

Angular Part 2: Budget Application

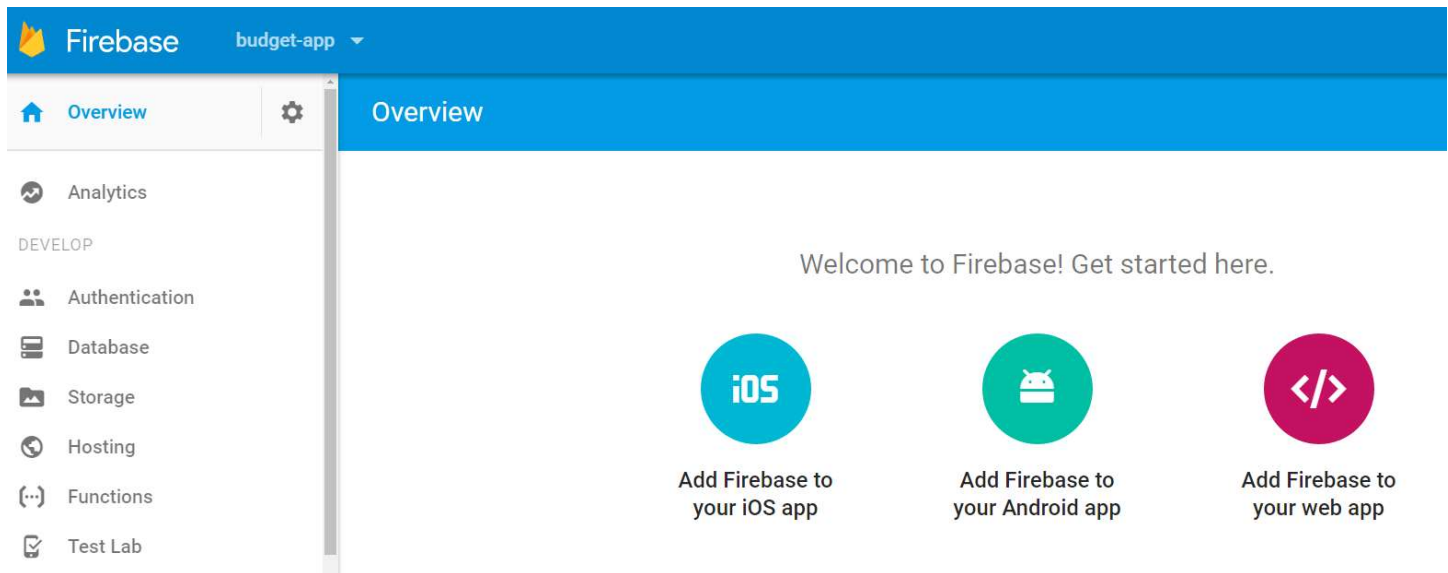
STEP 1: Start Project, add dependencies and config database

create new project	ng new angular-part2
change to project directory	cd angular-part2
install dependencies	npm install --save firebase primeng font-awesome
Add PrimeNG Stylesheets	edit src/ and add the following to the styles section (~line 23) "../node_modules/primeng/resources/primeng.min.css", "../node_modules/primeng/resources/themes/omega/theme.css", "../node_modules/font-awesome/css/font-awesome.min.css"

Add Firebase

Firebase Console Overview Tab click the Add Firebase to your web app.

Click the copy button to copy the code to the clipboard



Import the firebase	Edit src/app/app.component.ts and add the following import (~line 2) import * as firebase from "firebase";
Add database connection Note: Replace the config with your own on the clipboard	Edit src/app/app.component.ts and add the following constructor (~line 12) constructor() { let config = { apiKey: "AIzaSyAr3Bg2tJBrf_c9o6W0EK1B17RiHbulhPw", authDomain: "budget-app-7f40c.firebaseio.com",

