

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

No need. Spring automatically generate DB with assigned name. Just make sure MongoDB daemon is running.

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 and JSON content. 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.

The field check in JSON content does not proper implement to check which field is optional or mandatory.

2. Code redundancy

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

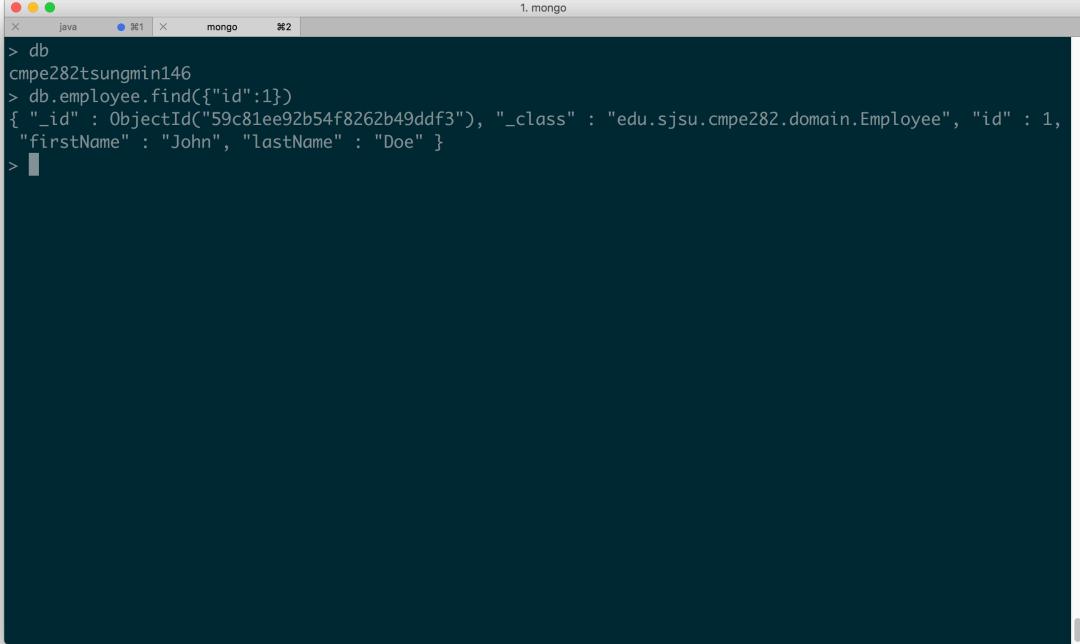
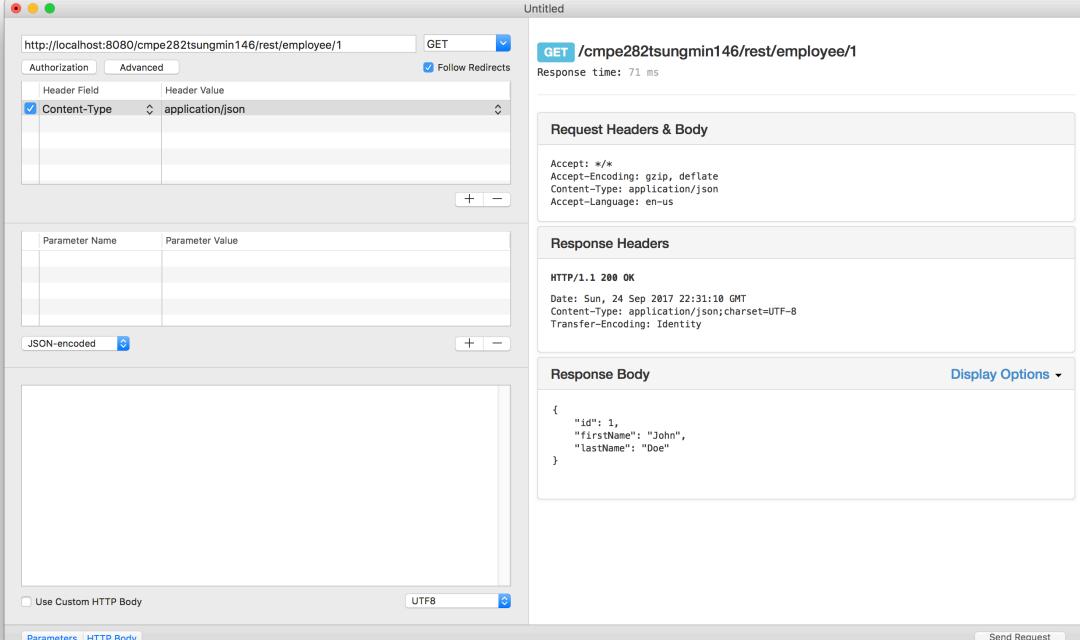
PUT API for employee and project

