Curso de Consumo de APIs REST con Angular

Nicolas Molina Monroy

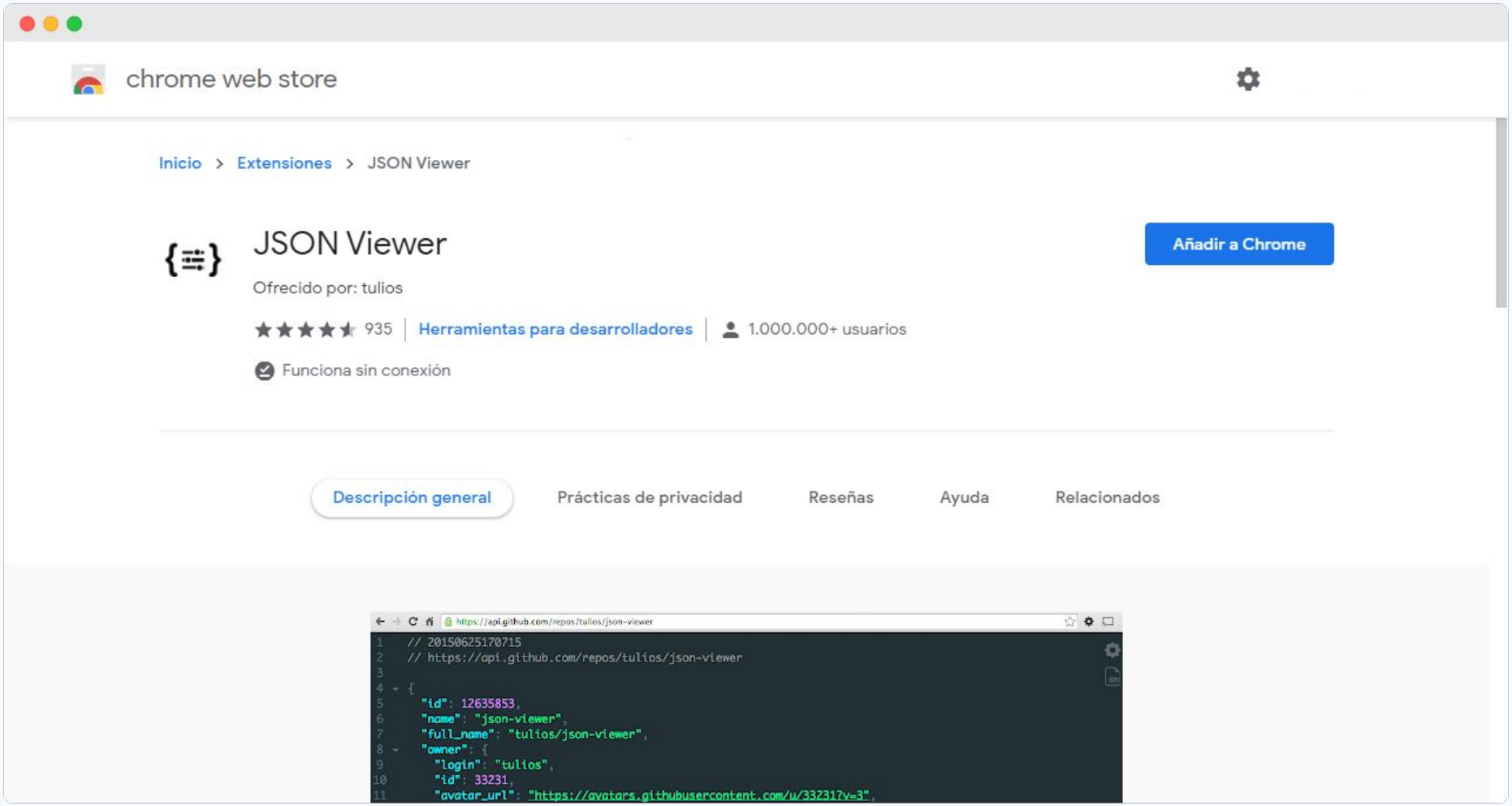
@nicobytes | Google Dev Expert

Angular Platform

Angular Http

@angular/common/http

JSON Viewer


chrome web store
Inicio > Extensiones > JSON Viewer
JSON Viewer
Ofrecido por: tulios
935
Herramientas para desarrolladores
1.000.000+ usuarios
Funciona sin conexión
Añadir a Chrome
Descripción general
Prácticas de privacidad
Reseñas
Ayuda
Relacionados
https://api.github.com/repos/tulios/json-viewer
// 20150625170715
// https://api.github.com/repos/tulios/json-viewer
{
"id": 12635853,
"name": "json-viewer",
"full_name": "tulios/json-viewer",
"owner": {
"login": "tulios",
"id": 33231,
"avatar_url": "https://avatars.githubusercontent.com/u/33231?v=3",

Solicitudes GET

Conectando a un API

Detalle producto

Implementando Slides

Solicitudes POST

Data Transfer Object

```
{
    "title": "Nuevo producto",
    "price": 121,
    "categoryId": 1,
    "description": "bla bla bla",
    "image": "https://placeimg.com/..."
}
```

Response


```
{
  "title": "asa",
  "price": 121,
  "description": "asas",
  "image": "https://placeimg.com/...",
  "category": {
    "id": 1,
    "name": "Clothes"
  },
  "id": 51
}
```

