</talentlabs>

Express Lecture 5

Basic Express Routing

</talentlabs>

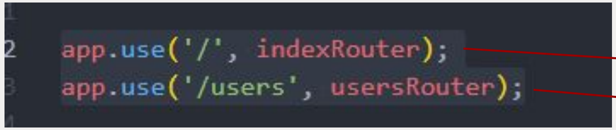
Agenda

- More on Router and Route
- More on HTTP Get request
- Use Url Parameter for rendering

More on Router and Route

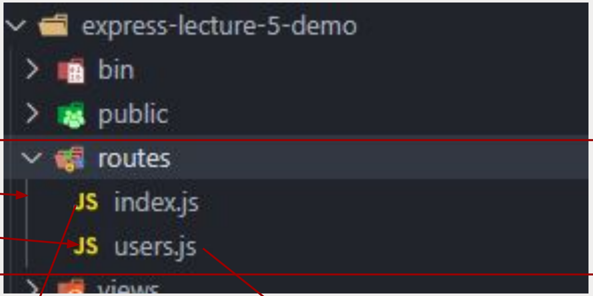
Router

1. Routers are **used** by the our Express Application in the **app.js** file.

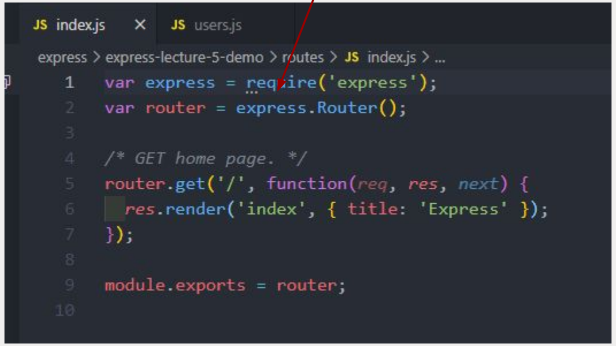


```
app.use('/', indexRouter);
app.use('/users', usersRouter);
```

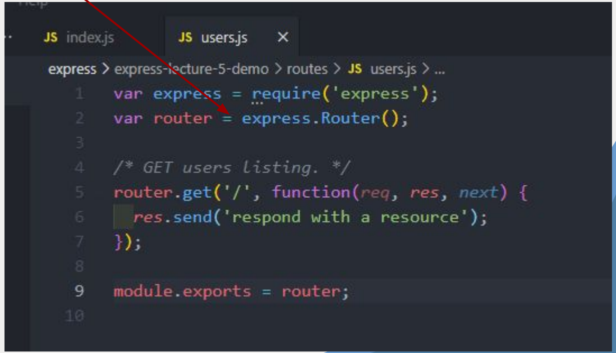
2. Each file in the **"/routes"** folder actually contains an **Express Router**. We are **building routes to a router.**



2 Routers



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

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

module.exports = router;
```

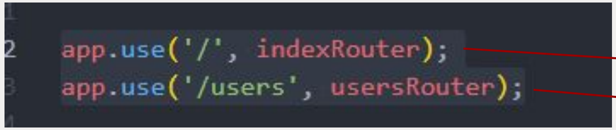


```
var express = require('express');
var router = express.Router();

/* GET users listing. */
router.get('/', function(req, res, next) {
  res.send('respond with a resource');
});

