

CMPE282 Lab1: Restful NoSQL

Supervised by Prof Kong Li

Q1.

a. List technologies, softwares (including version), and platforms for dev tools, REST client, REST server, and NoSQL database.

- OS: Mac OSX 10.12.6
- Dev tools: IntelliJ IDEA ULTIMATE 2017.2
- REST Client: Rested 2.7
- RSST Server: Spring-boot 1.5.7, Spring 4.3.11, Tomcat 8.5.20 (Spring embeded)
- NoSQL database: MongoDB (3.4.9)

b. How did you build your REST server?

Use Spring build into standalone jar

c. Is the format of the data object based on XML or JSON?

JSON

d. A sample HTTP request URI and request body for POST to create a new employee based on the XML format or JSON format (depending on your answer to c). Also indicate if there is any additional setup (e.g., HTTP header, etc.).

Request URI:

POST <http://localhost:8080/cmpe282tsungmin146/rest/employee>

Add addtional header to indicate this is JSON format:

Content-Type: application/json

Request body:

```
{ "id": 1, "firstName" : "John", "lastName": "Doe" }
```

e. If the NoSQL needs to create DB objects beforehand

Frist, make user mongod daemon is startup, then use MongoDB shell to declare DB name.

```
1 | use cmpe282tsungmin146
```

Q2. Functionality matrix:

	HTTP method + URI	HTTP Success	HTTP Error	Status
1	GET /.../rest/employee/m	OK (200)	Not found (404)	Done
2	GET /.../rest/project/n	OK (200)	Not found (404)	Done
3	POST /.../rest/employee	Created (201)	Conflict (409)	Done
4	POST /.../rest/project	Created (201)	Conflict (409)	Done
5	PUT /.../rest/employee/m	OK (200)	Not found (404)	Done
6	PUT /.../rest/project/n	OK (200)	Not found (404)	Done
7	DELETE /.../rest/employee/m	OK (200)	Not found (404)	Done
8	DELETE /.../rest/project/n	OK (200)	Not found (404)	Done
9	GET /.../rest/employee	OK (200)	Not found (404)	Done
10	GET /.../rest/project	OK (200)	Not found (404)	Done

List known issues

1. Exception handle and parameter validation

For current implementation, use annotation **@Validated** and **@Valid** to check URL parameter check and JSON input check.

It avoids a lot of condition statement in business logic. However, the exception just direct throw by validation may cause hard to figure out the error message.

2. Code redundancy

A lot of code logic is the almost the same. For example:

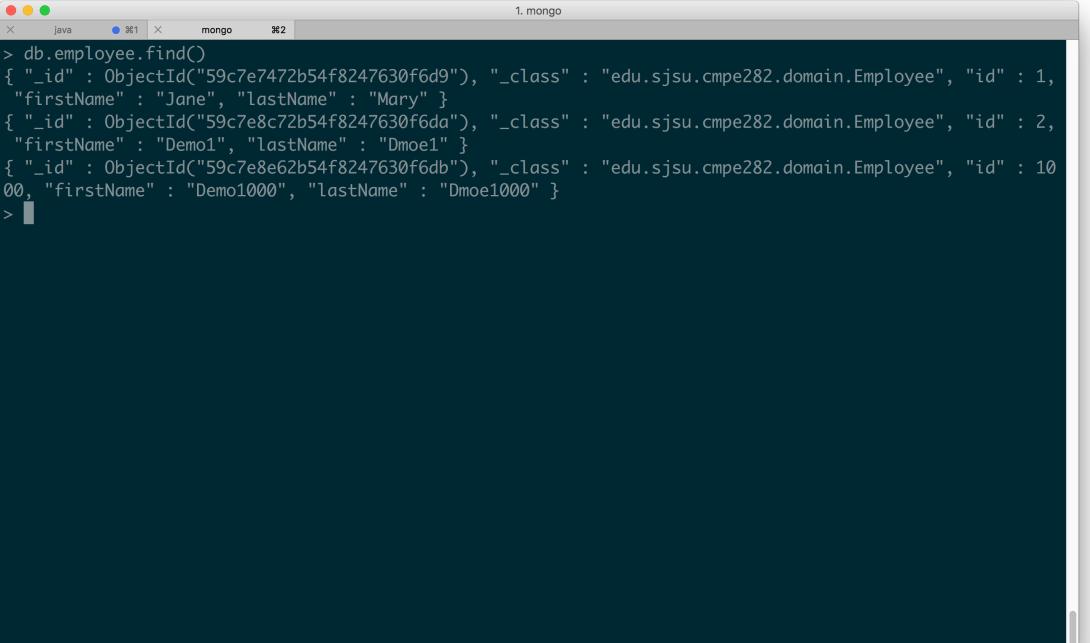
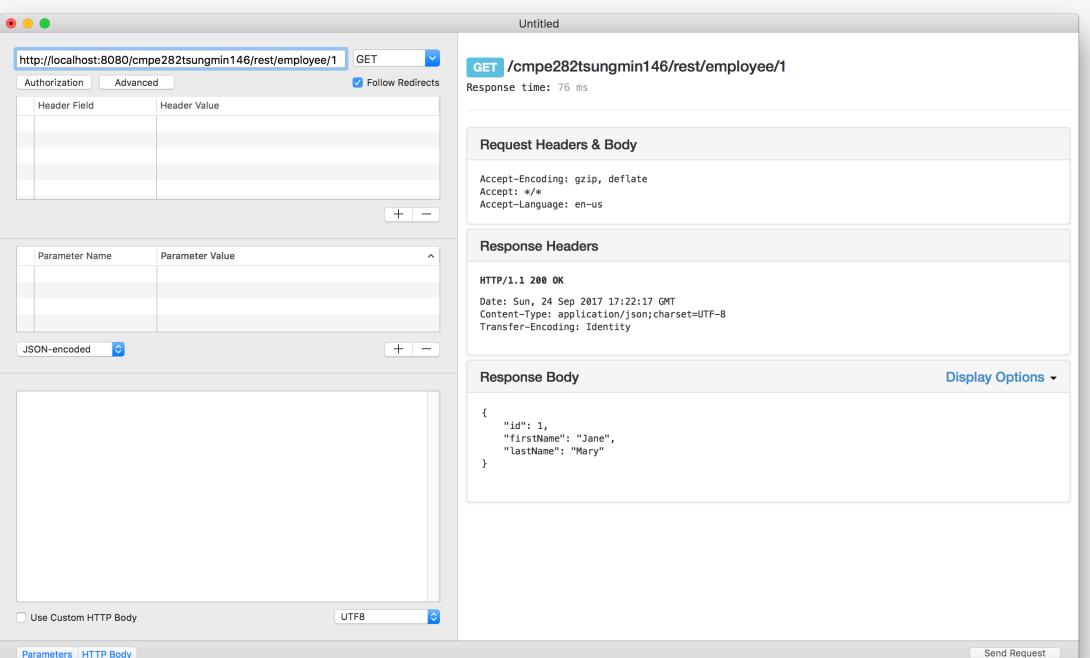
PUT API for employee and project

- * Check JSON content is valid for object
- * Retrieve form NoSQL DB test object exists
- * Update object data
- * Save object back

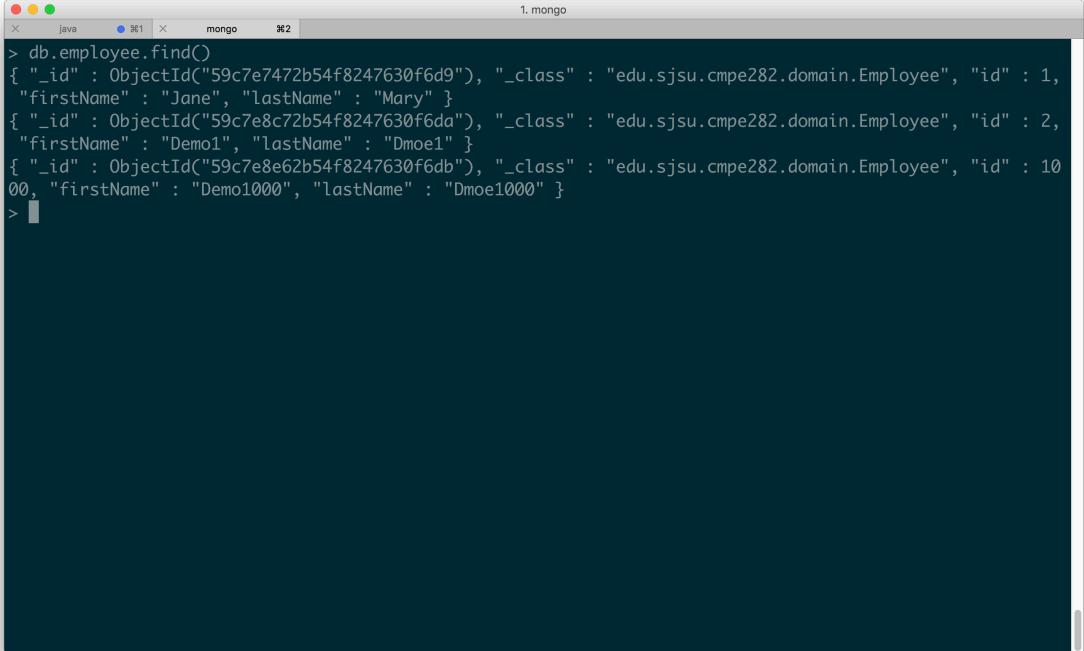
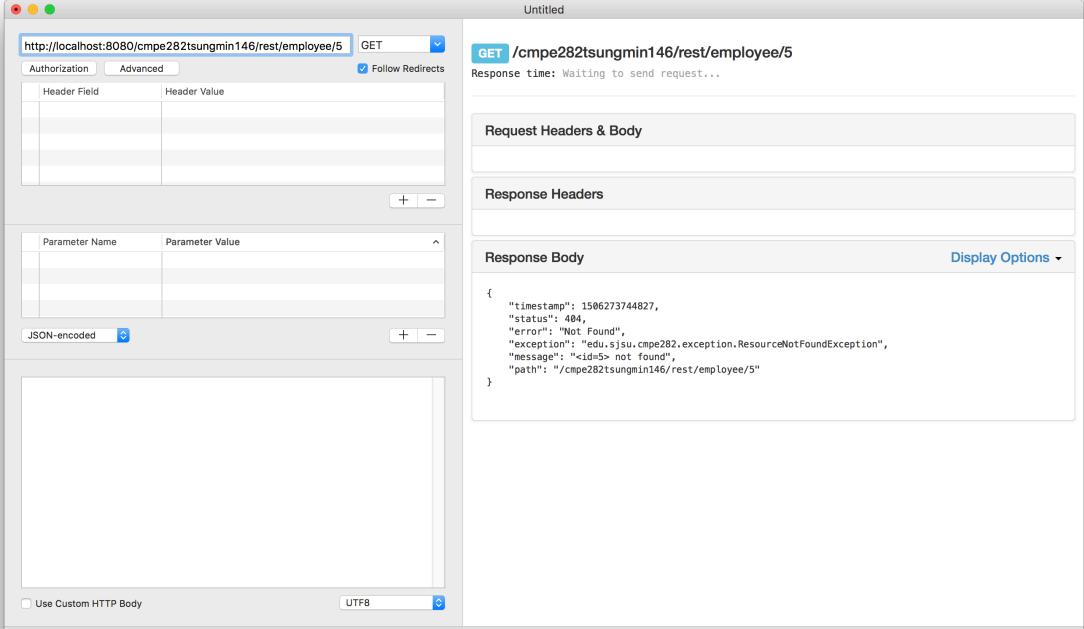
The part of business code should be encapsulated to achieve code reuse.

GET /.../rest/employee/m

Success: Query with exist Id=1

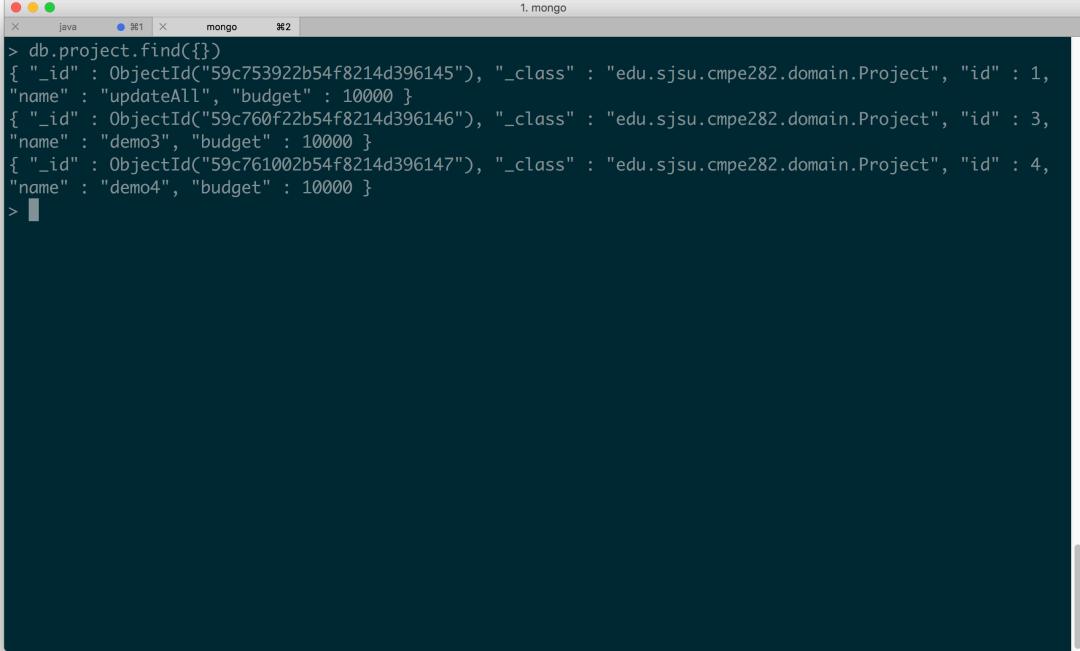
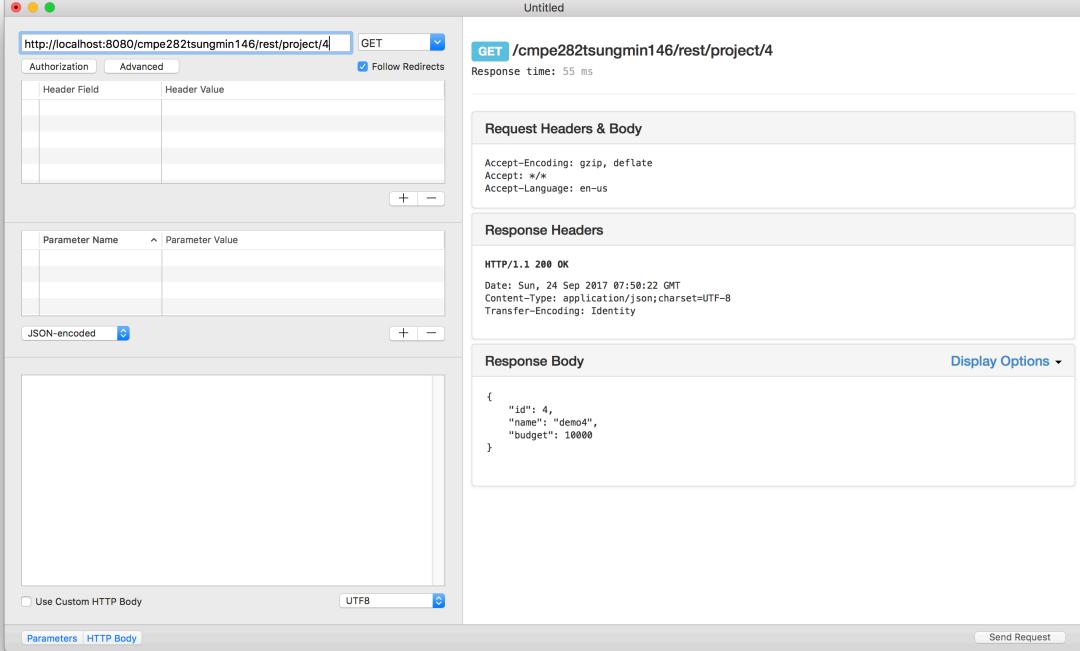
	<h2>Screen Shot</h2>  <pre>java mongo 1. mongo > db.employee.find() { "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "Jane", "lastName" : "Mary" } { "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2, "firstName" : "Demo1", "lastName" : "Dmoe1" } { "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 100, "firstName" : "Demo1000", "lastName" : "Dmoe1000" } ></pre>
MongoDB	 <p>Untitled</p> <p>GET /cmpe282tsungmin146/rest/employee/1 Response time: 76 ms</p> <p>Request Headers & Body</p> <p>Accept-Encoding: gzip, deflate Accept: */* Accept-Language: en-us</p> <p>Response Headers</p> <p>HTTP/1.1 200 OK Date: Sun, 24 Sep 2017 17:22:17 GMT Content-Type: application/json;charset=UTF-8 Transfer-Encoding: Identity</p> <p>Response Body</p> <p>{ "id": 1, "firstName": "Jane", "lastName": "Mary" }</p>
MongoDB	

