

# DAM - Accés a dades

## Tema 4 - Servicis i recursos web

Roberto Sanz Requena

[rsanz@florida-uni.es](mailto:rsanz@florida-uni.es)

# Índex

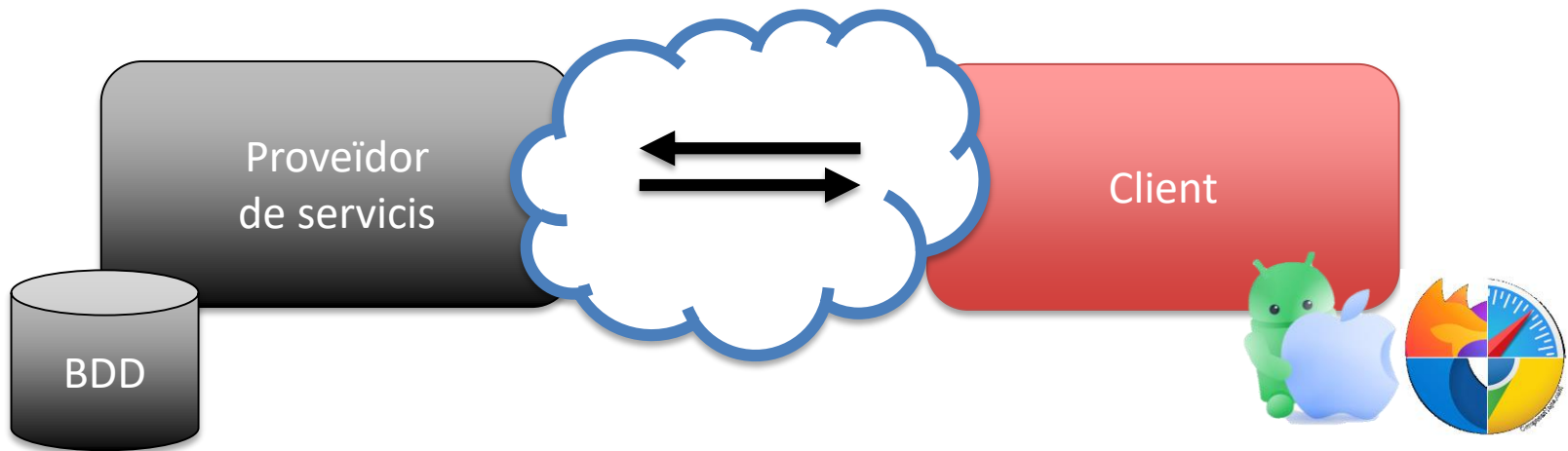
1. Context
2. Servicis web
3. REST APIs
4. Desenvolupament d'un client REST basat en web
5. Desenvolupament d'un servici REST API

# 1. Context

- Accés a dades
  - Fitxers
    - Entorn local
  - Bases de dades relacionals (SQL)
    - Molt estructurat, llenguatge de consultes potent, limitacions d'escalabilitat
  - Bases de dades No-SQL
    - Orientació a objectes, escalabilitat, distribució
- Tendència actual: augment d'aplicacions basades en Internet que generen i consumeixen dades contínuament
  - Necessitat de facilitar l'accés i la integració de dades entre aplicacions