module.exports = router;
```

</talentlabs>

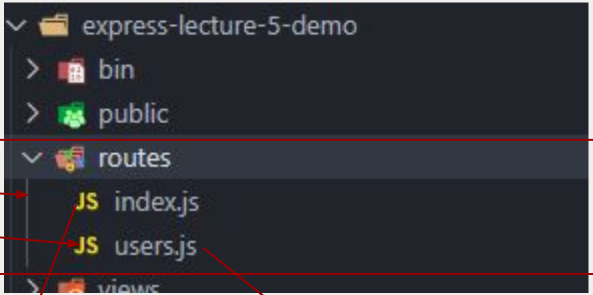
Route

1. Routers are **used** by the our Express Application in the **app.js** file.



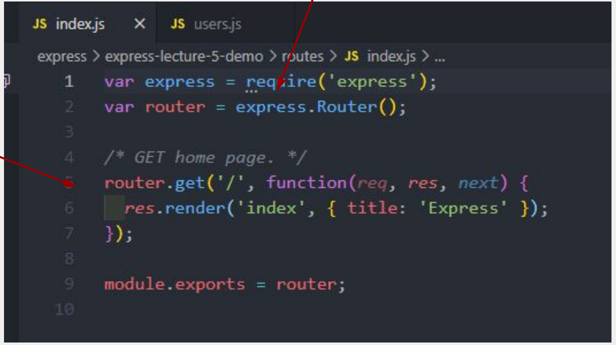
```
app.use('/', indexRouter);
app.use('/users', usersRouter);
```

2. Each file in the **“/routes”** folder actually contains an **Express Router**. We are **building routes to a router.**



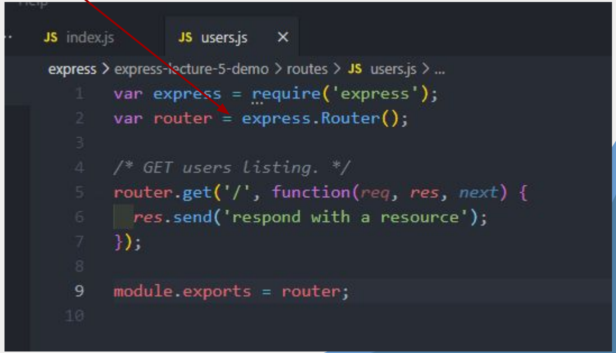
2 Routers

3. We can defines **Routes** under a **Router**.



express > express-lecture-5-demo > routes > JS index.js > ...

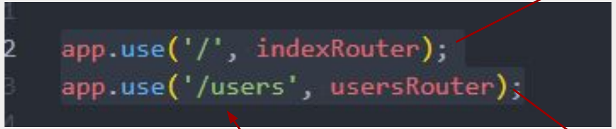
```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

module.exports = router;
```



express > express-lecture-5-demo > routes > JS users.js > ...

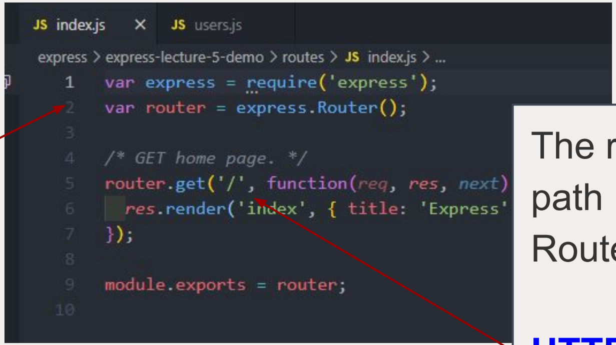
```
var express = require('express');
var router = express.Router();

/* GET users listing. */
router.get('/', function(req, res, next) {
  res.send('respond with a resource');
});

module.exports = router;
```

Router + Route Path



```
app.use('/', indexRouter);
app.use('/users', usersRouter);
```

When we **add a Router to the Express Application in app.js**, we can **specify a path for the Router**

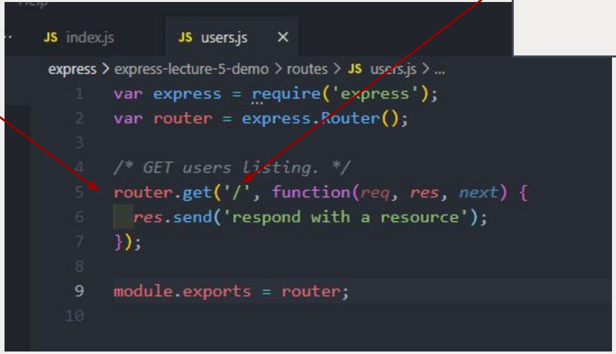


```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next)
  res.render('index', { title: 'Express'
});

module.exports = router;
```

The resultant matching path is = Router path + Route path.

HTTP GET /

HTTP GET /users/



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

/* GET users listing. */
router.get('/', function(req, res, next) {
  res.send('respond with a resource');
});

module.exports = router;
```

More on HTTP Get request

HTTP GET request parameters

- A HTTP GET request usually carries data in
  - Headers (not discuss in this chapter)
  - **Path**, such as
    - https://www.youtube.com/watch**?v=iYM2zFP3Zn0**
    - https://www.facebook.com/**node.express**

Why these are parameters?

If we update these values, we will land on a different page or video!

