

---

# Integrations with 3rd Party Libraries

Siddharth Ajmera @SiddAjmera

---

# Content:

- Adding MDB
- Adding Angular Material
- Adding Firebase
- Adding a Record to a Firebase Collection
- Reading Records from a Firebase Collection
- Updating a Record on a Firebase Collection
- Deleting a Record from a Firebase Collection

# Adding MDB

- Create an Angular CLI Project with SCSS as a base: `ng new YourProjectName --style=scss`
- `npm i angular-bootstrap-md chart.js@2.5.0 font-awesome hammerjs --save`
- In the `AppModule`:  

```
import { MDBBootstrapModule } from 'angular-bootstrap-md';  
imports: [  
    MDBBootstrapModule.forRoot()  
]
```
- In `.angular-cli.json`:  

```
"styles": [  
    "../node_modules/font-awesome/scss/font-awesome.scss",  
    "../node_modules/angular-bootstrap-md/scss/bootstrap/bootstrap.scss",  
    "../node_modules/angular-bootstrap-md/scss/mdb-free.scss",  
    "../styles.scss"  
],  
"scripts": [  
    "../node_modules/chart.js/dist/Chart.js",  
    "../node_modules/hammerjs/hammer.min.js"  
],
```
- In `tsconfig.json`, "include": `["node_modules/angular-bootstrap-md/**/*.ts", "src/**/*.ts"]`
- `ng serve -o`

# Adding Firebase

In the AppModule

- `import { AngularFireModule } from 'angularfire2';`
- `import { AngularFireDatabaseModule } from 'angularfire2/database';`
- `imports: [  
    AngularFireModule.initializeApp(environment.firebaseConfig),  
    AngularFireDatabaseModule  
]`

In a Service you want to use to interact with Firebase

- `import { AngularFireDatabase, AngularFireList } from 'angularfire2/database';`
- `constructor(private firebase: AngularFireDatabase) {  
    this.list = this.firebase.list('collectionName');  
}`

# Reading Records

- The list method returns an instance of AngularFireList which has a snapshotChanges method on it.
- Calling this method returns an Observable that could be subscribed to.
- The subscription returns a data snapshot which is a list of records in the collection.
- Each item in that list has a payload and a key.
- The key represents a unique Id with which a document in the collection is identified.
- toJSON method can be called on the payload to get the data in the form of JSON.
- `this.firebase.list('collectionName').snapshotChanges().subscribe(dataSnapshot => {  
 this.list = [];  
 dataSnapshot.forEach(snapshot => {  
 const item = snapshot.payload.toJSON();  
 item['$key'] = snapshot.key;  
 this.list.push(item as Item);  
 });  
});`

# Adding a Record

- We can use the push method on an AngularFireList and pass it an object containing the data we want to add to our collection.
- The object that we pass to the push method shouldn't have a key named \$key.
- `this.firebase.list('collectionName').push(newRecord)`


# Updating a Record

- We can use the update method on an AngularFireList to update a record in the collection.
- The update method accepts a \$key which is a unique identifier for that record. It also accepts the an object containing the data we want to update the specified object in our collection with.
- The object that we pass to the push method shouldn't have a key named \$key.
- `this.firebaseio('collectionName').update($key, updatedRecord)`

# Delete a Record

- We can use the remove method on an AngularFireList and pass it \$key which is a unique identifier for the record we want to delete.
- `this.firebase.list('collectionName').remove($key)`



A man in a dark blue pinstripe suit, light blue shirt, and dark tie is leaning forward with his arms resting on a dark wooden table. He is looking directly at the camera with a serious expression. The background is a dimly lit room with a large, ornate chandelier hanging from the ceiling. The text "I'm done here! Any Questions?" is overlaid in the upper right corner.

**I'm done here!  
Any Questions?**