Software Development for Web and Mobile

Communication between backend and frontend

Pepe García jgarciah@faculty.ie.edu

Plan for today

Today we will:

- Learn how to communicate with http services
- Do some exercises in class

Fetch

fetch is the way of calling HTTP services from Javascript.

```
fetch("url"); //done!
```

Fetch

We can customize our request using the second parameter:

```
fetch(
    "url"






);
```

Fetch

We can customize our request using the second parameter:

```
fetch(
    "url",
    {
        method: "POST",
        headers: {},
        body: "this is the body"
    }
);
```

Fetch

but, how do we use the data returned from the server?

let's open the console and see what does the following snippet return.

```
let a = fetch("http://google.com");
```

Promises

Promises are the solution used in JS for when we don't want to **block the program** while a long running task is made.

By using promises, we create **asynchronous** code.

Promises

fetch uses **Promises** to work asynchronously.

Promises can be in different states:

- **pending**: the promise hasn't finished yet.
- **fulfilled**: the promise finished correctly.
- **rejected**: there was an error in the promise

Promises

Promises

We use the methods **then** and **catch** to handle the different outcomes of the promise (**fulfilled** and **rejected** respectively)

```
fetch("https://google.com")
  .then((result) =>
    console
      .log("the promise is fulfilled, and returned" + result)
  )
  .catch((error) =>
    console
      .log("the promise failed with " + error)
  )
```

Back to fetch

To get the JSON response from fetch we need to use promise's **then** method:

```
fetch("http://api.open-notify.org/iss-now.json")
  .then(data => data.json())
  .then(json => console.log(json))
```

Practice

Let's do this couple of exercises in class!

Exercises

1. **Exercise 1** Let's create a webpage that displays information of a random movie. We'll use a couple of APIs to do it! `8f63893f`
2. **Exercise 2** Let's implement the frontend for an application like Twitter.