

Repaso ExpressJS

```
npm i express cors
```

y con servidor de plantillas:

```
npm i express cors ejs
```

Tipos de servidores

- **API:** Con `server.get()` (o post o put...) y devuelven JSON con `res.json(data)`.
- **Estáticos:** Configurando Express con `server.use(express.static(carpeta))`;
- **Dinámicos:** Con `server.get()` (solo get) y `res.render()`.

Endpoint

Es la parte de la dirección que va detrás del número del puerto.

- `http://localhost:3000 [/]`
- `http://localhost:3000 [/adalabers]`
- `http://localhost:3000 [/api/kitten]`

NOTA: Sin los [corchetes], claro

Verbos (method)

- **GET:** páginas, imágenes, css, js, API listado datos
- **POST:** enviar datos (servidor crea y se guarda un dato nuevo) (datos se envían por body)
- **PUT:** enviar datos (servidor modifica sus datos) (datos se envían por body)
- **DELETE:** servidor borre uno de sus datos (se le indica cual poniendo el id en URL params)

Cada ruta o endpoint con un método, se atiende con una función en Express:

```
server.get('/', (req, res) => {});

server.get('/adalabers', (req, res) => {
  // Recuperamos una lista
  res.json(adalabersList);
});

server.post('/adalabers', (req, res) => {
  // Guardamos lo que nos llegue por req.body
  if( ok ) {
    res.json(newIdJson);
  }
});
```

```

    else {
      res.json(errorJson);
    }
  });

server.put('/adalabers', (req, res) => {
  // Actualizamos el objeto que nos indiquen con lo que nos llegue por
  req.body
  if( ok ) {
    res.json(okJson);
  }
  else {
    res.json(errorJson, 409);
  }
});

server.delete('/adalabers', (req, res) => {
  // Borramos el objeto que nos indiquen
  if( ok ) {
    res.json(okJson);
  }
  else {
    res.json(errorJson, 409);
  }
});

```

Params

Tipos	pet	req	Tipo servidor	Verbos	Comentarios
Query	Dirección detrás ?	req.query	Serv. dinámicos. API	Todos	?nombre=valor
Body	body	req.body	API	POST y PUT	* Ver más abajo
URL	Dirección parte de la ruta	req.params	Serv. dinámicos. API	Todos	Como React Router /path/:nombre
Header	headers	req.headers req.header()	Serv. dinámicos. API	Todas	Siempre llegan en minúsculas

Uso de cada tipo de parámetros

- **Query:** API (paginación, search, fields) Dinámicos (personalizar el contenido)
- **Body:** API (Enviar datos)

- **URL:** API Dinámicos (Identificar qué quiero mostrar)
 - **Header:** API, Dinámicos (Autenticación), API (tipo de contenido (xml, json, imagen))
-

Cómo usar query params

En la dirección: `http://localhost:3000/adalabers?nombre=valor&nombre2=valor2` hay dos parámetros:

- Uno con nombre `nombre` y valor `'valor'`
- Otro con nombre `nombre2` y valor `'valor2'`

Se puede usar en HTML

```
<a href="http://localhost:3000/adalabers?
nombre=valor&nombre2=valor2">Enlace</a>
```

o en Javascript con fetch:

```
fetch('http://localhost:3000/adalabers?nombre=valor&nombre2=valor2')
  .then()
  .then()
```

y en el **servidor** se recuperan con:

```
server.get('/adalabers', (req, res) => {
  req.query.nombre2
})
```

Cómo usar body params

No va en la dirección.

Solo se puede usar en Javascript con fetch (usando el objeto de config):

```
fetch(
  'http://localhost:3000/adalabers',
  {
    body: JSON.stringify(data)
  }
)
  .then()
  .then()
```

y en el **servidor** se recuperan con:

```
server.get('/adalabers', (req, res) => {  
  req.body = data;  
});
```

IMP! Config Express para usar peticiones con body params

Poner en el servidor

```
server.use(express.json( {limit: '10mb'} ));
```

Poner un header param especial en el fetch:

```
'Content-Type': 'application/json'
```

Cómo usar URL param

En la dirección: `http://localhost:3000/adalabers/:nombre` hay una parte variable que le damos el nombre `nombre`

Se puede usar en HTML

```
<a href="http://localhost:3000/adalabers/alicia">Enlace</a>
```

y en este caso, la variable `nombre` toma el valor `'alicia'`

o en Javascript con fetch:

```
fetch('http://localhost:3000/adalabers/celia')  
  .then()  
  .then()
```

y en el **servidor** se recuperan con:

```
server.get('/adalabers/:nombre', (req, res) => {  
  req.params.nombre  
});
```

Cómo usar param header

No va en la dirección.

Solo se puede usar en Javascript con fetch (usando el objeto de config):

```
fetch(  
  'http://localhost:3000/adalabers/',  
  {  
    headers: {  
      nombre: valor,  
      auth: 'KATAKROKER',  
      usuario: 'MariCarmen',  
      pass: 'KJADS9JKSsa'  
    }  
  })  
.then()  
.then()
```

y en el **servidor** se recuperan con:

```
server.get('/adalabers/', (req, res) => {  
  req.headers.nombre  
  req.header('auth')  
});
```

IMP! Peticiones con header params

Los valores siempre llegan en minúsculas al servidor.