Solicitudes
PUT & PATCH

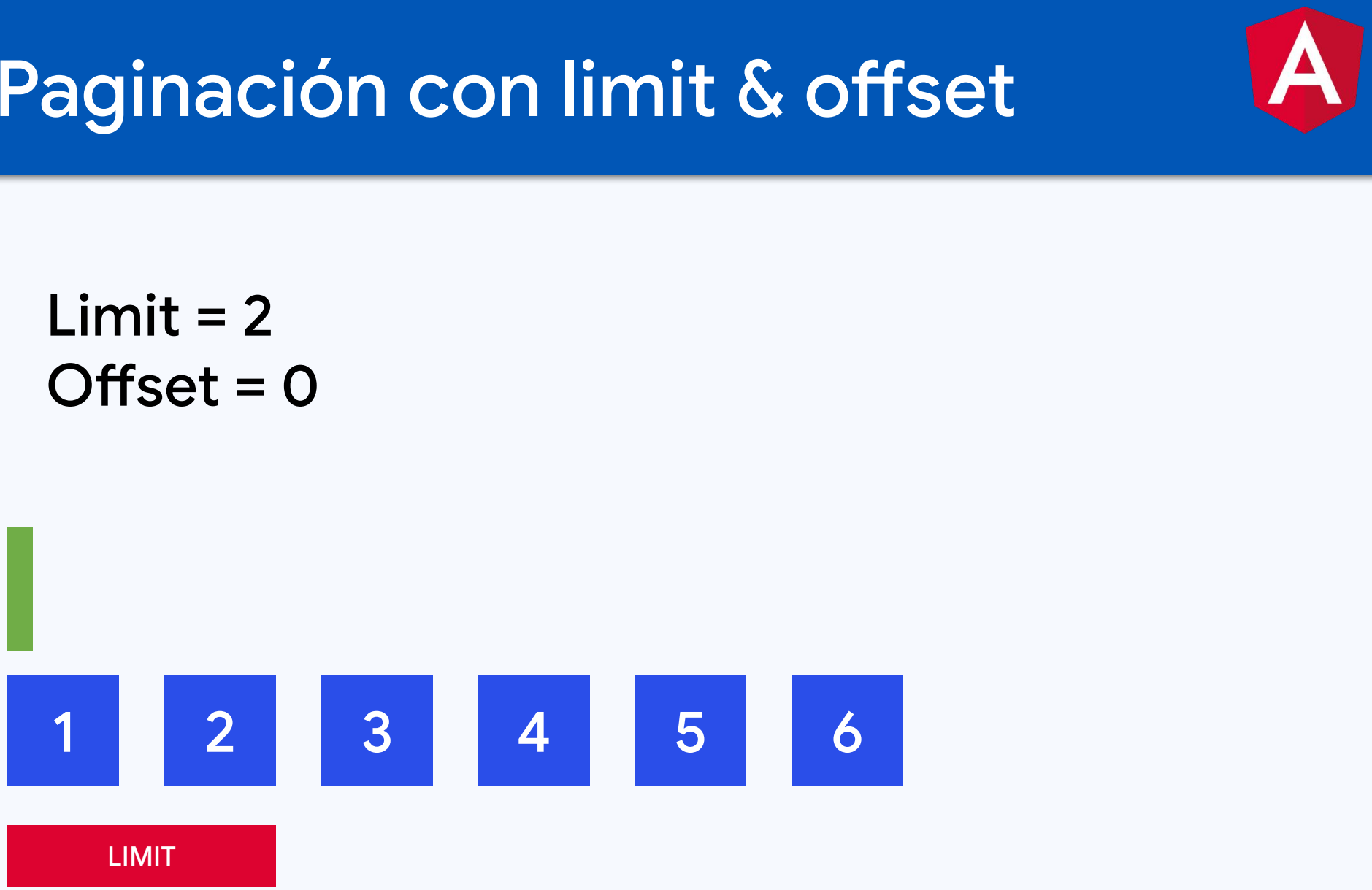
Solicitudes DELETE

Parámetros URLS

Parámetros URLS

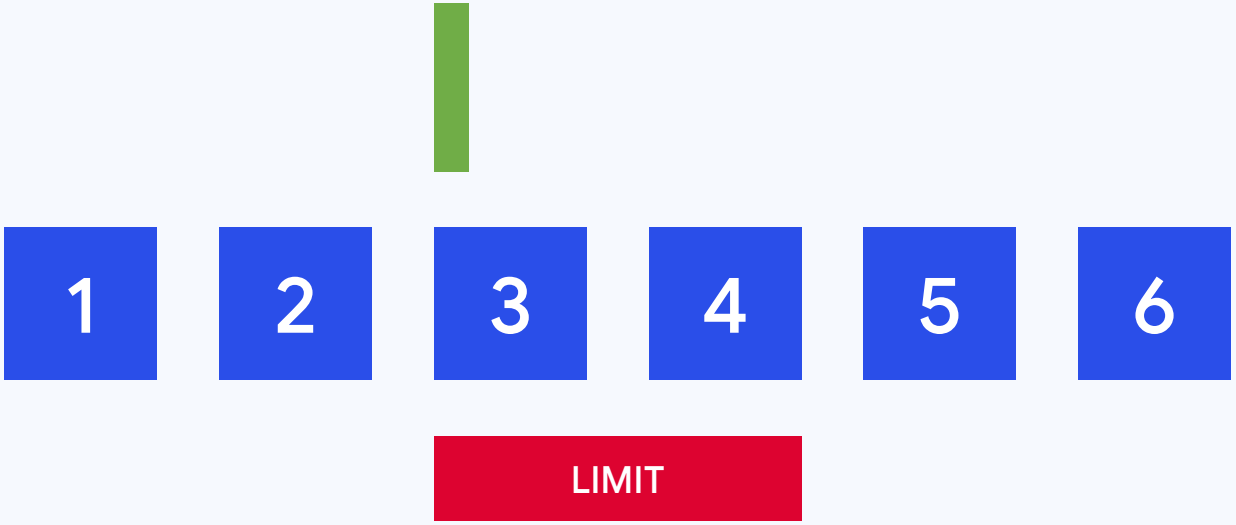
```
myapi.com/products&page=1
myapi.com/products&price=100
myapi.com/products&price_min=100?price_max=300
```

