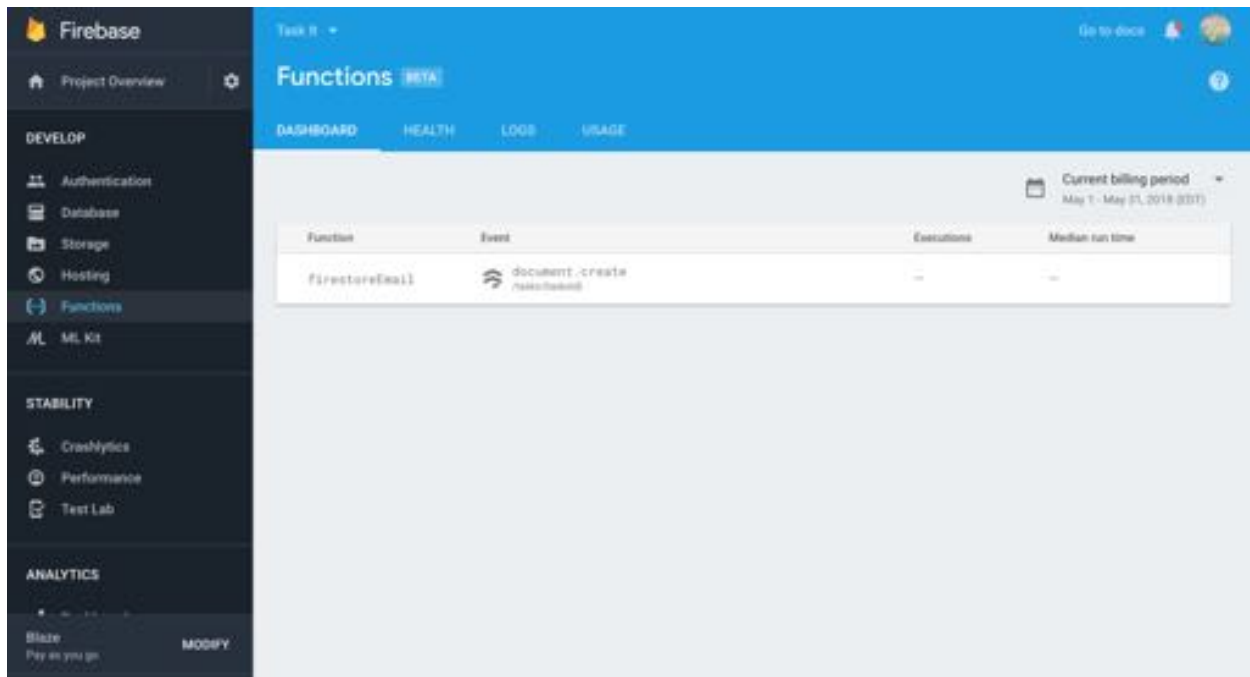
Firestore Hosting is a static and dynamic web hosting service that launched on May 13, 2014. It supports hosting static files such as CSS, HTML, JavaScript and other files, as well as dynamic Node.js support through Cloud Functions. The service delivers files over a content delivery network (CDN) through HTTP Secure (HTTPS) and Secure Sockets Layer encryption (SSL). Firestore partners with Fastly, a CDN, to provide the CDN backing Firestore Hosting.



- The hosting in the firestore is done with the help of the firestore tools, using firestore init command we need to initialize the hosting services.
- Running the firestore init command creates a firestore.json settings file in the root of your project directory.
- When we initialize firestore asks for a directory to use as the public root, here in angular it is dist folder where the build production version of our angular app files will be present.
- So, one has to use ng build –prod in the respective application folder to take build of the application first and then we are ready to deploy the same using firestore deploy.
- From the Hosting panel in the Firestore Console you can see a full history of deployments.
- To rollback to a previous deploy, hover over its entry in the list, click the overflow menu icon, then click "Rollback".

Firestore Cloud Functions:

