

Angular & Real Time Databases

Pierdomenico Reitano - Full Stack Developer @ Lotrèk



Cosa useremo?

Sfrutteremo la Angular per
l'interfaccia della nostra app.

Utilizzeremo Cloud Firestore
come database.



<https://github.com/lotrekagency/Todolistfirebase>

Firebase

Firebase è una piattaforma di sviluppo app e si posiziona sopra alla Google Cloud Platform. E' nato inizialmente come un servizio BaaS, ovvero Backend as a Service, e si è evoluto con il tempo diventando una vera e propria piattaforma.

No utilizzeremo in particolare Cloud Firestore e Hosting



Firebase

Angular

Angular è un framework JavaScript patrocinato da Google con il chiaro scopo di semplificare lo sviluppo client-side, introducendo una sintassi di tipo dichiarativo che si mescola al classico HTML in un meta-linguaggio facile da usare e veloce da apprendere.



Prepariamo l'ambiente

Prepariamo l'ambiente

Firestore-Tools

Tramite la CLI ufficiale possiamo gestire e deployare la nostra app firebase

```
> npm install -g firebase-tools
```

Prepariamo l'ambiente

Angular CLI

Per Gestire il nostro progetto Angular

```
> npm install -g @angular/cli
```


Prepariamo l'ambiente

AngularFire

La libreria ufficiale per Firebase e
Angular

```
> npm install angularfire2 firebase --save
```

Iniziamo!

Dashboard di Firebase

Creiamo un nuovo progetto su
Firebase seguendo la procedura
guidata.

The image shows a composite of screenshots from the Firebase console. On the left is the sidebar with navigation links: Project Overview, Authentication, Database, Storage, Hosting, Functions, STABILITY, ANALYTICS, GROW, and Spark. The main area displays the 'Database' section with a 'Cloud Firestore BETA' toggle. Overlaid on this are two modals. The top modal, 'Aggiungi dati', shows a collection path '/' and a document name 'todos'. The bottom modal, 'Aggiungi Firebase all'applicazione web', provides a code snippet for initializing Firebase in a web application. Below the code, it lists resources for further learning: 'Get Started with Firebase for Web Apps', 'Firebase Web SDK API Reference', and 'Firebase Web Samples'. At the bottom of the bottom modal, there is a table for defining document fields.

Aggiungi dati

1 Aggiungi raccolta 2 Primo documento

Percorso raccolta

/

Nome raccolta

todos

Aggiungi Firebase all'applicazione web

Copia e incolla lo snippet seguente in fondo al codice HTML, prima di altri tag script.

```
<script src="https://www.gstatic.com/firebasejs/4.6.2/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey:
    authDomain:
    databaseURL:
    projectId:
    storageBucket:
    messagingSenderId:
  };
  firebase.initializeApp(config);
</script>
```

Aggiungi dati

1 Aggiungi raccolta

Percorso documento

/todos

ID documento

ID AUTOMATICO

Campo	Tipo	Valore
description	string	devo fare questo
done	boolean	false

ANNULLA SALVA

Comunichiamo Con Firebase

Questi sono i parametri che Firebase
richiede per poter interagire con il
nostro DB

```
export const firebaseConfig = {  
  apiKey: '',  
  authDomain: '',  
  databaseURL: '',  
  projectId: '',  
  storageBucket: '',  
  messagingSenderId: ''  
};
```

