Forms &
Services in Angular

Forms in Angular

- Angular provides two different approaches to handling user input through forms:
  - **Template-Driven Forms**
  - **Reactive Forms**
- These two belong to the @angular/forms library and share a series of form control classes.
- They deviate in terms of philosophy and programming technique.

Template-Driven Forms

- The most straightforward approach to building forms in Angular is to take advantage of the directives provided for you.
- Directives such as NgForm, NgModel, etc...
- Places most of the form handling logic within that form's template.
- Need to import the **FormsModule** from '@angular/forms' to use template-driven forms.

Reactive Forms

- Also known as model-driven forms.
- places form handling logic within a component's class properties and provides interaction through observables.
- No need to use directives such as NgForm, NgModel, etc...
- Need to import the **ReactiveFormsModule** from '@angular/forms' to use reactive forms.

Which one is the best? Reactive or template-driven?

- Neither one.
- They are two different architectural paradigms with their own advantages and disadvantages.
- You can choose the approach that suits you the best.
- You are free to even decide using both in the same application.

Custom Services in Angular

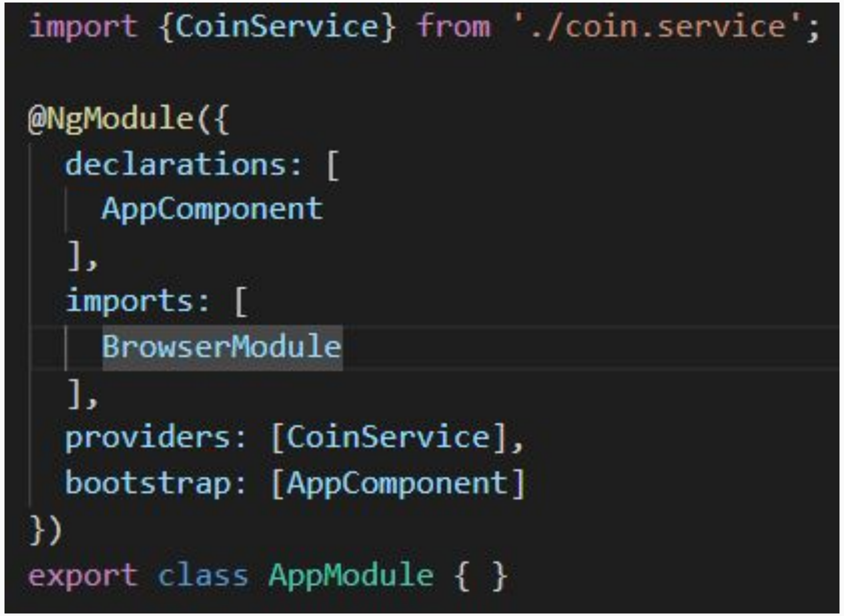
Custom Services

- Angular services are singleton objects which get instantiated only once during the lifetime of an application.

- The main objective of a service is to organize and share business logic or datawith different components of an Angular application.

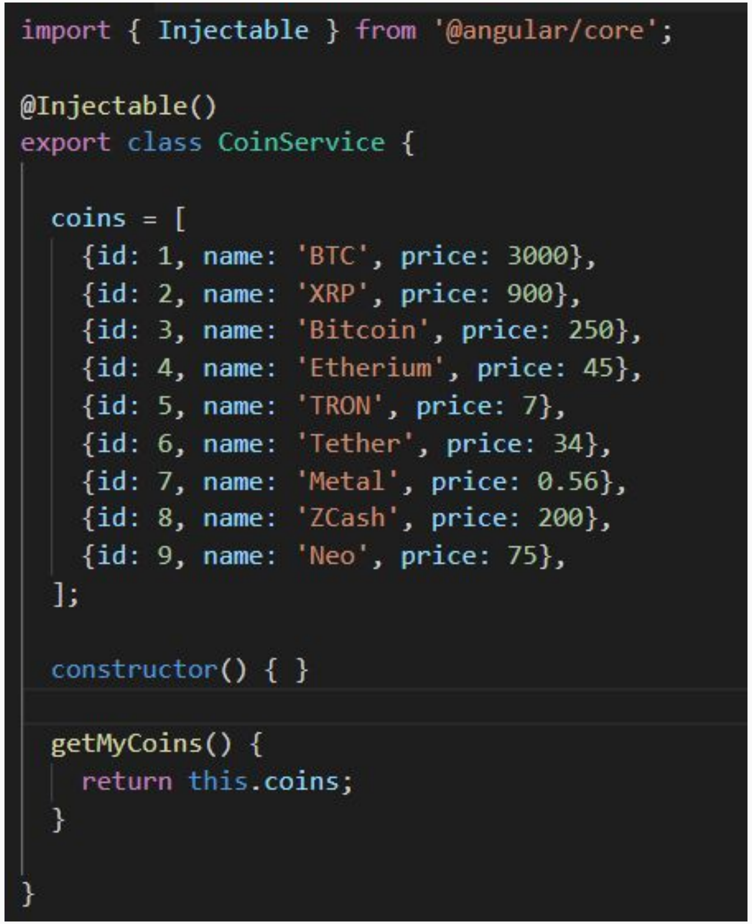
- @Injectable

Defining a service



```
import {CoinService} from './coin.service';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [CoinService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```



```
import { Injectable } from '@angular/core';

@Injectable()
export class CoinService {

  coins = [
    {id: 1, name: 'BTC', price: 3000},
    {id: 2, name: 'XRP', price: 900},
    {id: 3, name: 'Bitcoin', price: 250},
    {id: 4, name: 'Etherium', price: 45},
    {id: 5, name: 'TRON', price: 7},
    {id: 6, name: 'Tether', price: 34},
    {id: 7, name: 'Metal', price: 0.56},
    {id: 8, name: 'ZCash', price: 200},
    {id: 9, name: 'Neo', price: 75},
  ];

  constructor() { }

  getMyCoins() {
    return this.coins;
  }

}
```

Thank You!