

El redireccionamiento a otro recurso es una operación muy utilizada. Sobre todo, es muy importante en **SEO**, porque cambiar las cosas de sitio, no suele ser bien aceptada por los buscadores como **Google** o **Bing**. En su lugar, las buenas prácticas **SEO** recomiendan redireccionar los indexadores a las nuevas ubicaciones y, posteriormente, cuando los buscadores ya son conocedores del cambio, suprimir la URI definitivamente. Si no lo hacemos así, podríamos acabar perdiendo toda la indexación en los buscadores de los recursos movidos.

Esta lección es posiblemente de las más cortas del curso. Comienza con una introducción del concepto de redireccionamiento y con la presentación de los códigos de estado **3xx** a través de los cuales llevar a cabo los redireccionamientos. Después, se presenta el campo de cabecera **Location**, ampliamente utilizado en los redireccionamientos. Y finalmente, los métodos del objeto respuesta con los que llevarlos a cabo.

Al finalizar la lección, el estudiante sabrá:

- En qué consiste el redireccionamiento.
- Para qué usar el campo de cabecera **Location**.
- Cómo redactar una respuesta de redireccionamiento mediante el objeto **res** de **Express**.

INTRODUCCIÓN

El **redireccionamiento** (*redirect*) es la operación mediante la cual la aplicación comunica al cliente que se dirija a otro recurso para obtener lo solicitado o para completar o continuar la petición actual. Se consigue mediante los códigos de respuesta **3xx**:

- **300 Multiple Choices**. El servidor proporciona varias URLs a las que el cliente se debe remitir para continuar el procesamiento de la petición actual.

Cuando el servidor desea recomendar una URL de la lista, lo hará mediante el campo de cabecera **Location**.

- **301 Moved Permanently**. Informa al cliente que el recurso se ha movido permanentemente a otra ubicación, la cual se indica en el campo de cabecera **Location**.

Por buenas prácticas, se recomienda adjuntar una página con el enlace a la nueva ubicación en el cuerpo de la respuesta.

- **302 Found**. Es similar al **301 Moved Permanently**, pero ahora el recurso se ha movido sólo temporalmente a otra ubicación. Indicada en el campo de cabecera **Location**.
- **303 See Other**. El servidor remite al cliente a otra ubicación, indicada en el campo de cabecera **Location**.
- **304 Not Modified**. Se utiliza para indicar que el recurso no ha sufrido ninguna modificación desde la fecha proporcionada por el cliente.

El cliente puede remitir una petición, por ejemplo, con un campo de cabecera **If-Modified-Since** en la que indica la fecha de la última modificación del recurso. Si el recurso no ha cambiado, el servidor responde con este código, sin remitir el contenido del recurso. En otro caso, debe hacerlo con un **200 OK** con el contenido del recurso en el cuerpo del mensaje.

- **305 Use Proxy**. Indica que el acceso al recurso se debe hacer a través del *proxy* indicado en el campo de cabecera **Location**.
- **307 Temporary Redirect**. Similar al **302 Found**, pero el cliente debe mantener el mismo método **HTTP** en su nueva petición.

- **308 Permanent Redirect**. Similar al **301 Moved Permanently**, pero el cliente debe mantener el mismo método **HTTP** en su nueva petición.

Es importante observar que algunos códigos de respuesta requieren que se adjunte con el mensaje campos de cabecera específicos como, por ejemplo, **Location** que se adjunta en casi todos los casos.

CAMPO DE CABECERA LOCATION

Cuando la aplicación remite al cliente a otro recurso, generalmente, se recomienda indicar su URL mediante un campo de cabecera **Location**:

Location: URI

Esta URL puede ser interna o bien externa a la aplicación. Veamos unos ejemplos ilustrativos:

```
Location: /index.html
Location: http://www.google.es
```

OBJETO DE RESPUESTA

Express proporciona varios métodos para ayudar a redactar respuestas de redireccionamiento. Todos ellos, ubicados en el objeto respuesta.

Método **res.location()**

Mediante el método **location()**, podemos fijar el valor del campo de cabecera **Location**:

location(path)

| Parámetro | Tipo de datos | Descripción |
|-------------|---------------|---|
| path | string | Ruta local o URI externa al recurso redirigido. |

También es posible fijar **Location** mediante el método **set()**. Pero por convenio y buenas prácticas, se recomienda **location()**. Ejemplos:

```
//mediate location()
res.location("/index.html");

//mediate set()
res.set("Location", "/index.html");
```

Método **res.redirect()**

El redireccionamiento lo podemos llevar a cabo mediante el método **redirect()** de la respuesta:

redirect(status, path)
redirect(path)

| Parámetro | Tipos de datos | Descripción |
|---------------|----------------|--|
| status | number | Código de respuesta. Valor predeterminado: 302 . |
| path | string | Ruta o URI del recurso redirigido. |

Veamos algunos ejemplos ilustrativos:

```
res.redirect("/index.html");
res.redirect(302, "/index.html");
res.redirect("http://otro.dominio.com/index.html");
```

Cuando se indica como **path** el texto **back**, **Express** redirige al recurso indicado por el campo de cabecera **Referer** de la petición **HTTP** en curso. Ejemplo:

```
res.redirect("back"); //similar a: res.redirect(req.get("Referer"));
```