# HttpModule – Angular

## Pasos para utilizarlo

1. Crear un servicio mediante Angular-CLI



1. Instalar el módulo HttpModule mediante npm



1. Dentro del servicio importar las clases *Http* y *Response* de *@angular/http*



**Nota:**

* Http es un objeto que hace de cliente HTTP
* Response se utiliza como tipo a las respuestas de las peticiones HTTP

1. Creamos una instancia de *Http* mediante inyección de dependencias en el constructor del servicio



1. El objeto *Http* posee métodos para realizar peticiones (*request*) para los distintos tipos de métodos de petición HTTP:
   1. get
   2. post
   3. put
   4. delete
   5. patch
   6. options

Todos estos métodos devuelven un objeto del tipo *observable* <*response>* (que es un genérico de TypeScript)

Para escuchar los valores del *observable* necesitamos suscribirnos

**IMPORTANTE:**

**Dentro de app.module.ts debemos importar el modulo HttpModule**

****

### Convertir el *observable a promesa*

Los objetos *observable* (RxJS) poseen un método que convierten el observable en un objeto *promesa* por medio de *toPromise()*

**Nota:** las promesas se utilizan para acceder a recursos que pueden demorar en obtenerse. Representan un valor que pueden estar disponible ahora, en el futuro o nunca.

El objeto *promise* posee dos métodos:

* then(función\_que\_se\_ejecuta\_cuando\_se\_obtiene\_la\_promesa)
* catch(función\_que\_se\_ejecuta\_cuando\_ocurre\_un\_error)

#### Ejemplo de una petición get utilizando promesas



##### Explicación:

* Llamamos al método *get ()* del objeto *Http* y pasamos como parámetro la URL del recurso que queremos consumir. Este método nos devuelve un objeto *Observable*.
* Los objetos *observables* poseen un método para convertirlos en objetos *Promise*.
* A su vez los objetos *Promise* poseen un método *then* que nos permite especificar una función o método que se va a ejecutar cuando la petición se realizó correctamente.
* También los objetos *Promise* posee método *catch* que nos permiten especificar una función que se va a ejecutar en caso de que haya ocurrido algún error.
* Tanto *then* como *catch* le van a pasar a las funciones que llaman un objeto *response*

Estos serían los métodos que llamamos desde la promesa



En caso de que la respuesta sea Ok convertimos el response en un JSON