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

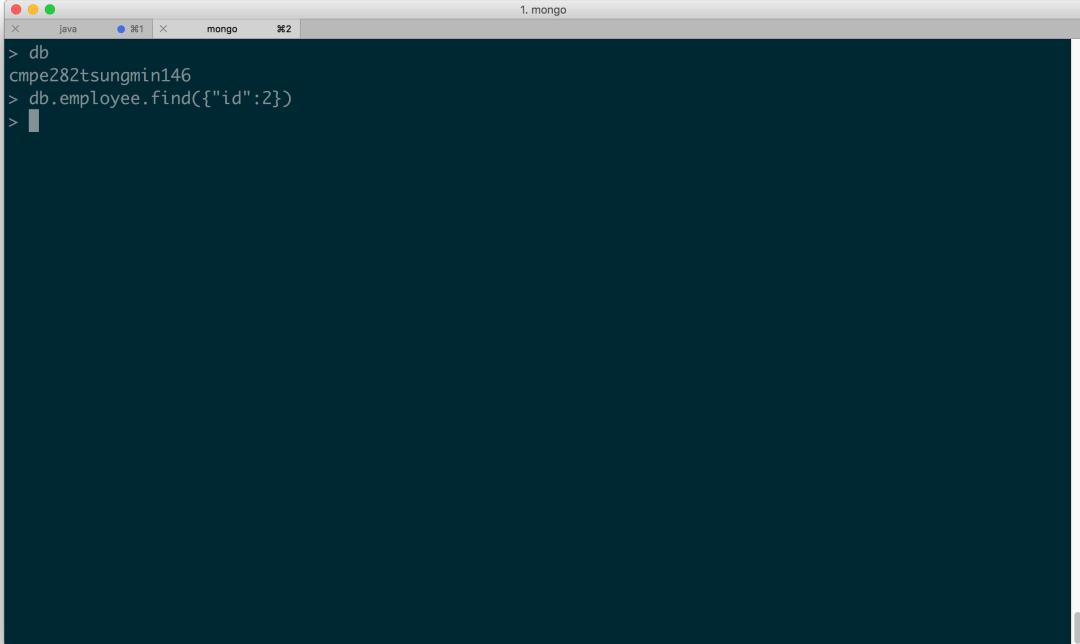
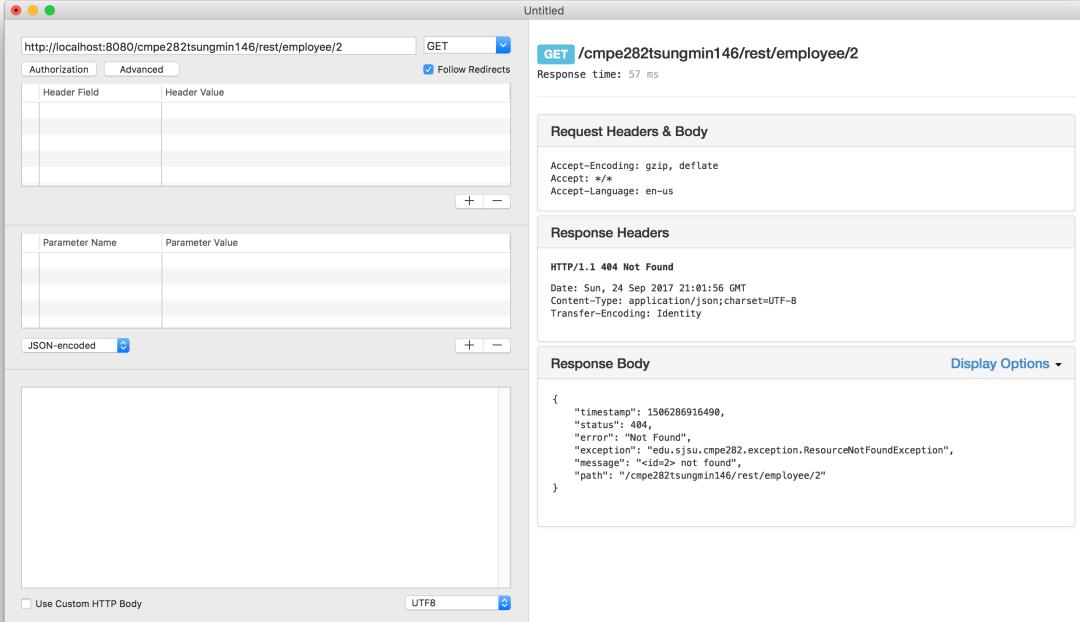
The part of business code should be encapsulated to achieve code reuse.

GET /.../rest/employee/m

Success: Get employee Id=1

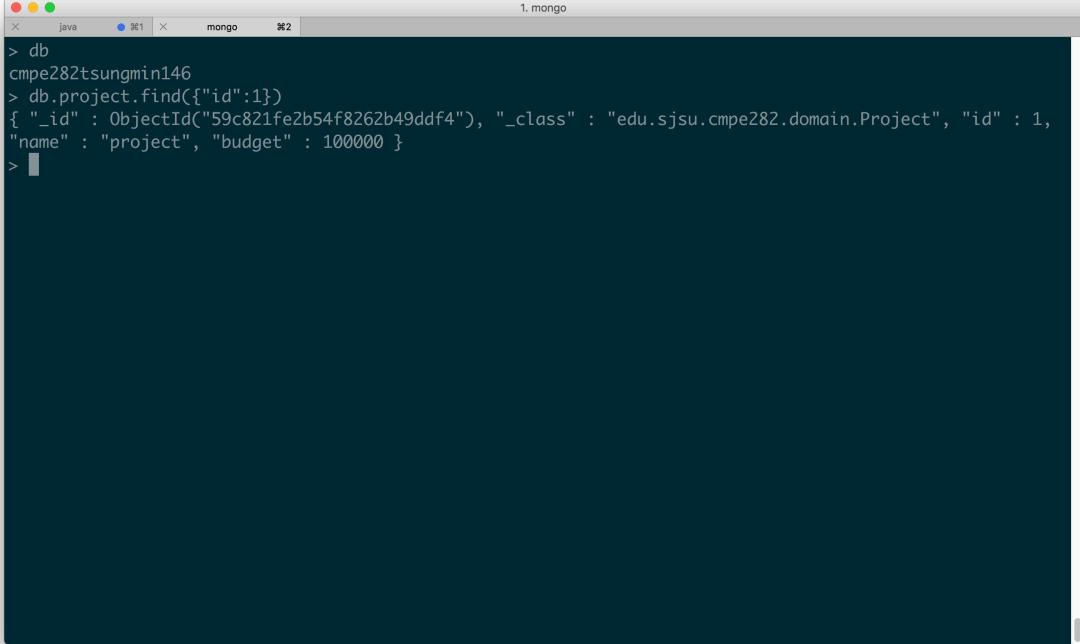
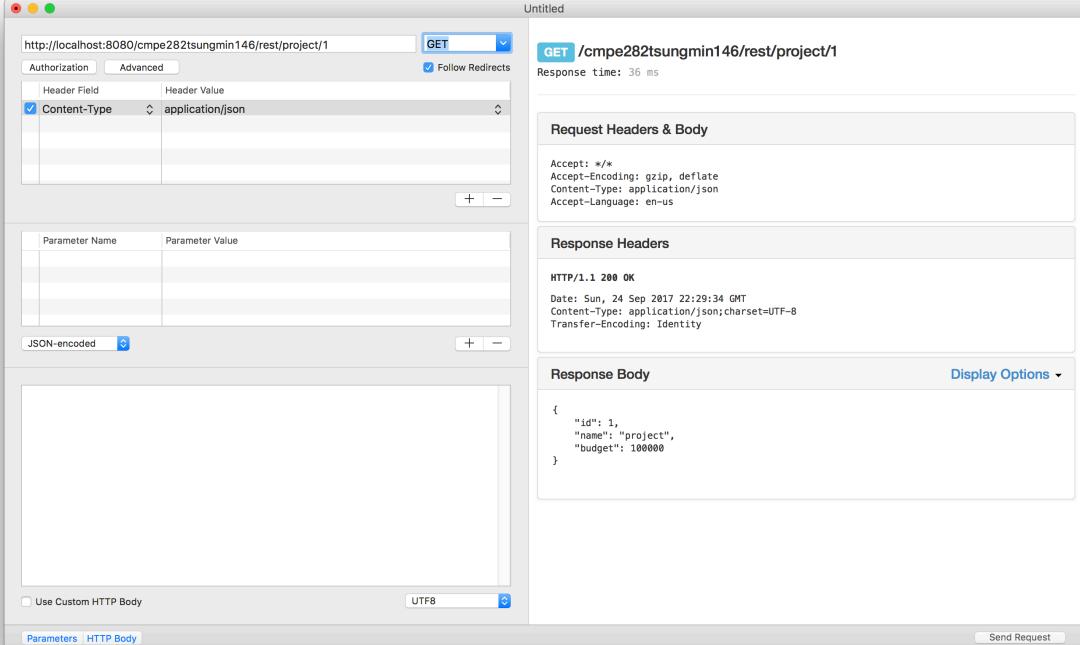
	Screen Shot
MongoDB	 A screenshot of the MongoDB shell window titled "mongo". It shows the command `db.employee.find({id:1})` and its result, which is a document with fields _id, _class, id, firstName, and lastName.
Client	 A screenshot of a REST client interface. The URL is set to `http://localhost:8080/cmpe282tsungmin146/rest/employee/1` and the method is `GET`. The response status is `200 OK` with a response time of 71 ms. The response body contains the same JSON document as the MongoDB output, representing an employee with ID 1, first name John, and last name Doe.