Error: Query with non exist Id=5

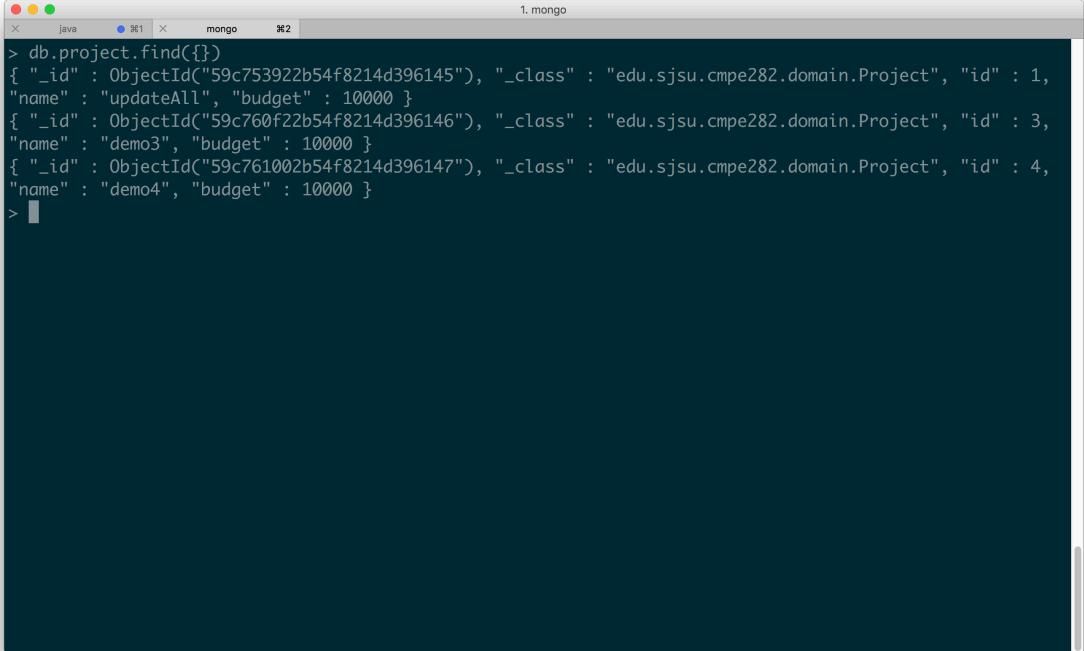
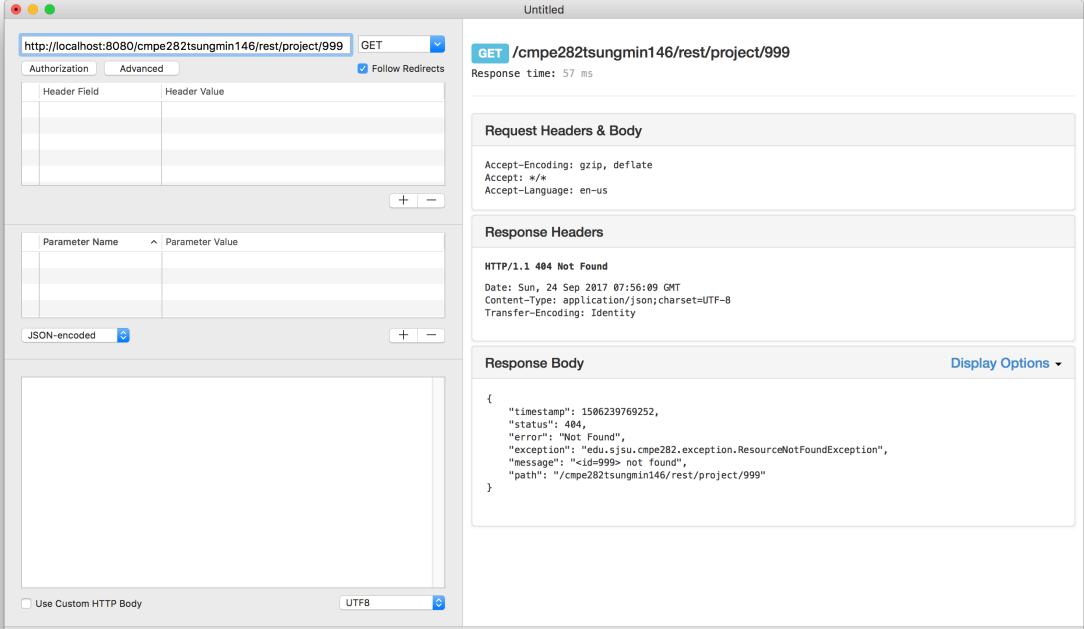
	<p style="text-align: center;">Screen Shot</p>  <pre>> db.employee.find() { "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "Jane", "lastName" : "Mary" } { "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2, "firstName" : "Demo1", "lastName" : "Dmoe1" } { "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 100, "firstName" : "Demo1000", "lastName" : "Dmoe1000" } ></pre>
MongoDB	
MongoDB	 <p>The screenshot shows a Postman interface with the following details:</p> <ul style="list-style-type: none">Request URL: <code>http://localhost:8080/cmpe282tsungmin146/rest/employee/5</code>Method: GETHeaders:<ul style="list-style-type: none">AuthorizationAdvancedFollow Redirects (checked)Request Headers & Body (empty)Response Headers (empty)Response Body:<pre>{ "timestamp": 1506273744827, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id=5> not found", "path": "/cmpe282tsungmin146/rest/employee/5" }</pre>Display Options dropdownSend Request button

GET /.../rest/project/n

Success: Query with exist Id=4

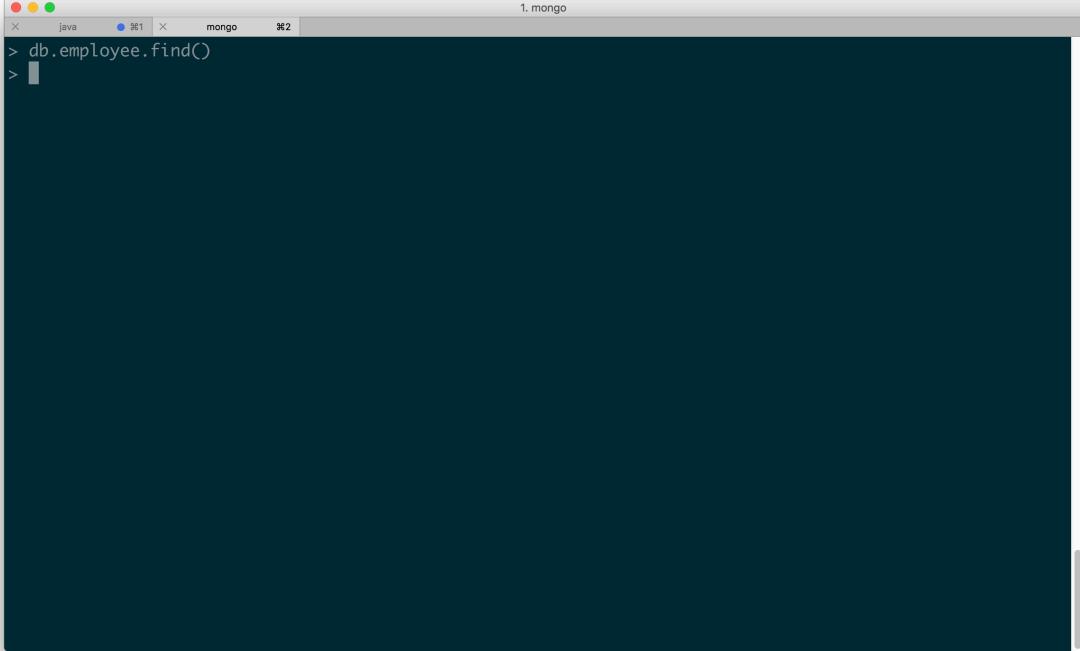
	Screen Shot
MongoDB	 <pre>> db.project.find({}) { "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 } { "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 } { "_id" : ObjectId("59c761002b54f8214d396147"), "name" : "demo4", "budget" : 10000 }</pre>
MongoDB	 <p>Request Headers & Body</p> <p>Accept-Encoding: gzip, deflate Accept: */* Accept-Language: en-us</p> <p>Response Headers</p> <p>HTTP/1.1 200 OK Date: Sun, 24 Sep 2017 07:50:22 GMT Content-Type: application/json; charset=UTF-8 Transfer-Encoding: Identity</p> <p>Response Body</p> <pre>{ "id": 4, "name": "demo4", "budget": 10000 }</pre>

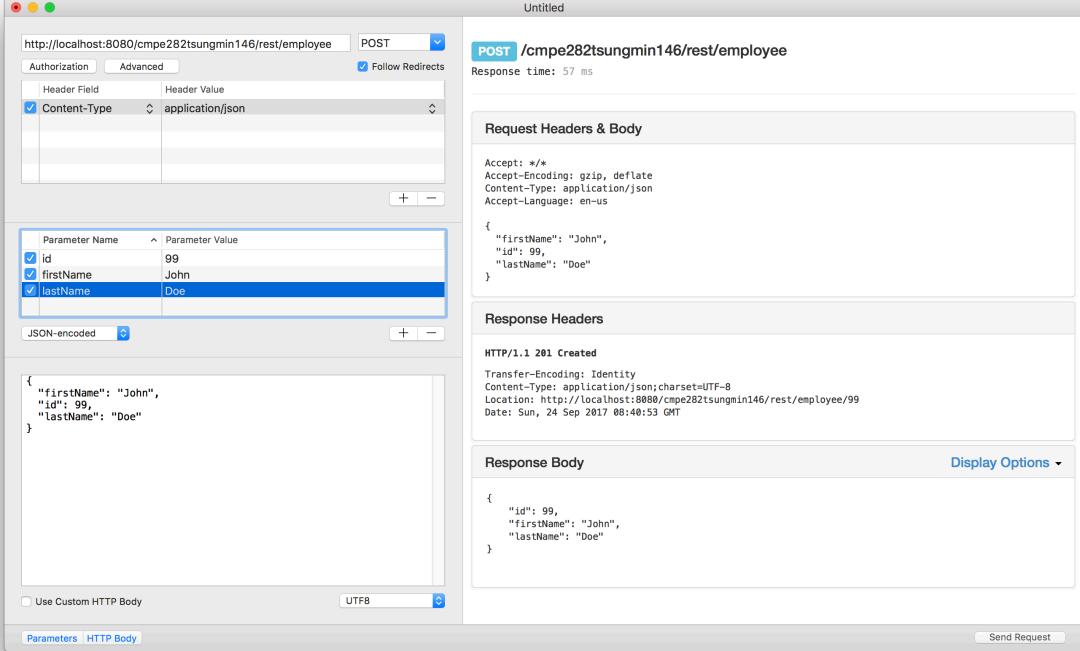
Error: Query with non exist Id=999

	<h3>Screen Shot</h3>  <pre>> db.project.find({}) { "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 } { "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 } { "_id" : ObjectId("59c761002b54f8214d396147"), "name" : "demo4", "budget" : 10000 } > </pre>
MongoDB	
MongoDB	 <p>Request URL: http://localhost:8080/cmpe282tsungmin146/rest/project/999 GET</p> <p>Request Headers & Body</p> <p>Response Headers</p> <p>HTTP/1.1 404 Not Found</p> <p>Response Body</p> <pre>{ "timestamp": 1506239769252, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id:999> not found", "path": "/cmpe282tsungmin146/rest/project/999" }</pre>

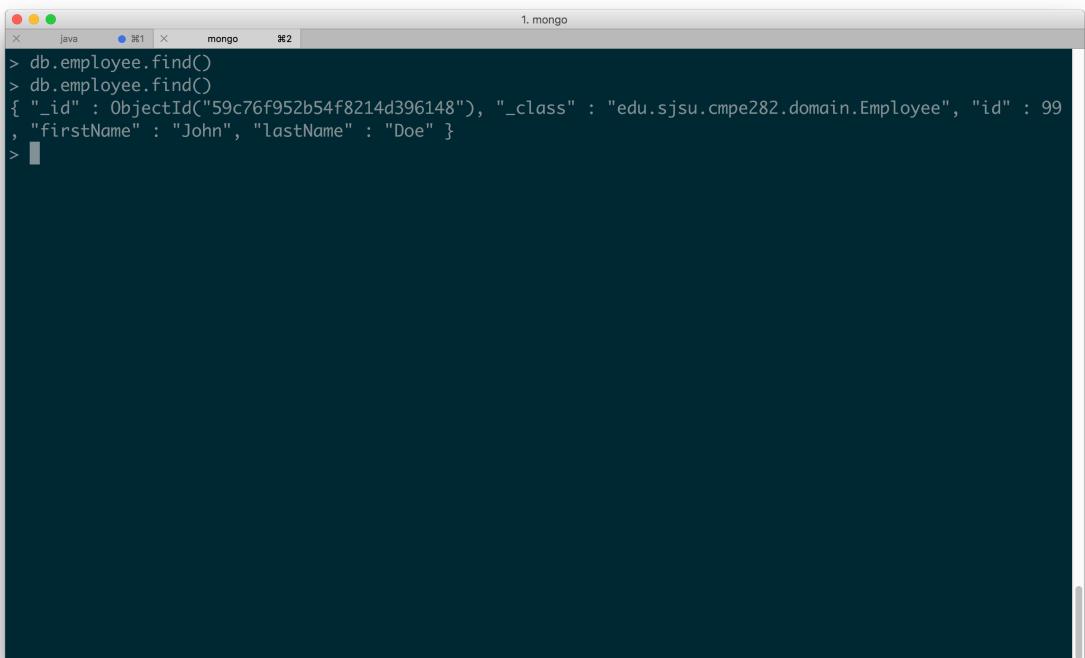
POST .../rest/employee

Success: Create employee based JSON context

MongoDB	Screen Shot
	 A screenshot of a macOS terminal window titled "mongo". It contains the command `db.employee.find()` followed by a blank line for the response.

