

# Petición de datos a php con httpclient

**Desarrollo frontend - Unidad 5**

# Creación del api php

— — —

- ¿Qué es un API?
  - Un API (Application Programming Interface, por sus siglas en inglés) es un conjunto de reglas y protocolos que permiten que distintos sistemas informáticos se comuniquen entre sí. Proporciona una interfaz bien definida y documentada que especifica cómo se pueden solicitar y utilizar determinadas funciones o servicios de un software o plataforma.

# Creamos la BD y la tabla

---

- CREATE DATABASE angular;
- CREATE TABLE t\_articulos (- id int AUTO\_INCREMENT,
- producto varchar(50),
- precio float,
- PRIMARY KEY (id)
- );

Agrega al menos 2 registros :)

# Creamos el proyecto PHP

— — —

- Creamos una carpeta llamada **productos** dentro de tu carpeta de servidor puede ser www, htdocs, html etc ..
- Creamos los archivos y carpetas:
  - index.php
  - conexion.php

# Archivo index.php

```
<?php
    header('Access-Control-Allow-Origin: *');
    header("Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept");

    include "conexion.php";
    $Conexion = new Conexion();
    $con = $Conexion->conectar();
    $query = "SELECT id, producto, precio FROM t_articulos";
    $result = mysqli_query($con, $query);
    $item = [];
    while ($ver = mysqli_fetch_assoc($result)){
        $item[] = $ver;
    }

    $data = json_encode($item);
    echo $data;
?>
```



# Explicación de header

---

Los encabezados de respuesta HTTP mencionados permiten el intercambio de recursos de origen cruzado (CORS) en una solicitud. 'Access-Control-Allow-Origin: ' indica acceso permitido desde cualquier origen. Puedes reemplazar ' ' con un dominio específico. 'Access-Control-Allow-Headers' especifica los encabezados permitidos en la solicitud, como Origin, X-Requested-With, Content-Type y Accept. Estos encabezados evitan bloqueos de seguridad y permiten solicitudes entre dominios.

# Creación de proyecto angular

— — —

- `ng new obtenerDatos`



# 1 - HttpClientModule

---

- Importamos el módulo HttpClientModule en el archivo app.module.ts:
  - `import {HttpClientModule} from '@angular/common/http';`
- En la sección de imports
  - `imports: [`
  - `BrowserModule,`
  - `HttpClientModule`
  - `],`

## 2 - Creamos una interfaz

— — —

- Creamos una carpeta llamada interfaces
- Creamos el archivo **articulos.interface.ts**
- Le agregamos el código de la imagen

```
export interface articulo {  
  id : number,  
  producto : string,  
  precio : number  
}
```

# 3 - Código de app.component.ts

— — —

```
import { HttpClient } from '@angular/common/http';
import { Component } from '@angular/core';
import { articulo } from '../interfaces/articulos.interface';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {

  public articulos: articulo[] = [];

  constructor(private http: HttpClient) { }

  ngOnInit() {
    let url = "http://localhost/productos/";
    this.http.get<articulo[]>(url)
      .subscribe(
        resultado => {
          this.articulos = resultado;
          console.log(this.articulos);
        }
      );
  }
}
```

## 4 - Muestra en consola

— — —

```
▶ System Error: NETWORK_DISCONNECTED_BY_CLIENT  
▼ (2) [{...}, {...}] ⓘ  
  ▶ 0: {id: '1', producto: 'Coca', precio: '18'}  
  ▶ 1: {id: '2', producto: 'Papas', precio: '15'}  
    length: 2  
  ▶ [[Prototype]]: Array(0)  
>
```

# Actividad

— — —

- Mostrar en una tabla html los datos obtenidos desde la petición httpclient
- Si la tabla no tiene datos que muestre un mensaje de no hay datos
- Subir código fuente a github y agregar a la asignación
- Crear documento que muestre todo lo que hiciste y tus resultados de tu prueba.