Error: Get employee Id=2

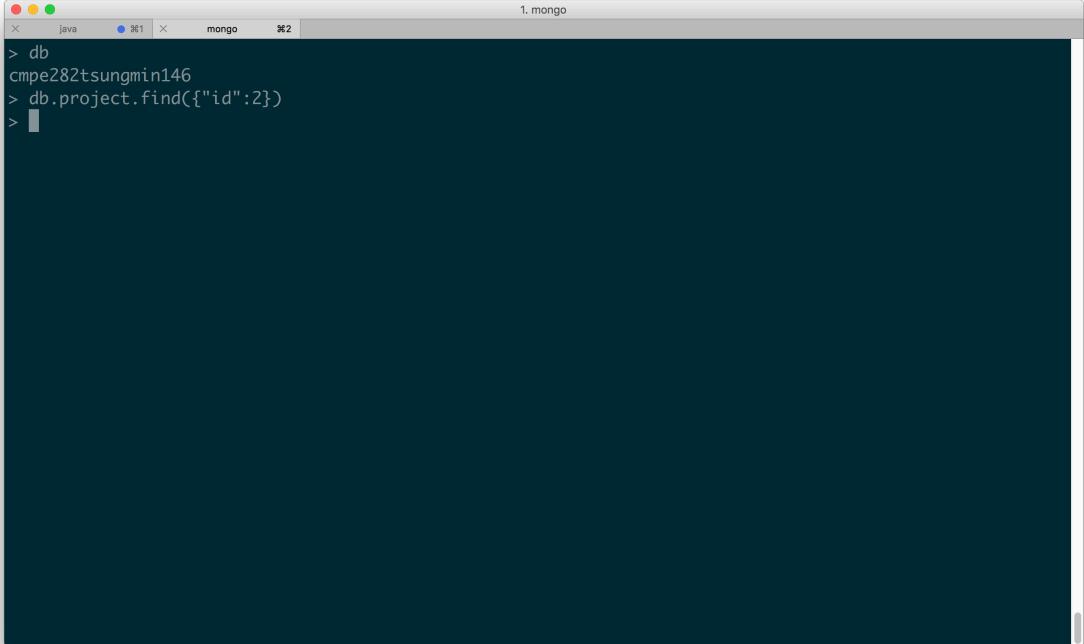
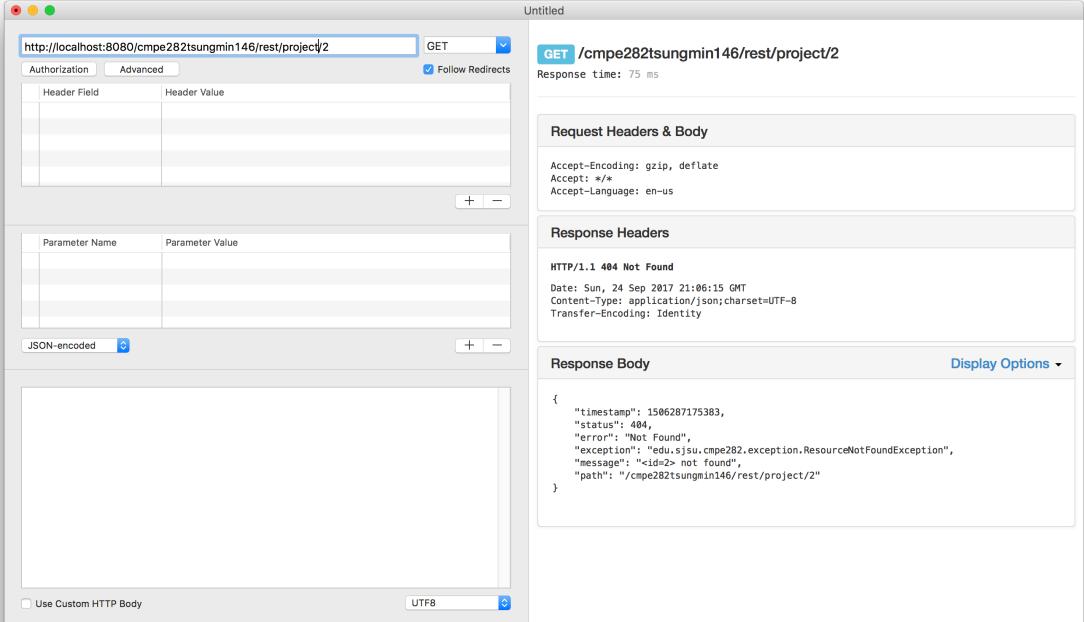
	<h3>Screen Shot</h3>  <pre>1. mongo > db cmpe282tsungmin146 > db.employee.find({"id":2}) > </pre>
MongoDB	 <p>The screenshot shows a cURL interface with the following details:</p> <ul style="list-style-type: none">Method: GETURL: <code>http://localhost:8080/cmpe282tsungmin146/rest/employee/2</code>Headers:<ul style="list-style-type: none">AuthorizationAdvanced<input checked="" type="checkbox"/> Follow RedirectsRequest Headers & Body:<ul style="list-style-type: none">Accept-Encoding: gzip, deflateAccept: */*Accept-Language: en-usResponse Headers:<ul style="list-style-type: none">HTTP/1.1 404 Not FoundDate: Sun, 24 Sep 2017 21:01:56 GMTContent-Type: application/json;charset=UTF-8Transfer-Encoding: IdentityResponse Body:<pre>{ "timestamp": 1506286916490, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "id=2 not found", "path": "/cmpe282tsungmin146/rest/employee/2" }</pre>

GET /.../rest/project/n

Success: Get project Id=1

	Screen Shot
MongoDB	 A screenshot of a Mac OS X terminal window titled "mongo". The terminal shows a mongo shell session with the command `db.project.find({"id":1})` and its response, which is a single document: { "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "project", "budget" : 100000 }.
Client	 A screenshot of a REST client application window titled "Untitled". The request URL is "http://localhost:8080/cmpe282tsungmin146/rest/project/1" and the method is "GET". The response status is "200 OK" and the response time is "36 ms". The "Request Headers & Body" section shows the headers: Accept: */*, Accept-Encoding: gzip, deflate, Content-Type: application/json, Accept-Language: en-us. The "Response Headers" section shows the headers: HTTP/1.1 200 OK, Date: Sun, 24 Sep 2017 22:29:34 GMT, Content-Type: application/json; charset=UTF-8, Transfer-Encoding: Identity. The "Response Body" section displays the JSON response: { "id": 1, "name": "project", "budget": 100000 }.

