

En el contexto de un servidor utilizando Express.js en Node.js, los objetos `req` (request) y `res` (response) son fundamentales para manejar las solicitudes y respuestas HTTP. Aquí te presento algunos de los métodos y propiedades más comunes de estos objetos, junto con una explicación de lo que resulta de cada operación.

Objeto `req` (request)

1. **`req.params`:**
 - **Descripción:** Contiene los parámetros de la ruta definidos en la URL.
 - **Ejemplo:** Para la ruta `/movies/:id`, si se accede a `/movies/123`, `req.params.id` será `'123'`.
2. **`req.query`:**
 - **Descripción:** Contiene los parámetros de consulta de la URL, después del signo de interrogación (?).
 - **Ejemplo:** Para la URL `/movies?genre=comedy`, `req.query.genre` será `'comedy'`.
3. **`req.body`:**
 - **Descripción:** Contiene los datos enviados en el cuerpo de la solicitud. Utilizado principalmente en solicitudes POST y PUT.
 - **Ejemplo:** Si se envía `{ "name": "Inception" }` en una solicitud POST, `req.body.name` será `'Inception'`.
 - **Nota:** Para que `req.body` funcione, **es necesario utilizar middleware** como `express.json()` o `express.urlencoded()`.
4. **`req.headers`:**
 - **Descripción:** Contiene los encabezados HTTP enviados con la solicitud.
 - **Ejemplo:** `req.headers['content-type']` puede devolver `'application/json'`.
5. **`req.method`:**
 - **Descripción:** Indica el método HTTP utilizado en la solicitud (GET, POST, PUT, DELETE, etc.).
 - **Ejemplo:** `req.method` puede devolver `'GET'` o el método recibido.
6. **`req.url`:**
 - **Descripción:** Devuelve la URL de la solicitud.
 - **Ejemplo:** `req.url` puede devolver `'/movies?genre=comedy'`.

Objeto **res** (response)

1. **res.send():**
 - **Descripción:** Envía una respuesta al cliente. Puede enviar texto, HTML, JSON, etc.
 - **Ejemplo:** `res.send('Hello World')` enviará 'Hello World' al cliente.
2. **res.json():**
 - **Descripción:** Envía una respuesta JSON al cliente.
 - **Ejemplo:** `res.json({ message: 'Hello World' })` enviará un objeto JSON `{ "message": "Hello World" }`.
3. **res.status():**
 - **Descripción:** Establece el código de estado HTTP para la respuesta.
 - **Ejemplo:** `res.status(404).send('Not Found')` enviará una respuesta con el código de estado 404 y el mensaje 'Not Found'.
4. **res.redirect():**
 - **Descripción:** Redirige al cliente a otra URL.
 - **Ejemplo:** `res.redirect('/login')` redirigirá al cliente a la ruta `/login`.
5. **res.set():**
 - **Descripción:** Establece un encabezado HTTP en la respuesta.
 - **Ejemplo:** `res.set('Content-Type', 'application/json')` establecerá el tipo de contenido de la respuesta como JSON.
6. **res.end():**
 - **Descripción:** Finaliza la respuesta sin ningún dato.
 - **Ejemplo:** `res.end()` finaliza la respuesta y el cliente no recibirá ningún cuerpo de respuesta.