

PROJECTE

Participants: Toni Susany, Iber Tallon i Bryan Alvitres

Assignatura: M14 Projectes

January 22, 2024

ÍNDEX

- ```

 var transitionDuration = Util.getTransitionDuration(
 $(activeElement).one(Util.TRANSITION_END, function () {
 $(nextElement).removeClass(className + " " +
 $.fn.jquery.split('.')[0] + ".transition");
 $(activeElement).removeClass(className);
 })
);

 if (!$.support.transition) return;
 setTimeout(function () {
 return $(_this._element).trigger(CustomEvent);
 }, 0);
 $.emulateTransitionEnd(transitionDuration);
} else {
 $(activeElement).removeClass(className);
 $(nextElement).addClass(className);
}

```

# 1. Introducció

Avui explorarem la creació d'una pàgina web que implementa una API-REST i autenticació OAuth. A través d'aquest projecte, construirem un exemple fictici inicial, seguit d'una aplicació que s'entronca amb el servei ben conegut de Spotify. Aquesta aventura en el desenvolupament web no només ens portarà a la comprensió de les APIs en REST, AJAX i altres conceptes clau, sinó que també ens mostrarà com connectar-se amb recursos externs com Spotify per enriqueir la funcionalitat de la nostra pàgina web.

## 2. Què és una API en REST?

Una arquitectura per aprofitar el protocol HTTP en aplicacions distribuïdes.

- ▶ REST o RESTful: Reemplaça architectures com SOAP o XML-RPC.
- ▶ Sis restriccions per a servidors web REST.

Conceptes claus:

- ▶ **Client-Server:** Comunicació sense emmagatzemar informació del client.
- ▶ **Uniform Interface:** Defineix la interfície client-servidor.
  - ▶ Identificació dels recursos (via URI o URL).
  - ▶ Manipulació dels recursos.
  - ▶ Missatges autodescriptius.
  - ▶ HATEOAS (Hypermedia As The Engine Of Application State).

### 3. AJAX

- ▶ Utilitzat per aplicacions web dinàmiques.
- ▶ Processa peticions i respostes del servidor REST.

#### Propietat `readyState` en AJAX

- ▶ **0**: Petició no inicialitzada
- ▶ **1**: Connexió de servidor establerta
- ▶ **2**: Petició rebuda
- ▶ **3**: Petició en procés
- ▶ **4**: Petició finalitzada i resposta llesta

#### Codis de resposta HTTP comuns

- ▶ **200**: OK (Petició satisfactòria)
- ▶ **201**: Creat (En peticions POST, recurs creat)

### 3. AJAX (ERORS)

#### Errors del client (Status 400)

- ▶ **400**: Bad Request (Error de sintaxi)
- ▶ **401**: Unauthorized (Autorització incorrecta)
- ▶ **403**: Forbidden (Accés prohibit)
- ▶ **404**: Not Found (Recurs no trobat)

#### Errors del servidor (Status 500)

- ▶ **500**: Internal Server Error (Error intern del servidor)
- ▶ **503**: Service Unavailable (Servidor no disponible)

## 4. Com es vincula REST amb HTTP

### Mètodes per HTTP:

- ▶ **GET:** Demana dades d'un recurs específic.
- ▶ **POST:** Recull dades i crea o actualitza un recurs.
- ▶ **PUT:** Envia dades per crear o actualitzar un recurs.
- ▶ **DELETE:** Esborra un recurs determinat.

### Mètode amb AJAX:

- ▶ **XMLHttpRequest:** Actualitza pàgina sense recarregar completament.

#### Funcionament:

- ▶ **Usuari:** Realitza cerca.
- ▶ **Client:** Envia petició al servidor.
- ▶ **Servidor:** Executa aplicació i retorna resposta al client.

## 5. JQuery

- ▶ Framework de JavaScript
- ▶ Llibreria JavaScript per manipular objectes HTML.
- ▶ Facilita la manipulació d'HTML a través de tots els navegadors.
- ▶ Simplificació de peticions AJAX



## 6. JSON

- ▶ **Definició:** Format lleuger de intercanvi de dades.
- ▶ **Objectius:** Fàcil de llegir i escriure per humans, fàcil de generar i interpretar per les màquines.
- ▶ **Estructura:** Col·lecció de parells clau/valor.

### Funcionalitat General

- ▶ **Notació d'Objectes:** JSON permet representar objectes JavaScript.
- ▶ **Integració:** Comunicació eficient entre servidor (JSON) i client (JavaScript).