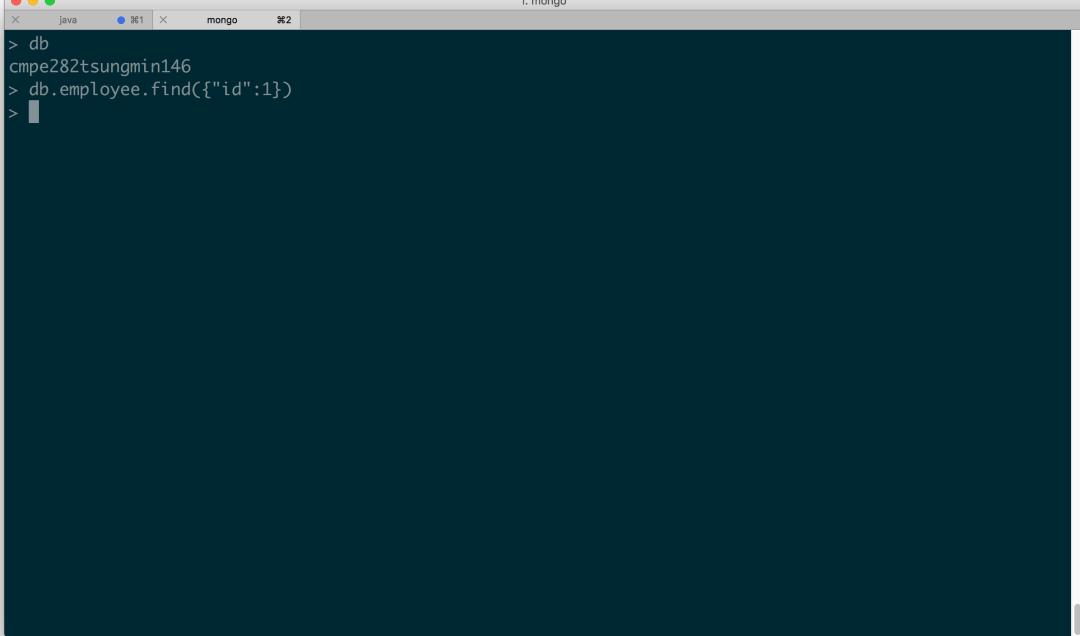
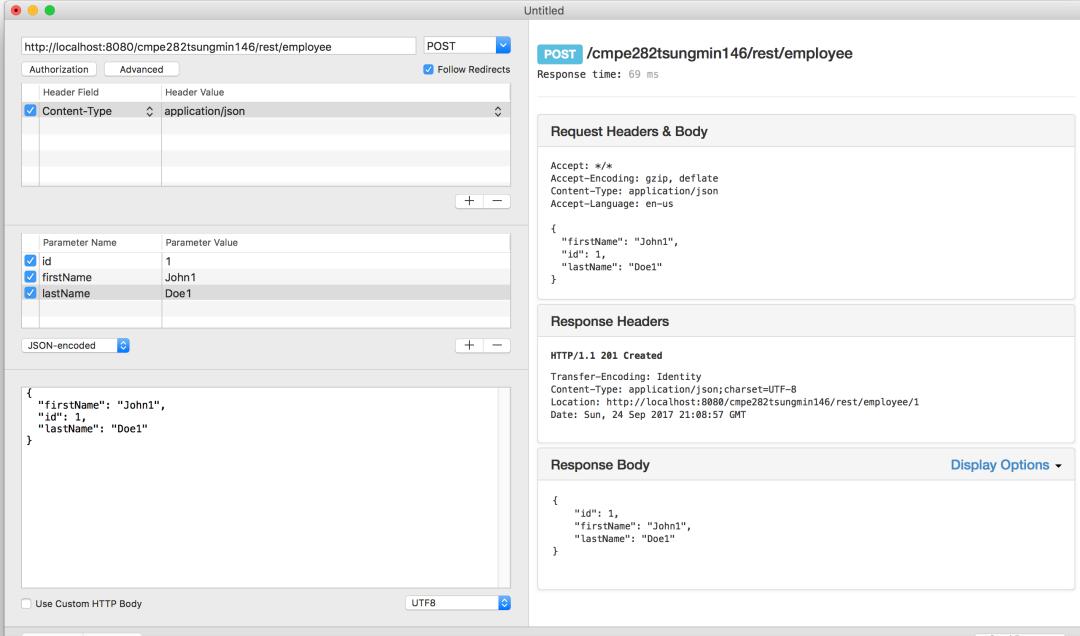
Error: Get project Id=2

	<h3>Screen Shot</h3>  <p>A screenshot of a Mac OS X terminal window titled "mongo". It contains the following MongoDB shell commands:</p> <pre>> db cmpe282tsungmin146 > db.project.find({"id":2}) > </pre>
MongoDB	 <p>A screenshot of a Charles proxy tool interface. The "Request" tab shows a GET request to "http://localhost:8080/cmpe282tsungmin146/rest/project/2". The "Response" tab displays the following details:</p> <p>Request Headers & Body</p> <pre>Accept-Encoding: gzip, deflate Accept: */* Accept-Language: en-us</pre> <p>Response Headers</p> <pre>HTTP/1.1 404 Not Found Date: Sun, 24 Sep 2017 21:06:15 GMT Content-Type: application/json;charset=UTF-8 Transfer-Encoding: Identity</pre> <p>Response Body</p> <pre>{ "timestamp": 1506287175383, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "id=2 not found", "path": "/cmpe282tsungmin146/rest/project/2" }</pre>