Paginación con limit & offset



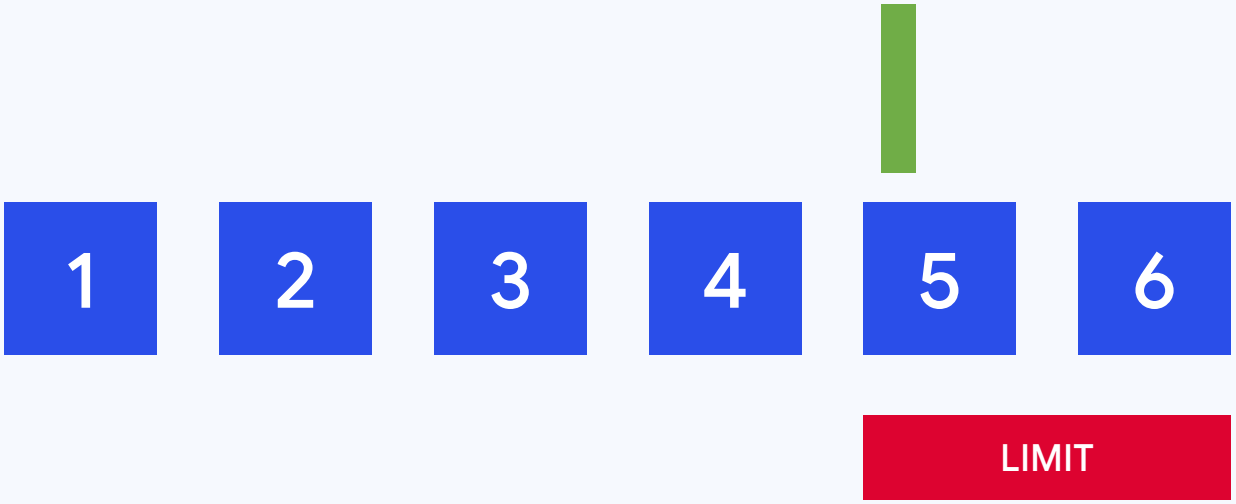
Paginación con limit & offset

Limit = 2
Offset = 2



Paginación con limit & offset

Limit = 2
Offset = 4



Observable & Promise

Promise

```
fetch('my-api-goes-here')
.then(resp => resp.json());
//
const data = await fetch('url');
const json = await data.json()
```

Observable

```
const data$ = new Observable("stuff here")

data$.subscribe(data => {
  // do some stuff here
}
```

Promises & Observables

Promesa

- Emite un solo valor
- Simplicidad

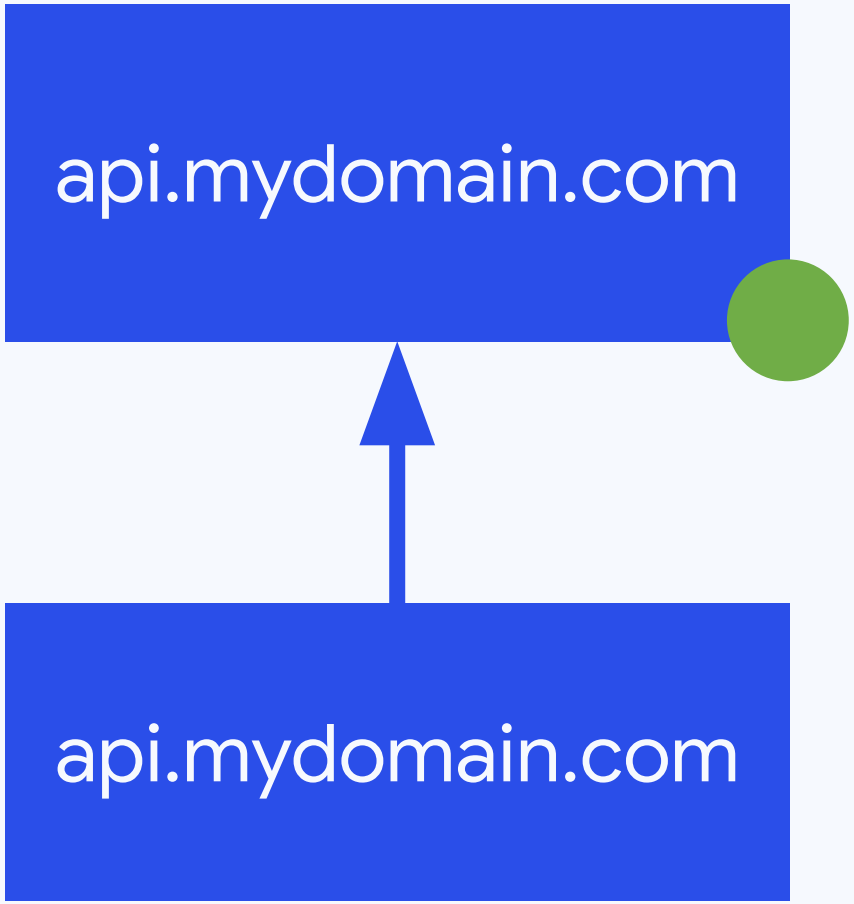
Observable

- Stream de datos (puede emitir múltiples valores).
- Es posible escuchar constantemente: eventos, responsive, fetchs.
- Se puede cancelar.

