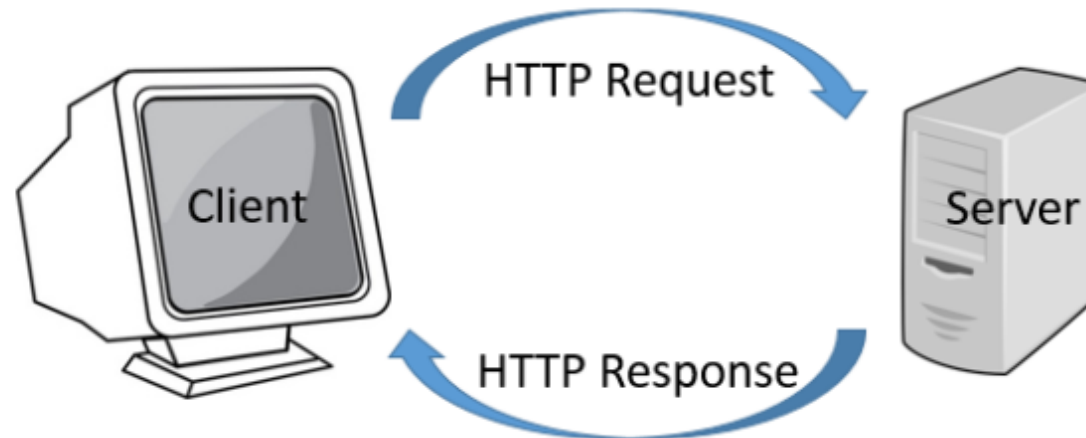


JS

AXIOS

How to request APIs

**HTTP is the format used to structure
request and responses on communication
between a client and a server**



In Javascript we use **Axios** to send an HTTP **request** and handle their **responses**,

⇒ **axios**

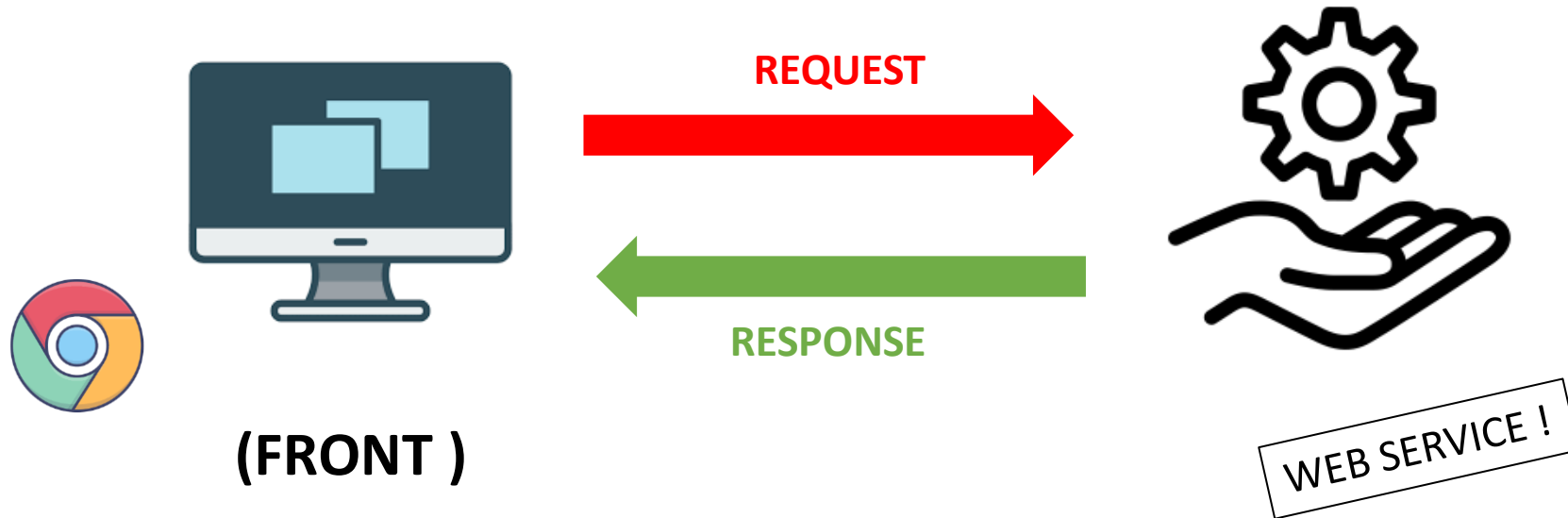


OBJECTIVES



- ✓ Understand what is an **API** and how to use them
- ✓ Understand the anatomy of an **HTTP request**
- ✓ Be able to define requests

On your website, you can ask
a web service to help you



What is an API ?



