

7.- HttpClient

7.1.- Solicitud de respuesta

Se puede declarar el tipo de objeto para la respuesta del servidor, para que el consumo de la salida sea más fácil y más obvio. Para especificar el tipo de objeto de respuesta, primero se debe de definir una interfaz con las propiedades requeridas.

```
export interface Config {  
  heroesUrl: string;  
  textfile: string;  
}
```

Es importante recordar que debe ser una *interface* y no una clase.

En la petición http:

```
getConfig() {  
  // now returns an Observable of Config  
  return this.http.get<Config>(this.configUrl);  
}
```

Y en la devolución de la llamada en el componente:

```
config: Config;
```

```
showConfig() {  
  this.configService.getConfig()  
    // clone the data object, using its known Config shape  
    .subscribe((data: Config) => this.config = { ...data });  
}
```

7.2.- Leer la respuesta completa

A veces necesitamos leer la respuesta completa del servidor, para ello primero en la petición http:

```
getConfigResponse(): Observable<HttpResponse<Config>> {  
  return this.http.get<Config>(  
    this.configUrl, { observe: 'response' });  
}
```

En la devolución de la llamada:

```
showConfigResponse() {  
  this.configService.getConfigResponse()  
    // resp is of type `HttpResponse<Config>`  
    .subscribe(resp => {  
      // display its headers  
      const keys = resp.headers.keys();  
      this.headers = keys.map(key =>  
        `${key}: ${resp.headers.get(key)}`);  
  
      // access the body directly, which is typed as `Config`.  
      this.config = { ... resp.body };  
    });  
}
```

7.3.- Peticiones de datos no JSON

No todas las peticiones a las APIs son para obtener datos JSON. Ejemplo de una petición para descargar un archivo de texto:

```
getTextFile(filename: string) {  
  // The Observable returned by get() is of type Observable<string>  
  // because a text response was specified.  
  // There's no need to pass a <string> type parameter to get().  
  return this.http.get(filename, { responseType: 'text' })  
    .pipe(  
      tap( // Log the result or error  
        data => this.log(filename, data),  
        error => this.logError(filename, error)  
      )  
    );  
}
```