



</talentlabs>

Express Lecture 8

Express Restful API Basic



</talentlabs>

Agenda

- Restful API review
- Customers routes
- Customers/:id routes

Restful API Review

</talentlabs>



Meaningful HTTP Path and Method -> Meaningful Endpoint

A **combination of (Endpoint, HTTP method)** defines the API behaviour.

Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

Only an example, there is no 100% correct answer, the key here is to make them readable.

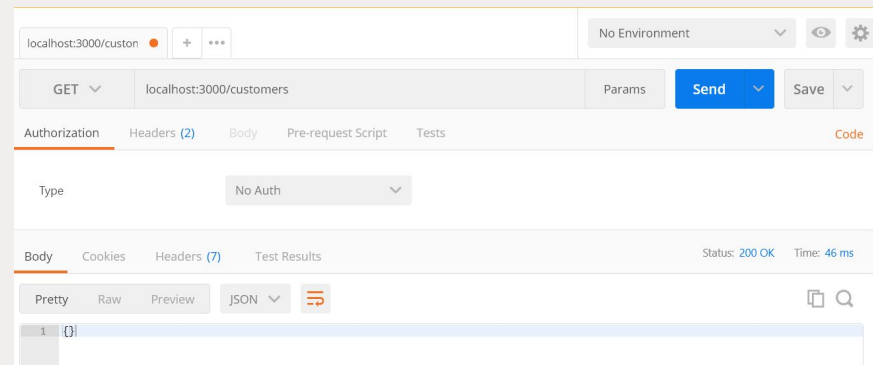
Customers Routes

</talentlabs>



Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

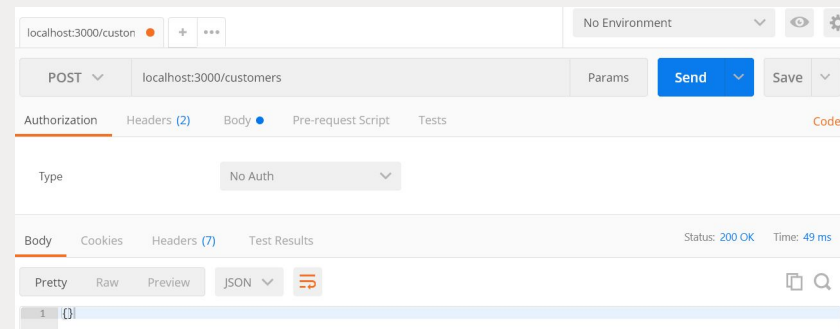
```
// GET /customers
router.get('/customers', function(req, res, next) {
  console.log("Retrieve all customers.")
  res.json({})
});
```



Retrieve all customers.
GET /customers 200 7.358 ms - 2

Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

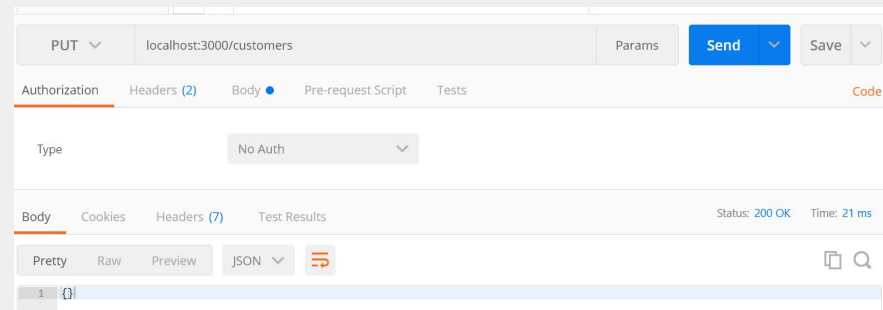
```
// POST /customers
router.post('/customers', function(req, res, next) {
  console.log("Create a new customer.")
  res.json({})
});
```



Create a new customer.
POST /customers 200 16.656 ms -

Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

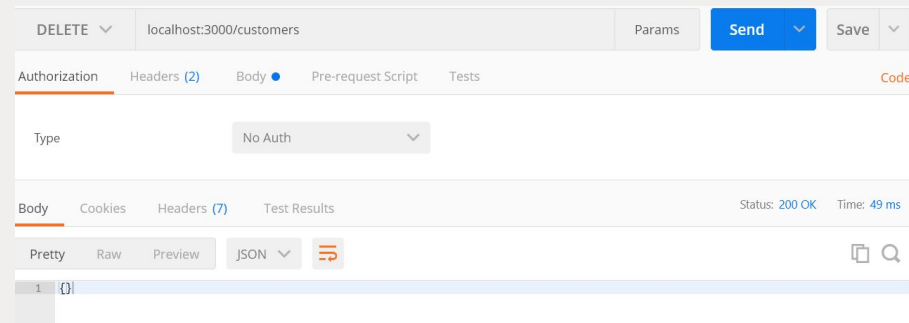
```
// PUT /customers
router.put('/customers', function(req, res, next) {
  console.log("Bulk update customers.")
  res.json({})
});
```