Create	Screen Shot
	 A screenshot of a REST client interface. The request URL is `http://localhost:8080/cmpe282tsungmin146/rest/employee` with a `POST` method. The request body is a JSON object with parameters: `id: 99`, `firstName: John`, and `lastName: Doe`. The response shows a `201 Created` status with headers like `Content-Type: application/json; charset=UTF-8` and a location header pointing to the created resource's URL. The response body matches the request body.

MongoDB



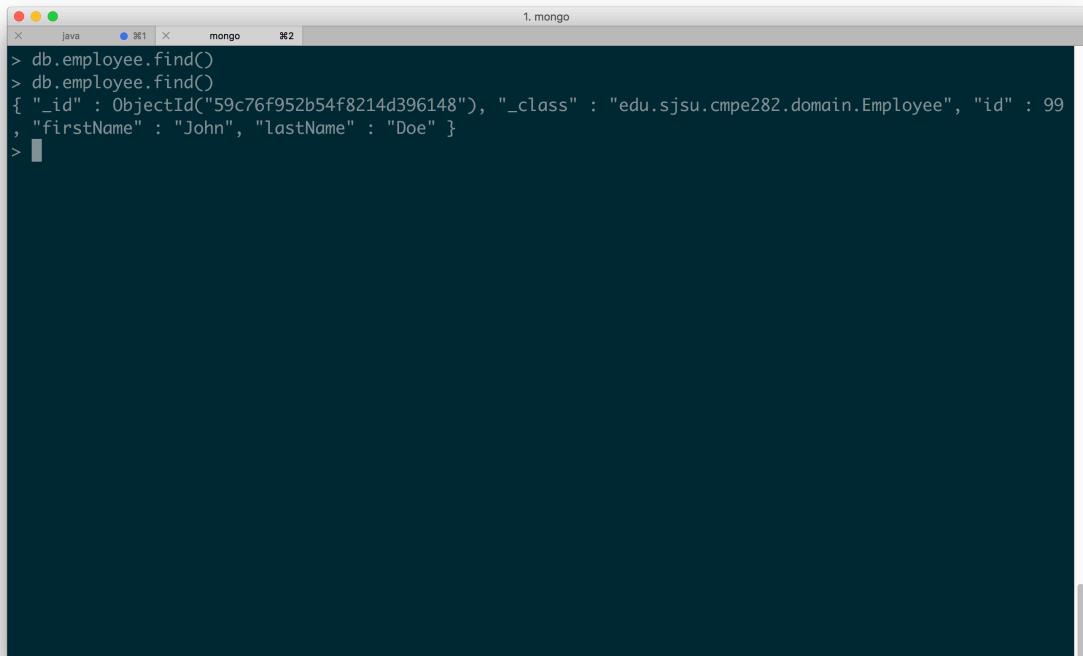
A screenshot of a terminal window titled "mongo". The window contains the following MongoDB shell session:

```
> db.employee.find()
> db.employee.find()
{ "_id" : ObjectId("59c76f952b54f8214d396148"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
>
```

Error: Create employee with the same Id=99

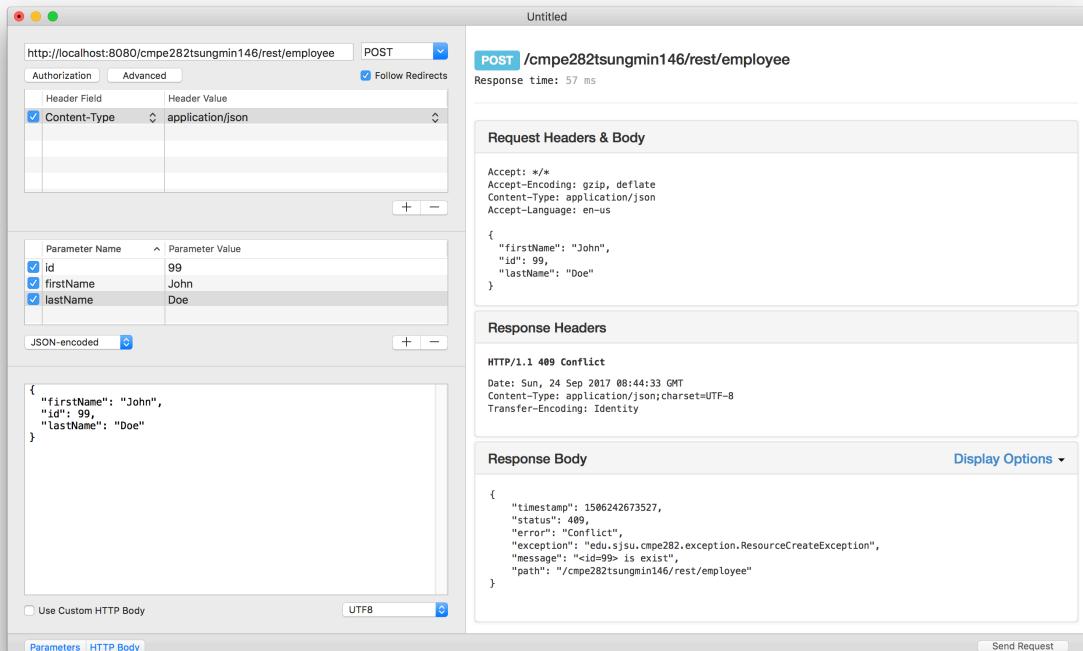
MongoDB

Screen Shot



```
> db.employee.find()
> db.employee.find()
{ "_id" : ObjectId("59c76f952b54f8214d396148"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
>
```

Conflict



POST /cmpe282tsungmin146/rest/employee

Request Headers & Body

```
Accept: */*
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: en-us

{
  "firstName": "John",
  "id": 99,
  "lastName": "Doe"
}
```

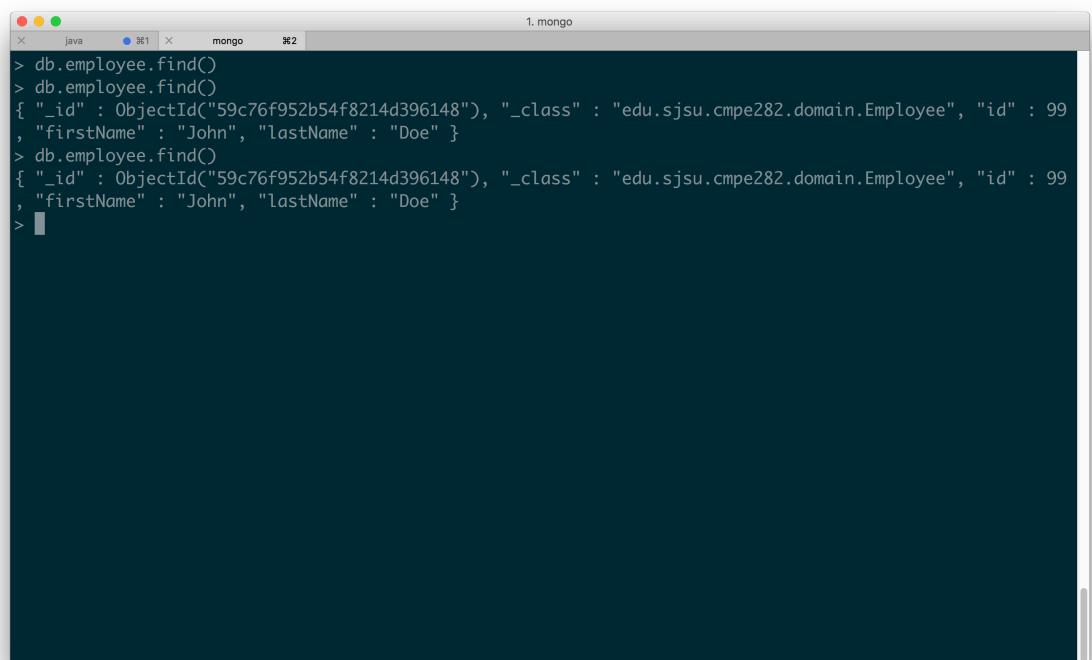
Response Headers

```
HTTP/1.1 409 Conflict
Date: Sun, 24 Sep 2017 08:44:33 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: Identity
```

Response Body

```
{
  "timestamp": 1506242673527,
  "status": 409,
  "error": "Conflict",
  "exception": "edu.sjsu.cmpe282.exception.ResourceCreateException",
  "message": "<id:99> is exist",
  "path": "/cmpe282tsungmin146/rest/employee"
}
```

MongoDB

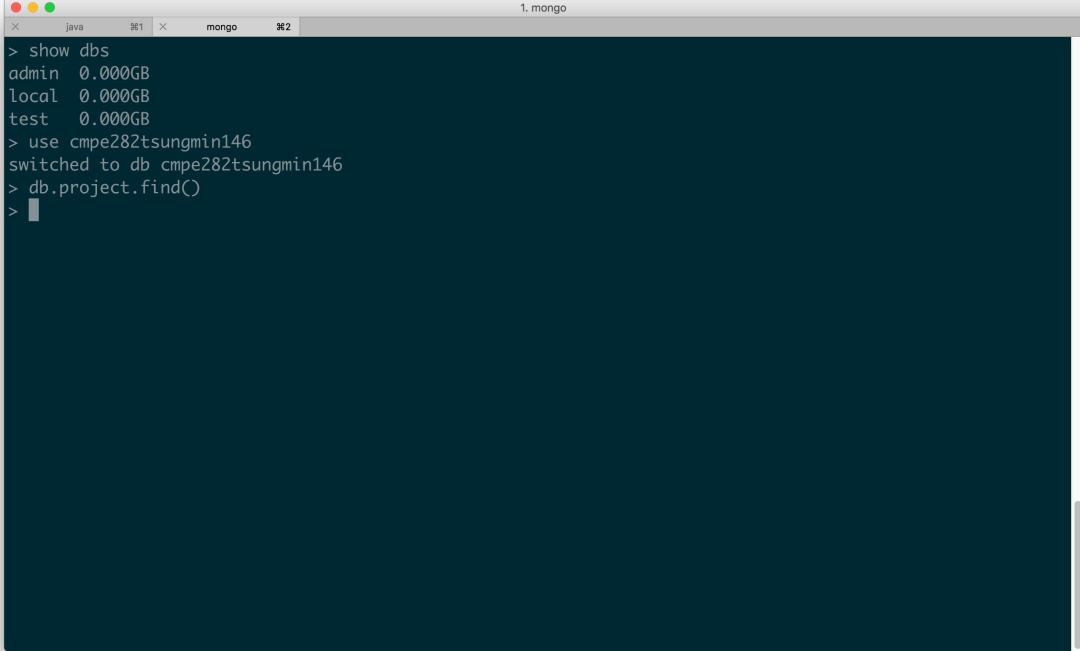
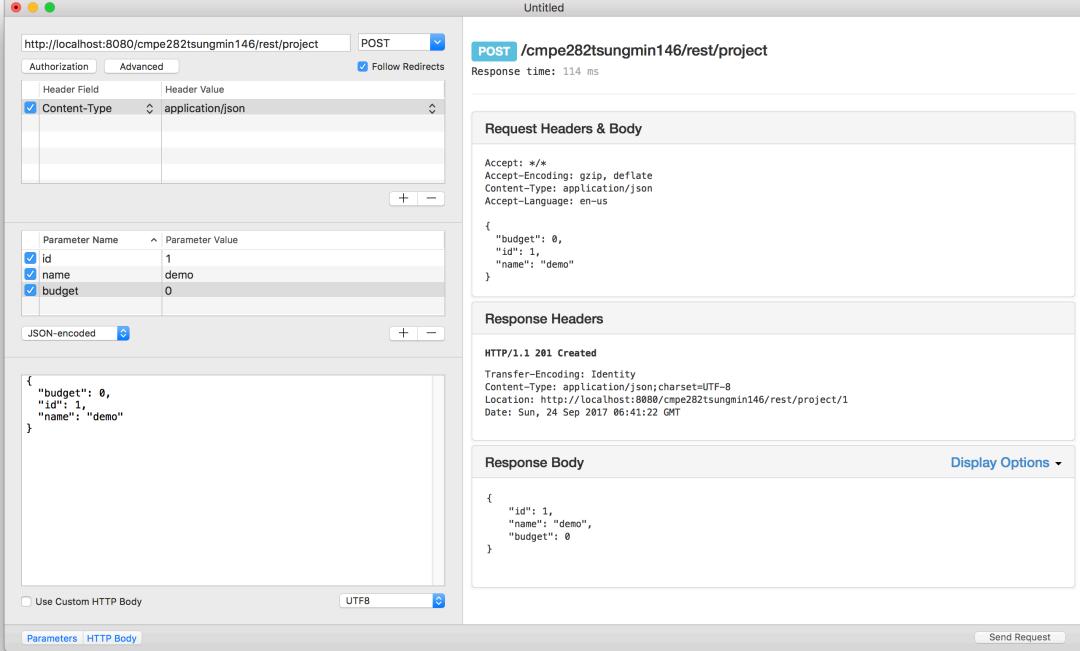


A screenshot of a terminal window titled "1. mongo". The window contains the following MongoDB shell session:

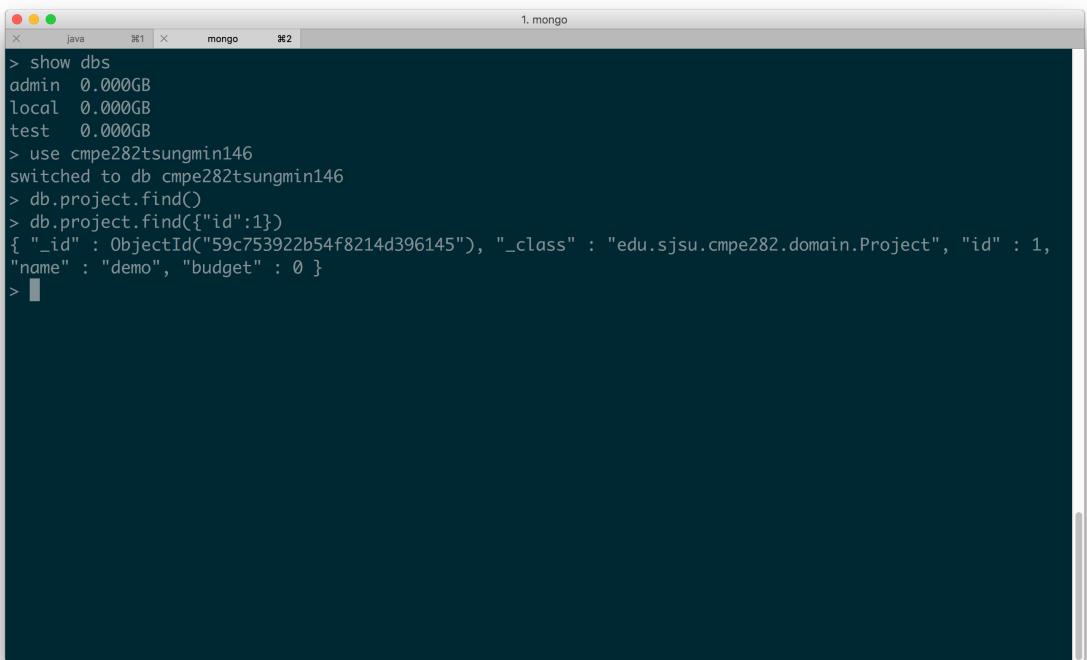
```
> db.employee.find()
> db.employee.find()
{ "_id" : ObjectId("59c76f952b54f8214d396148"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c76f952b54f8214d396148"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
>
```