Angular Part 2: Budget Application

```
databaseURL: "https://budget-app-7f40c.firebaseio.com",
storageBucket: "budget-app-7f40c.appspot.com",
messagingSenderId: "792611408752"
};

firebase.initializeApp(config);

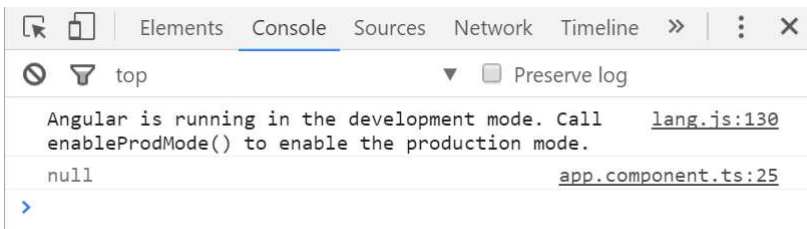
let root = firebase.database().ref();
root.on('value', function (snap){
  console.log(snap.val())
});
}
```

Verify Firebase

ng serve

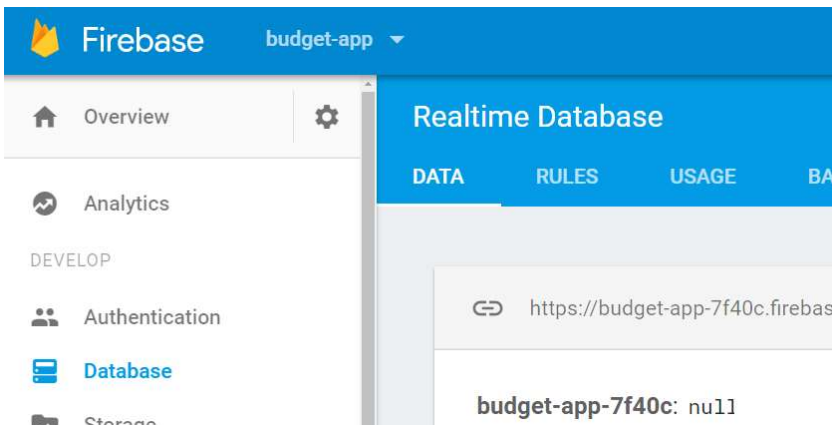
in browser goto localhost:4200 and open the developer tools console

you should see the value null on the console.



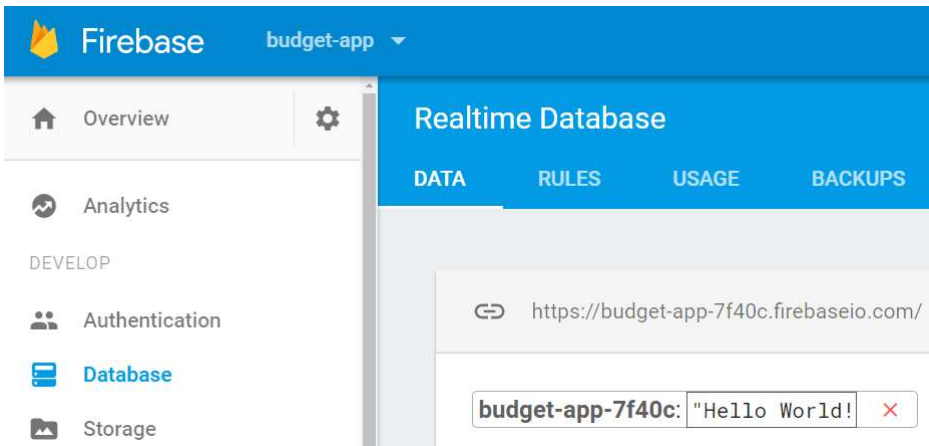
In the firebase console click database on the side menu

You will notice the value of the database in **null**



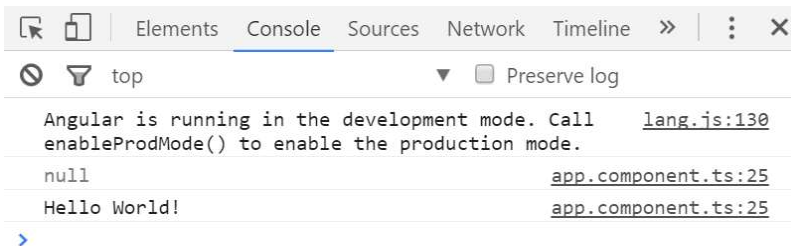
Angular Part 2: Budget Application

Click the null value and change it to **Hello World!**



Switch back to the app in the browser:

The application has been updated with the database change.



STEP -2 Initialize Some Real Data for the Application

The application is a budget application which needs a list of budget categories.

