

## OBJETO PETICIÓN (PRÁCTICA)

Tiempo estimado: 10min

El objeto de esta práctica es asentar y consolidar los conceptos presentados en la parte teórica de la lección.

Al finalizarla, el estudiante:

- Habrá consultado información de los mensajes de petición **HTTP**.
- Habrá instalado y usado **cURL** para enviar peticiones **HTTP**.

### objetivos

---

En esta práctica, vamos a crear una aplicación **Express** que devuelva una respuesta con la información principal de las peticiones recibidas.

### creación del proyecto

---

Primero, vamos a crear el proyecto de la aplicación.

1. Abrir una consola.
2. Crear el directorio de la práctica.
3. Ir al directorio de la práctica.
4. Crear el archivo **package.json** con el siguiente contenido:

```
{
  "name": "express-app",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "start": "node ./app.js"
  },
  "dependencies": {
    "express": "*"
  }
}
```

5. Crear el archivo **app.js** con el siguiente contenido:

```
"use strict";

//imports
const express = require("express");
const http = require("http");

//app
const app = express();

//config app
app.get("*", function(req, res) {
  res.set("Content-Type", "text/plain; charset=utf-8");

  res.write(`Versión HTTP: ${req.httpVersion}\n`);
  res.write(`Protocolo: ${req.protocol}\n`);
  res.write(`Método: ${req.method}\n`);
  res.write(`Dominio: ${req.get("Host")}\n`);
  res.write(`URL completa: ${req.originalUrl}\n`);
  res.write(`Ruta: ${req.path}\n`);
  res.write(`Cadena de consulta: ${JSON.stringify(req.query)}\n`);
});
```

```

    res.write(`User agent: ${req.get("User-Agent")}\n`);
    res.end();
  });

  //listen
  http.createServer(app).listen(8080, function() {
    console.log("Listening...");
  });

```

Observe cómo se accede a la información de la petición.

#### 6. Instalar dependencias:

```
> npm install
```

#### 7. Iniciar la aplicación Express:

```
> npm start
```

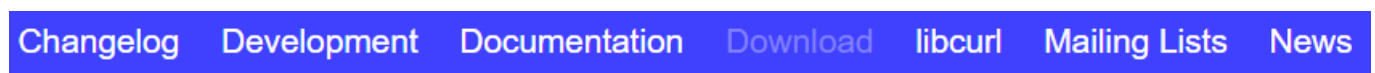
## EJECUCIÓN DE CONSULTAS MEDIANTE CURL

Una vez tenemos la aplicación en ejecución, vamos a enviarle algunos mensajes de petición **HTTP** mediante **cURL**.

### DESCARGA DE CURL PARA WINDOWS

Primero, vamos a descargar **cURL** para su instalación:

1. Abrir el navegador.
2. Ir a la página oficial de **cURL**, [curl.haxx.se](http://curl.haxx.se).
3. Hacer clic en **Download**:



4. Buscar la sección de **Windows** y descargar, por ejemplo, la versión **Win64 x86\_64 7zip**:

Win64 - Generic					
<a href="#">Win64 ia64 zip</a>	7.33.0	binary		Don Luchini	228 KB
<a href="#">Win64 ia64 zip</a>	7.33.0	binary	SSL	Don Luchini	697 KB
<a href="#">Win64 ia64 CAB</a>	7.49.1	binary	SSL	Stefan Kanthak	
<a href="#">Win64 x86_64 7zip</a>	7.49.1	binary	SSL	Darren Owen	
<a href="#">Win64 x86_64 CAB</a>	7.49.1	binary	SSL	Stefan Kanthak	
<a href="#">Win64 x86_64 7zip</a>	7.49.1	binary	SSL SSH	Viktor Szakáts	1.81 MB
<a href="#">Win64 2000/XP x86_64 zip</a>	7.46.0	binary	SSL SSH	Edward LoPinto	
<a href="#">Win64 2000/XP x86_64 MSI</a>	7.46.0	binary	SSL SSH	Edward LoPinto	

### INSTALACIÓN DE CURL EN WINDOWS

Una vez descargado, vamos a instalarlo:

1. Crear una carpeta en la que instalar **cURL**, por ejemplo, **/opt/cURL**.
2. Descomprimir el archivo descargado en la carpeta creada en el paso anterior.
3. Añadir al **PATH** del usuario la carpeta **bin** de **cURL**. En nuestro caso, **C:/opt/cURL/bin**.

4. Abrir una nueva consola.

5. Comprobar que hay acceso a **cURL**:

```
> curl --version
```

En **Windows 10**, existe el comando **Invoke-WebRequest**, el cual tiene asociado el alias **curl** ya que tiene un funcionamiento similar pero no igual. Para hacer uso de nuestra versión instalada, no hay más que suprimir este alias como sigue:

```
> Remove-Item alias:curl
```

## ENVÍO DE PETICIONES HTTP MEDIANTE CURL

Ahora, vamos a usar **cURL** para solicitar recursos y consultar las respuestas recibidas:

1. Ir a la consola.

2. Solicitar las páginas de inicio de **google.es** y **www.google.es**:

```
> curl google.es  
> curl www.google.es
```

Si está usando el alias de **Windows 10**:

```
> (curl http://www.google.es).RawContent
```

3. Echar un vistazo a las respuestas.

4. Solicitar mediante **GET** el recurso **/birds-of-tokyo**:

```
> curl http://localhost:8080/birds-of-tokyo
```

5. Echar un vistazo detenidamente al mensaje de respuesta.

6. Solicitar mediante **GET** el recurso **/travis?origin=Glasgow**:

```
> curl http://localhost:8080/travis?origin=glasgow
```

7. Echar un vistazo a la respuesta.