## 2. Servicis web

### Protocols i estàndars

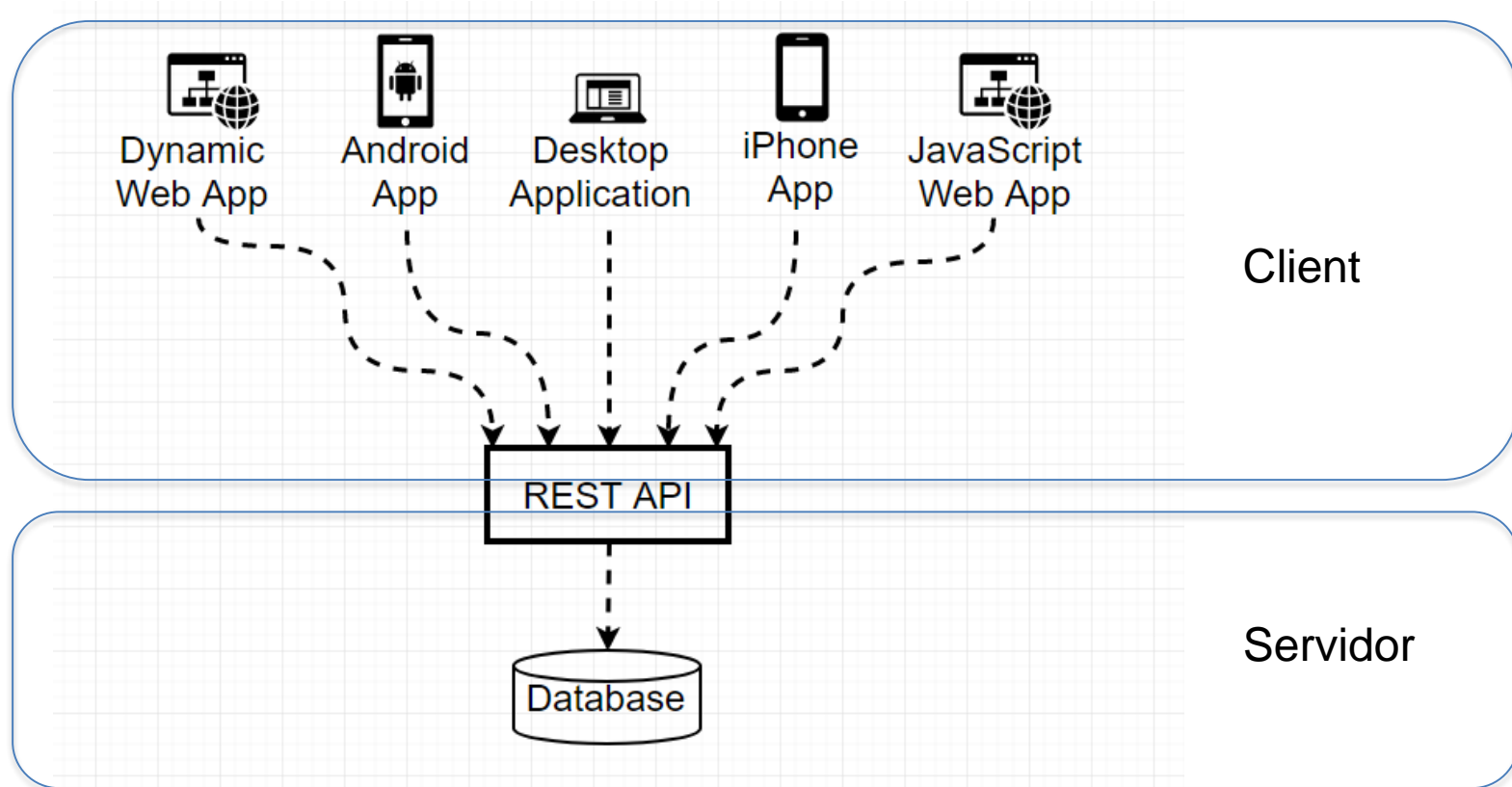


<https://www.ibm.com/docs/es/rsas/7.5.0?topic=overview-web-services-standards>

[https://es.wikipedia.org/wiki/Servicio\\_web](https://es.wikipedia.org/wiki/Servicio_web)

<http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/320>

### 3. REST APIs



<https://happycoding.io/tutorials/java-server/rest-api>

### 3. REST APIs

- Representational state transfer (transferència d'estat representacional)
- Arquitectura software que s'executa sobre **HTTP** per a obtenir dades o indicar l'execució d'operacions sobre les dades, en qualsevol format (XML, JSON, etc).
- Centrada en les dades (**recursos**), a diferència de protocols com SOAP
- 1 recurs = 1 **URI** (Uniform Resource Identifier) ( $\neq$  URL)
- **RESTful**: servicis que implementan REST i que en executar-los en un navegador tornen un contingut legible per un ésser humà
- Operacions GET, PUT, POST, DELETE (analogia CRUD)