POST /.../rest/project

Success: Create project based JSON context

	<p style="text-align: center;">Screen Shot</p>  <p>A screenshot of a Mac OS X terminal window titled "mongo". It contains the following MongoDB shell session:</p> <pre>> show dbs admin 0.000GB local 0.000GB test 0.000GB > use cmpe282tsungmin146 switched to db cmpe282tsungmin146 > db.project.find() > </pre>
MongoDB	
Create	 <p>A screenshot of the Postman application interface. The request URL is <code>http://localhost:8080/cmpe282tsungmin146/rest/project</code> and the method is <code>POST</code>. The request body is a JSON object:</p> <pre>{ "budget": 0, "id": 1, "name": "demo" }</pre> <p>The response status is <code>HTTP/1.1 201 Created</code>. The response headers include <code>Transfer-Encoding: Identity</code>, <code>Content-Type: application/json; charset=UTF-8</code>, and <code>Location: http://localhost:8080/cmpe282tsungmin146/rest/project/1</code>. The response body is identical to the request body.</p>

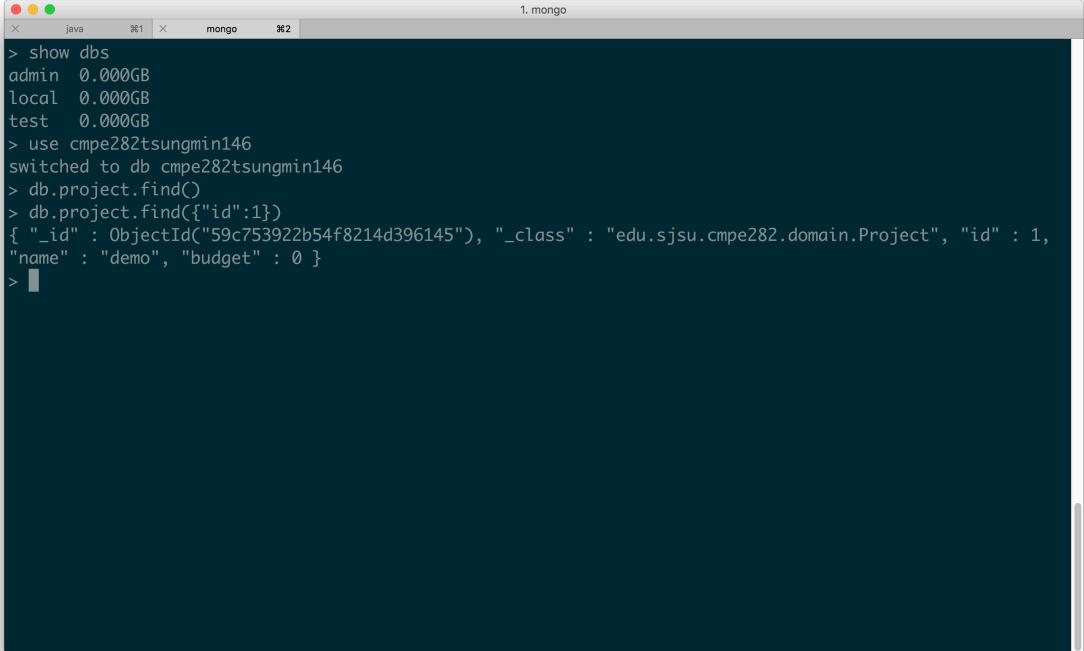
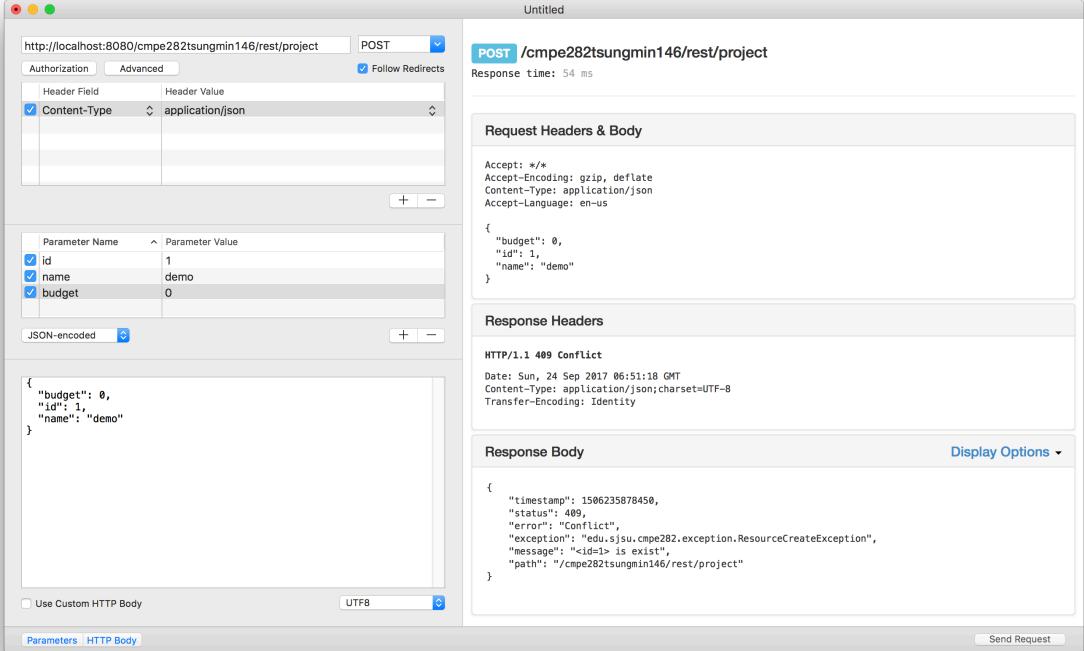
MongoDB



A screenshot of a terminal window titled "1. mongo". The window shows the following MongoDB shell session:

```
> show dbs
admin 0.000GB
local 0.000GB
test 0.000GB
> use cmpe282tsungmin146
switched to db cmpe282tsungmin146
> db.project.find()
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
"name" : "demo", "budget" : 0 }
> █
```

Error: Create project with the same Id=1

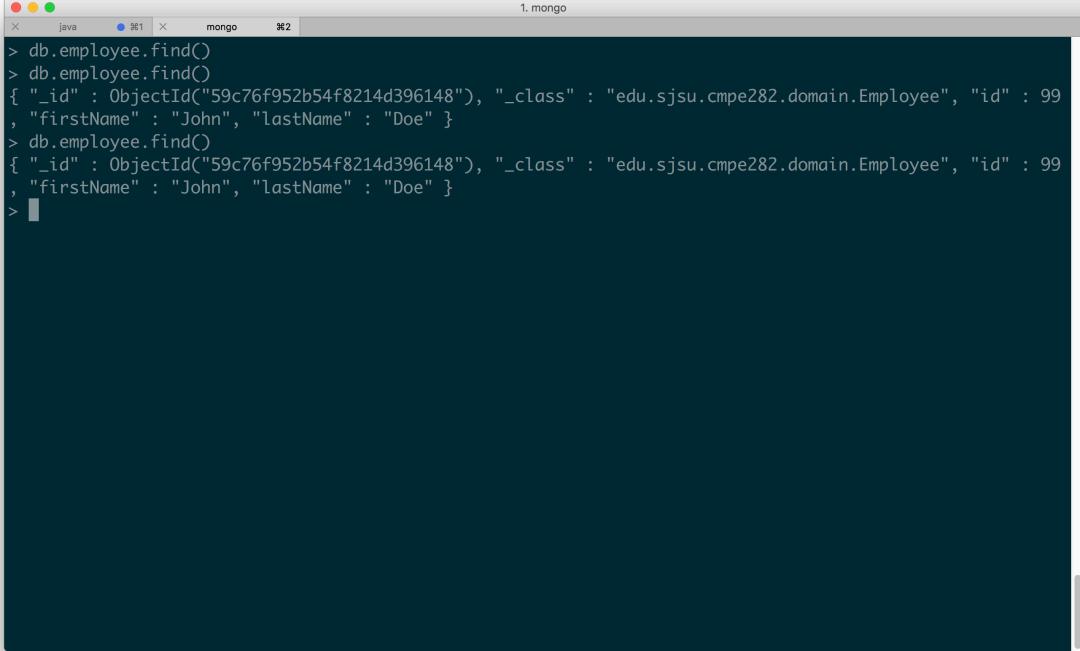
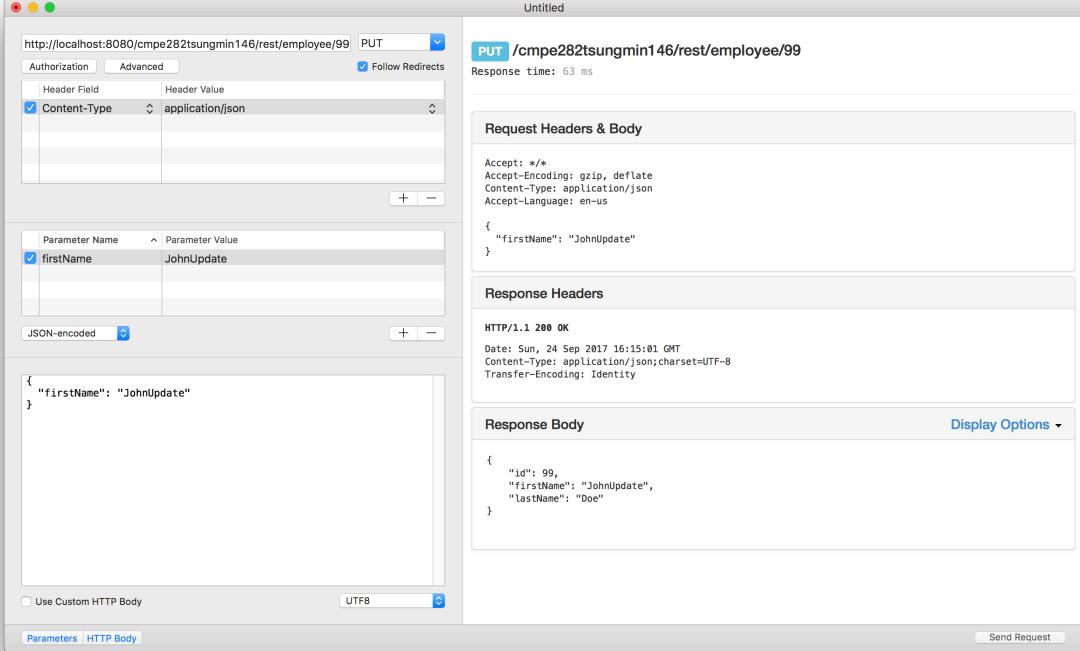
	<h3>Screen Shot</h3>  <p>The screenshot shows a MongoDB shell window titled "mongo". The command `db.project.find({id:1})` is run, returning a single document: { _id: ObjectId("59c753922b54f8214d396145"), _class: "edu.sjsu.cmpe282.domain.Project", id: 1, name: "demo", budget: 0 }.</p>
MongoDB	 <p>The screenshot shows a REST client interface with a POST request to <code>/cmpe282tsungmin146/rest/project</code>. The request body is a JSON object with parameters: { id: 1, name: "demo", budget: 0 }. The response status is 409 Conflict, with the message: "HTTP/1.1 409 Conflict Date: Sun, 24 Sep 2017 06:51:18 GMT Content-Type: application/json; charset=UTF-8 Transfer-Encoding: Identity". The response body contains a detailed error message: { "timestamp": 1506235878450, "status": 409, "error": "Conflict", "exception": "edu.sjsu.cmpe282.exception.ResourceCreateException", "message": "<id> is exist", "path": "/cmpe282tsungmin146/rest/project" }.</p>

MongoDB

```
1. mongo
> show dbs
admin 0.000GB
local 0.000GB
test 0.000GB
> use cmpe282tsungmin146
switched to db cmpe282tsungmin146
> db.project.find()
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
"name" : "demo", "budget" : 0 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
"name" : "demo", "budget" : 0 }
> █
```

PUT .../rest/employee/n

Success: Demo employee Id=99 partial update

	Screen Shot
MongoDB	 <p>The screenshot shows the MongoDB shell interface. The user has run the command `db.employee.find()` twice, and the results are displayed. Both queries return the same document:</p> <pre>> db.employee.find() > db.employee.find() { "_id" : ObjectId("59c76f952b54f8214d396148"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99 , "firstName" : "John", "lastName" : "Doe" } > db.employee.find() { "_id" : ObjectId("59c76f952b54f8214d396148"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99 , "firstName" : "John", "lastName" : "Doe" }</pre>
Update first name	 <p>The screenshot shows a REST client tool with the following details:</p> <ul style="list-style-type: none">Request URL: <code>http://localhost:8080/cmpe282tsungmin146/rest/employee/99</code>Method: PUTHeaders: Content-Type: application/jsonBody (Request Headers & Body): <code>{ "firstName": "JohnUpdate" }</code>Response Headers: HTTP/1.1 200 OK, Date: Sun, 24 Sep 2017 16:15:01 GMT, Content-Type: application/json; charset=UTF-8, Transfer-Encoding: IdentityResponse Body: <code>{ "id": 99, "firstName": "JohnUpdate", "lastName": "Doe" }</code>

MongoDB

```
1. mongo
> db.employee.find()
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "JohnUpdate", "lastName" : "Doe" }
> 
```

Update
last name

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://localhost:8080/cmpe282tsungmin146/rest/employee/99`
- Method:** `PUT`
- Headers:** `Content-Type: application/json`
- Parameters:** `lastName: DoeUpdate`
- Request Body:** `{"lastName": "DoeUpdate"}`
- Response Headers:**
 - `HTTP/1.1 200 OK`
 - `Date: Sun, 24 Sep 2017 16:27:26 GMT`
 - `Content-Type: application/json; charset=UTF-8`
 - `Transfer-Encoding: Identity`
- Response Body:** `{ "id": 99, "firstName": "JohnUpdate", "lastName": "DoeUpdate" }`

MongoDB

```
1. mongo
> db.employee.find()
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "JohnUpdate", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "JohnUpdate", "lastName" : "DoeUpdate" }
>
```