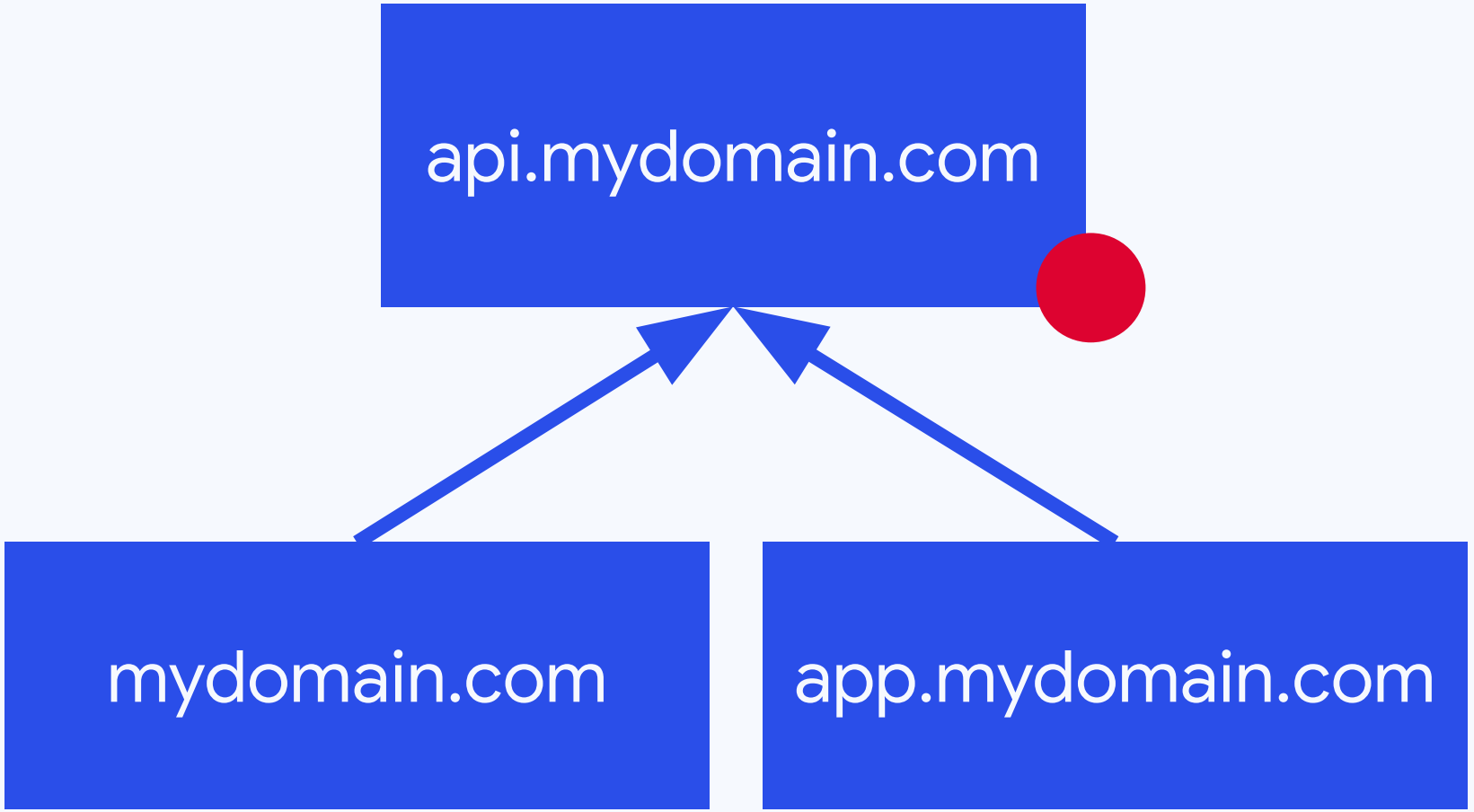
Reintentar petición

El problema de CORS

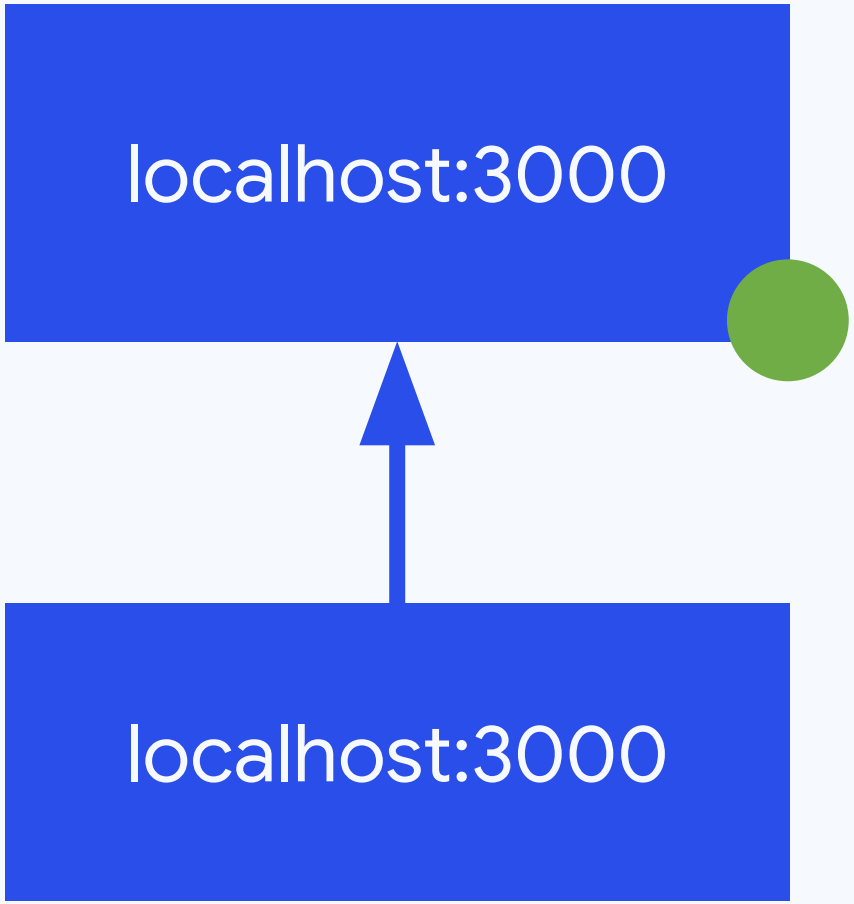