Comunichiamo Con Firebase

Inizializziamo la comunicazione con Firebase all'interno del nostro `app.module.ts`, passando la configurazione precedentemente creata.

```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    ...,
    AngularFireModule.initializeApp(firebaseConfig),
    AngularFirestoreModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Comunichiamo Con Firebase

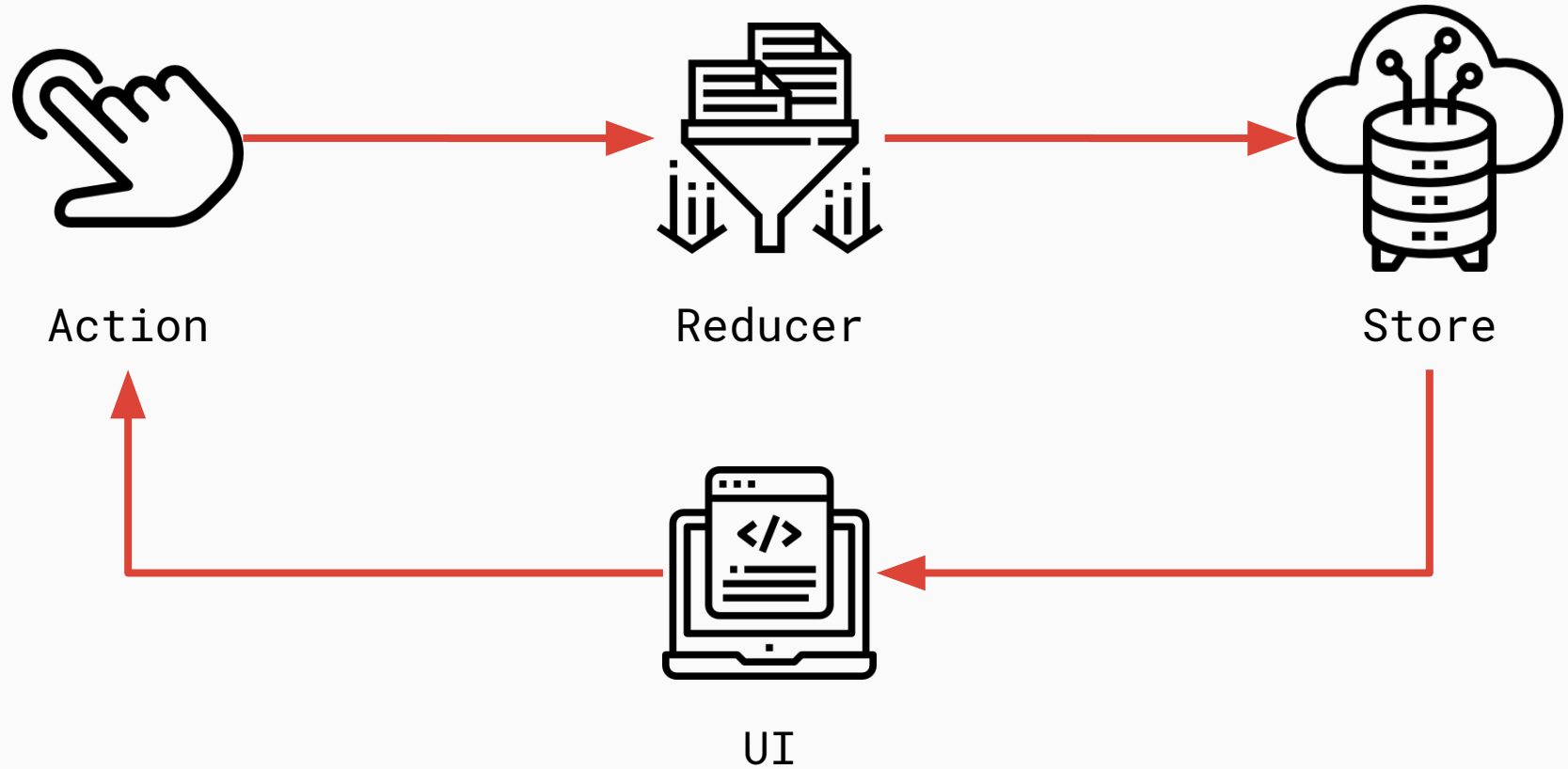
Prendiamo i dati da firebase
utilizzando il metodo collection.

```
//imports and declarations
...
interface Todos {
  description: string;
  done: boolean;
}

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

ngOnInit() {
  this.todosCollection = this.afs.collection('todos');
  this.todos = this.todosCollection.snapshotChanges()
    .map(actions => {
      return actions.map(a => {
        const data = a.payload.doc.data() as Todo;
        const id = a.payload.doc.id;
        return { id, data };
      });
    });
}
```

Come Funziona?



Creazione Elemento

```
addTodo() {  
  this.afs.collection('todos').add({  
    'description': this.description,  
    'done': false  
  });  
}
```

In questo caso utilizziamo il metodo `collection` per aggiungere un nuovo elemento al nostro DB

Update Elemento

In questo caso utilizziamo il metodo `doc`, che permette di recuperare direttamente un elemento, passando alla funzione `id` e valore della checkbox.

```
updateTodo(id, event) {  
  const checkedVal = event.target.checked;  
  this.afs.doc('todos/' + id).update({  
    'done': checkedVal  
  });  
}
```

Rimozione Elemento

Semplicemente passiamo l'id del todo che vogliamo eliminare usando sempre il metodo doc.

```
removeTodo(id){  
  this.afs.doc('todos/' + id).delete();  
}
```

Il nostro template

```
<header>
  <h1>To do list by</h1>
  
</header>
<!--Form aggiunta todo-->
<form>
  <input type="text" [(ngModel)]="description" name="description" placeholder="Cosa devo fare ...">
  <button type="submit" (click)="addTodo()"> Aggiungi Todo </button>
</form>
<!--Lista dei todo-->
<main>
  <ul>
    <li *ngFor="let todo of todos | async">
      <input [checked]="todo.data.done" type="checkbox" (change)="updateTodo(todo.id, $event)" />
      <b [class.done]="todo.data.done" >{{todo.data.description}}</b>
      <button *ngIf="todo.data.done" (click)="removeTodo(todo.id)">&times;</button>
    </li>
  </ul>
</main>
```

Tempo di Deploy!

Deploy

Firebase mette a disposizione un hosting, per utilizzarlo basta attivarlo direttamente dalla dashboard.

Todolist ▾

Hosting

DASHBOARD UTILIZZO

Configura hosting

1 Installa — 2 Implementa

Apri una finestra del terminale e accedi a o crea una directory per il tuo sito

Accedi a Google: `$ firebase login`

Avvia il progetto: `$ firebase init`

Aggiungi i file statici alla directory di implementazione (quella predefinita è public)

Implementa il sito web: `$ firebase deploy`

FINE

Quali Sono i Vantaggi?

Cloud Firestore è sicuramente un prodotto interessante, permette di togliersi la preoccupazione di impostare, mantenere e soprattutto pagare un server per il Back End della propria App (ovviamente se usiamo il piano free)

Ottimo per sviluppare le proprie idee se non si hanno grosse competenze sistemiche.



Però?

È ancora in beta.

**Non ci sono grossi problemi,
molti sviluppatori stanno
iniziando ad utilizzarlo non solo
per sperimentare.**



Comparativa Cloud Firestore vs. Real Time Databases

<https://firebase.google.com/docs/firestore/rtdb-vs-firestore>

Questions?



Grazie!

Contatti:

pierdomenico@lotrek.it

 @89mino

 <https://github.com/mino89>

<https://github.com/lotrekagency/Todolistfirebase>