Update
all

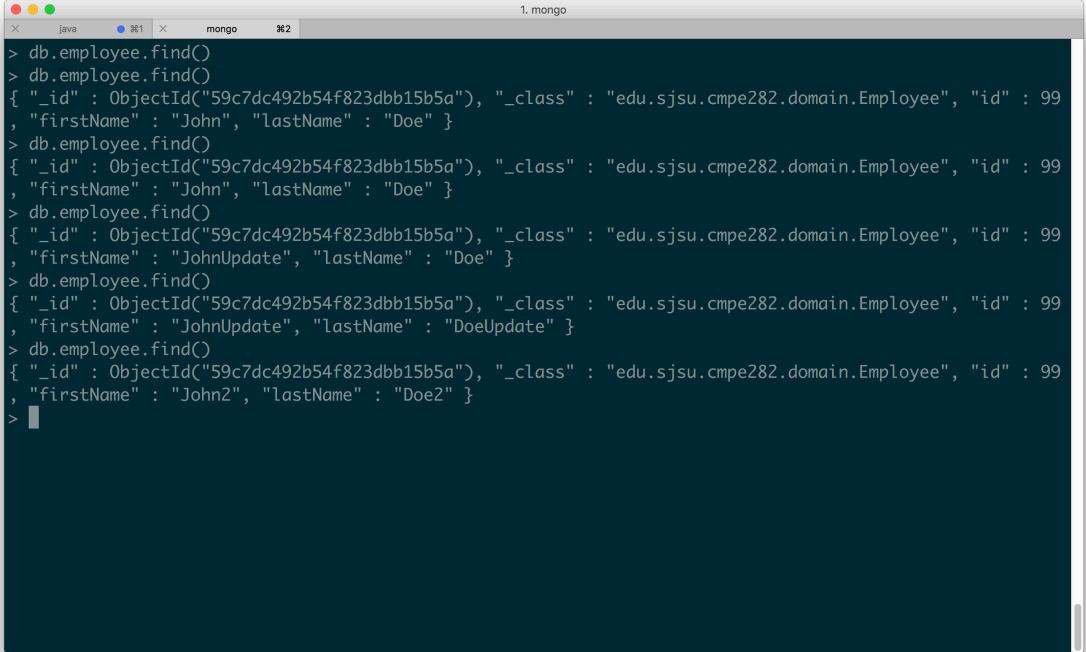
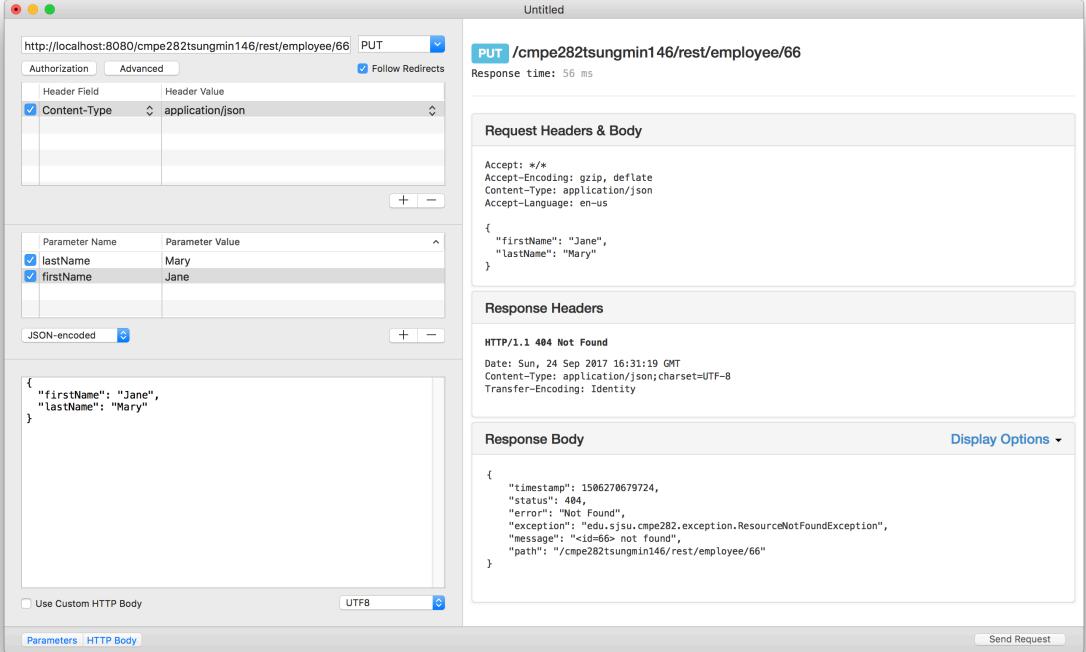
The screenshot shows a REST client interface with the following details:

- Request URL:** `http://localhost:8080/cmpe282tsungmin146/rest/employee/99`
- Method:** PUT
- Headers:** Content-Type: application/json
- Parameters:** firstName: John2, lastName: Doe2
- Request Body:** `{ "firstName": "John2", "lastName": "Doe2" }`
- Response Headers:** HTTP/1.1 200 OK, Date: Sun, 24 Sep 2017 16:29:16 GMT, Content-Type: application/json; charset=UTF-8, Transfer-Encoding: Identity
- Response Body:** `{ "_id": 99, "firstName": "John2", "lastName": "Doe2" }`

MongoDB

```
1. mongo
> db.employee.find()
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "JohnUpdate", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "JohnUpdate", "lastName" : "DoeUpdate" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John2", "lastName" : "Doe2" }
> █
```

Error: Employee Id=4 does not exist error, employ in JSON Id=10 does not match request URL

	Screen Shot
MongoDB	 A screenshot of a macOS terminal window titled "mongo". It shows several MongoDB shell commands and their results. The results are identical for each command, showing a document with _id 99, firstName "John", and lastName "Doe". <pre>> db.employee.find() > db.employee.find() { "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99 , "firstName" : "John", "lastName" : "Doe" } > db.employee.find() { "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99 , "firstName" : "John", "lastName" : "Doe" } > db.employee.find() { "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99 , "firstName" : "JohnUpdate", "lastName" : "Doe" } > db.employee.find() { "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99 , "firstName" : "JohnUpdate", "lastName" : "DoeUpdate" } > db.employee.find() { "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99 , "firstName" : "John2", "lastName" : "Doe2" } ></pre>
Id not exist	 A screenshot of the Postman application interface. It shows a PUT request to "http://localhost:8080/cmpe282tsungmin146/rest/employee/66". The request body contains parameters: lastName: "Mary" and firstName: "Jane". The response status is 404 Not Found, with the following error message: <pre>HTTP/1.1 404 Not Found Date: Sun, 24 Sep 2017 16:31:19 GMT Content-Type: application/json; charset=UTF-8 Transfer-Encoding: Identity { "timestamp": 1506270679724, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id=> not found", "path": "/cmpe282tsungmin146/rest/employee/66" }</pre>

MongoDB

```
1. mongo
> db.employee.find()
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "JohnUpdate", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "JohnUpdate", "lastName" : "DoeUpdate" }
> db.employee.find()
{ "_id" : ObjectId("59c7dc492b54f823dbb15b5a"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 99
, "firstName" : "John2", "lastName" : "Doe2" }
> db.employee.find({"id":66})
>
```

URL and
JSON id
not match

http://localhost:8080/cmpe282tsungmin146/rest/employee/1 PUT

Header Field Header Value

Content-Type application/json

Parameter Name Parameter Value

lastName MaryUpdate

firstName JaneUpdate

id 10

Request Headers & Body

Accept: */*

Accept-Encoding: gzip, deflate

Content-Type: application/json

Accept-Language: en-US

{ "firstName": "JaneUpdate",
"id": 10,
"lastName": "MaryUpdate"
}

Response Headers

HTTP/1.1 404 Not Found

Date: Sun, 24 Sep 2017 17:13:28 GMT

Content-Type: application/json; charset=UTF-8

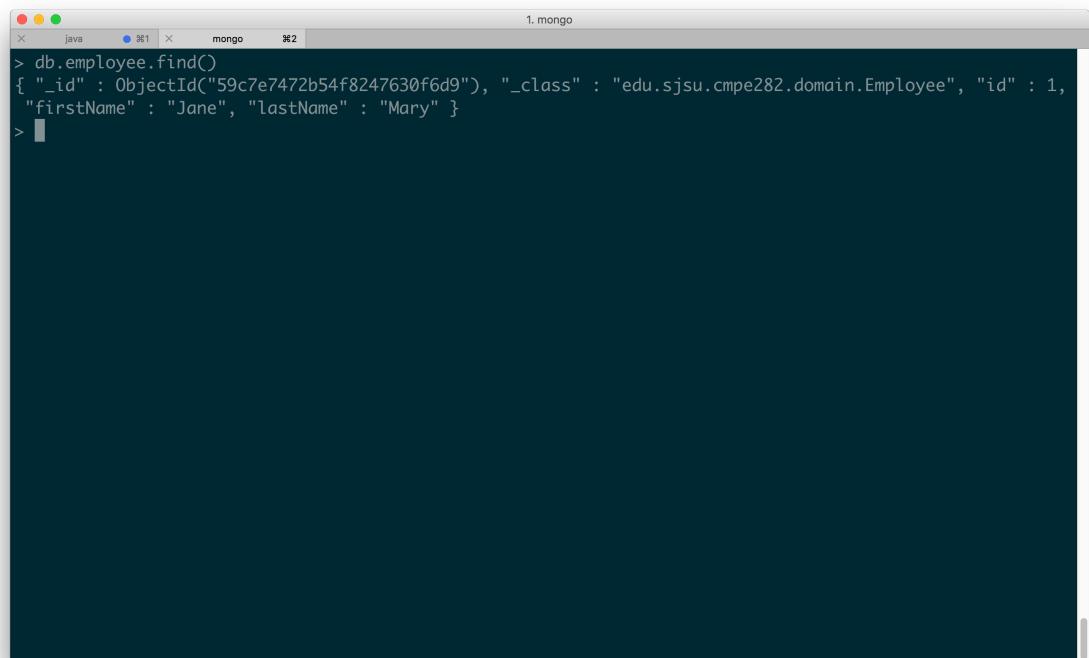
Transfer-Encoding: Identity

Response Body

{
"timestamp": 1506273209536,
"status": 404,
"error": "Not Found",
"exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException",
"message": "REQUEST: <id=1> not match with JSON: <id=10>",
"path": "/cmpe282tsungmin146/rest/employee/1"
}

Send Request

MongoDB



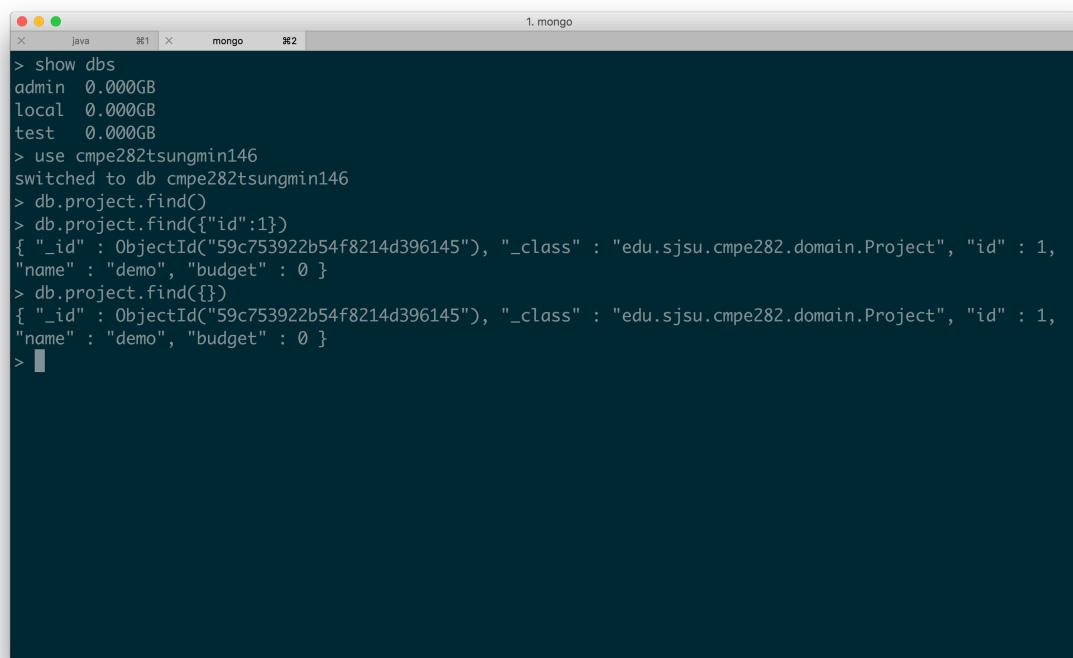
```
> db.employee.find()
{ "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "._class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "Jane", "lastName" : "Mary" }
> █
```

PUT ../rest/project/n

Success: Demo project Id=1 partial update

MongoDB

Screen Shot



```
> show dbs
admin 0.000GB
local 0.000GB
test 0.000GB
> use cmpe282tsungmin146
switched to db cmpe282tsungmin146
> db.project.find()
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "._class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "._class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> █
```

Update
name

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://localhost:8080/cmpe282tsungmin146/rest/project/1`
- Method:** PUT
- Headers:** Content-Type: application/json
- Request Body:** `{ "name": "demoPartialUpdate" }`
- Response Headers:** HTTP/1.1 200 OK, Date: Sun, 24 Sep 2017 07:10:06 GMT, Content-Type: application/json; charset=UTF-8, Transfer-Encoding: Identity
- Response Body:** `{ "id": 1, "name": "demoPartialUpdate", "budget": 0 }`

MongoDB

The screenshot shows a MongoDB shell session with the following commands and output:

```
> show dbs
admin 0.000GB
local 0.000GB
test 0.000GB
> use cmpe282tsungmin146
switched to db cmpe282tsungmin146
> db.project.find()
> db.project.find({ "id": 1 })
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> db.project.find({ "id": 1 })
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demoPartialUpdate", "budget" : 0 }
>
```

Update budget

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://localhost:8080/cmpe282tsungmin146/rest/project/1`
- Method:** `PUT`
- Headers:** `Content-Type: application/json`
- Parameters:** `budget: 10`
- Response Headers:**
 - `HTTP/1.1 200 OK`
 - `Date: Sun, 24 Sep 2017 07:12:37 GMT`
 - `Content-Type: application/json; charset=UTF-8`
 - `Transfer-Encoding: Identity`
- Response Body:**

```
{  
    "_id": 1,  
    "name": "demoPartialUpdate",  
    "budget": 10  
}
```

MongoDB

The screenshot shows a MongoDB shell session with the following commands and output:

```
> show dbs
admin 0.000GB
local 0.000GB
test 0.000GB
> use cmpe282tsungmin146
switched to db cmpe282tsungmin146
> db.project.find()
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demoPartialUpdate", "budget" : 0 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demoPartialUpdate", "budget" : 10 }
```