CORS



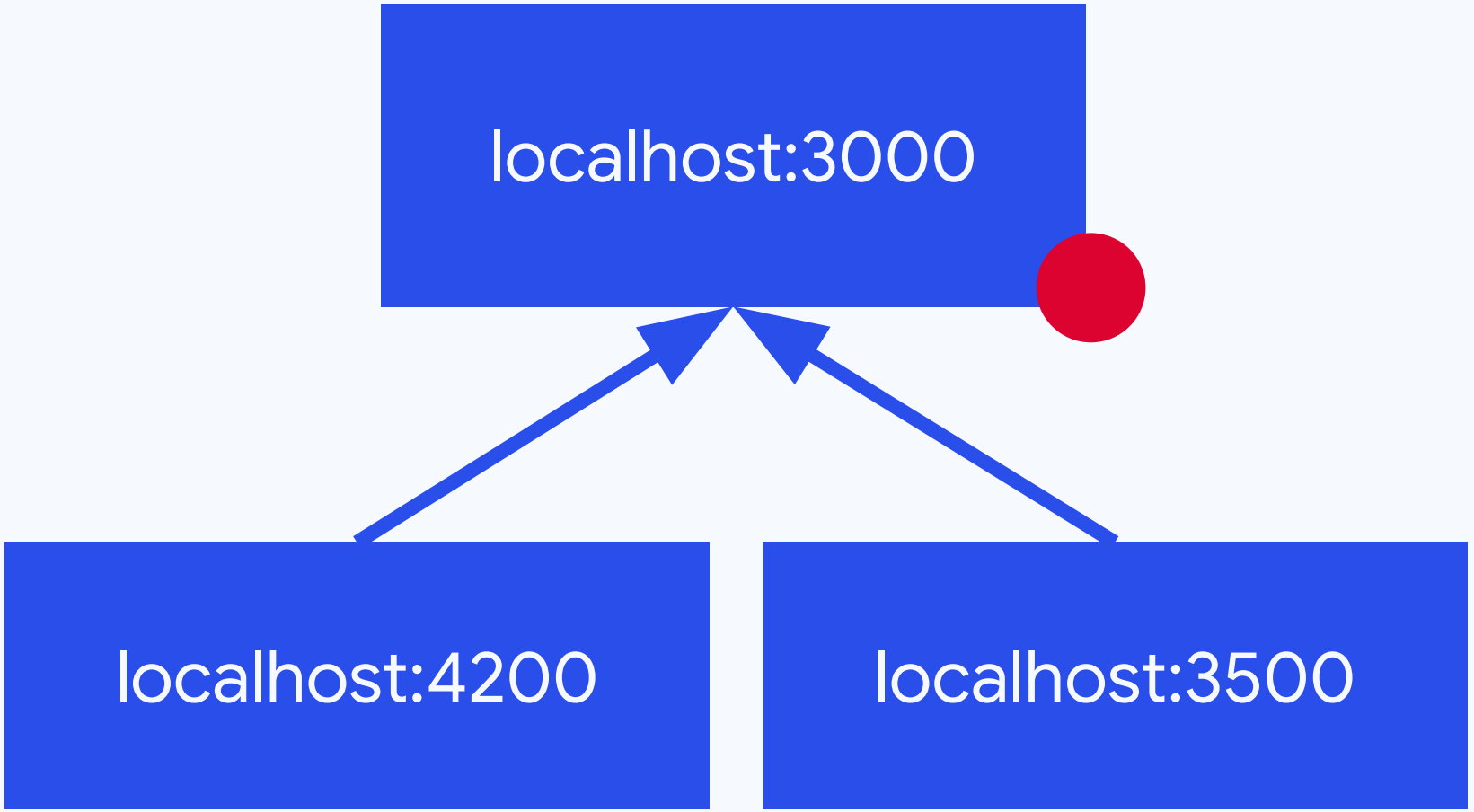
CORS



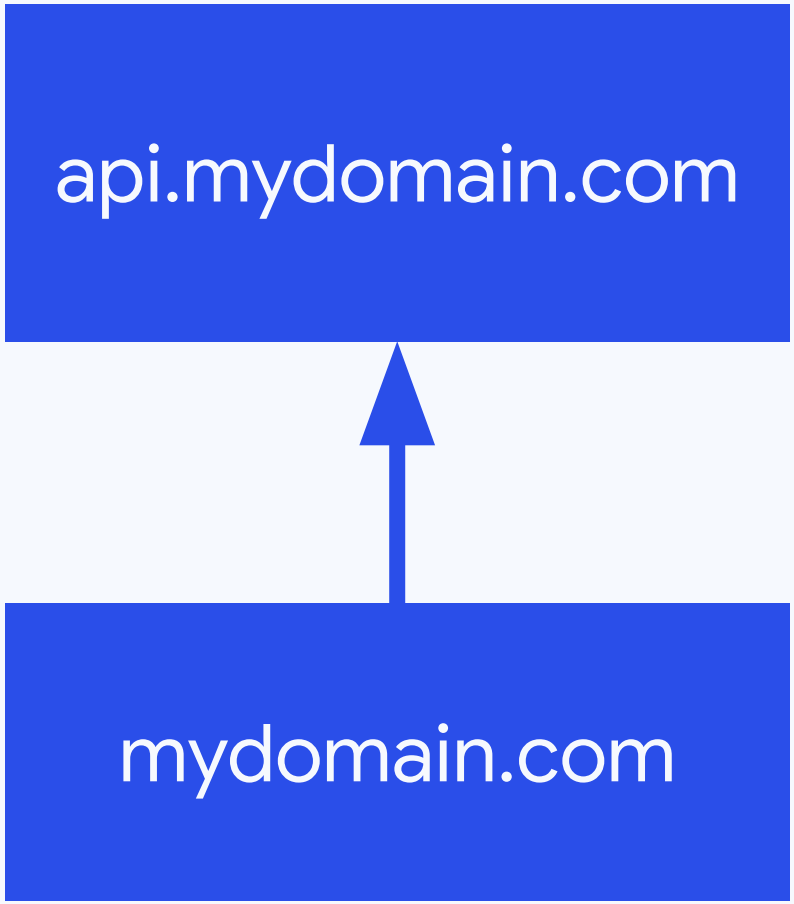
CORS



CORS



CORS



CORS

*

[mydomain.com, app.mydomain.com]

[..., localhost:4200]

