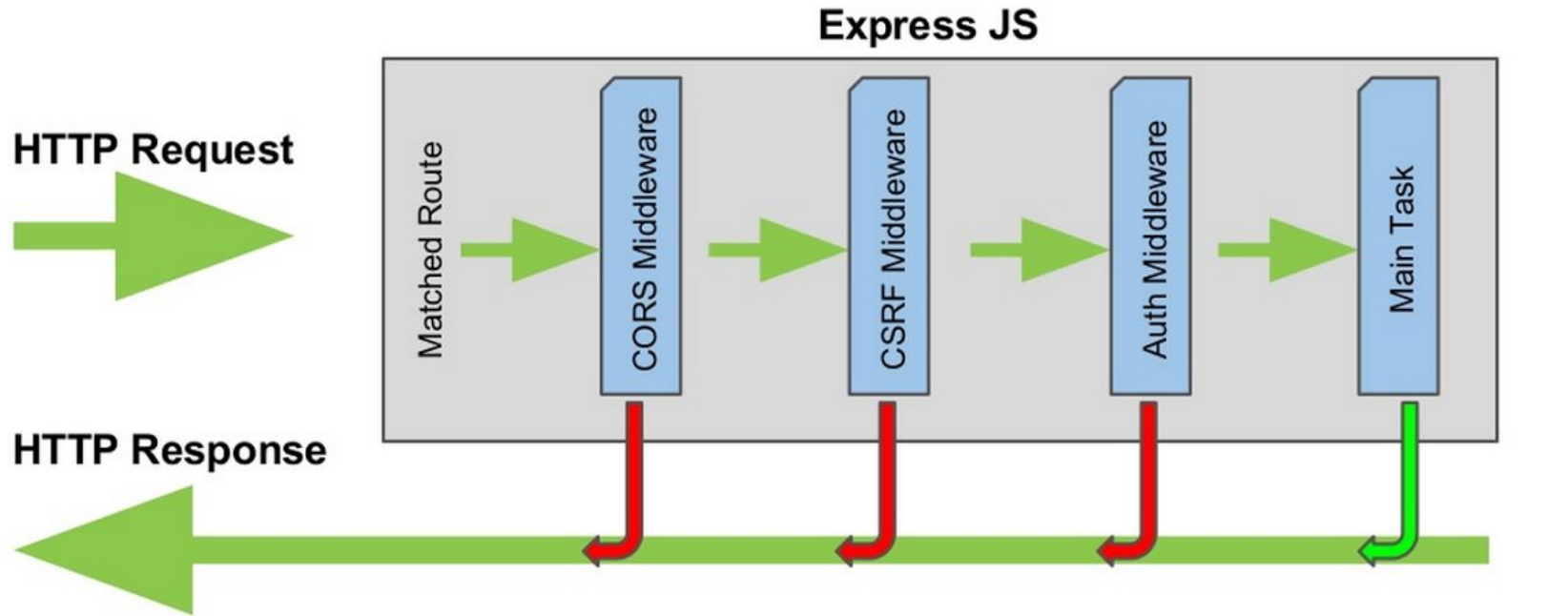


# Routing en Express

...

Sebastián Oliva  
Ignacio Ferreira

# Http Request - Response en aplicaciones Express



# Definición básica de una ruta

```
app.METHOD(PATH, HANDLER)
```

# Request Methods

- GET
- HEAD
- POST
- PUT
- DELETE
- OPTIONS
- PATCH

# Aplicación básica de Express

```
var express = require('express')
var app = express();

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(3000);
```

# Métodos de respuesta

Método	Descripción
<code>res.download()</code>	Solicita un archivo para descargarlo.
<code>res.end()</code>	Finaliza el proceso de respuesta.
<code>res.json()</code>	Envía una respuesta JSON.
<code>res.jsonp()</code>	Envía una respuesta JSON con soporte JSONP.
<code>res.redirect()</code>	Redirecciona una solicitud.
<code>res.render()</code>	Representa una plantilla de vista.
<code>res.send()</code>	Envía una respuesta de varios tipos.
<code>res.sendFile()</code>	Envía un archivo como una secuencia de octetos.
<code>res.sendStatus()</code>	Establece el código de estado de la respuesta y envía su representación de serie como el cuerpo de respuesta.

