

RUTAS EN EXPRESS

“

A través del **sistema** de **ruteo** de Express podemos definir, de manera sencilla, cómo va a responder nuestra aplicación según el **método** HTTP y la **ruta** que esté llegando al **servidor**.



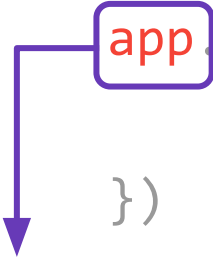
DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:

```
app.get('/', function (req , res) {  
    res.send('¡Hola mundo!');  
})
```

DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:



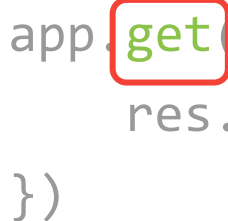
```
app.get('/', function (req , res) {  
  res.send('¡Hola mundo!');  
})
```

app

Variable que guarda la **ejecución** de **Express**.

DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:



```
app.get('/', function (req , res) {  
  res.send('¡Hola mundo!');  
})
```

Método

Escribimos el **método HTTP** que queremos atender: *get*, *post*, *put*, *patch* ó *delete*.

DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:



```
app.get('/', function (req , res) {  
    res.send('¡Hola mundo!');  
})
```

Path

String que hará referencia a la **ruta** en sí (url que llegará por petición).

DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:



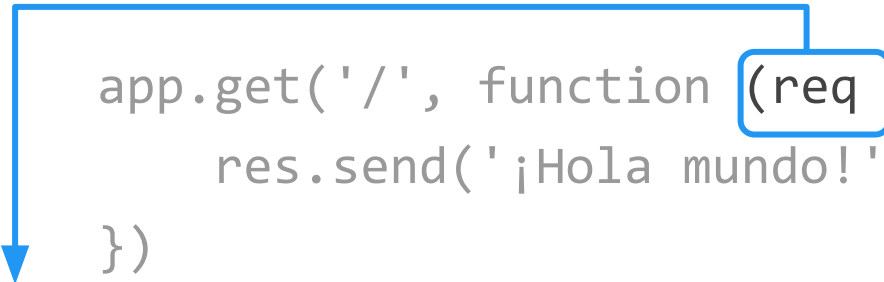
```
app.get('/', function (req , res) {  
    res.send('¡Hola mundo!');  
})
```

Handler

Callback que se encargará de definir qué acción tomar cuando se acceda a la ruta definida.

DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:



```
app.get('/', function (req, res) {  
    res.send('¡Hola mundo!');  
})
```

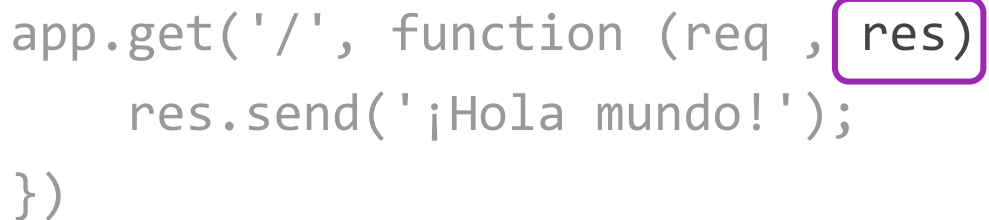
The diagram consists of a blue line that starts from the right side of the `(req, res)` parameter list in the code block above, extends horizontally to the right, then turns 90 degrees downward, and finally turns 90 degrees to the left to point at the **Request** definition below.

Request *(primer parámetro del handler)*

Es un objeto literal con múltiples métodos y propiedades. Representa al **request** solicitado.

DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:



```
app.get('/', function (req, res) {  
    res.send('¡Hola mundo!');  
})
```

Response *(segundo parámetro del handler)*

Es un objeto literal con múltiples métodos y propiedades. Representa al **response** que dará el servidor.

DEFINIENDO UNA RUTA

A través de Express, contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:

```
app.get('/', function (req , res) {  
  res.send('¡Hola mundo!');  
})
```

A red line originates from the `res.send()` call in the code block above. It extends horizontally to the left, then turns 90 degrees downward, ending in a red arrowhead pointing towards the 'Lógica de la ruta' label.

Lógica de la ruta

Definimos la **lógica** que va a manejar la ruta definida. Se estila dar la respuesta que verá el cliente en su navegador.