Curso de Angular 6

Unidad Didáctica 04: Servicios

Ayuntamiento de Vitoria-Gasteiz
Vitoria-Gasteizko Udala

http://cursosdedesarrollo.com/

Índice de contenidos

- Introducción
- Creando un Servicio
- Asociando el Servicio
- Cliente HTTP
- Presentación de Datos
- Referencias
- Conclusiones

http://cursosdedesarrollo.com/

Introducción

Dentro de Angular normalmente es necesario crear servicios que nos permitan el acceso a los datos

http://cursosdedesarrollo.com/

Creando un Servicio

Para crear un servicio será necesario ejecutar otra vez la creación de un componente con el comando:

ng generate service nombre_servicio

por ejemplo

ng g service todo

http://cursosdedesarrollo.com/

Creando un Servicio

Nos ofrecerá una salida similar a la siguiente:

```
$ ng g service todo
  installing service

    create src/app/todo.service.spec.ts

    create src/app/todo.service.ts

    WARNING Service is generated but not provided, it
  must be provided to be used
```


Creando un Servicio

Luego podemos volver a ejecutar el

ng serve

y volver a entrar a http://localhost:4200/

Creando un Servicio

Dentro del directorio app nos encontraremos los dos ficheros generados, el principal es el todo.service.ts

http://cursosdedesarrollo.com/

Creando un Servicio

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

@Injectable({

  providedIn: 'root'

})

export class TodoService {

  constructor() {

    console.log('TodoService inicializado...');

  }

}
```

http://cursosdedesarrollo.com/

Asociando el Servicio

De cara a que se pueda cargar el servicio desde la aplicación será necesario modificar el app.module.ts

http://cursosdedesarrollo.com/

Asociando el Servicio

app.module.ts

```
import {TodoService} from “./todo.service";

@NgModule({

  ...,

  providers: [TodoService]

})
```

http://cursosdedesarrollo.com/

Asociando el Servicio

Debido a que queremos incluir el servicio dentro de un componente, también deberemos asociarlo allí, en este caso desde el TodosComponent

Asociando el Servicio

todos.component.ts

```
import {TodoService} from "../todo.service"

    export class TodosComponent implements OnInit {

        ...

        constructor(private _todoService:TodoService) {}
```


Asociando el Servicio

Entonces al arrancar la aplicación debería poder decir lo siguiente en el log:

Servicio Todo inicializado...

Cliente HTTP

Uno de los servicios más utilizados es el cliente de HTTP que permitirá realizar consultas HTTP desde los componentes o servicios Angular

http://cursosdedesarrollo.com/

Cliente HTTP

Pongamos el ejemplo de utilización de una petición get a la URL:

http://cursosdedesarrollo.com/pactometro/resultados.json

http://cursosdedesarrollo.com/

Cliente HTTP

Esta dirección nos dará los resultados de las elecciones generales y tiene una estructura comparable a un array de objetos

http://cursosdedesarrollo.com/

Cliente HTTP

Será conveniente que creemos una clase para manejar los datos de cada partido:

ng g class Partido

```
export class Partido {
  nombre: string;
  dipu: number;
  imagen:string;
}
```

http://cursosdedesarrollo.com/

Cliente HTTP

Dentro del app.module.ts normalmente suele estar incluida la inclusión e importación del módulo de peticiones HTTP :

```
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  imports: [
    ...,
    HttpClientModule,
  ],
  ...
})
```

http://cursosdedesarrollo.com/

Cliente HTTP

Luego deberemos importarlo para usarlo desde algún servicio de la app

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

import {HttpClient} from '@angular/common/http';

import {catchError} from 'rxjs/operators';

import {Partido} from './partido';

import {Observable} from 'rxjs';

…

export class TodoService {

  url = 'http://cursosdedesarrollo.com/pactometro/resultados.json';

  constructor(private _httpClient: HttpClient) {console.log('TodoService inicializado...');}

  getData() {

    return this._httpClient.get<Observable<Partido[]>>(this.url)

      .pipe(catchError(this.handleError('get', [])));

  }
```

http://cursosdedesarrollo.com/

Cliente HTTP

```
private handleError (operation = 'operation', result?) {
  return (error: any): any[] => {

    // TODO: send the error to remote logging infrastructure
    console.error(error); // log to console instead

    // TODO: better job of transforming error for user consumption
    console.log(`${operation} failed: ${error.message}`);

    // Let the app keep running by returning an empty result.
    return [];
  };
 }
}
```

http://cursosdedesarrollo.com/

Cliente HTTP

Como puede verse es necesario gestionar la url de consulta y una llamada a una función que permita la consulta de los datos, en este caso getData()

http://cursosdedesarrollo.com/

Cliente HTTP

Después dentro del componente todos.component.ts deberemos indicar que queremos llamar al servicio, definiendo una propiedad en la clase que llame al método que devuelve la promesa, sólo podrá hacerse si se ha inyectado previamente

```
// Nueva propiedad para almacenar los datos de la petición

partidos = this._todoService.getData().toPromise();

constructor(protected _todoService: TodoService) { }
```

http://cursosdedesarrollo.com/

Cliente HTTP

Se almacenarán los resultados en la propiedad partidos que luego utilizaremos desde la vista

http://cursosdedesarrollo.com/

Cliente HTTP

Una vez manejados los datos ya los podemos utilizar desde la vista del controlador todos.component.html

```
<h3>Listado de partidos</h3>

<ul>
  <li *ngFor="let resultado of partidos | async">

    {{resultado.nombre}}: {{resultado.dipu}} <img src="http://
cursosdedesarrollo.com/pactometro/img/{{resultado.imagen}}" width="90px"
height="40px"/>

  </li>

</ul>
```

http://cursosdedesarrollo.com/

Referencias

Simple Angular 2 App with Angular CLI: https://www.youtube.com/watch?v=QMQbAoTLJX8

Tutorial HTTP client:

https://angular.io/docs/ts/latest/guide/server-communication.html

http://cursosdedesarrollo.com/

Conclusiones

Hemos visto cómo manejar servicios y unas buenas prácticas de creación

http://cursosdedesarrollo.com/

Datos de Contacto

http://www.cursosdedesarrollo.com
david@cursosdedesarrollo.com

http://cursosdedesarrollo.com/

Licencia

David Vaquero Santiago

http://cursosdedesarrollo.com/