



Fernando Saez

Capturar datos del pathname

<http://www.miserver.com/user/358>

Pathname: /user/358

```
const router = function (pathname, req, res) {
```

```
  //Solo sirve para un para un parámetro al final de la url
```

```
  let path = pathname.split("/");
```

```
  const param1 = path.pop();
```

```
  const pathSinParam1 = path.join("/");
```

```
  if(pathSinParam === '/user'){  
    handleRequest(param1)  
  }
```

Capturar query de la URL

Parámetros que pueden llegar en el pathname

<http://www.ventas.com/pedido?articulo=serrucho&cantidad=3>

```
const http=require('http');  
const {URL}=require('url');  
const servidor=http.createServer((req,res) => {
```

```
const baseUrl = "http://" + req.headers.host;  
const objUrl = new URL(req.url, baseUrl);  
const params = objUrl.searchParams;
```

```
console.log(params.get("articulo"));  
console.log(params.get("cantidad"));
```

```
....
```

Interacción con Formularios

First Name	Last Name	
Email		
Password	Confirm Password	
Address		
(Suite, Apt,...)		
Zip Code	City	State

Estructura

Campos (Input)

Rótulos

Botones

Feedback

Asistencia

Validación

Leer desde formulario 1 - POST

```
<form action="recuperardatos" method="post">
```

```
....
```

```
</form>
```

Servimos el formulario
que se encuentra en una
página estática HTML

```
const http=require('http');  
const url=require('url');  
const fs=require('fs');  
const querystring = require('querystring');  
const servidor=http.createServer((req ,res) => {  
    const baseUrl = "http://" + req.headers.host;  
    const objUrl = new URL(req.url, baseUrl);  
    let camino='public'+ objUrl.pathname;  
    if (camino=='public/') camino='public/index.html';  
    encaminar(req,res,camino);  
});  
servidor.listen(8888);
```

Leer desde formulario 2 - POST

```
function encaminar (req,res,camino) {  
  switch (camino) {  
    case 'public/recuperardatos': {  
      recuperar(req,res);  
      break;  
    }  
    default : {  
      fs.stat(camino, error => {  
        if (!error) {  
          fs.readFile(camino,(error, contenido) => {  
            //atiende a index...  
          });  
        } else { /*recurso inexistente*/ }  
      });  
    }  
  }  
}
```

2 posibilidades para camino:

[public/index.html](#)

[public/recuperardatos](#)

Leer desde formulario 3 - POST

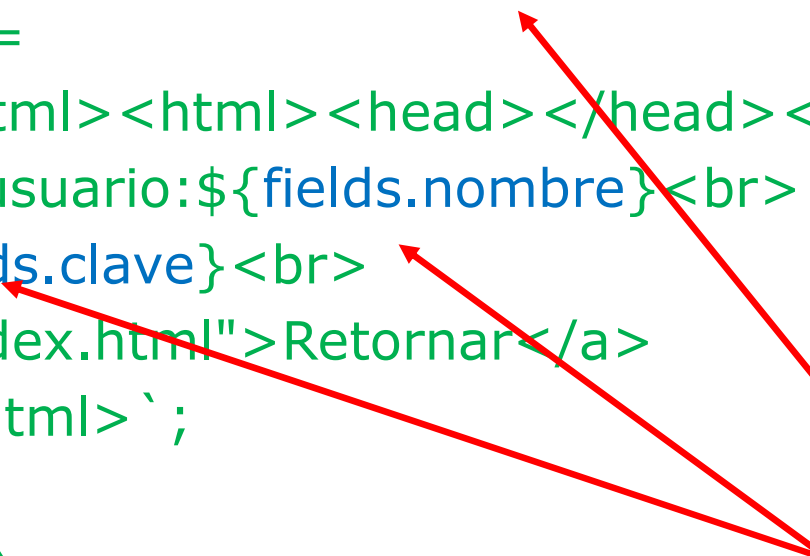
```
function recuperar(req,res) {  
  let info = "";  
  req.on('data', datosparciales => {  
    info += datosparciales;  
  });  
  req.on('end', () => {  
    const formulario = querystring.parse(info);  
    res.writeHead(200, {'Content-Type': 'text/html'});  
    const pagina =  
    `<!doctype html><html><head></head><body>  
      Nombre de usuario:${formulario['nombre']<br>  
      Clave:${formulario['clave']}<br>  
      <a href="index.html">Retornar</a>  
    </body></html>`;   
    res.end(pagina);  
  }); }  
}
```