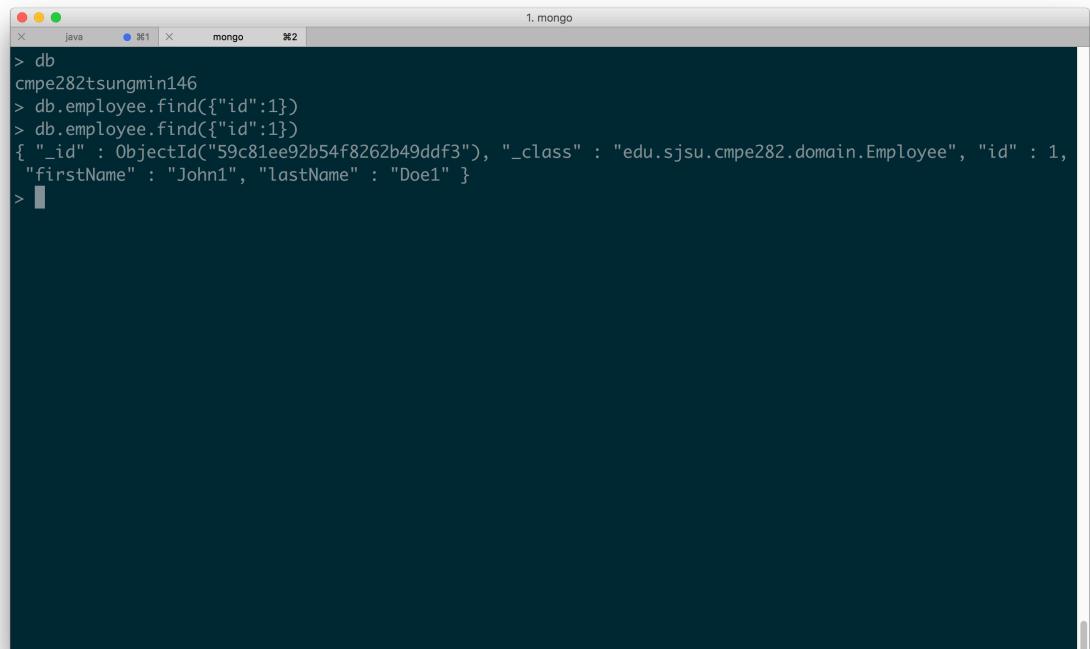
POST /.../rest/employee

Success: Create employee based JSON:

```
{"id":1,"firstName":"John1","lastName":"Doe1"}
```

	<p style="text-align: center;">Screen Shot</p>  <p>The screenshot shows a MongoDB terminal window titled 'mongo'. The command `db.employee.find({"id":1})` is run, and the response is an empty array, indicating the document was successfully created.</p>
MongoDB	
	<p style="text-align: center;">Client</p>  <p>The screenshot shows a REST client interface. The request URL is <code>http://localhost:8080/cmpe282tsungmin146/rest/employee</code>. The request method is <code>POST</code>. The request body is a JSON object with fields <code>id</code>, <code>firstName</code>, and <code>lastName</code> set to 1, John1, and Doe1 respectively. The response status is <code>HTTP/1.1 201 Created</code>. The response body contains the newly created employee document with the same fields and values.</p>
Client	

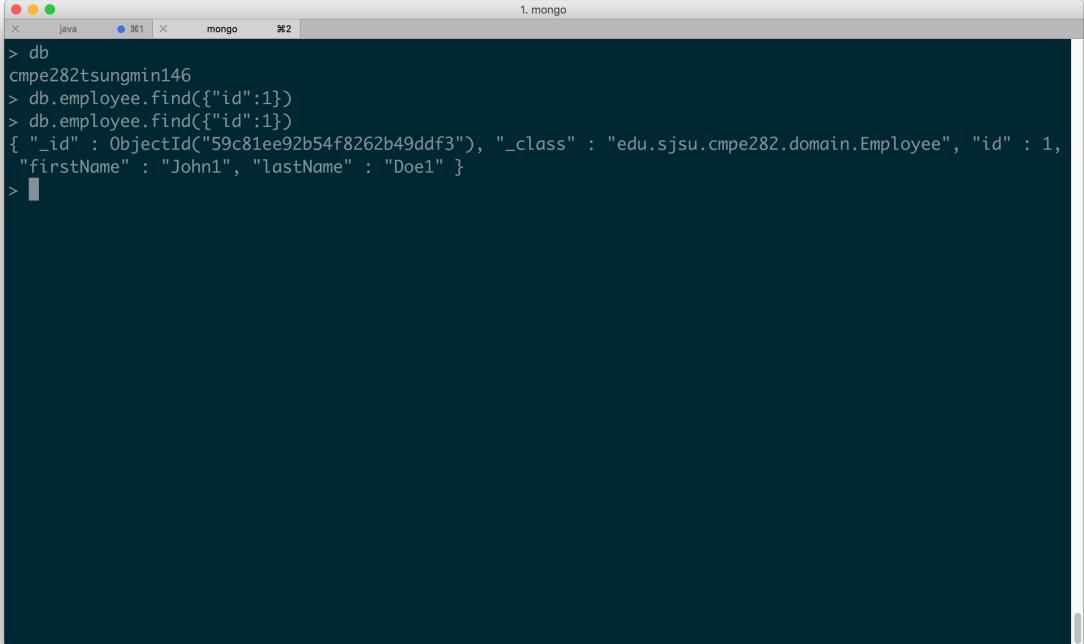
MongoDB



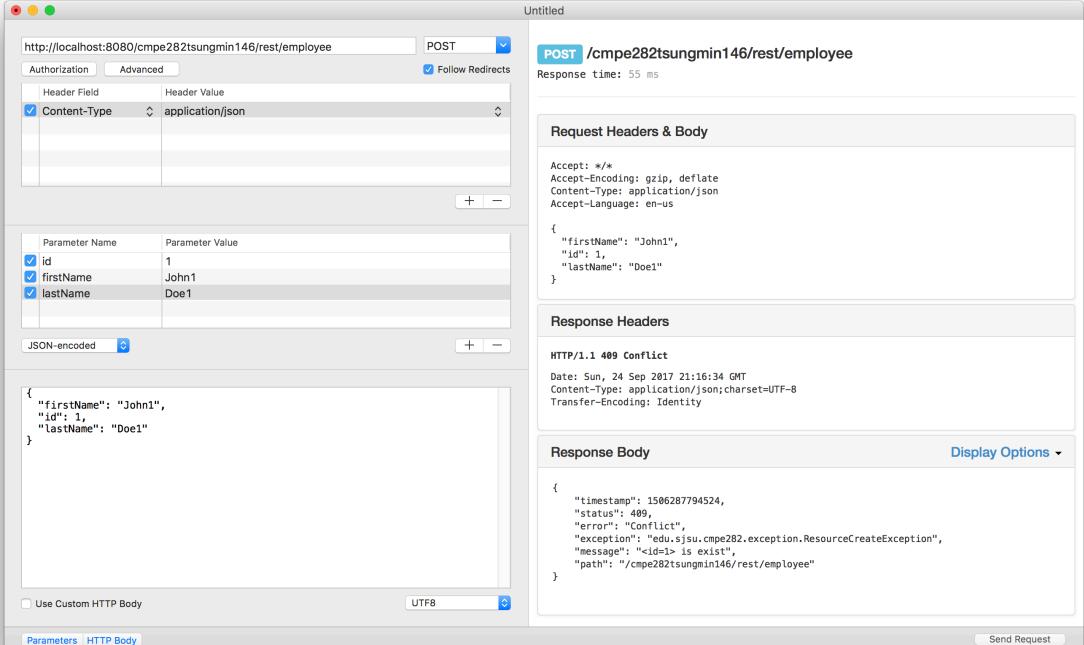
A screenshot of a terminal window titled "1. mongo". The window shows the MongoDB shell interface. The user has run the command `db.employee.find({id:1})` twice, and the output is a single document:

```
> db.cmpe282tsungmin146
> db.employee.find({"id":1})
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1", "lastName" : "Doe1" }
>
```