El problema de CORS

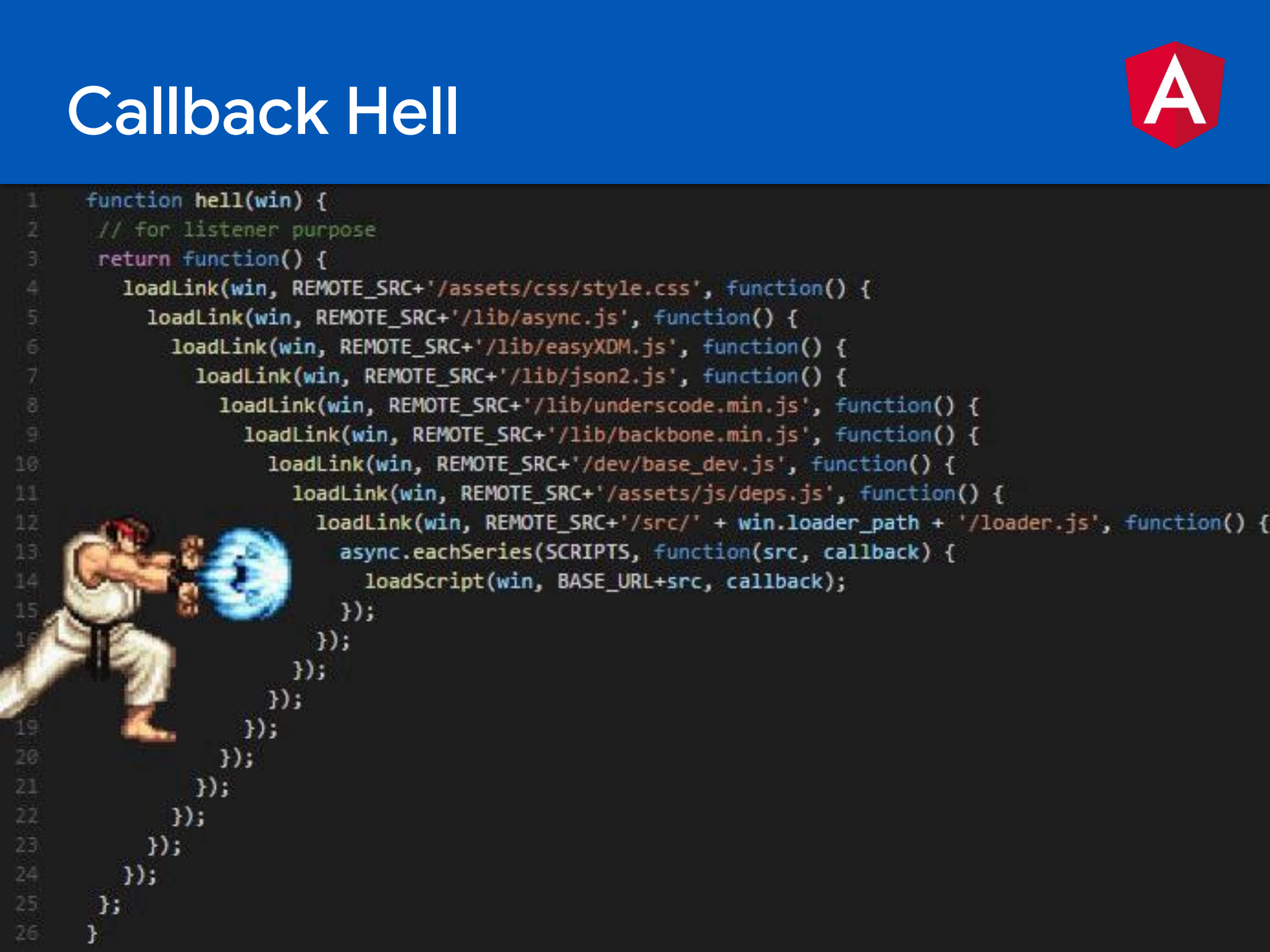
Manejo de ambientes

Manejo de errores

Transformar peticiones

Evitando el Callback Hell

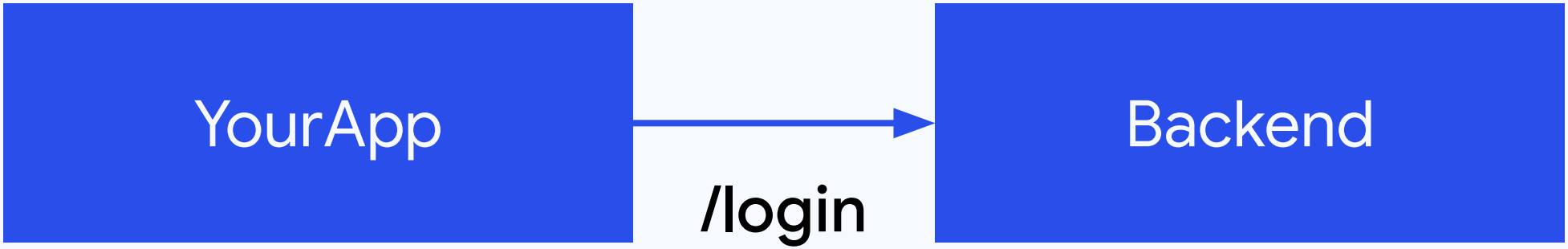
Callback Hell



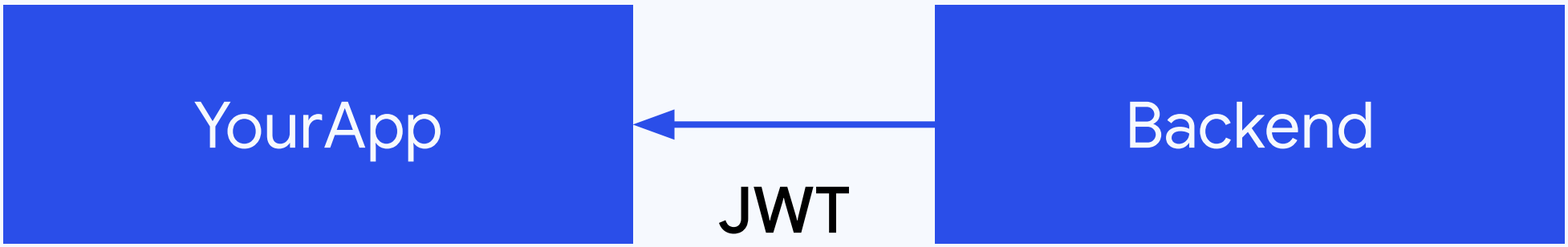
```
function hell(win) {
 // for listener purpose
 return function() {
   loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
     loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
       loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
         loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
           loadLink(win, REMOTE_SRC+'/lib/underscode.min.js', function() {
             loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
               loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
                 loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
                   loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
                     async.eachSeries(SCRIPTS, function(src, callback) {
                       loadScript(win, BASE_URL+src, callback);
                     });
                   });
                 });
               });
             });
           });
         });
       });
     });
   });
 };
}
```

