

## Unknown Title

---



To retrieve data from Firestore in Angular, you need to use the Firestore SDK, which is provided by Firebase. First, you need to install Firebase SDK by running the following command in your terminal:

sql

```
npm install firebase @angular/fire --save
```

Then you need to import the necessary modules and initialize the Firebase app in your Angular project. You can add the following code to your `app.module.ts` file:

typescript

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';

// Import the Firebase and AngularFire modules
import { AngularFireModule } from '@angular/fire';
import { environment } from '../environments/environment';
import { AngularFireFirestoreModule } from '@angular/fire/firestore';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,

    // Initialize Firebase app with your project credentials
    AngularFireModule.initializeApp(environment.firebase),

    // Import the AngularFireFirestore module
    AngularFireFirestoreModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Next, you can create a service to fetch data from Firestore. Here is an example service that retrieves a collection of products from Firestore:

typescript

```
import { Injectable } from '@angular/core';
import { AngularFireStore } from '@angular/fire/firestore';
import { Observable } from 'rxjs';
import { Product } from '../product';

@Injectable({
  providedIn: 'root'
})
export class ProductService {

  constructor(private firestore: AngularFireStore) { }

  // Fetch all products from Firestore
  getProducts(): Observable<Product[]> {
    return this.firestore.collection<Product>('products').valueChanges();
  }
}
```

In this service, we inject the `AngularFirestore` service and define a `getProducts` method that returns an observable of an array of `Product` objects. We use the `valueChanges` method of the collection object to get a stream of data from the Firestore collection.

Finally, you can use this service in your components to fetch data from Firestore. Here is an example component that uses the `ProductService` to fetch data and display it in a template:

typescript

```
import { Component } from '@angular/core';
import { Observable } from 'rxjs';
import { Product } from '../product';
import { ProductService } from '../product.service';

@Component({
  selector: 'app-product-list',
  template: `
    <h2>Product List</h2>
    <ul>
      <li *ngFor="let product of products$ | async">{{ product.name }}</li>
    </ul>
  `
})
export class ProductListComponent {
  constructor(private productService: ProductService) {}
}
```

```
  })  
  export class ProductListComponent {  
    products$: Observable<Product[]>;  
  
    constructor(private productService: ProductService) { }  
  
    ngOnInit() {  
      // Fetch all products from Firestore using ProductService  
      this.products$ = this.productService.getProducts();  
    }  
  }  
}
```

In this component, we inject the `ProductService` and use it to fetch data from Firestore. We use the `async` pipe to subscribe to the `products$` observable and display the product names in a list.