Bulk update customers.
PUT /customers 200 1.311 ms - 2

Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

```
// DELETE /customers
router.delete('/customers', function(req, res, next) {
  console.log("Bulk delete customers.")
  res.json({})
});
```



Bulk delete customers.
PUT /customers 200 1.311 ms - 2

Summary

- We have demonstrated how to set up Express route to **match a HTTP method and url pair!**
- For now, the handler is empty and the response is always a empty JSON.

```
// GET /customers
router.get('/customers', function(req, res, next) {
  console.log("Retrieve all customers.")
  res.json({})
});

// POST /customers
router.post('/customers', function(req, res, next) {
  console.log("Create a new customer.")
  res.json({})
});

// PUT /customers
router.put('/customers', function(req, res, next) {
  console.log("Bulk update customers.")
  res.json({})
});

// DELETE /customers
router.delete('/customers', function(req, res, next) {
  console.log("Bulk delete customers.");
  res.json({})
});
```

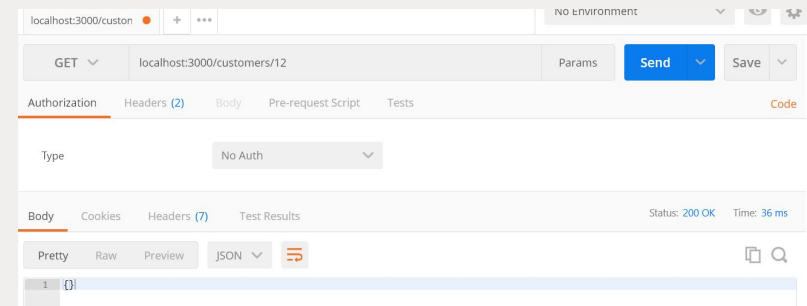
Customers/:id Routes

</talentlabs>



Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

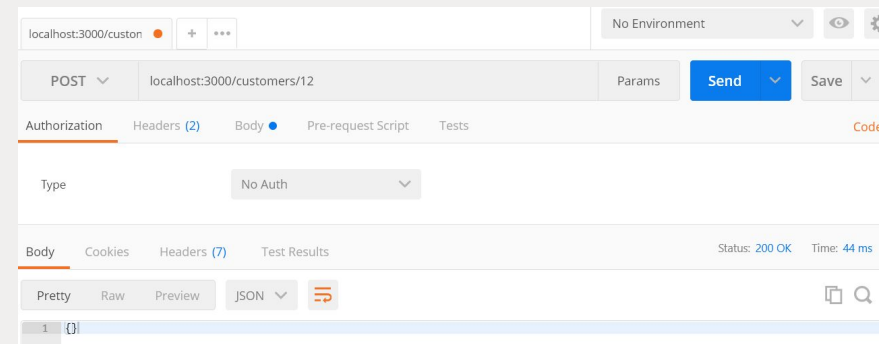
```
// GET /customers/:id
router.get('/customers/:id', function(req, res, next) {
  console.log("Retrieve a customer with id " + req.params["id"])
  res.json({})
});
```



Retrieve a customer with id 12
GET /customers/12 200 7.576 ms - 2

Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

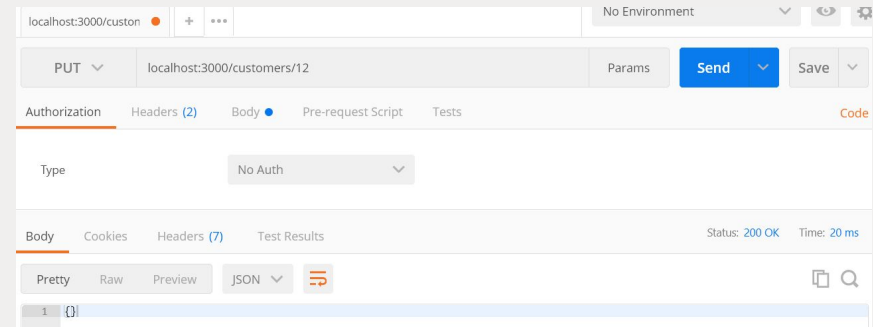
```
// POST /customers/:id
router.post('/customers/:id', function(req, res, next) {
  console.log("Create a customer with id " + req.params["id"])
  res.json({})
});
```