[https://es.wikipedia.org/wiki/Transferencia\\_de\\_Estado\\_Representacional](https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional)  
<https://restfulapi.net/http-methods/>

# 3. REST APIs

## Tipus de recursos

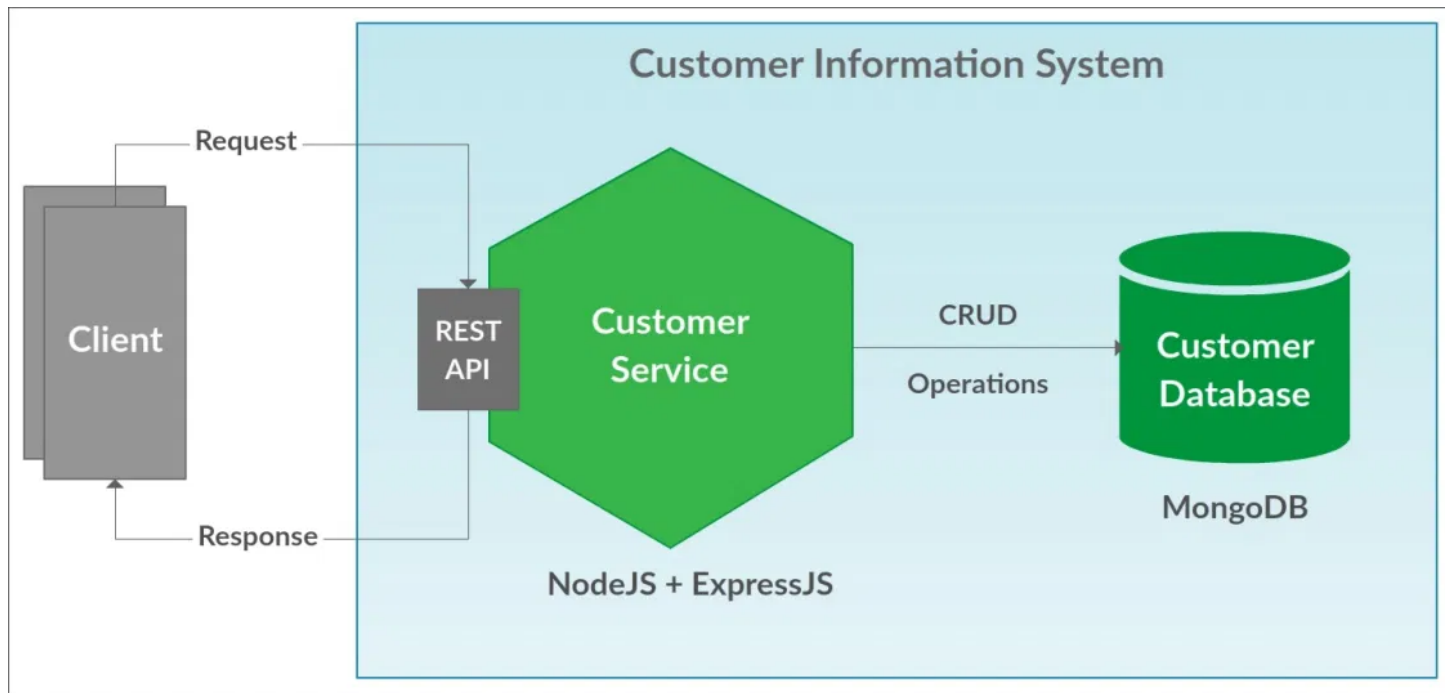
- Document (document): objecte o registre únic d'una BDD  
`http://api.example.com/user-management/users/{id}`
- Collection (col·lecció): directori d'objectes gestionat pel servidor  
`http://api.example.com/user-management/users`
- Store (magatzem): directori d'objectes gestionat pel client  
`http://api.example.com/song-management/users/{id}/playlists`
- Controller (controlador): funcions que admeten/tornen paràmetres  
`http://api.example.com/song-management/users/{id}/playlist/play`