nombre=juan&clave=123456

querystring.parse

{
 nombre:'juan',
 clave:'123456'
}

Post Data con formidable

```
const formidable = require("formidable");  
function recuperar(req,res) {  
  var form = new formidable.IncomingForm();  
  form.parse(req, function (err, fields, files) {  
    const pagina=  
    `<!doctype html><html><head></head><body>  
    Nombre de usuario:${fields.nombre}<br>  
    Clave:${fields.clave}<br>  
    <a href="index.html">Retornar</a>  
    </body></html>`;   
  })  
  res.end(pagina);  
}
```

Cuando los
datos del
formulario estén
disponibles

Atención!! formidable no soporta Promesas

formidable y sync await

```
const formidable = require("formidable");
```

```
sync function recuperar(req,res) {  
  var form = new formidable.IncomingForm();
```

```
  const formfields = await new Promise( function (resolve, reject) {  
    form.parse(req, (err, fields, files) => {  
      if (err) reject(err);  
      else resolve(fields);  
    })  
  })  
  console.log("campo nombre es: " + formfields.nombre);  
  console.log("campo clave es: " + formfields.clave);  
}
```

Upload de archivos 1

➤ npm install formidable

```
<form method="post" action="/subir"  
enctype="multipart/form-data">
```

```
Nombre: <input type="text" name="nombre">
```

```
Apellido: <input type="text" name="apellido">
```

```
Seleccione la foto:
```

```
<input type="file" name="foto1">
```

```
<br>
```

```
<input type="submit" value="confirmar">
```

```
</form>
```

Instancia de IncomingForm

```
const form = formidable.IncomingForm();  
console.log(form);
```

```
const uploadFolder = path.join(__dirname, "public", "files");
```

```
// Basic Configuration  
form.multiples = true;  
form.maxFileSize = 50 * 1024 * 1024; // 5MB  
form.uploadDir = uploadFolder;  
console.log(form);
```

```
IncomingForm {  
  _events: [Object: null prototype] {},  
  _eventsCount: 0,  
  _maxListeners: undefined,  
  error: null,  
  ended: false,  
  maxFields: 1000,  
  maxFieldsSize: 20971520,  
  maxFileSize: 209715200,  
  keepExtensions: false,  
  uploadDir: 'C:\\Users\\SARTHA~1\\AppData\\Local\\Temp',  
  encoding: 'utf-8',  
  headers: null,  
  type: null,  
  hash: false,  
  multiples: false,  
  bytesReceived: null,  
  bytesExpected: null,  
  _parser: null,  
  _flushing: 0,  
  _fieldsSize: 0,  
  _fileSize: 0,  
  openedFiles: [],  
  [Symbol(kCapture)]: false  
}
```

Upload de archivos 2

```
function subir(req,res){
  const form=new formidable.IncomingForm();
  form.uploadDir='upload';
  form.keepExtensions: true,
  let dataform = await new Promise(function (resolve, reject) {
    form.parse(req, function (error, fields, files) {
      if (error) reject(error);
      else {
        //1.- renombro el archivo como el original
        //fs.renameSync(files.cv.path, path.join("tmp", files.cv.name));
        //2.- renombro el archivo al nombre + apellido
        const nombreCompleto = fields.nombre + "_" + fields.apellido + ".jpg";
        files.cv.name = nombreCompleto;
        fs.renameSync(files.cv.path, path.join("tmp", nombreCompleto));
        resolve([fields, files]);
      }
    });
  });
  res.writeHead(200, { "content-type": "text/html" });
  res.write("<p> Nombre: " + dataform[0].nombre + "</p>");
  res.write("<p> Apellido: " + dataform[0].apellido + "</p>");
  res.write("<p> archivo: " + dataform[1].foto1.name + "</p>");
  res.end();
}
```

Listar archivos subidos

```
function listar(respuesta) {  
  fs.readdir('./public/upload/',(error, archivos) => {  
    let fotos="";  
    for(let x=0;x<archivos.length;x++) {  
      fotos += `<br>`;   
    }  
    respuesta.writeHead(200,  {'Content-Type':  'text/html'  
});  
    respuesta.write(`<!doctype  
html><html><head></head>  
      <body>${fotos}<a href="index.html">  
      Retornar</a></body></html>` );  
    respuesta.end();  
  });  
}
```