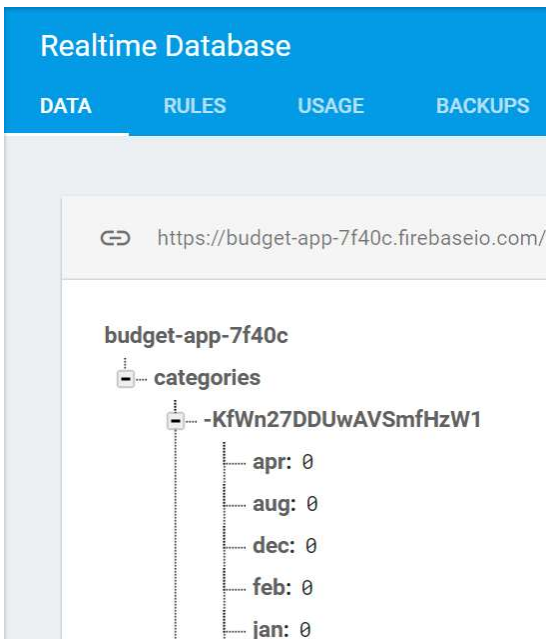
<p>initDB function will initialize database with categories</p>	<p>Edit src/app/app.component.ts and add initDB method (~line 29)</p> <pre>initDB() { let categories = firebase.database().ref('categories'); let monthlyBudget = firebase.database().ref('monthly-budget'); let budgetCategories = ['Mortgage/Rent', 'Electricity', 'Mobile Phone', 'Cable', 'Groceries', 'Entertainment', 'Water/Sewer', 'Auto Loan', 'Dining Out', 'Auto Ins', 'HO Ins', 'Rainy Day Fund', 'Vacation Fund', 'Retirement'] let months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'] budgetCategories.forEach(category => {</pre>
-----------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Angular Part 2: Budget Application

	<pre>const categoryRef = categories.push({name: category}); months.forEach(month => { monthlyBudget.push({month: month, amount: 0, category: categoryRef.key}) }) }) }</pre>
Button to initialize db	Edit src/app/app.component.html add a button (~line 4) <pre><button (click)="initDB()">Initialize DB</button></pre>

Return to the browser and click the InitDB Button

Switch to firebase database view and the data will be listed



Step 3: Create App UI

Import PrimeNG Components to be used

Add PrimeNG Imports	Edit src/app/app.module.ts add imports at (~line 5) <pre>import {InputModule} from 'primeng/primeng'; import {ButtonModule} from 'primeng/primeng'; import {ChartModule} from 'primeng/primeng'; import {DataTableModule, SharedModule} from 'primeng/primeng';</pre>
Import PrimeNG modules	Edit src/app/app.module.ts add imports at (~line 20)

Angular Part 2: Budget Application

	<pre> InputTextModule, ButtonModule, ChartModule, DataTableModule, SharedModule </pre>
--	------------------------------------------------------------------------------------------------------------------------------------

NOTE: stop and start ng-serve as these changes are not automatically reloaded

	Edit src/app/app.component.html change the existing InitDB Button to PrimeNG and add DataTable
Button using PrimeNG Style	<pre> <button pButton type="button" (click)="initDB()" label="Initialize DB"></button> </pre>
PrimeNG dataTable	<pre> <p-dataTable [value]="budgetData" [editable]="true" (onEditComplete)="updateData(\$event) "> </pre>
Columns to be displayed with numbers editable	<pre> <p-column field="name" header="Category" ></p-column> <p-column field="jan" header="January" [editable]="true" ></p- column> <p-column field="feb" header="February" [editable]="true"></p-column> <p-column field="mar" header="March" [editable]="true"></p-column> <p-column field="apr" header="April" [editable]="true"></p-column> <p-column field="may" header="May" [editable]="true"></p-column> <p-column field="jun" header="June" [editable]="true"></p-column> <p-column field="jul" header="July" [editable]="true"></p-column> <p-column field="aug" header="August" [editable]="true"></p-column> <p-column field="sep" header="September" [editable]="true"></p- column> <p-column field="oct" header="October" [editable]="true"></p-column> <p-column field="nov" header="November" [editable]="true"></p-column> <p-column field="dec" header="December" [editable]="true" ></p- column> </p-dataTable> </pre>

Test DataGrid

Change the value of one of the values and check the database.

Change a value in database and check the App DataTable.