Update
all

HTTP/1.1 200 OK

Date: Sun, 24 Sep 2017 07:15:14 GMT

Content-Type: application/json; charset=UTF-8

Transfer-Encoding: Identity

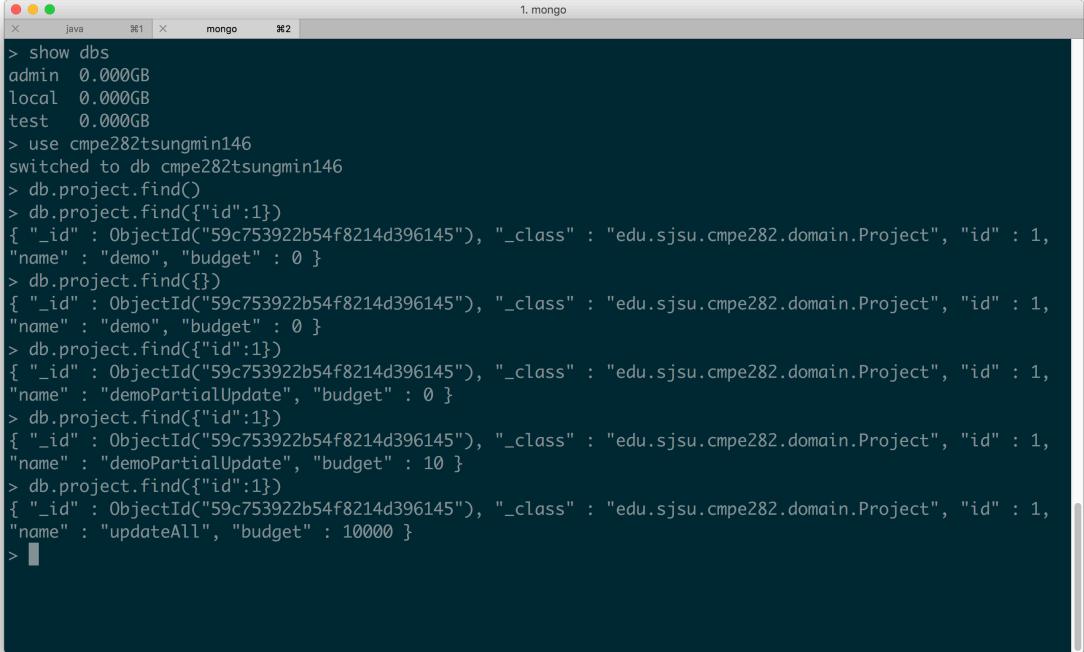
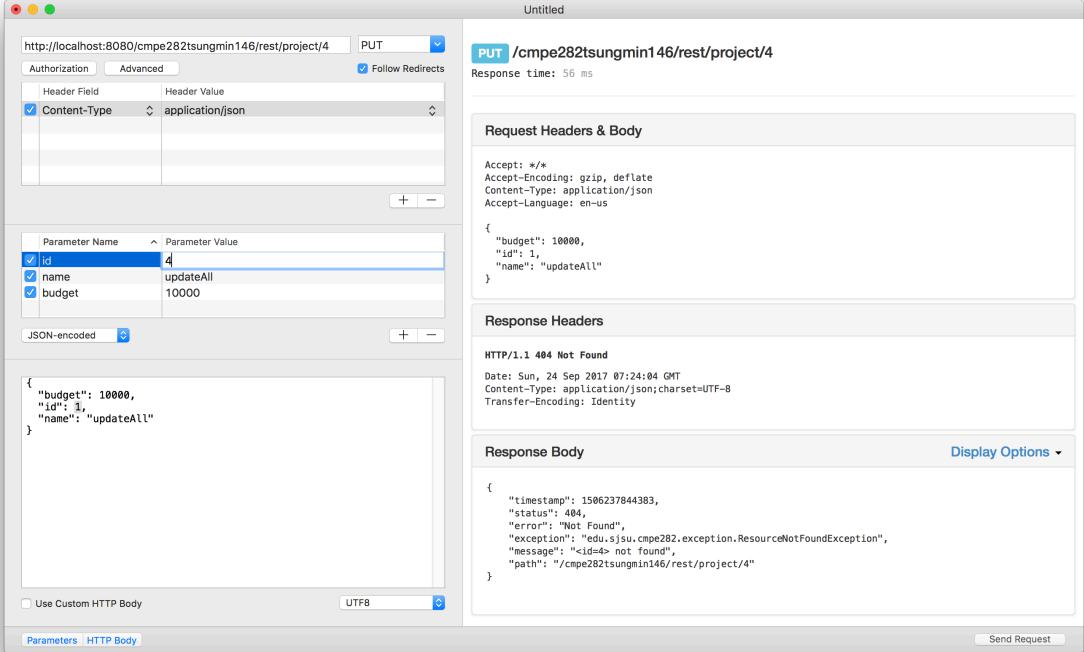
{
 "budget": 10000,
 "id": 1,
 "name": "updateAll"
}

{
 "id": 1,
 "name": "updateAll",
 "budget": 10000
}

MongoDB

```
> show dbs
admin 0.000GB
local 0.000GB
test 0.000GB
> use cmpe282tsungmin146
switched to db cmpe282tsungmin146
> db.project.find()
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demo", "budget" : 0 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demoPartialUpdate", "budget" : 0 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "demoPartialUpdate", "budget" : 10 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "updateAll", "budget" : 10000 }
```

Error: Project Id=4 does not exist error, project in JSON Id=1 does not match request URL

	Screen Shot
MongoDB	 <pre> > show dbs admin 0.000GB local 0.000GB test 0.000GB > use cmpe282tsungmin146 switched to db cmpe282tsungmin146 > db.project.find() > db.project.find({"id":1}) { "_id" : ObjectId("59c753922b54f8214d396145"), "._class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "demo", "budget" : 0 } > db.project.find({}) { "_id" : ObjectId("59c753922b54f8214d396145"), "._class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "demo", "budget" : 0 } > db.project.find {"id":1} { "_id" : ObjectId("59c753922b54f8214d396145"), "._class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "demoPartialUpdate", "budget" : 0 } > db.project.find {"id":1} { "_id" : ObjectId("59c753922b54f8214d396145"), "._class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "demoPartialUpdate", "budget" : 10 } > db.project.find {"id":1} { "_id" : ObjectId("59c753922b54f8214d396145"), "._class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "updateAll", "budget" : 10000 } > </pre>
Id not exist	 <p>Request Headers & Body</p> <pre> Accept: */* Accept-Encoding: gzip, deflate Content-Type: application/json Accept-Language: en-us { "budget": 10000, "id": 1, "name": "updateAll" } </pre> <p>Response Headers</p> <pre> HTTP/1.1 404 Not Found Date: Sun, 24 Sep 2017 07:24:04 GMT Content-Type: application/json; charset=UTF-8 Transfer-Encoding: Identity </pre> <p>Response Body</p> <pre> { "timestamp": 1506237844383, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id=> not found", "path": "/cmpe282tsungmin146/rest/project/4" } </pre>

MongoDB

```
1. mongo
> db.project.find({"id":4})
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "updateAll", "budget" : 10000 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "updateAll", "budget" : 10000 }
>
```

URL and
JSON id
not match

http://localhost:8080/cmpe282tsungmin146/rest/project/1 PUT

Header Field Header Value

Content-Type application/json

Parameter Name	Parameter Value
<input checked="" type="checkbox"/> id	4
<input checked="" type="checkbox"/> name	notMatchId
<input checked="" type="checkbox"/> budget	10000

Request Headers & Body

Accept: */*
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: en-us

{
 "budget": 10000,
 "id": 4,
 "name": "notMatchId"
}

Response Headers

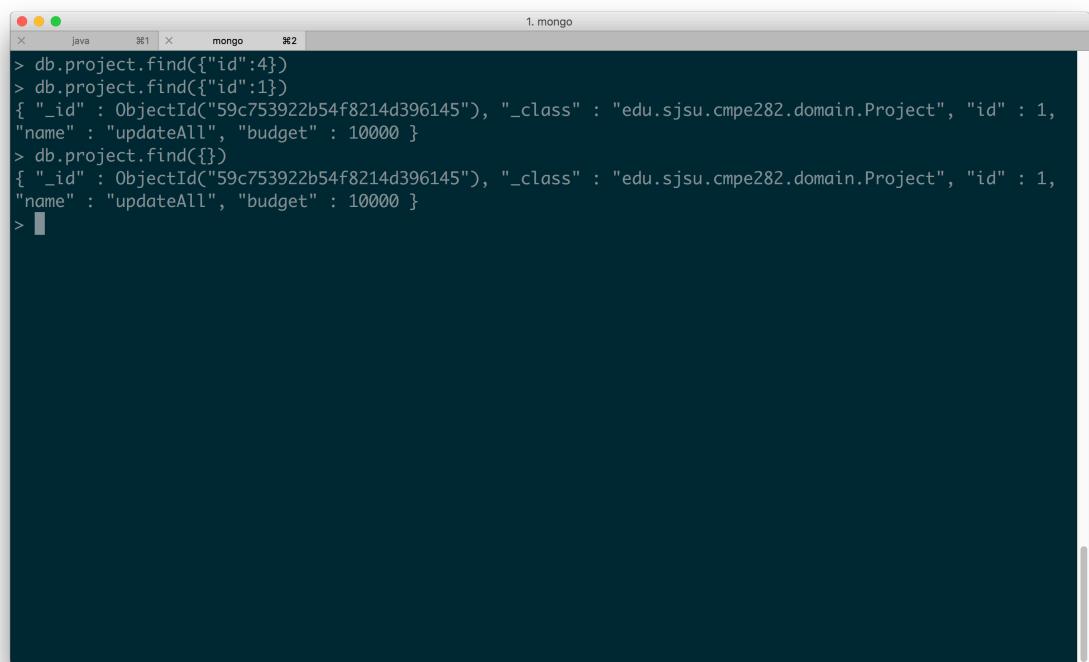
HTTP/1.1 404 Not Found

Date: Sun, 24 Sep 2017 07:34:28 GMT
Content-Type: application/json; charset=UTF-8
Transfer-Encoding: Identity

Response Body

{
 "timestamp": 1506238460180,
 "status": 404,
 "error": "Not Found",
 "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException",
 "message": "REQUEST: <id=1> not match with JSON: <id=4>",
 "path": "/cmpe282tsungmin146/rest/project/1"
}

MongoDB

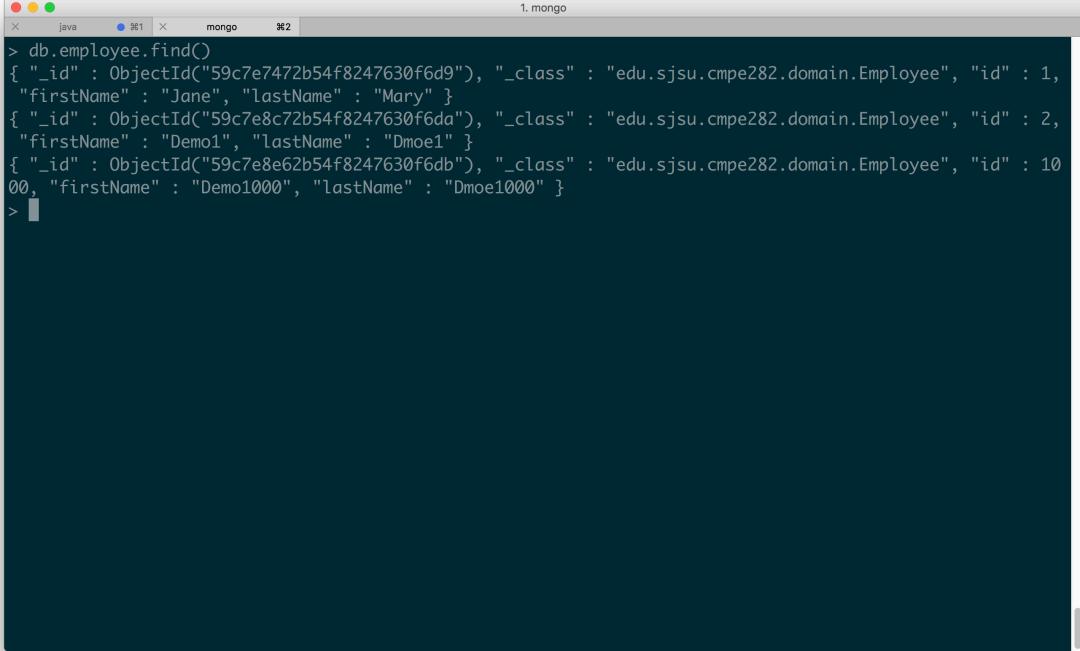


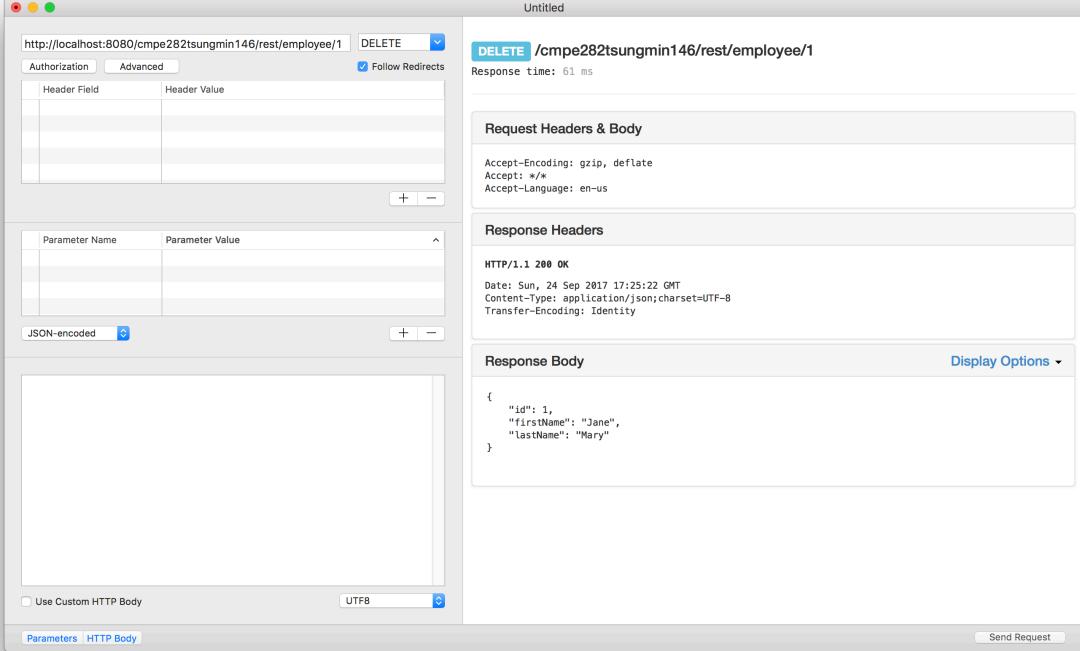
A screenshot of a terminal window titled "mongo". The window contains the following MongoDB shell session:

```
> db.project.find({"id":4})
> db.project.find({"id":1})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
"name" : "updateAll", "budget" : 10000 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
"name" : "updateAll", "budget" : 10000 }
>
```

DELETE /.../rest/employee/n

Success: Delete with Id=1

MongoDB	Screen Shot
	 <pre>> db.employee.find() { "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "Jane", "lastName" : "Mary" } { "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2, "firstName" : "Demo1", "lastName" : "Dmoe1" } { "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000, "firstName" : "Demo1000", "lastName" : "Dmoe1000" } ></pre>

Delete	Screen Shot
	 <p>HTTP Request:</p> <pre>DELETE /cmpe282tsungmin146/rest/employee/1</pre> <p>Request Headers & Body</p> <pre>Accept-Encoding: gzip, deflate Accept: */* Accept-Language: en-us</pre> <p>Response Headers</p> <pre>HTTP/1.1 200 OK Date: Sun, 24 Sep 2017 17:25:22 GMT Content-Type: application/json; charset=UTF-8 Transfer-Encoding: Identity</pre> <p>Response Body</p> <pre>{ "id": 1, "firstName": "Jane", "lastName": "Mary" }</pre>

MongoDB

```
1. mongo
> db.employee.find()
{ "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "Jane", "lastName" : "Mary" }
{ "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2,
  "firstName" : "Demo1", "lastName" : "Dmoe1" }
{ "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000,
  "firstName" : "Demo1000", "lastName" : "Dmoe1000" }
> db.employee.find()
{ "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2,
  "firstName" : "Demo1", "lastName" : "Dmoe1" }
{ "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000,
  "firstName" : "Demo1000", "lastName" : "Dmoe1000" }
>
```

Error: Delete with non exist Id=9

Screen Shot

MongoDB

```
1. mongo
> db.employee.find()
{ "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "Jane", "lastName" : "Mary" }
{ "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2,
  "firstName" : "Demo1", "lastName" : "Dmoe1" }
{ "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000,
  "firstName" : "Demo1000", "lastName" : "Dmoe1000" }
> db.employee.find()
{ "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2,
  "firstName" : "Demo1", "lastName" : "Dmoe1" }
{ "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000,
  "firstName" : "Demo1000", "lastName" : "Dmoe1000" }
>
```

No
records

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://localhost:8080/cmpe282tsungmin146/rest/employee/9`
- Method:** `DELETE`
- Headers:** `Accept-Encoding: gzip, deflate`, `Accept: */*`, `Accept-Language: en-us`
- Response Headers:**
 - `HTTP/1.1 404 Not Found`
 - `Date: Sun, 24 Sep 2017 17:31:51 GMT`
 - `Content-Type: application/json; charset=UTF-8`
 - `Transfer-Encoding: Identity`
- Response Body:**