</talentlabs>

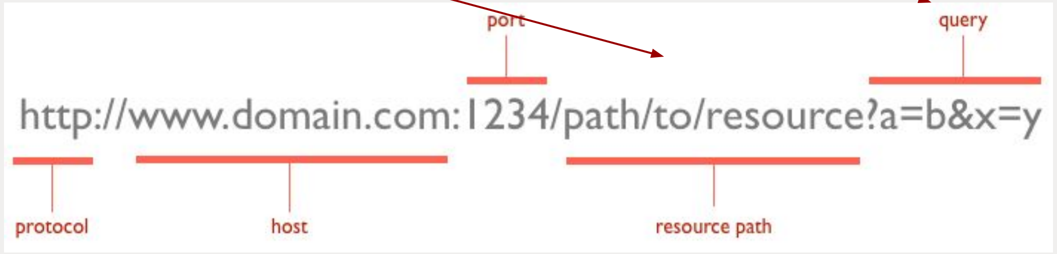
HTTP GET request parameters

- A HTTP GET request usually carries data in
  - Headers (not discuss in this chapter)
  - **Path**, such as
    - https://www.youtube.com/watch**?v=iYM2zFP3Zn0**
    - https://www.facebook.com/**node.express**

Again we don't need to memorize all the parts.

When we pass a value **via the resource path section**, it is called an **Route parameter**.

When we pass a value **via the query section**, it is called an **query string**.



</talentlabs>

Route Params

With Express, we can define a part of the path as the Route Param like this with **a : prefix**

Then we can get it with **req.params**

```
router.get('/demo/:a', function(req, res, next) {
  console.log("URL Params", req.params)
  res.render('index', { title: 'Express' });
});
```

```
http://localhost:3000/demo/hello/
> URL Params { a: 'hello' }

http://localhost:3000/demo/123/
> URL Params { a: '123' }
```

These values are used as the Route Param.

</talentlabs>

Route Params (name of the param)

With Express, we can define a part of the path as the Route Param like this with **a : prefix**

Then we can get it with **req.params**

```
router.get('/demo/:my_param, function(req, res, next) {
  console.log("URL Params", req.params)
  res.render('index', { title: 'Express' });
});
```

```
http://localhost:3000/demo/hello/
> URL Params { my_param: 'hello' }

http://localhost:3000/demo/123/
> URL Params { my_param: '123' }
```

These values are used as the Route Param.

Route Params (more than 1 param)

With Express, we can define a part of the path as the Route Param like this with **a : prefix**

Then we can get it with **req.params**

```
router.get("/demo/:a/test/:b", function (req, res, next) {
  console.log("URL Params", req.params);
  res.render("index", { title: "Express" });
});
```

```
http://localhost:3000/demo/hello/test/abc
> URL Params { a: 'hello', b: 'abc' }

http://localhost:3000/demo/123/test/456
> URL Params { a: '123', b: '456' }
```

These values are used as the Route Param.

</talentlabs>

Query

Queries are free-form, we don't need to pre-define them in the path.

Then we can get it with **req.query**

```
router.get("/demo2", function (req, res, next) {
  console.log("URL Params", req.params);
  console.log("Queries", req.query);
  res.render("index", { title: "Express" });
});
```

```
http://localhost:3000/demo2?a=1&b=2&video=10

> URL Params {}
> Queries { a: '1', b: '2', video: '10' }
```

Query + Route Param

```
router.get("/demo3/:a", function (req, res, next) {
  console.log("URL Params", req.params);
  console.log("Queries", req.query);
  res.render("index", { title: "Express" });
});
```

```
http://localhost:3000/demo3/hello
> URL Params { a: 'hello' }
> Queries {}

http://localhost:3000/demo3/hello?a=10&b=20
> URL Params { a: 'hello' }
> Queries { a: '10', b: '20' }
```

Queries are optional!
Route Params are required!

Query + Route Param Data type

```
router.get("/demo3/:a", function (req, res, next) {
  console.log("URL Params", req.params);
  console.log("Queries", req.query);
  res.render("index", { title: "Express" });
});
```

```
http://localhost:3000/demo3/hello
> URL Params { a: 'hello' }
> Queries {}

http://localhost:3000/demo3/hello?a=10&b=20
> URL Params { a: 'hello' }
> Queries { a: '10', b: '20' }
```

They are all in string type!
‘Hello’, ‘10’, ‘20’

</talentlabs>

Query + Route Param Data type

```
router.get("/demo4/:a", function (req, res, next) {
  console.log("URL Params", req.params);
  console.log("Queries", req.query);
  console.log(req.query["a"] + req.query["b"])
  res.render("index", { title: "Express" });
});
```

```
http://localhost:3000/demo4/hello?a=10&b=20
> URL Params { a: 'hello' }
> Queries { a: '10', b: '20' }
> 1020
```

They are all in string type!
‘Hello’, ‘10’, ‘20’

