



red.es

PROFESIONALES
DIGITALES

Formación
Continua

El FSE invierte en tu futuro



UNIÓN EUROPEA

HttpClient



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Solicitud de respuesta

config.ts

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

config.service.ts

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

config.component.ts

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



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

El FSE invierte en tu futuro

Respuesta completa

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

```
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 };  
    });  
}
```



red.es

PROFESIONALES
DIGITALES

Formación
Continua



UNIÓN EUROPEA

***El FSE** invierte en tu futuro*

Peticiones no JSON

```
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)  
      )  
    );  
}
```