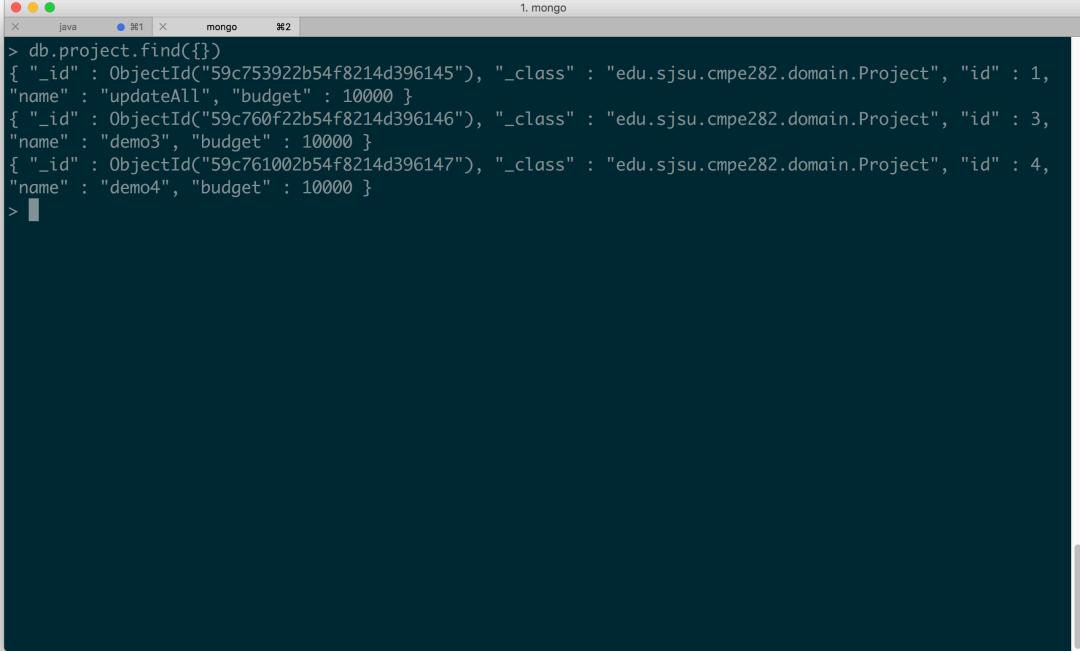
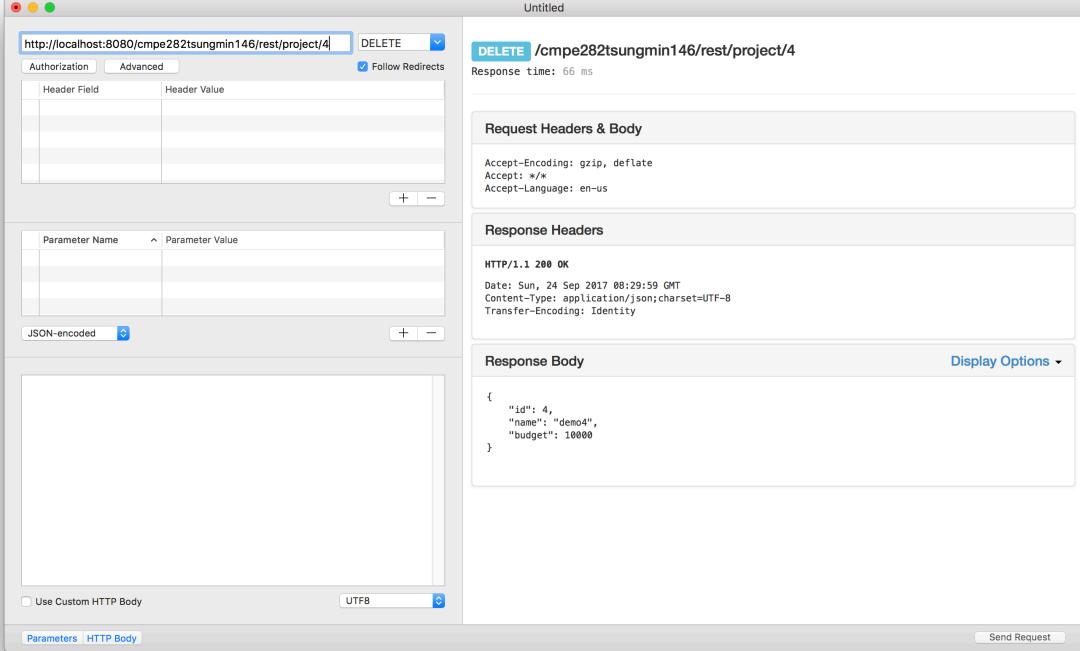
```
{  "timestamp": 1506274311263,  "status": 404,  "error": "Not Found",  "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException",  "message": "<id=9> not found",  "path": "/cmpe282tsungmin146/rest/employee/9"}
```

MongoDB

```
> db.employee.find()
{ "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "Jane", "lastName" : "Mary" }
{ "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2, "firstName" : "Demo1", "lastName" : "Dmoe1" }
{ "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000, "firstName" : "Demo1000", "lastName" : "Dmoe1000" }
> db.employee.find()
{ "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2, "firstName" : "Demo1", "lastName" : "Dmoe1" }
{ "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000, "firstName" : "Demo1000", "lastName" : "Dmoe1000" }
> db.employee.find()
{ "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2, "firstName" : "Demo1", "lastName" : "Dmoe1" }
{ "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1000, "firstName" : "Demo1000", "lastName" : "Dmoe1000" }
```

DELETE /.../rest/project/n

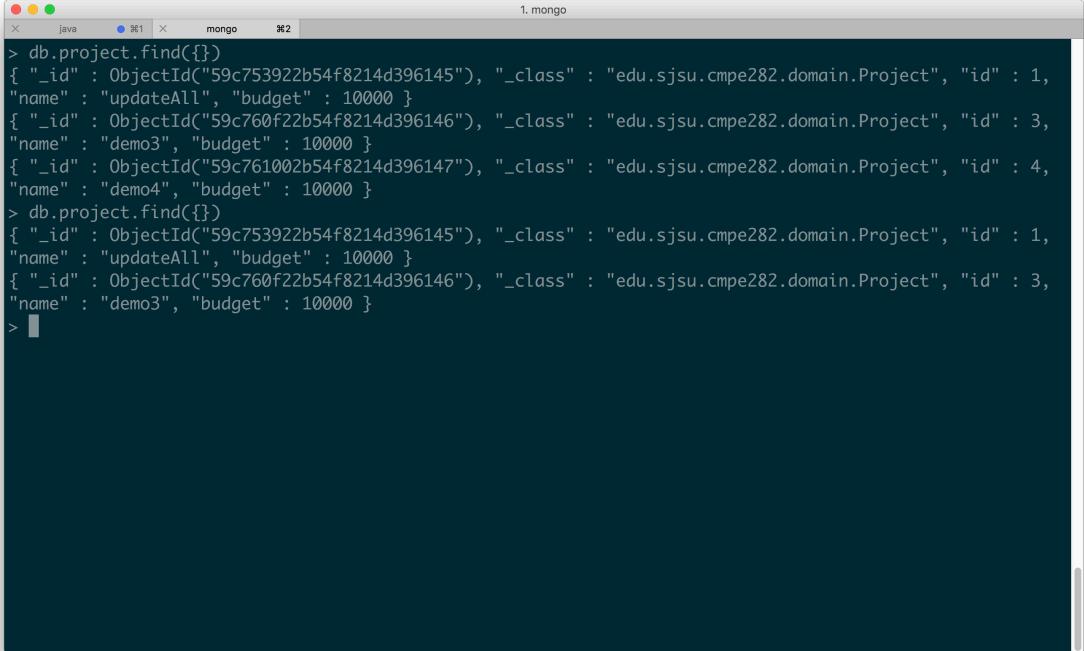
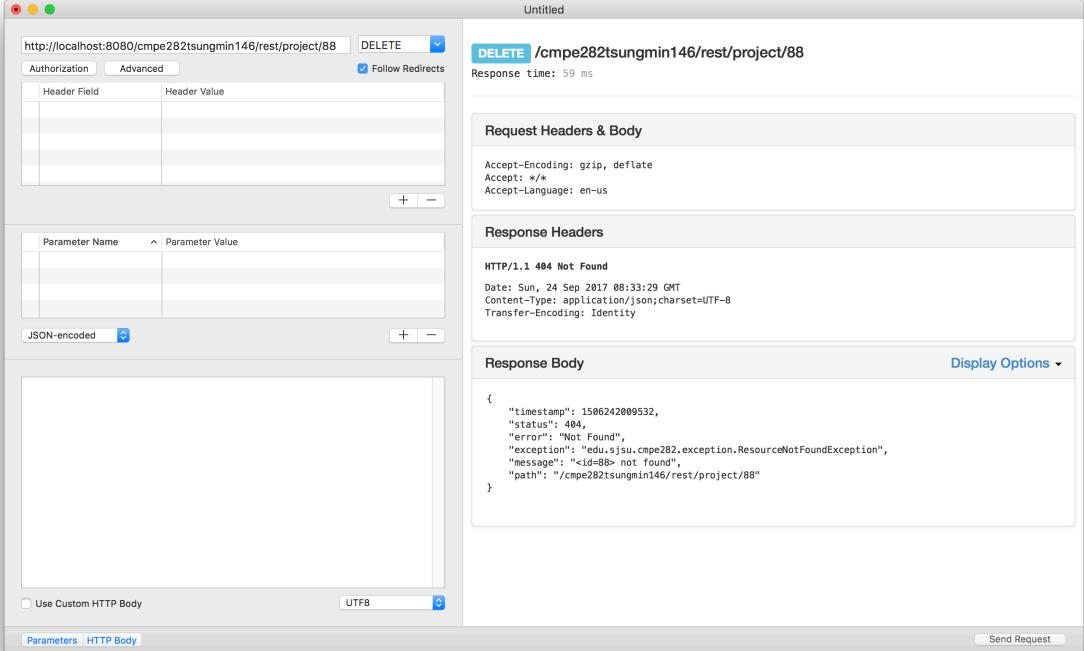
Success: Delete with Id=4

	Screen Shot
MongoDB	 A screenshot of a Mac OS X terminal window titled "mongo". It contains the following MongoDB shell session: <pre>> db.project.find({}) { "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 } { "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 } { "_id" : ObjectId("59c761002b54f8214d396147"), "name" : "demo4", "budget" : 10000 } ></pre> The terminal has three tabs open: "java", "mongo", and another tab whose title is partially visible.
Delete	 A screenshot of a REST client interface titled "Untitled". The URL field shows "http://localhost:8080/cmpe282tsungmin146/rest/project/4". The method dropdown shows "DELETE". The "Request Headers & Body" section shows the following headers: <pre>Accept-Encoding: gzip, deflate Accept: */* Accept-Language: en-us</pre> The "Response Headers" section shows: <pre>HTTP/1.1 200 OK Date: Sun, 24 Sep 2017 08:29:59 GMT Content-Type: application/json;charset=UTF-8 Transfer-Encoding: Identity</pre> The "Response Body" section displays the JSON object: <pre>{ "id": 4, "name": "demo4", "budget": 10000 }</pre>

MongoDB

```
1. mongo
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 }
{ "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 }
{ "_id" : ObjectId("59c761002b54f8214d396147"), "name" : "demo4", "budget" : 10000 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 }
{ "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 }
>
```

Error: Delete with non exist Id=88

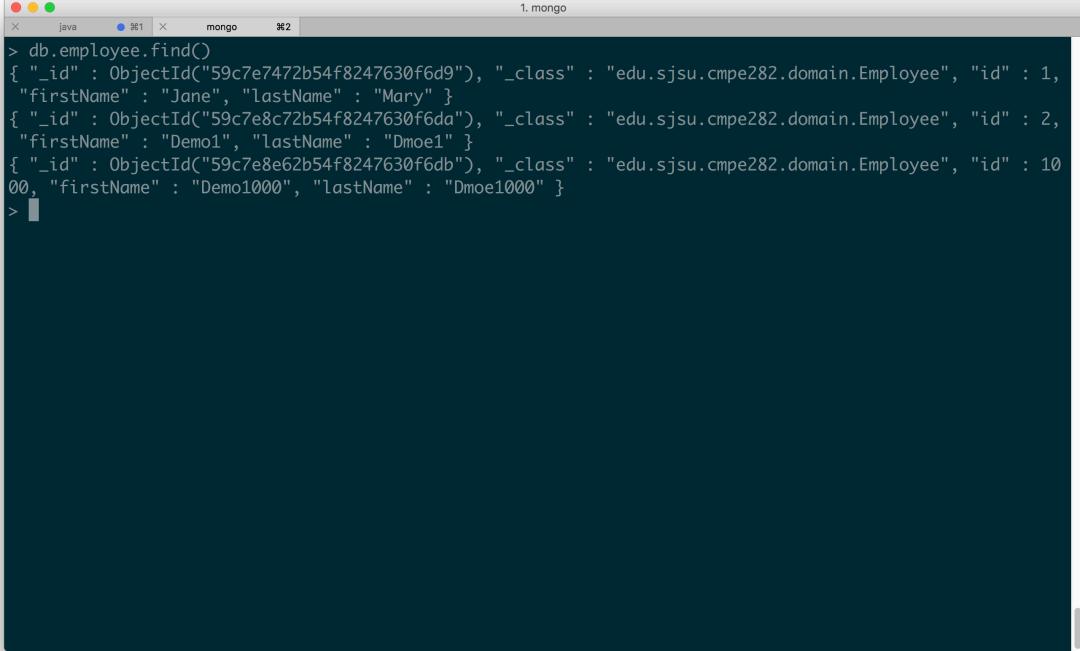
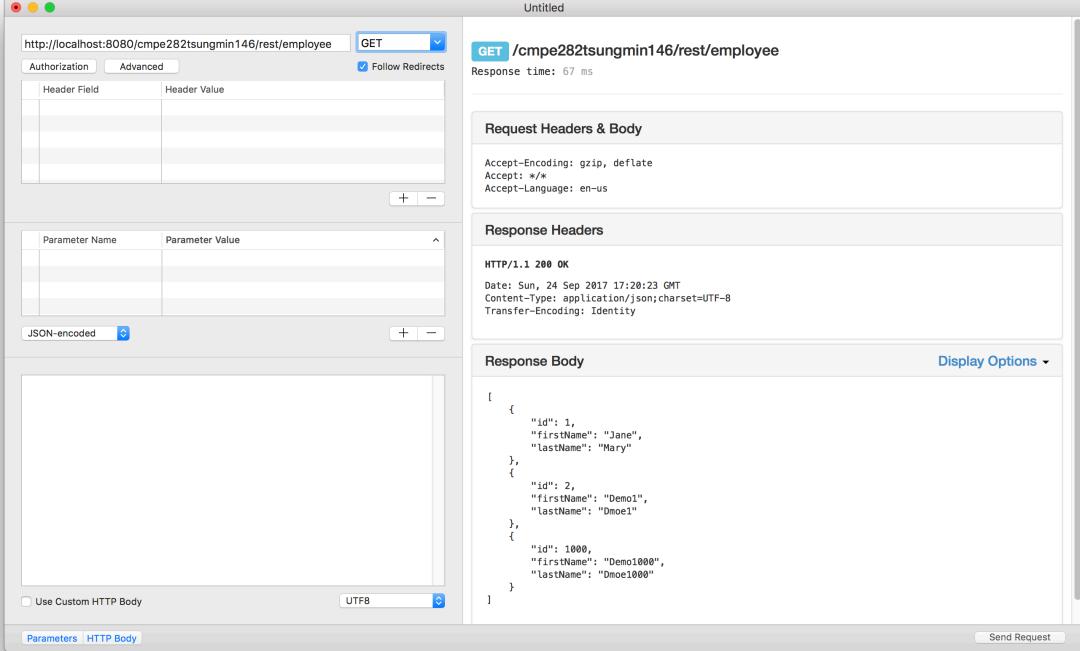
	<p style="text-align: center;">Screen Shot</p>  <p>A screenshot of a Mac OS X terminal window titled "mongo". It contains the following MongoDB shell session:</p> <pre>> db.project.find({}) { "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 } { "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 } { "_id" : ObjectId("59c761002b54f8214d396147"), "name" : "demo4", "budget" : 10000 } > db.project.find({}) { "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 } { "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 } ></pre>
MongoDB	
No records	 <p>A screenshot of a REST client interface. The URL is <code>http://localhost:8080/cmpe282tsungmin146/rest/project/88</code>. The response shows a 404 Not Found error with the following JSON body:</p> <pre>{ "timestamp": 1506242009532, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id=88> not found", "path": "/cmpe282tsungmin146/rest/project/88" }</pre>

