

Procesamiento GET



Las peticiones que se hacen por **GET** son todas aquellas donde la información viene **directamente** en la **URL** a la que se hace el pedido.



Manejar peticiones **GET**

Comúnmente usamos el método GET para:

- Retornar **vistas**.
- Retornar **archivos**.
- Retornar **datos**.

Cuando definimos una ruta podemos hacerlo directamente sobre la **ejecución de Express**, implementar un **sistema de ruteo** y también **incorporar controladores** que se encarguen de manejarlas.

En todos los casos las rutas recibirán dos parámetros:

- Un string con la ruta que estaremos procesando.
- Un callback donde definiremos qué lógica ejecutaremos cuando el cliente pida esa ruta.

Manejar peticiones GET

{}

```
// Enrutador
router.get("/peliculas", (req, res) => { res.render('peliculas') } );

// Enrutador con controlador
const controller = {
  todas: (req, res) => { res.render('peliculas') }
};
router.get("/peliculas", controller.todas );

// Sobre la ejecución de Express
app.get("/peliculas", (req, res) => { res.render('peliculas') } );
```

Manejar peticiones GET

```
// Enrutador
router.get("/peliculas", (req, res) => { res.render('peliculas') } );

// Enrutador con controlador
const controller = {
  todas: (req, res) => { res.render('peliculas') }
};

router.get("/peliculas", controller.todas );

// Sobre la ejecución de Express
app.get("/peliculas", (req, res) => { res.render('peliculas') } );
```

The diagram illustrates how different Express.js routing methods eventually lead to a callback function. Three code snippets are shown on the left, each with a yellow box highlighting the callback function. Arrows from these boxes point to a yellow box on the right labeled 'CALLBACK'.

- The first snippet shows `router.get("/peliculas", (req, res) => { res.render('peliculas') });` with the callback function highlighted.
- The second snippet shows `router.get("/peliculas", controller.todas);` with `controller.todas` highlighted.
- The third snippet shows `app.get("/peliculas", (req, res) => { res.render('peliculas') });` with the callback function highlighted.

Query string

Es una **cadena** de **texto** —conocida como cadena de consulta— que viaja en la **URL** al momento de hacer una **petición** al servidor mediante GET.

- Comienza al final de la ruta con el signo **?**.
- Está formado por uno o más pares **clave=valor**.
- En el caso de haber más de un par, son separados por el caracter **&**.



Query string

Para **acceder** al *query string* dentro del **callback** que maneja la petición lo haremos a través de la propiedad **query** del objeto **request**.

Esta propiedad es un **objeto literal**, cuyas **claves** y **valores** serán las mismas que viajen en la URL.

Es importante tener en cuenta que todos los datos que lleguen serán de tipo string.

```
{}
```

```
// https://www.youtube.com/watch?video=dQw4w9WgXcQ&time=30  
console.log(req.query.video)  
// 'dQw4w9WgXcQ'
```

DigitalHouse>
Coding School