# All method

```
app.all('/all', (req, res, next) => {  
  res.send('This is all route')  
})
```

```
app.all('*', requireAuthentication, loadUser);
```

# Tipos de rutas

- `'/'`
- `'/about'`
- `'/api/fishes.json'`
- `'/users/:user_id'`



# Tipos de rutas

- ?
- +
- \*
- ()

```
app.get('/gatitos?', (req, res) => {  
  res.send('Muchos gatitos!')  
})
```

```
app.get('/grito+', (req, res) => {  
  res.send('Esto es un gritooooo!')  
})
```

```
app.get('/hola*chao', (req, res) => {  
  res.send('hola amigo')  
})
```

```
app.get('/perrit(os)?(as)?', (req, res) => {  
  res.send('Es un mamifero')  
})
```

# Expresiones regulares

```
app.get(/mon/, (req, res) => {  
  res.send('Probablemente es un digimon')  
})
```

# Manejo de rutas: funciones anidadas

```
// doble función anidada
app.get('/doblefucion', (req, res, next) => {
  console.log('Verificando usuario...')
  next()
}, (req, res) => {
  res.send('Información requerida por el usuario')
})
```

# Manejo de rutas: arreglo de funciones

```
// arreglo de funciones anidadas
var login = (req, res, next) => {
  console.log('Verificando usuario...')
  next()
}

var validation = (req, res, next) => {
  console.log('Validando información...')
  next()
}

var send = (req, res, next) => {
  res.send('Información requerida por el usuario')
}

app.get('/arreglofuncion', [login, validation, send])
```

# Manejo de rutas: mixtas

```
// Mixta
```

```
app.get('/mixfuncion', [login, validation], (req, res) => {  
  res.send('Información requerida por el usuario')  
})
```

# Router

- Mini-aplicación
- Solo se encarga de los middlewares y rutas

```
app.route('/bears')  
  .get((req, res) => {  
    res.send('Toma todos los osos.')  
  })  
  .post((req, res) => {  
    res.send('Guardando un oso.')  
  })
```

# Router: como archivo externo

```
// router.js
var express = require('express');
var router = express.Router();

// Se le entrega una middleware específico para este router
router.use(function timeLog(req, res, next) {
  console.log('Algo está pasando...');
  next();
});

// Se define el home de esta ruta
router.get('/', function(req, res) {
  res.send('Perros Welcome page');
});

// se le agrega una página específica a esta ruta
router.get('/about', function(req, res) {
  res.send('Acerca de los perros');
});

module.exports = router;
```

```
//index.js
// Se importa el router
var dogs = require('./router.js')
// Se le entrega como handler de la ruta específica
app.use('/dogs', dogs)
```

# Params

```
// router.js
var express = require('express');
var router = express.Router();

// Se toma el parámetro que ingresa
router.param('user_id', function(req, res, next, id) {
  console.log('buscando usuario de parámetro:', id);
  console.log('buscando usuario de parámetro:', req.params.user_id)
  req.user = {
    name: 'Usuario1',
    id: id
  }
  next();
});

router.get('/:user_id', function (req, res) {
  res.send('Nombre: ' + req.user.name);
});

module.exports = router;
```

```
// index.js
// Se importa el router
var users = require('./userRouter.js')
// Se entrega como handler de la ruta
app.use('/users', users)
```



# Headers

```
// Header
app.get('/header1', (req, res) => {
  res.append('Content-Type', 'text/plain');
  res.send('enviado con headers');
})

app.get('/header2', (req, res) => {
  res.set({
    'Content-Type': 'application/json'
  });
  res.json('enviado con headers')
})
```

Nota: no es buena idea combinar los métodos de seteo de headers

# Referencias

- <http://expressjs.com/en/api.html#router>
- <http://expressjs.com/es/guide/routing.html>
- <http://www.restapitutorial.com/lessons/httpmethods.html>

# Routing en Express

...

Sebastián Oliva  
Ignacio Ferreira