MongoDB

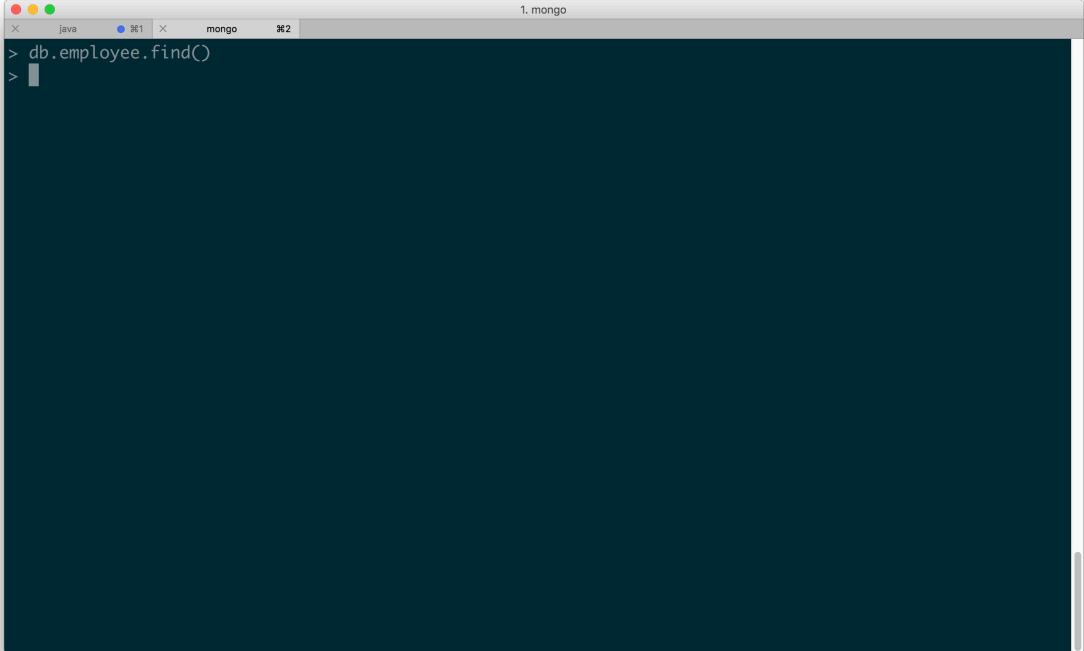
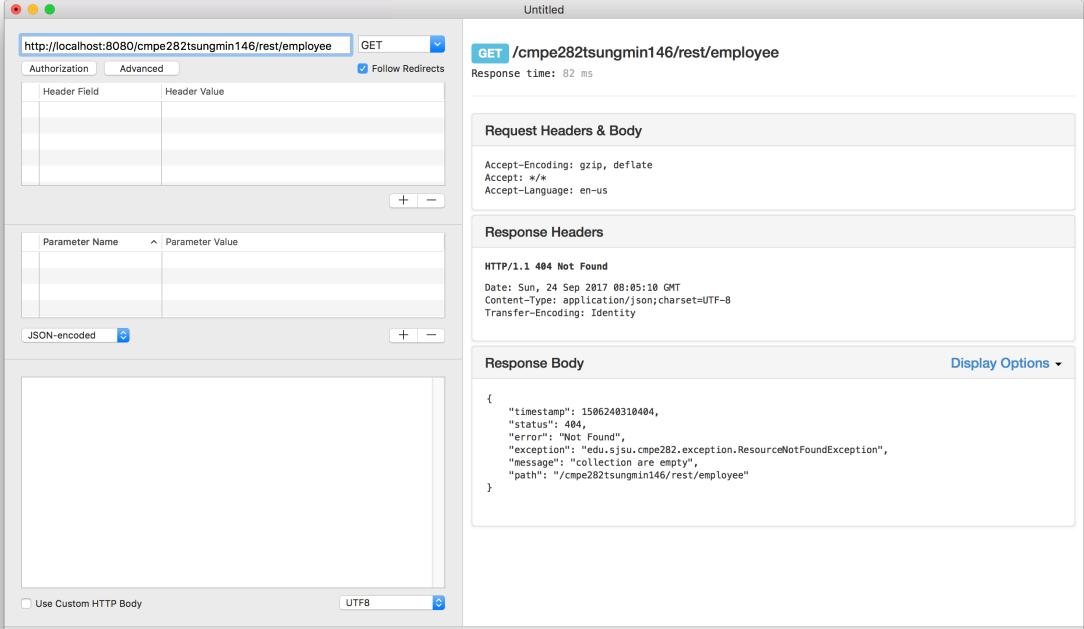
```
1. mongo
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 }
{ "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 }
{ "_id" : ObjectId("59c761002b54f8214d396147"), "name" : "demo4", "budget" : 10000 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 }
{ "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 }
> db.project.find({})
{ "_id" : ObjectId("59c753922b54f8214d396145"), "name" : "updateAll", "budget" : 10000 }
{ "_id" : ObjectId("59c760f22b54f8214d396146"), "name" : "demo3", "budget" : 10000 }
>
```

GET /.../rest/employee

Success: Get all employee

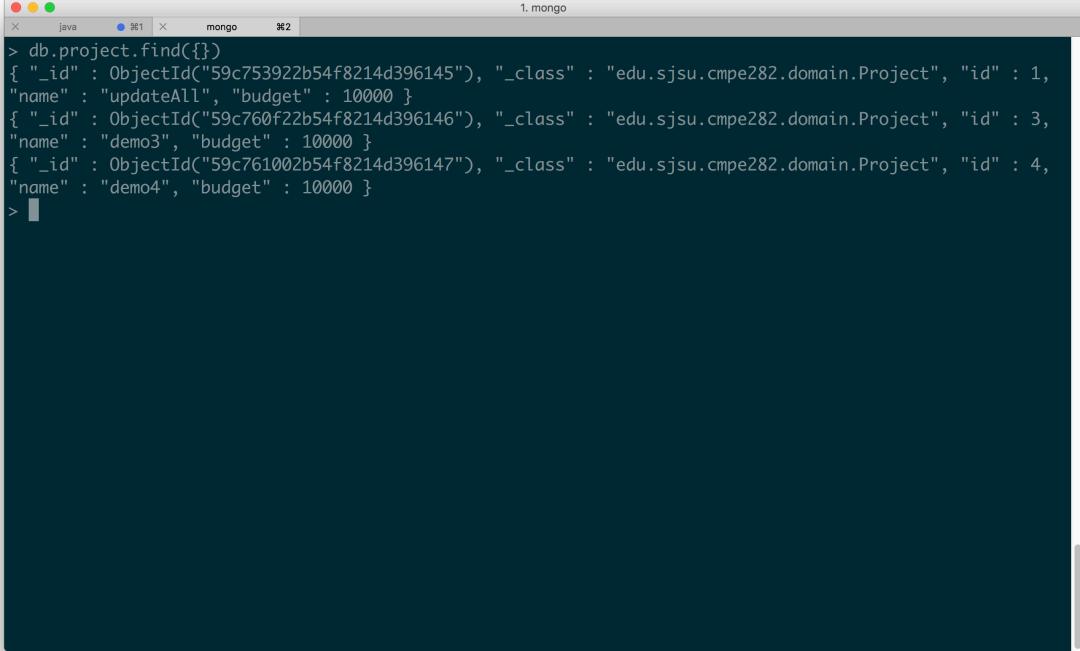
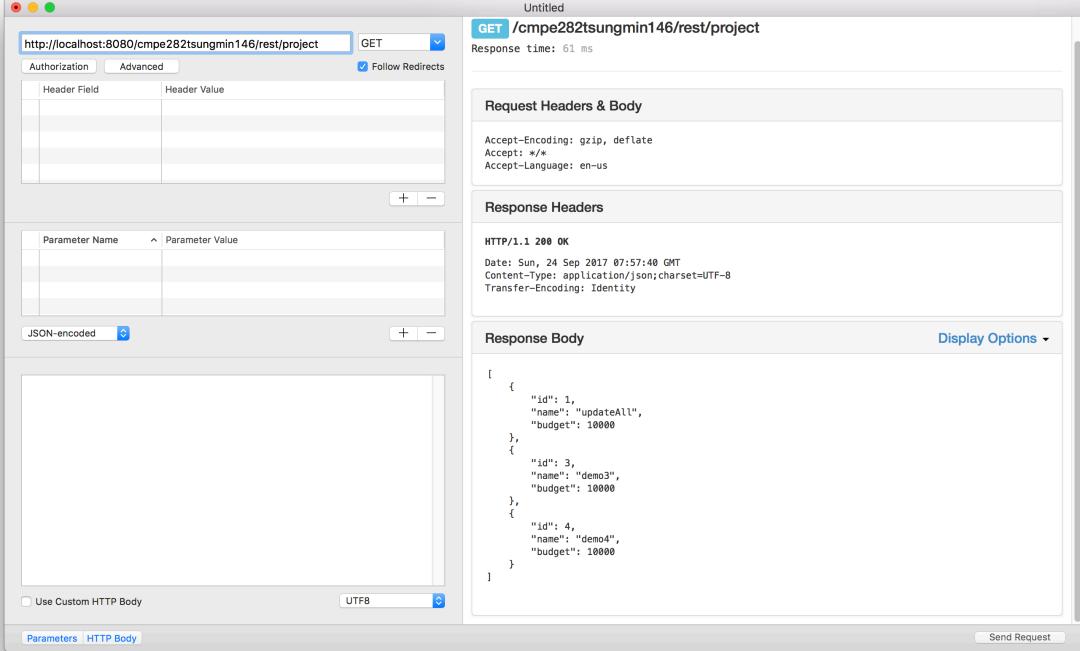
MongoDB	<p>Screen Shot</p>  <pre>> db.employee.find() { "_id" : ObjectId("59c7e7472b54f8247630f6d9"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "Jane", "lastName" : "Mary" } { "_id" : ObjectId("59c7e8c72b54f8247630f6da"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 2, "firstName" : "Demo1", "lastName" : "Dmoe1" } { "_id" : ObjectId("59c7e8e62b54f8247630f6db"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 100, "firstName" : "Demo1000", "lastName" : "Dmoe1000" } ></pre>
No records	 <p>Untitled</p> <p>http://localhost:8080/cmpe282tsungmin146/rest/employee GET</p> <p>Request Headers & Body</p> <p>Response Headers</p> <p>Response Body</p> <pre>[{"id": 1, "firstName": "Jane", "lastName": "Mary"}, {"id": 2, "firstName": "Demo1", "lastName": "Dmoe1"}, {"id": 100, "firstName": "Demo1000", "lastName": "Dmoe1000"}]</pre>

Error: Not collection data

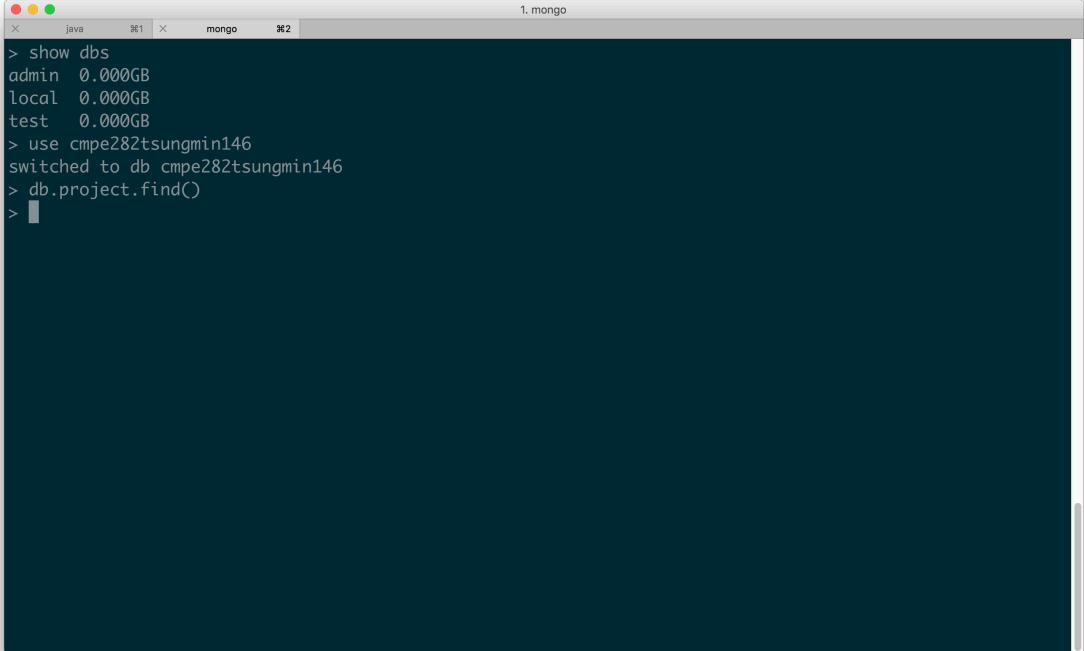
MongoDB	<p style="text-align: center;">Screen Shot</p>  <p>A screenshot of a Mac OS X terminal window titled "mongo". It contains the command "> db.employee.find()" followed by a blank line, indicating no results were found.</p>
No records	 <p>A screenshot of a REST client interface. The request URL is "http://localhost:8080/cmpe282tsungmin146/rest/employee". The response status is "HTTP/1.1 404 Not Found". The response body contains a JSON object:</p> <pre>{ "timestamp": 1506240310404, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "collection are empty", "path": "/cmpe282tsungmin146/rest/employee"}</pre>

GET /.../rest/project

Success: Get all project

	Screen Shot
MongoDB	 <p>The screenshot shows the MongoDB shell interface. A query is run: <code>> db.project.find({})</code>. The results are displayed as an array of four documents:</p> <pre>> db.project.find({}) [{"_id": ObjectId("59c753922b54f8214d396145"), "name": "updateAll", "budget": 10000}, {"_id": ObjectId("59c760f22b54f8214d396146"), "name": "demo3", "budget": 10000}, {"_id": ObjectId("59c761002b54f8214d396147"), "name": "demo4", "budget": 10000}]</pre>
No records	 <p>The screenshot shows a REST client interface with the following details:</p> <ul style="list-style-type: none">URL: <code>http://localhost:8080/cmpe282tsungmin146/rest/project</code>Method: GETRequest Headers & Body:<ul style="list-style-type: none">Accept-Encoding: gzip, deflateAccept: */*Accept-Language: en-usResponse Headers:<ul style="list-style-type: none">HTTP/1.1 200 OKDate: Sun, 24 Sep 2017 07:57:48 GMTContent-Type: application/json; charset=UTF-8Transfer-Encoding: IdentityResponse Body:<pre>[{"id": 1, "name": "updateAll", "budget": 10000}, {"id": 3, "name": "demo3", "budget": 10000}, {"id": 4, "name": "demo4", "budget": 10000}]</pre>

Error: Not collection data

MongoDB	<p>Screen Shot</p> 
No records	