A web API (Application Programming Interface)

Gives you the menu :



WEB API

Endpoints (URL) to send requests to **web service**
parameters to specify for each request

`https://www.smashingmagazine.com/tag/javascript/?query1=value1&query2=value2`

Root-endpoint = host of server +

Path

PARAMETERS

= ENDPOINT

ACTIVITY 1

 10 MIN

Example of API : omdbapi

- 1 – Go to <http://www.omdbapi.com/>
- 2 – What are the **web services** available ? What are their **endpoints** ?
- 3 – What are the **parameters**, and their **roles** ?

OMDb API

The Open Movie Database

The OMDb API is a RESTful web service to obtain movie information, all content and images on the site are contributed and maintained by our users.

If you find this service useful, please consider making a [one-time donation](#) or [become a patron](#).



Poster API

The Poster API is only available to patrons.

Currently over 280,000 posters, updated daily with resolutions up to 2000x3000.

Example of API : omdbapi

Service 1 : movies

Service 2 : posters

Send all data requests to:

`http://www.omdbapi.com/?apikey=[yourkey]&`

Poster API requests:

`http://img.omdbapi.com/?apikey=[yourkey]&`

Endpoints

Parameters and roles ...

Parameters

By ID or Title

Parameter	Required	Valid Options	Default Value	Description
i	Optional*		<empty>	A valid IMDb ID (e.g. tt1285016)
t	Optional*		<empty>	Movie title to search for.
type	No	movie, series, episode	<empty>	Type of result to return.
y	No		<empty>	Year of release.
plot	No	short, full	short	Return short or full plot.
r	No	json, xml	json	The data type to return.
callback	No		<empty>	JSONP callback name.
v	No		1	API version (reserved for future use).

*Please note while both "i" and "t" are optional at least one argument is required.

By Search

A special parameter : the **API key**

Send all data requests to:

`http://www.omdbapi.com/?apikey=[yourkey]&`

Poster API requests:

`http://img.omdbapi.com/?apikey=[yourkey]&`

API key

= **Security** for accessing the service

Can be private/public with access restriction...

Generate API Key

Account Type

☐ Patreon

☒ **FREE!** (1,000 daily limit)

ACTIVITY 2

🕒 10 MIN

Find the URL !

Send in message to the conversation, what is the corresponding URL for each service ! **The first who is correct wins the point !**

QUERY	URL
1- Get info of « Avengers »	1: http://www.omdbapi.com/?apikey=[yourkey]&t=Avengers

Get ready !

ACTIVITY 2


🕒 10 MIN

Find the URL !

Send in message to the conversation, what is the corresponding URL for each service ! The first who is correct wins the point !

QUERY	URL
1- Get all info of « Fight Club »	
2- Get all movies from 2013 with « jedi » in title	
3- Get the full plot of the serie named « utopia »	
4- Search all posters of series with title « potatoe »	
5- get the second page of search for all movies with « friend » in their title, get the result in xml type	

ACTIVITY 2

 10 MIN

Find the URL !

QUERY	URL
1- Get the info of « Fight Club »	http://www.omdbapi.com/?apikey=[yourkey]&t=Fight+Club
2- Search all movies from 2013 with « jedi » in title	http://www.omdbapi.com/?apikey=[yourkey]&s=jedi&type=movie&y=2013
3- Get the full plot of the serie named « utopia »	http://www.omdbapi.com/?apikey=[yourkey]&t=utopia&type=serie&plot=full
4- Search all posters of series with title « potatoe »	http://www.img.omdbapi.com/?apikey=[yourkey]&s=potatoe&type=serie
5- get the second page of search for all movies with « friend » in their title, get the result in xml type	http://www.omdbapi.com/?apikey=[yourkey]&s=friend&type=movie&page=2&r=xml

How to use a service

1 – Read the API doc, chose the operation you want to do

EX : I want to get information about a movie : Fight Club

2 – Analyze the URL, and parameters you need to give

EX : url is <http://www.omdbapi.com/>

parameters are : **apikey=mykey** **type=movie** **t=Fight Club**

3.0 – Try your request (with the REST plugin)

3.1 – Setup your URL in your js code, and give it to axios

How to use a service

Sometimes parameters are variables of your code !