Error: Create employee with the same Id=1

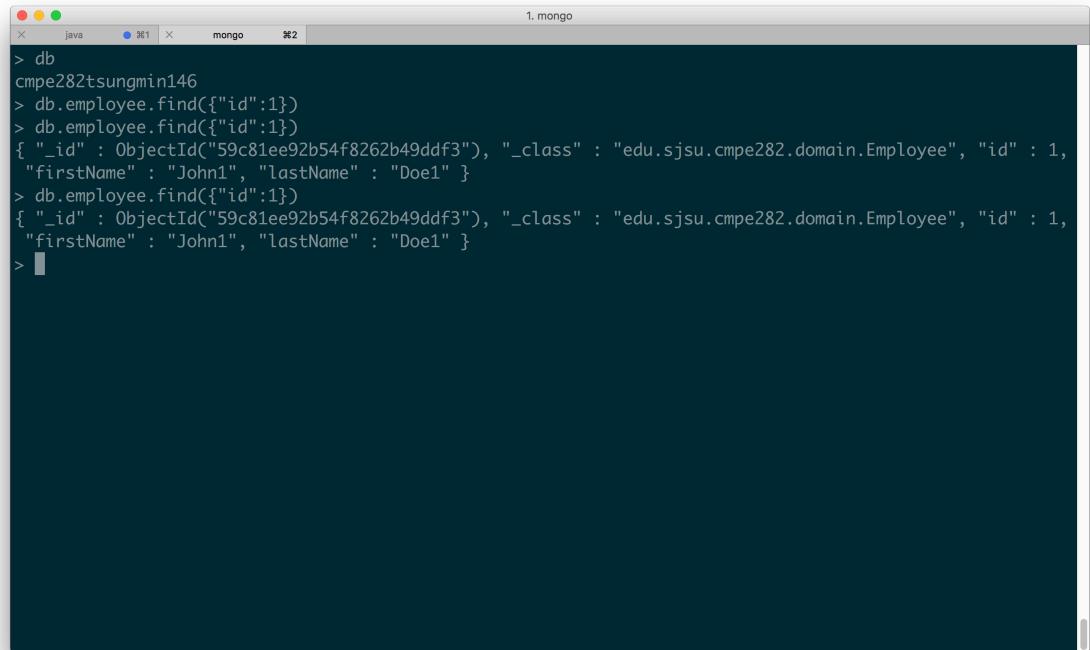
	<h3>Screen Shot</h3>  <pre>> db cmpe282tsungmin146 > db.employee.find({"id":1}) > db.employee.find({"id":1}) { "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "John1", "lastName" : "Doe1" } ></pre>
--	---

MongoDB

Client	 <p>http://localhost:8080/cmpe282tsungmin146/rest/employee</p> <p>POST</p> <p>Authorization Advanced Follow Redirects</p> <p>Header Field Header Value</p> <p>Content-Type application/json</p> <p>Parameter Name Parameter Value</p> <p><input checked="" type="checkbox"/> id 1</p> <p><input checked="" type="checkbox"/> firstName John1</p> <p><input checked="" type="checkbox"/> lastName Doe1</p> <p>Request Headers & Body</p> <p>Accept: */*</p> <p>Accept-Encoding: gzip, deflate</p> <p>Content-Type: application/json</p> <p>Accept-Language: en-us</p> <p>{ "firstName": "John1", "id": 1, "lastName": "Doe1" }</p> <p>Response Headers</p> <p>HTTP/1.1 409 Conflict</p> <p>Date: Sun, 24 Sep 2017 21:16:34 GMT</p> <p>Content-Type: application/json;charset=UTF-8</p> <p>Transfer-Encoding: Identity</p> <p>Response Body</p> <p>{ "timestamp": 1506287794524, "status": 409, "error": "Conflict", "exception": "edu.sjsu.cmpe282.exception.ResourceCreateException", "message": "<id1> is exist", "path": "/cmpe282tsungmin146/rest/employee" }</p>
--------	--

Java Application

MongoDB

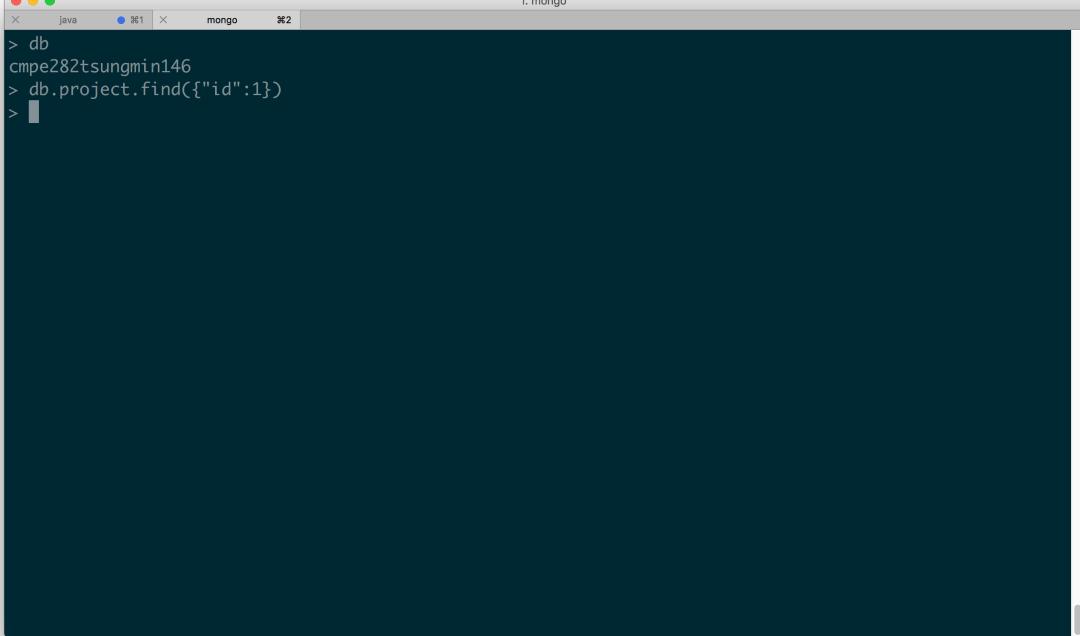
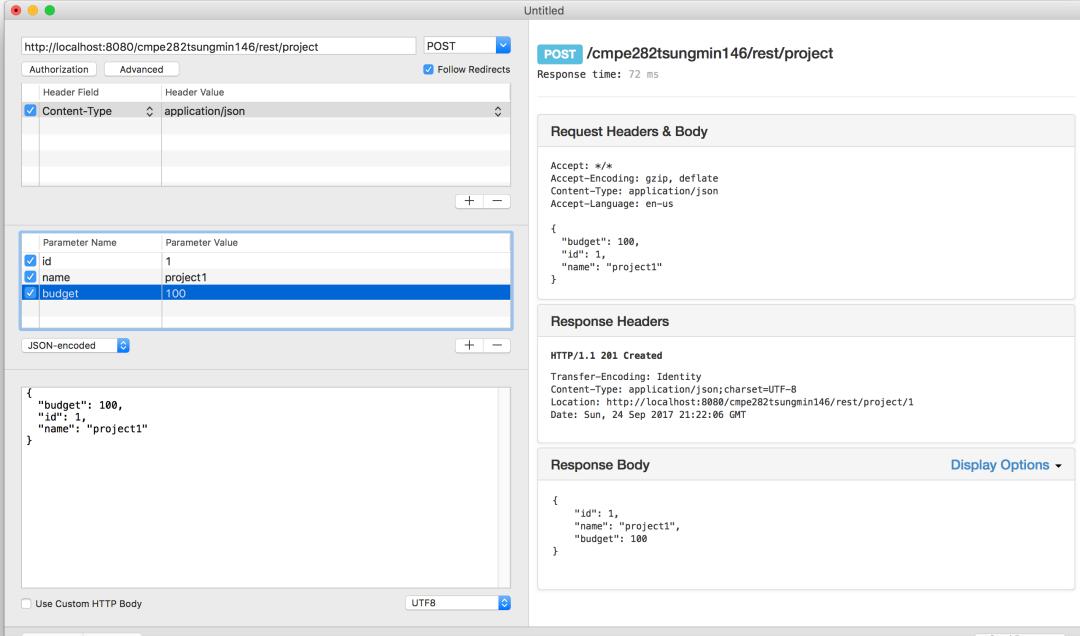