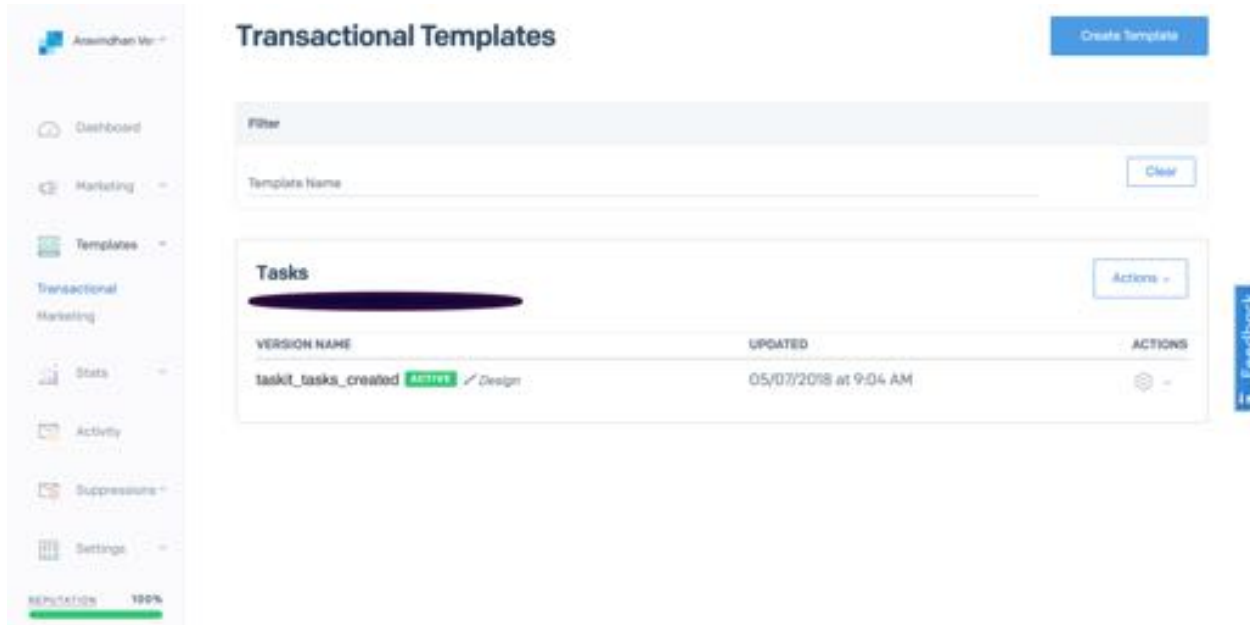
We can extend the app with custom backend code without needing to manage and scale our own servers. Functions can be triggered by events, which are emitted by Firebase products, Google Cloud services, or third parties, using web hooks.



- Again firebase functions can be initialized using firebase tools and firebase init command.
- Run firebase login to log in via the browser and authenticate the firebase tool.
- Go to your Firebase project directory.
- Run firebase init functions. The tool gives you an option to install dependencies with npm. It is safe to decline if you want to manage dependencies in another way.
- The tool gives you two options for language support: JavaScript, Typescript. See Write Functions with Typescript for more information.
- Here I have used send grid API call to utilize the firebase cloud, where an email is sent whenever a task is posted or asked.
- Here firestore trigger function is used, i.e. whenever a task is created a trigger is will activate the onCreate event which will be captured and respective send email function will be executed with relevant data.
- You can view the logs in the logs tab in the firebase tab for trigger execution and the results.

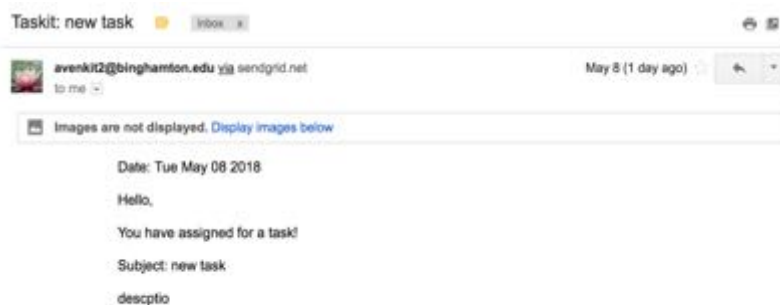
Email service - [Send Grid](#):

Send Grid is a platform for transactional and marketing email for large set of user. It is a 3rd party email service that can be consumed for communication purpose. Here we have used it to communicate the task assignments and updates to respective user.



- The key things we need to consume the send grid email services, we require send grid account, for demonstration purpose I have used send grid trial account where no more than 100 emails can be sent in a day.
- Once account is created, one has to create API key and copy it as it is shown to user only once for security reasons.
- Then one has to create a template for the email here one can use substitution parameters as {{Param}} in the template and configure subject, from, to email address, content respectively.

Example of a received email:



Future Improvements:

- Enable SMS notification for requests and updates of the task, using twilio 3rd party services.
- Form groups, teams and circles among your friends, family and colleagues.
- You can have one to one or groups, teams, and circles chat option using firebase real time database.
- More elaborate status assignments can be accomplished and various end user requirements like emoji, snap images and even memes can be added.
- As of now only front end web application based on Angular is integrated with firebase cloud backend but, in future we can add mobile frontends as well.
- That is to integrate the same firebase backend project instance with various platforms such as iOS and Android.

Thank you