EXAMPLE (S5)

```
let user_name = Edouard;  
let user_url = 'https://api.genderize.io/?name=' + user_name;  
axios.get(user_url).then(afterRequest);
```

ACTIVITY 3 20 MIN

Instructions are in the code

Anatomy of a request

Header

What is inside ?
Who sends it ?
Is it authorized ? (key)



Body Only for Post/Put

The data you send
(usually JSON or XML)

Delivery Address
= URL

Sometimes parameters are
in the URL instead of
header/body

Example

URL

[https://api.hubapi.com/crm/v3/objects/companies/search?hapikey=\[mykey\]](https://api.hubapi.com/crm/v3/objects/companies/search?hapikey=[mykey])

Header

KEY	VALUE
Postman-Token ⓘ	<calculated when request is sent>
Content-Type ⓘ	application/json
Content-Length ⓘ	<calculated when request is sent>
Host ⓘ	<calculated when request is sent>
User-Agent ⓘ	PostmanRuntime/7.26.8
Accept ⓘ	*/*
Accept-Encoding ⓘ	gzip, deflate, br
Connection ⓘ	keep-alive

Body

```
{
  "filterGroups": [
    {
      "filters": [{"value": "Tesla", "propertyName": "Name", "operator": "EQ"}]
    }
  ],
  "sorts": ["string"],
  "query": "string",
  "properties": ["string"],
  "limit": 0,
  "after": 0
}
```

With Axios :

Define the header object

```
const headers = {  
  'Content-Type': 'application/json',  
  'Authorization': 'my token'  
}  
  
axios.post('my URL', body, {headers: headers})  
  .then((response) => {  
    '...'  
  })  
  .catch((error) => {  
    '...'  
  })
```

Give body and header to axios

What about HTTP errors ?



#61954375

Read the STATUS

Every request returns a status code

01 **200+** means the request has **succeeded**.

Your request is wrong !

02 **300+** means the request is **redirected** to another URL

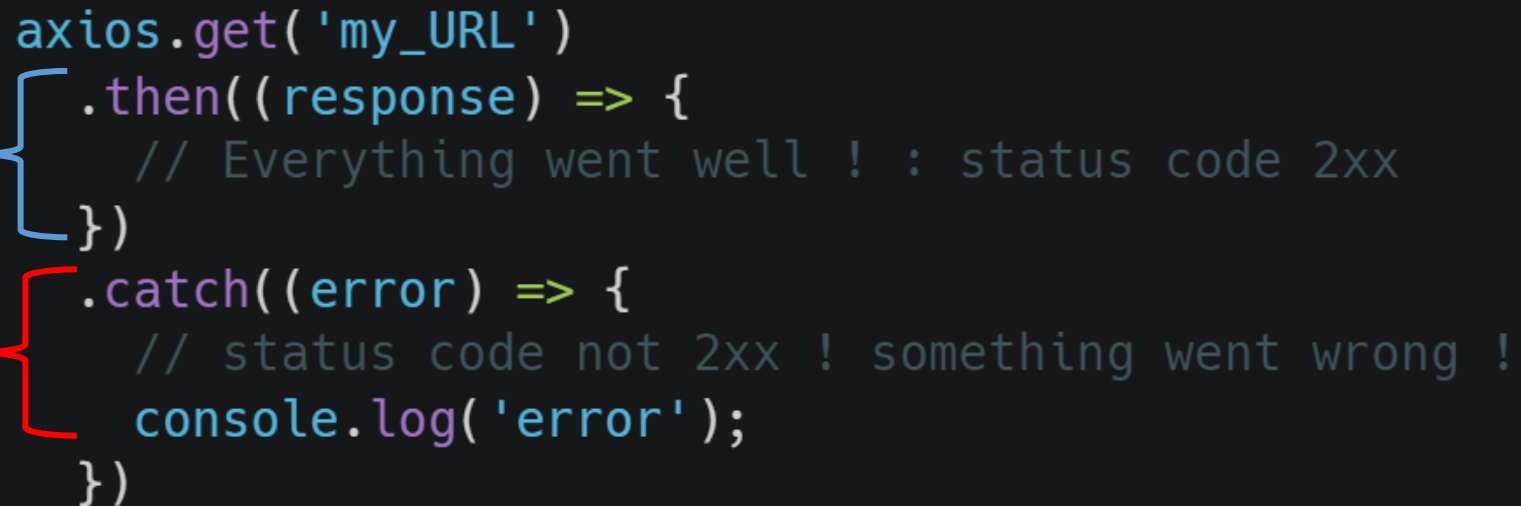
03 **400+** means an **error that originates from the client** has occurred

04 **500+** means an **error that originates from the server** has occurred

Axios error handling :

Status code 2xx

Not 2xx



```
axios.get('my_URL')
  .then((response) => {
    // Everything went well ! : status code 2xx
  })
  .catch((error) => {
    // status code not 2xx ! something went wrong !
    console.log('error');
  })
```

Note : in case of error, maybe there is still something in “error.response” !



NOW YOU KNOW



- ✓ What an **API** is, and how to use it
- ✓ How is made an **HTTP request**, and how to define it
- ✓ How to use **AXIOS** in a Javascript projects to execute **HTTP requests** and use **web services** !

Have fun practicing with fun API online !

Useful links :

- List of HTTP status codes:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

- list of public API you can use :

<https://mixedanalytics.com/blog/list-actually-free-open-no-auth-needed-apis/>

- Request bin online :

<https://requestbin.com/>