

Routes

Routing refers to determining how the application responds to a client request to a particular endpoint, which is a URI and a particular HTTP request method.

Route definition has the following structure:

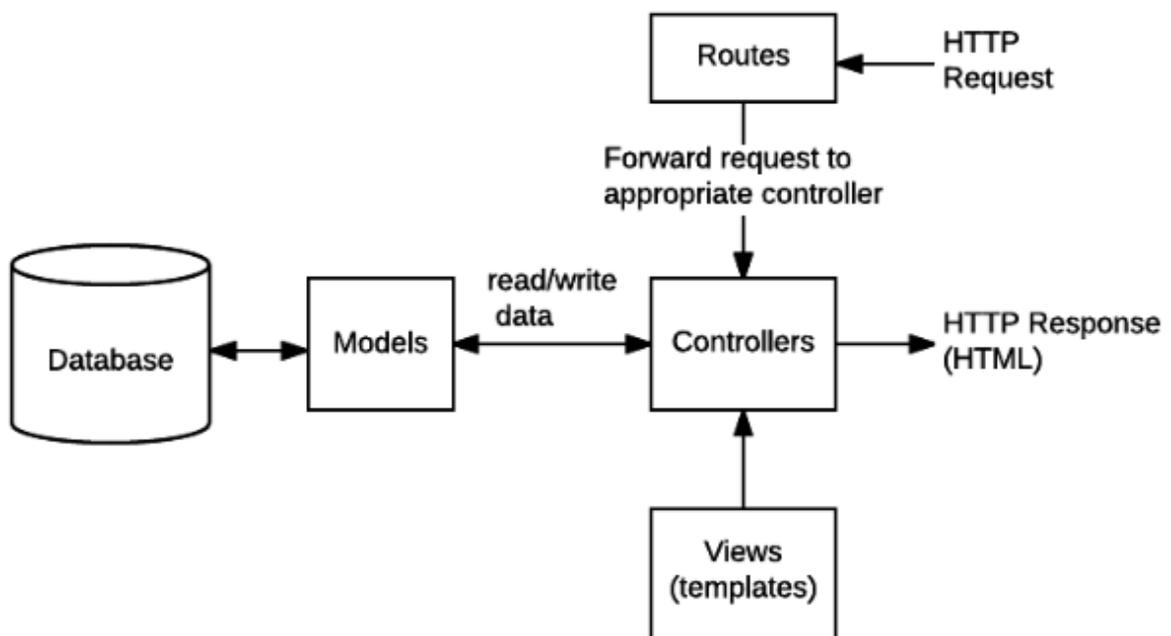
```
App.METHOD(PATH, HANDLER)
```

Where:

- app is the instance of express.
- METHOD is an HTTP request method, in lowercase.
- PATH is the path on the server.
- HANDLER is the function executed when the route is matched.

The diagram determines the flow of data:

- "Routes" to forward the supported requests to the appropriate controller functions.
- Controller functions to get the requested data from the models, create an HTML page displaying the data, and return it to the user to view in the browser.
- Views (templates) used by the controllers to render the data.



In this example, we are going to use Router middleware.

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

The following examples illustrate defining simple routes.

```
app.get('/', function (req, res) {  
  res.send('Hello World!')  
})
```

Respond to POST request on the root route (/), the application's home page:

```
app.post('/', function (req, res) {  
  res.send('Got a POST request')  
})
```

Respond to a PUT request to the /user route:

```
app.put('/user', function (req, res) {  
  res.send('Got a PUT request at /user')  
})
```

Respond to a DELETE request to the /user route:

```
app.delete('/user', function (req, res) {  
  res.send('Got a DELETE request at /user')  
})
```

- The callback function here calls send() on the response to return the string
- res.json() to send a JSON response
- res.sendFile() to send a file

Route parameters:

Route parameters used to capture the values specified at their position in the URL. The named segments are prefixed with a colon and then the name (e.g. `/:your_parameter_name/`).

The captured values are stored in the `req.params` object using the parameter names as keys (e.g. `req.params.your_parameter_name`).

```
app.get('/users/:userId/books/:bookId', function (req, res) {  
  // Access userId via: req.params.userId  
  // Access bookId via: req.params.bookId  
  res.send(req.params);  
})
```