Create a customer with id 12
 POST /customers/12 200 21.179 ms - 2

Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

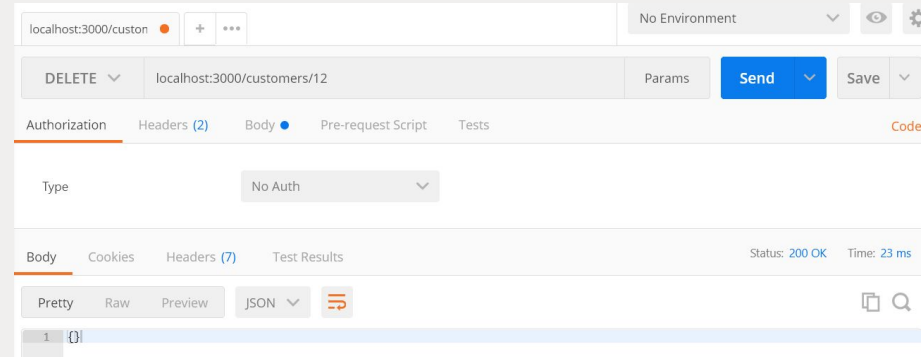
```
// PUT /customers/:id
router.put('/customers/:id', function(req, res, next) {
  console.log("Update a customer with id " + req.params["id"])
  res.json({})
});
```



Update a customer with id 12
PUT /customers/12 200 1.794 ms - 2

Endpoint	POST	GET	PUT	DELETE
/customers	Bulk / Create new customer(s)	Retrieve all customers	Bulk update of customers	Remove all customers
/customers/1 (/customers/:id)	Create a new customer with id = 1	Retrieve the details for customer 1	Update the details of customer 1 if it exists	Remove customer 1

```
// Delete /customers/:id
router.delete('/customers/:id', function(req, res, next) {
  console.log("Delete a customer with id " + req.params["id"])
  res.json({})
});
```



Delete a customer with id 12
DELETE /customers/12 200 1.794 ms - 2

Summary

- We have demonstrated how to set up Express route to **match a HTTP method and url pair!**
- For now, the handler is empty and the response is always a empty JSON.
- This time, we utilized the route parameter :id as well.

```
// GET /customers/:id
router.get('/customers/:id', function(req, res, next) {
  console.log("Retrieve a customer with id " + req.params["id"])
  res.json({})
});

// POST /customers/:id
router.post('/customers/:id', function(req, res, next) {
  console.log("Create a customer with id " + req.params["id"])
  res.json({})
});

// PUT /customers/:id
router.put('/customers/:id', function(req, res, next) {
  console.log("Update a customer with id " + req.params["id"])
  res.json({})
});

// Delete /customers/:id
router.delete('/customers/:id', function(req, res, next) {
  console.log("Delete a customer with id " + req.params["id"])
  res.json({})
});
```


JSON Response and status code

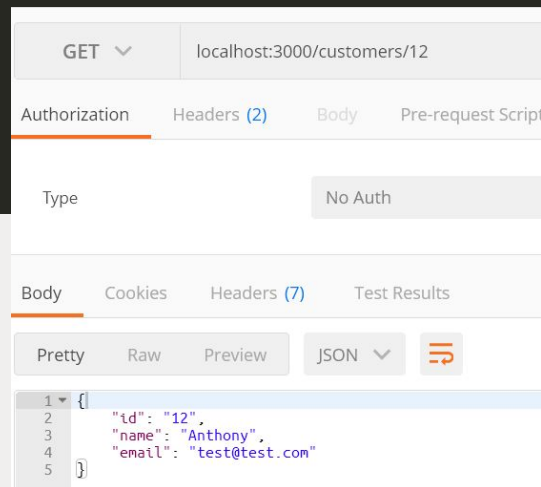
</talentlabs>



We are going to return a JSON response.

The JSON

```
// GET /customers/:id
router.get('/customers/:id', function(req, res, next) {
  console.log("Retrieve a customer with id " + req.params["id"])
  res.json({
    "id": req.params["id"],
    "name": "Anthony",
    "email": "test@test.com"
  })
});
```



We are going to return a JSON response.

The JSON

```
// GET /customers/:id
router.get('/customers/:id', function(req, res, next) {
  console.log("Retrieve a customer with id " + req.params["id"])
  res.json({
    "id": req.params["id"],
    "name": "Anthony",
    "email": "test@test.com"
  }, 200)
});
```

The Status code, default is 200 if it is not given.

Status: 200 OK Time: 22 ms



We are going to return a JSON response.

The JSON

```
// GET /customers/:id
router.get('/customers/:id', function(req, res, next) {
  console.log("Retrieve a customer with id " + req.params["id"])
  res.json({
    "id": req.params["id"],
    "name": "Anthony",
    "email": "test@test.com"
  }, 400)
});
```

The Status code, default is 200 if it is not given.

Status: 400 Bad Request Time: 40 ms

Summary

- Set up Express route to **match a HTTP method and url pair!**
- Set the JSON response.
- Set the HTTP status code.