A screenshot of a terminal window titled "1. mongo". The window shows the MongoDB shell interface. The user has run the command "db.employee.find({\"id\":1})" twice, resulting in two identical document outputs. The documents have an "_id" field (an ObjectId), a "_class" field ("edu.sjsu.cmpe282.domain.Employee"), an "id" field (1), a "firstName" field ("John1"), and a "lastName" field ("Doe1").

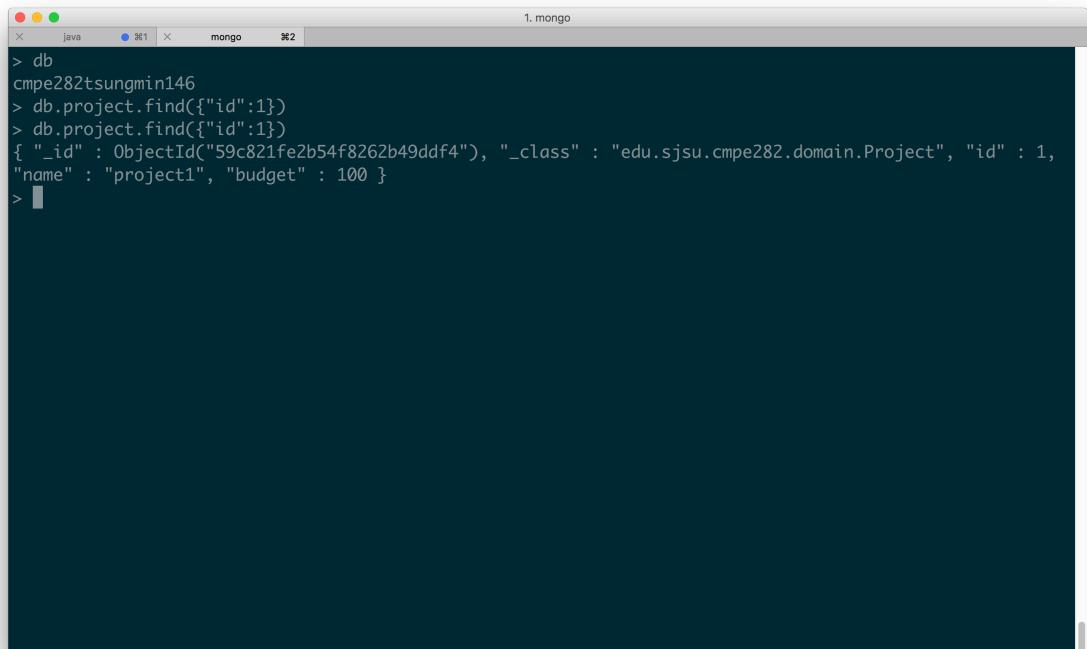
```
> db.employee.find({"id":1})
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1", "lastName" : "Doe1" }
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "_class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1", "lastName" : "Doe1" }
>
```

POST /.../rest/project

Success: Create project based JSON: {"id":1,"name":"project1", budget: 100}

	<p style="text-align: center;">Screen Shot</p>  <p>The screenshot shows a MongoDB terminal window titled 'mongo'. The command 'db.project.find({"id":1})' is run, and the response is an empty array, indicating the document was successfully created.</p>								
MongoDB									
Client	 <p>The screenshot shows a REST client interface with the following details:</p> <ul style="list-style-type: none">URL: http://localhost:8080/cmpe282tsungmin146/rest/projectMethod: POSTHeaders:<ul style="list-style-type: none">Content-Type: application/jsonRequest Body (Parameters):<table border="1"><tr><th>Parameter Name</th><th>Parameter Value</th></tr><tr><td>id</td><td>1</td></tr><tr><td>name</td><td>project1</td></tr><tr><td>budget</td><td>100</td></tr></table>Response Headers:<ul style="list-style-type: none">Accept: */*Accept-Encoding: gzip, deflateContent-Type: application/jsonAccept-Language: en-usResponse Body:<pre>{ "budget": 100, "id": 1, "name": "project1" }</pre>	Parameter Name	Parameter Value	id	1	name	project1	budget	100
Parameter Name	Parameter Value								
id	1								
name	project1								
budget	100								