## Importància d'anomenar bé els elements (endpoint URIs)

<https://restfulapi.net/resource-naming/>

### 3. REST APIs

#### Exemple d'arquitectura

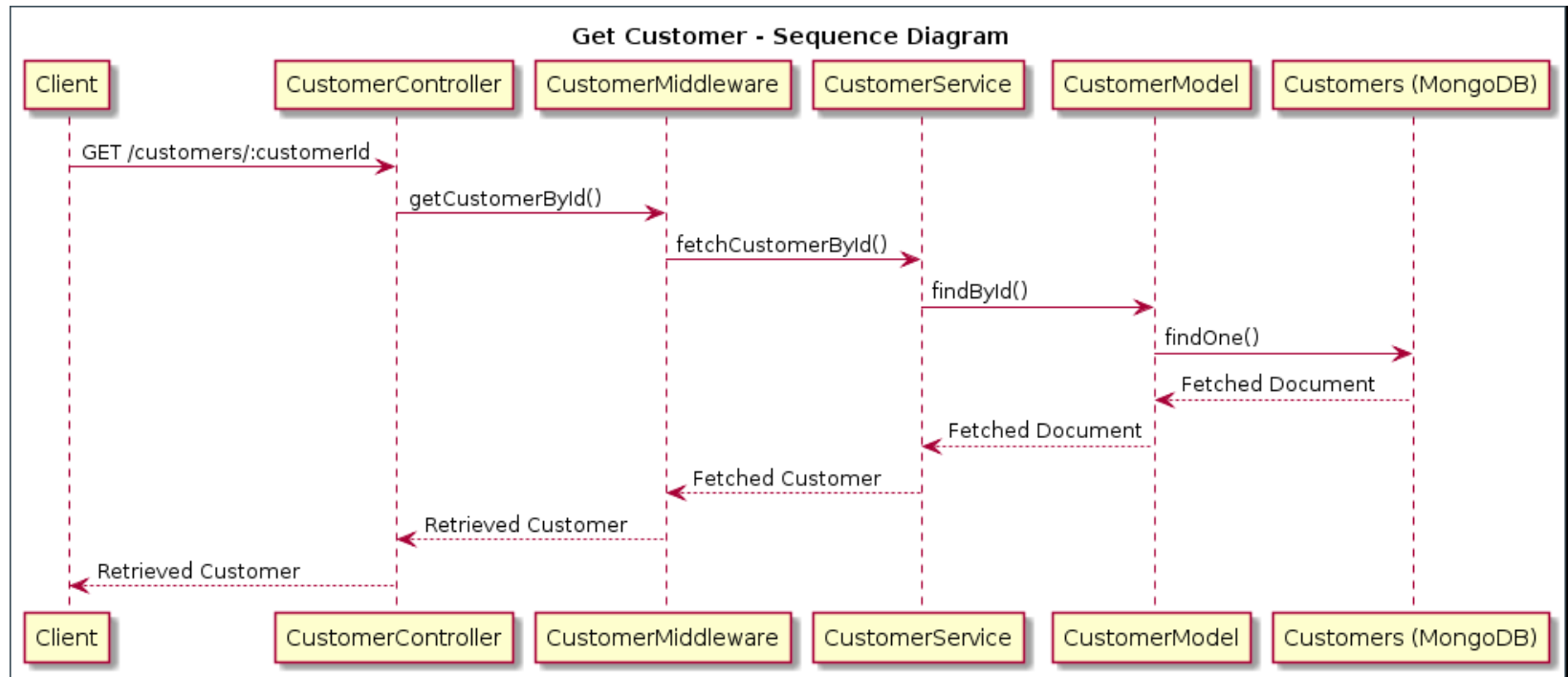


<https://novicedeveloper.com/architecture-design-rest-api-nodejs-api-tdd/>



# 3. REST APIs

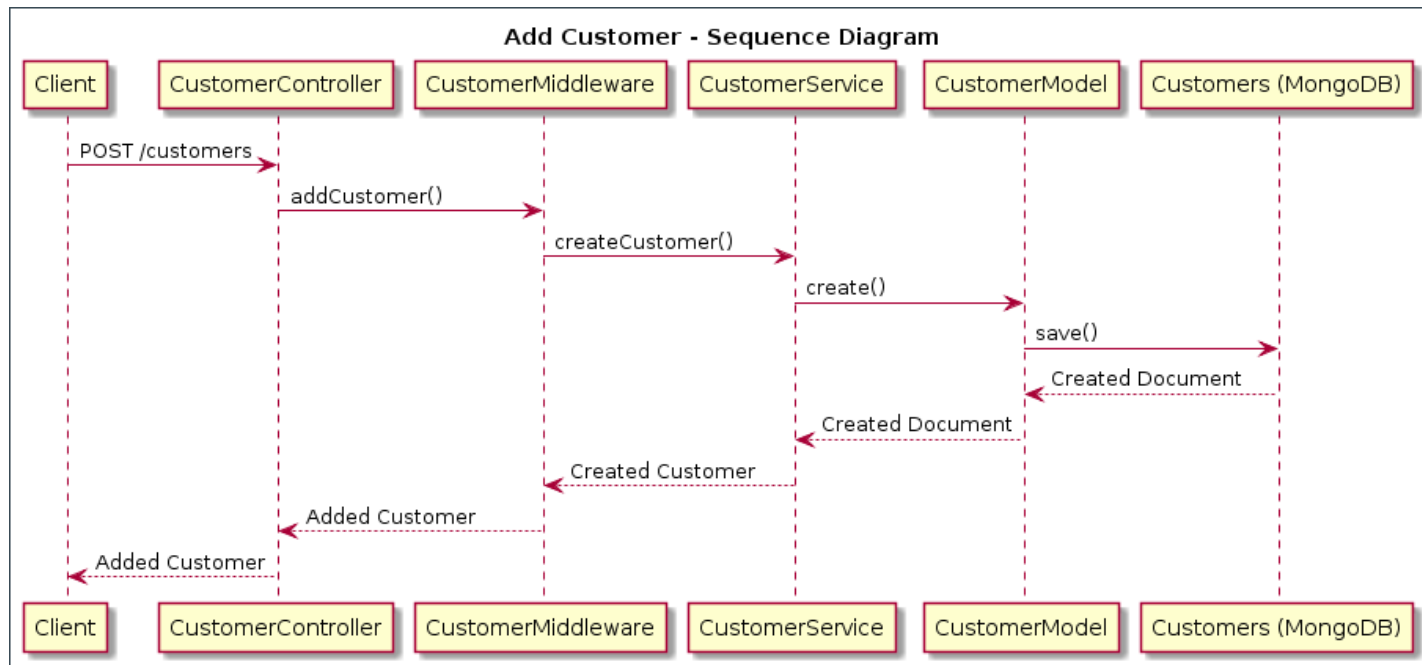
## Operació GET



<https://novicedeveloper.com/architecture-design-rest-api-nodejs-api-tdd/>

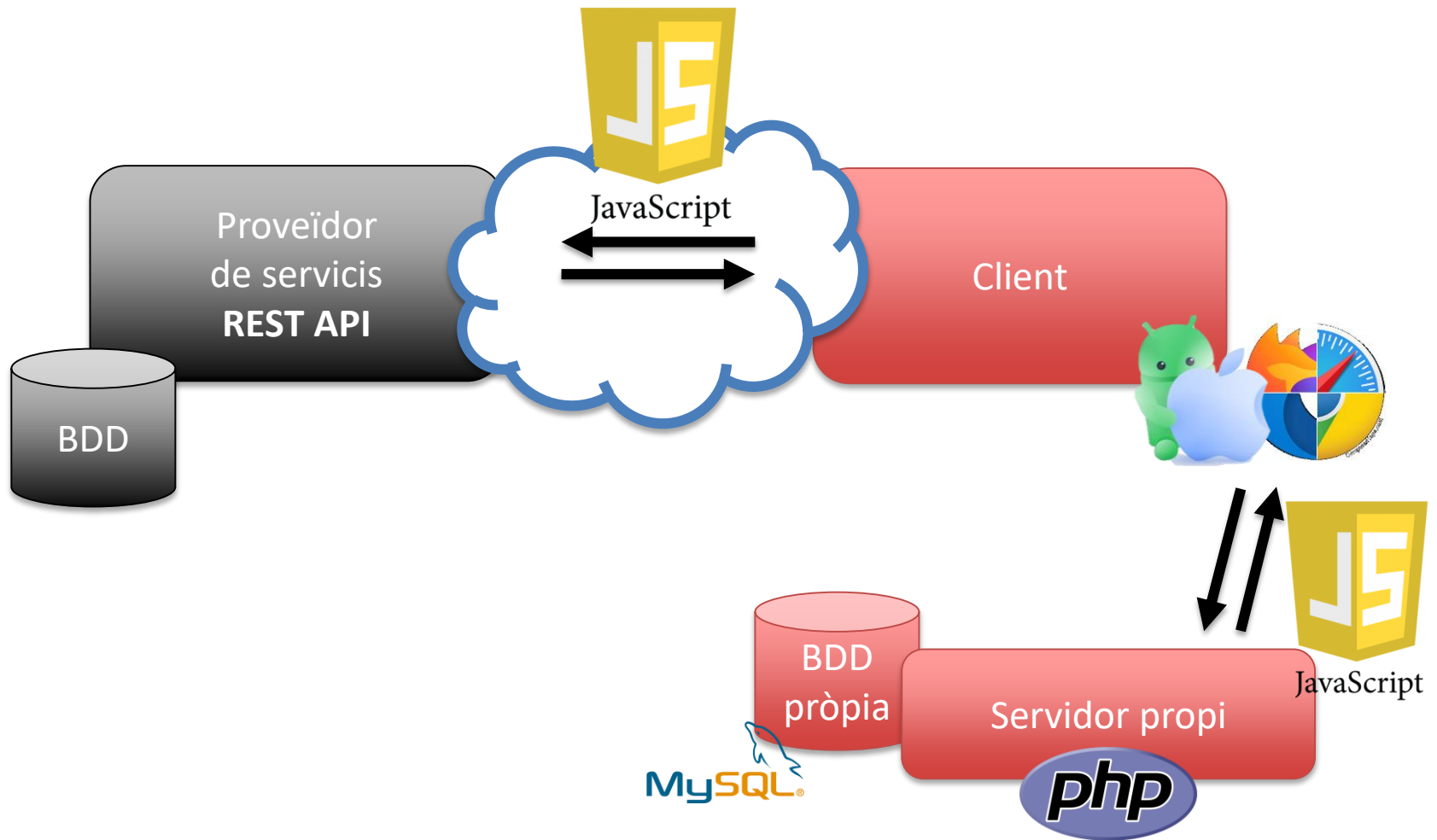
# 3. REST APIs

## Operació POST



<https://novicedeveloper.com/architecture-design-rest-api-nodejs-api-tdd/>

## 4. Desenvolupament d'un client REST basat en web



## 4. Desenvolupament d'un client REST basat en web

Passos:

- Peticions de recursos a través de REST → Javascript Axios
- Gestió d'objectes JSON → response.data
- Gestió de la interfície gràfica → HTML, Javascript (getElementById,...)

XMLHttpRequest  
fetch  
jQuery

També

<https://levelup.gitconnected.com/all-possible-ways-of-making-an-api-call-in-plain-javascript-c0dee3c11b8b>

```
<script src = "https://unpkg.com/axios/dist/axios.min.js"></script>

<script>

function getGroupInfo() {

    var strBusquedaGrupo = document.getElementById('strBusquedaGrupo').value;
    axios
    .get("https://theaudiodb.com/api/v1/json/1/search.php?s=" + strBusquedaGrupo)
    .then(response => {

        console.log(response.data);
        document.getElementById('strBiographyEN').value =
            response.data.artists[0].strBiographyEN;
        document.getElementById('imagenGrupo').src = response.data.artists[0].strArtistThumb;
        document.getElementById('imagenGrupo').style.display = "block";

    })
    .catch(error => {

        console.error(error);
        document.getElementById('strBiographyEN').value = "ERROR EN LA CONSULTA: " + error;
        document.getElementById('imagenGrupo').style.display = "none";

    });

}

</script>
```

## POST

```
await axios.post(  
  'localhost/...', //URL  
  { //Dades  
    title: title,  
    description: description,  
  },  
  { //Configuracio (opcional)  
    headers: {  
      "x-access-token": "token-value",  
    },  
  }  
);
```