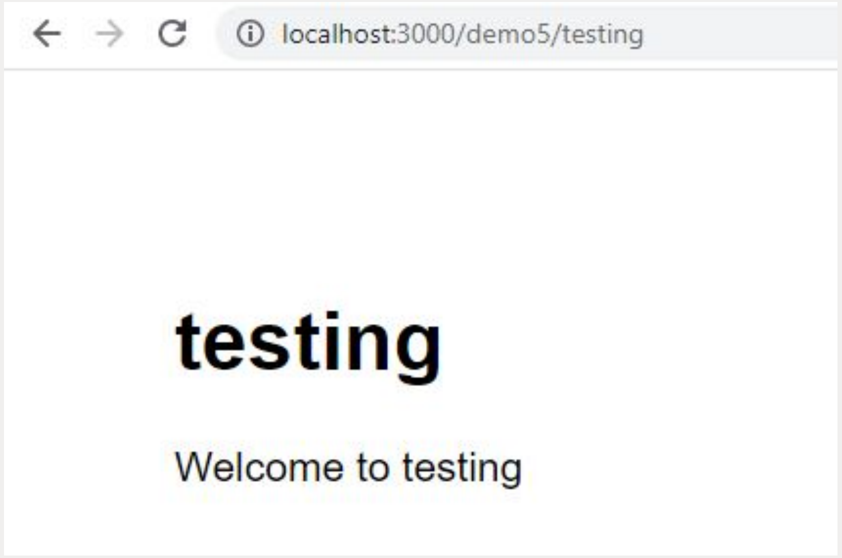
‘10’ + ‘20’ -> ‘1020’ String concatenation!

We can fix this by apply the **parseInt** or the **parseFloat** function.

</talentlabs>

Use Url Parameter for rendering

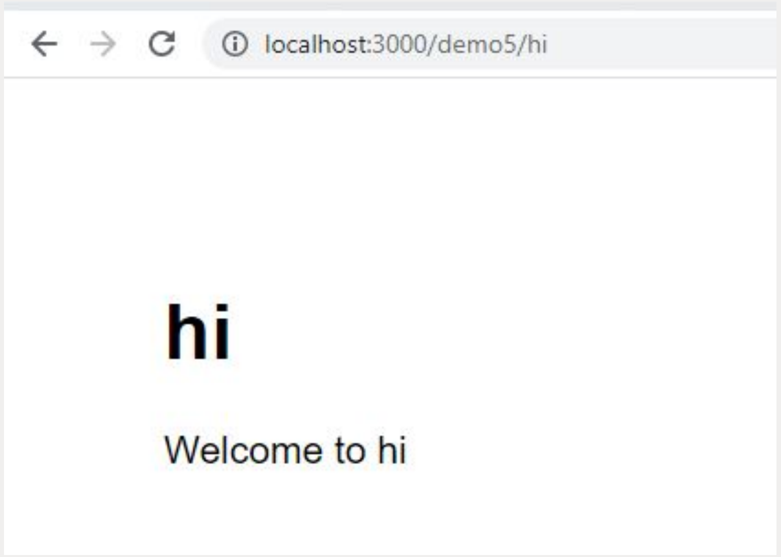
```
router.get("/demo5/:a", function (req, res, next) {
  console.log("URL Params", req.params);
  console.log("Queries", req.query);
  res.render("index", { title: req.params["a"] });
});
```



localhost:3000/demo5/testing

testing

Welcome to testing

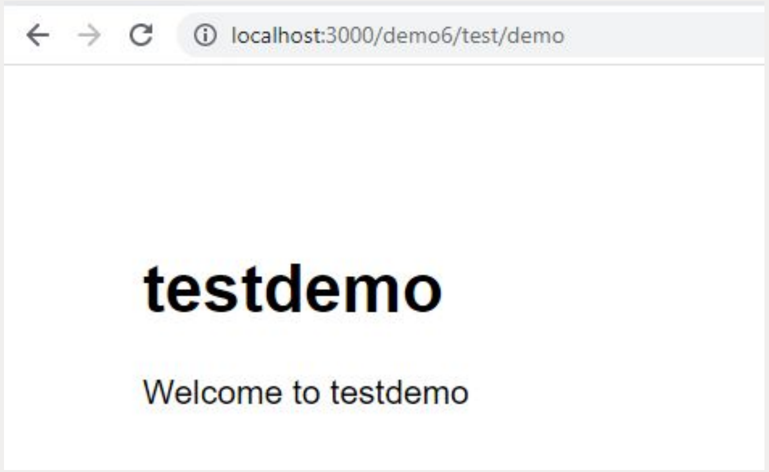


localhost:3000/demo5/hi

hi

Welcome to hi

```
router.get("/demo6/:a/:b/", function (req, res, next) {
  console.log("URL Params", req.params);
  console.log("Queries", req.query);
  res.render("index", { title: req.params["a"] + req.params["b"] });
});
```



localhost:3000/demo6/test/demo

testdemo

Welcome to testdemo