Monday :HttpClient

HttpClient

HttpClient is a mechanism used by Angular to enable communication to remote servers and backend services using the HTTP protocol. We will use a random quote API which is a backend service in our app to display quotes from the API on our app. This will give us an understanding of how to use HttpClient in Angular to access backend services and remote servers. To make HttpClient available everywhere in the app, we import it in our root modules as follows.

src/app/app.module.ts

```
...
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
...
@NgModule({
...
  imports: [
...
    FormsModule,
    HttpClientModule
  ],
...
})
```

At the top, we have imported the **HttpClientModule** and added it to the imports array. We are going to use this module to access a random quote API and display the quotes on our app. If you paste this **http://quotes.stormconsultancy.co.uk/random.json (http://quotes.stormconsultancy.co.uk/random.json)** link in your browser's address bar, you'll see a random quote that changes every time you refresh the webpage. You can install **JSONView (https://chrome.google.com/webstore/detail/jsonview/chklaanhfefbnpoihckbnefhakgolnmc)** , or **JSONFormatter (https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa)** from chrome web store to make the response more readable. The response looks something like this but the quote and author may vary:

{

- **author**: "James O. Coplien",
- **id**: 23,
- **quote**: "You should name a variable using the same care with which you name a first-born child.",
- **permalink**: "**http://quotes.stormconsultancy.co.uk/quotes/23 (http://quotes.stormconsultancy.co.uk/quotes/23)** "

}

The request returns a response that has four properties, author, id, quote, and permalink. We will make a request to the API and display the quotes in our app. We will display the author and quote only in our app. We, therefore, need to make Angular sieve out what we need by creating a quote class that will help us create quote instances. To create a class, let's execute this command in our terminal:

```
$ ng generate class quote-class/quote
```

This command creates a folder quote-class, a class file, quote.ts, and a test file quote.spec.ts. Inside the class file, let's define how we want our quote instances created.

src/app/quote-class/quote.ts

```
export class Quote {

  constructor(public author:string, public quote:string ){}
}
```

We have created a constructor inside the Quote class and defined two public properties, author and quote, both of the string type.

We can now make a HTTP request to the API in our goal component

src/app/goal/goal.component.ts

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
...
import { Quote } from '../quote-class/quote';
....
export class GoalComponent implements OnInit {

  goals:Goal[];
  alertService:AlertService;
  quote:Quote;
....
  constructor(goalService:GoalService, alertService:AlertService, private http:HttpClient) {
    this.goals = goalService.getGoals()
    this.alertService = alertService;
  }

  ngOnInit() {

    interface ApiResponse{
      author:string;
      quote:string;
    }

    this.http.get<ApiResponse>("http://quotes.stormconsultancy.co.uk/random.json").subscribe(data=>
{
      // Succesful API request
      this.quote = new Quote(data.author, data.quote)
    })
  }

}
```

At the top, we have imported HttpClient and the Quote class. Inside the component class, we have created a property quote an assigned it the type Quote. In the constructor function, we have created a property http which is of the type HttpClient. Before we make the request, we need to inform Angular the kind of response we'll receive from the API by defining an interface which we have named ApiResponse. Inside the interface, we have specified that we'll be expecting a property author and quote which are both of the type string. We have then made a request to the API with the get function passing in the API URL accompanied by the interface for the data we expect to receive. We have then called the subscribe function which has a data function that is executed when the request is successful. We then create a new quote instance with the properties we get from the response.

Let's display this quote on our app.

src/app/goal/goal.component.html

```
....
<p *ngIf='goals.length > 5'>You have too many goals</p>
  <div class="row">
    <blockquote class="blockquote text-center">
      <p class="mb-0">{{quote.quote}}</p>
      <footer class="blockquote-footer"><cite>{{quote.author}}</cite></footer>
    </blockquote>
  </div>
</div>
```

We have displayed our quote in a bootstrap blockquote, and if our server is still running, we can see a random quote displayed on our app below our goals.