## PUT

```
axios.put(  
  'localhost/...', //URL  
  { //Dades  
    title: title,  
    description: description,  
    published: true,  
  },  
  { //Configuracio (opcional)  
    headers: {  
      "x-access-token": "token-value",  
    },  
  }  
);
```

## DELETE

```
axios.delete(  
  'localhost/recurso', //URI  
  { //Configuracio (opcional)  
    headers: {  
      "x-access-token": "token-value",  
    },  
  }  
);
```



**Florida**

Grup Educatiu

## 4. Desenvolupament d'un client REST basat en web

Enviament de dades al servidor (Base de dades):

- Enviament de dades del client al servidor mitjançant POST → Javascript jQuery
- Script PHP en servidor → Connexió i guardat en MySQL
- Gestió del resultat en el client → Javascript jQuery

## 4. Desenvolupament d'un client REST basat en web

Aquest codi en el client web:

```
<script src = "https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script type = "text/javascript">
function guardar() {
    $.ajax({
        type: "POST", //metode POST per a enviar dades al servidor
        url: "ficheroPHP", // ruta del fitxer PHP del servidor
        data: datos, // dades a enviar (p.ex. {valor:valor, nom:nom})
        success: function (response) { //resultat del PHP del servidor
            alert(response);
        },
        error: function () {
            alert("Error");
        }
    });
}
</script>
```



## 4. Desenvolupament d'un client REST basat en web

<?php

```
if(isset($_POST["valor"])){
    $valor = $_POST["valor"];
    $nombre = $_POST["nombre"];
    $servidor = "localhost";
    $usuario = "root";
    $password = "";
    $dbname = "prueba";
    $conexion = mysqli_connect($servidor, $usuario, $password, $dbname);
    if (!$conexion) {
        echo "Error en la conexion a MySQL: ".mysqli_connect_error();
        exit();
    }
    $sql = "INSERT INTO tabla (colValor,colNombre) VALUES ('".$valor."','".$nombre."')";
    if (mysqli_query($conexion, $sql)) {
        echo "Registro insertado correctamente.";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conexion);
    }
}

?>
```

Aquest codi al servidor:  
xampp → htdocs

## 4. Desenvolupament d'un client REST basat en web

Estructura completa del client:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Mi Gramola</title>
    <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
    <script>ACÍ EL COS DEL SCRIPT PER A CONSULTA</script>
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
    <script>ACÍ EL COS DEL SCRIPT PER A GUARDAR</script>
  </head>
  <body>
    <h1>Mi Gramola</h1>
    <p>Grupo/Artista:</p>
    <p><input type="text" value="" id="strBusquedaGrupo">
    <button id="botonBusquedaGrupo" onclick="getGroupInfo()">Buscar</button>
    </p>
    <img id="imagenGrupo" src="" height="300" style="display:none;" />
    <p>Biograf&iacute;tea:</p>
    <textarea id="strBiographyEN" rows="10" style="width:100%"></textarea>
    <p>Discograf&iacute;tea:</p>
    <textarea id="strDiscography" rows="10" cols="50"></textarea>
    <p><button id="botonGuardar" onclick="guardarInfo()">Guardar</button></p>
  </body>
</html>
```

# Activitat Entregable 7 - Client web REST

Presentació de l'Activitat Entregable 7

REST APIs

[https://theaudiodb.com/api\\_guide.php](https://theaudiodb.com/api_guide.php)

<https://openlibrary.org/developers/api>

<http://www.omdbapi.com/>

<https://openweathermap.org/api>

<https://github.com/public-apis/public-apis>

<https://rapidapi.com/collection/list-of-free-apis>

Requereix API Key

Llista

## 5. Desenvolupament d'un servici REST API

Passos:

- Servici REST API per a gestió BDD cançons (HTTP)
- Operacions GET, POST, PUT y DELETE
- Base de dades
  - MySQL
  - MySQL mapejada amb Hibernate
  - MongoDB
- Entorn de proves (Postman)

**Tema 5**  
**Programación de**  
**servicios y procesos**