Login y manejo de Auth

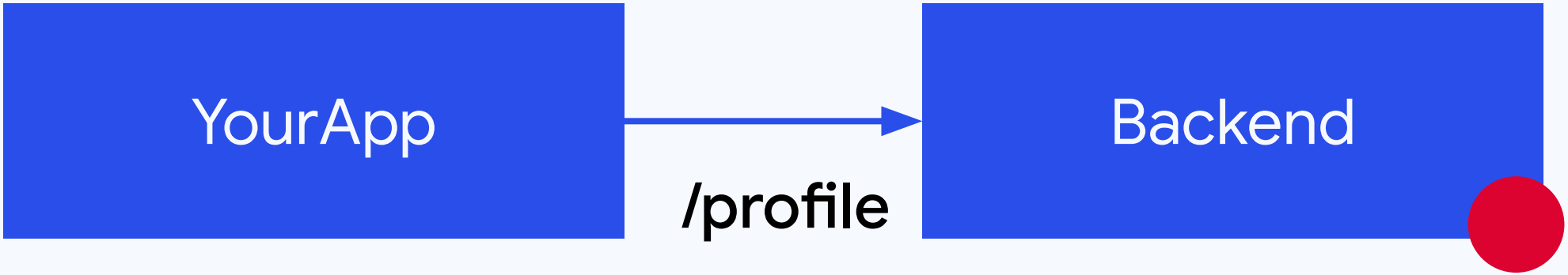
Manejo de Auth



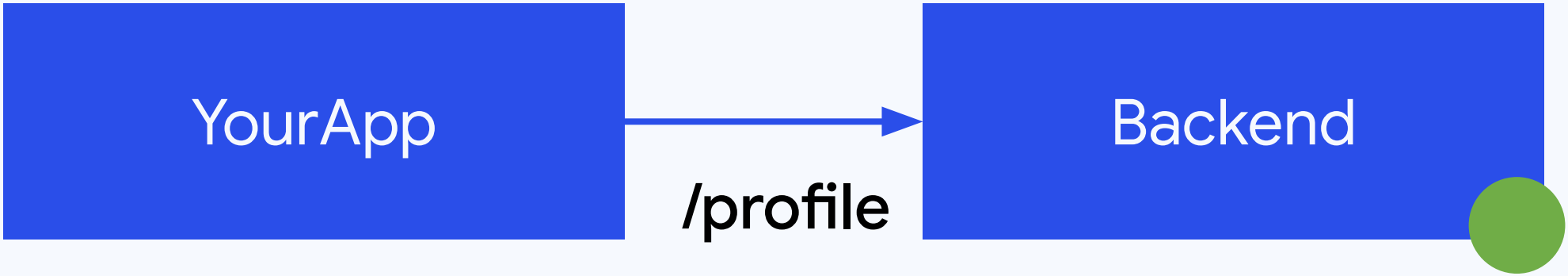
Manejo de Auth



Manejo de Auth



Manejo de Auth



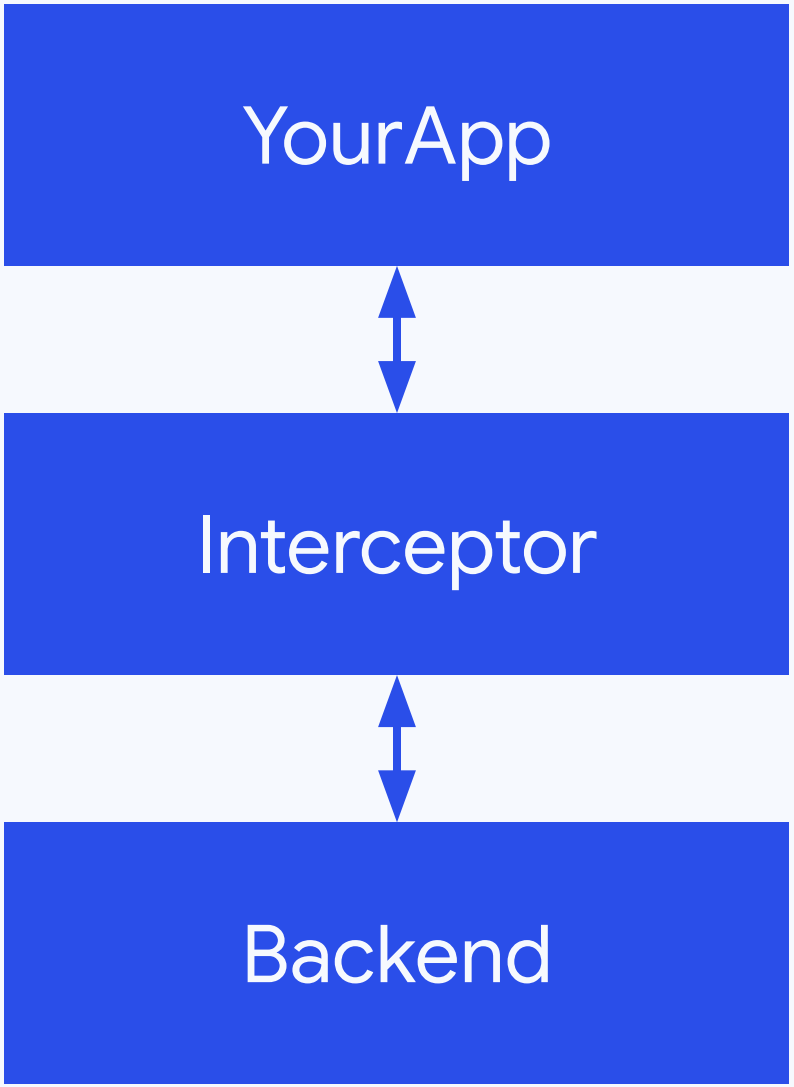
Customer Headers

Authorization

```
Authorization: <type> <credentials>
Authorization: Bearer ...
```

Interceptors

Interceptors



Enviar token con un interceptor

Token

- In-memory storage
- LocalStorage & SessionStorage
- Cookie storage

Creando un contexto a interceptor

Descarga de archivos con Http

Subida de archivos con Http