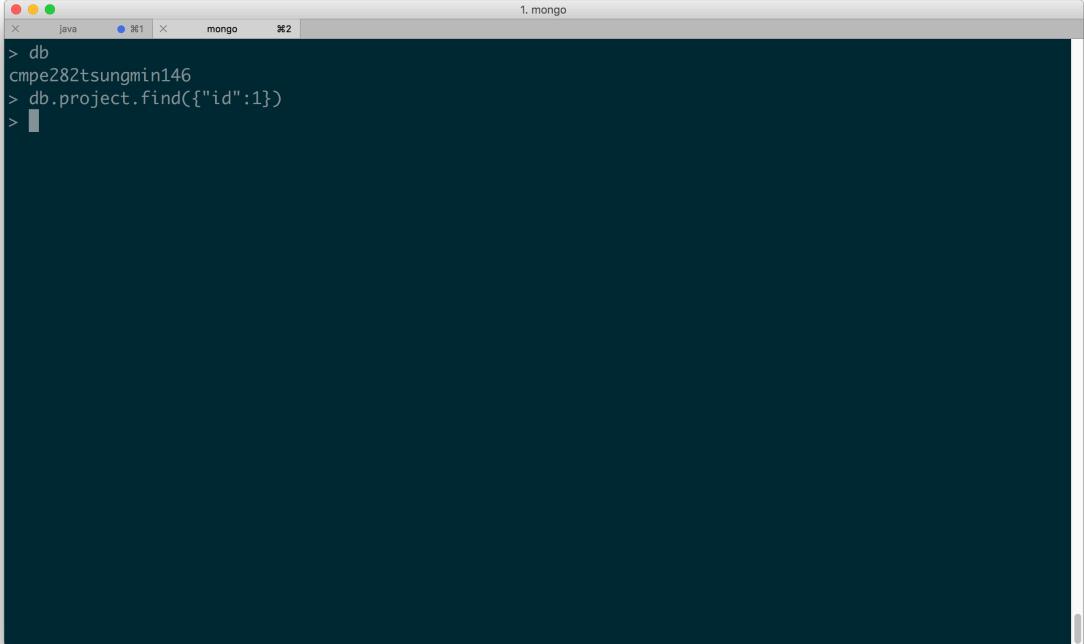
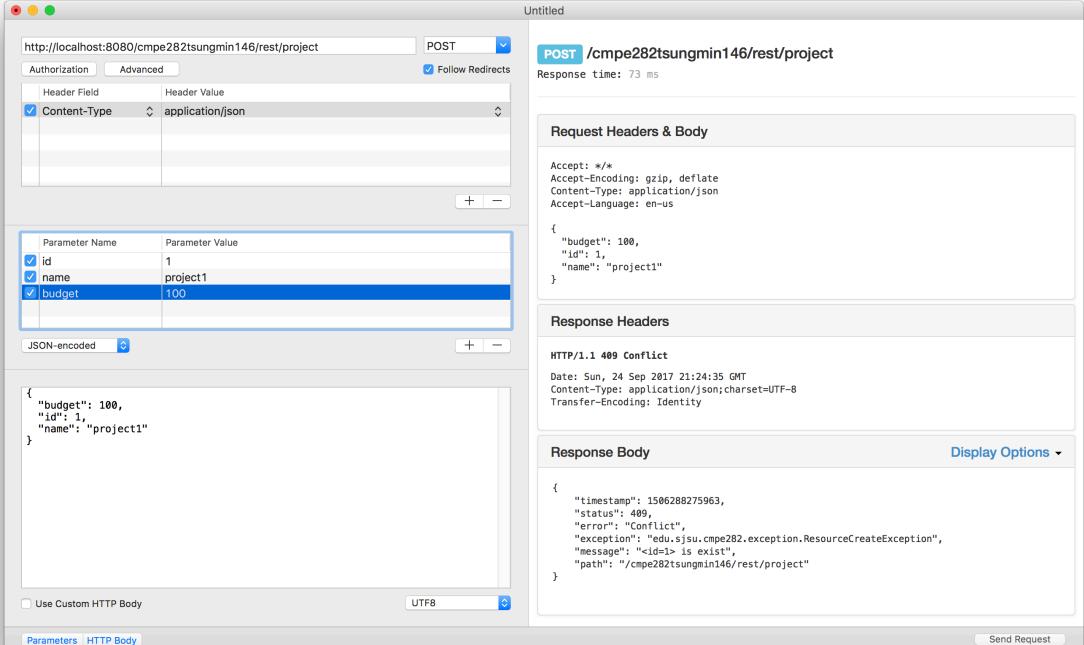
MongoDB



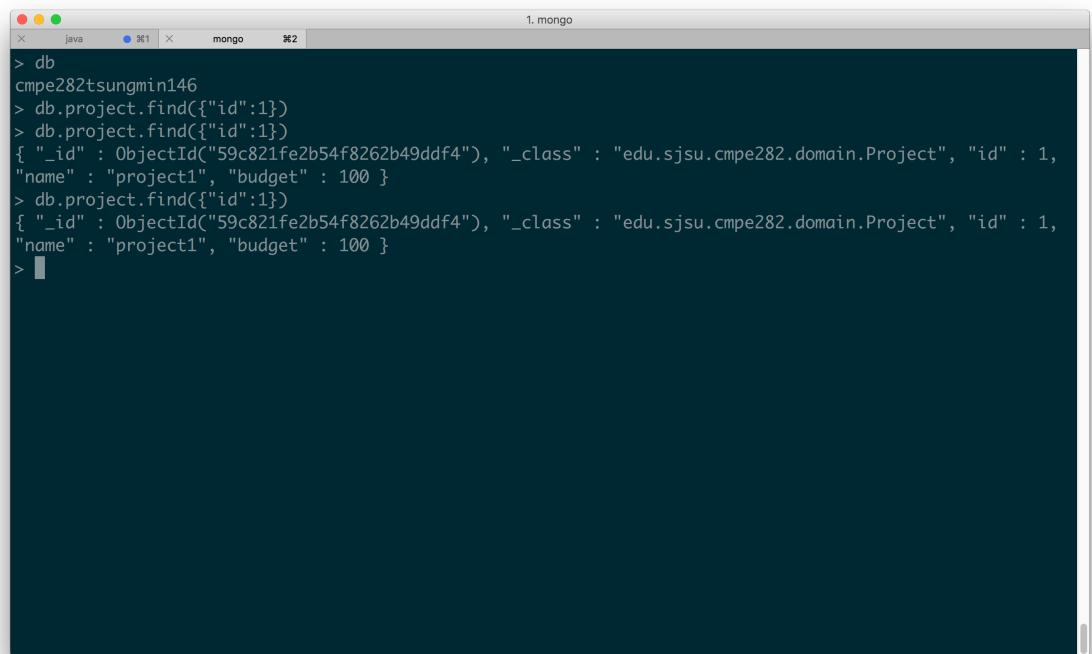
A screenshot of a terminal window titled "1. mongo". The window shows the MongoDB shell interface. The user has run the command "db.project.find({\"id\":1})" twice, and both executions return the same result: a document with "_id" as an ObjectId ("59c821fe2b54f8262b49ddf4"), "_class" as "edu.sjsu.cmpe282.domain.Project", "id" as 1, "name" as "project1", and "budget" as 100.

```
> db.project.find({"id":1})
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "project1", "budget" : 100 }
>
```

Error: Create project with the same Id=1

	<h3>Screen Shot</h3>  <pre>> db cmpe282tsungmin146 > db.project.find({"id":1}) > </pre>
MongoDB	 <p>Client</p> <p>POST /cmpe282tsungmin146/rest/project</p> <p>Request Headers & Body</p> <pre>Accept: */* Accept-Encoding: gzip, deflate Content-Type: application/json Accept-Language: en-us { "budget": 100, "id": 1, "name": "project1" }</pre> <p>Response Headers</p> <pre>HTTP/1.1 409 Conflict Date: Sun, 24 Sep 2017 21:24:35 GMT Content-Type: application/json;charset=UTF-8 Transfer-Encoding: Identity</pre> <p>Response Body</p> <pre>{ "timestamp": 1506288275963, "status": 409, "error": "Conflict", "exception": "edu.sjsu.cmpe282.exception.ResourceCreateException", "message": "<id1> is exist", "path": "/cmpe282tsungmin146/rest/project" }</pre>

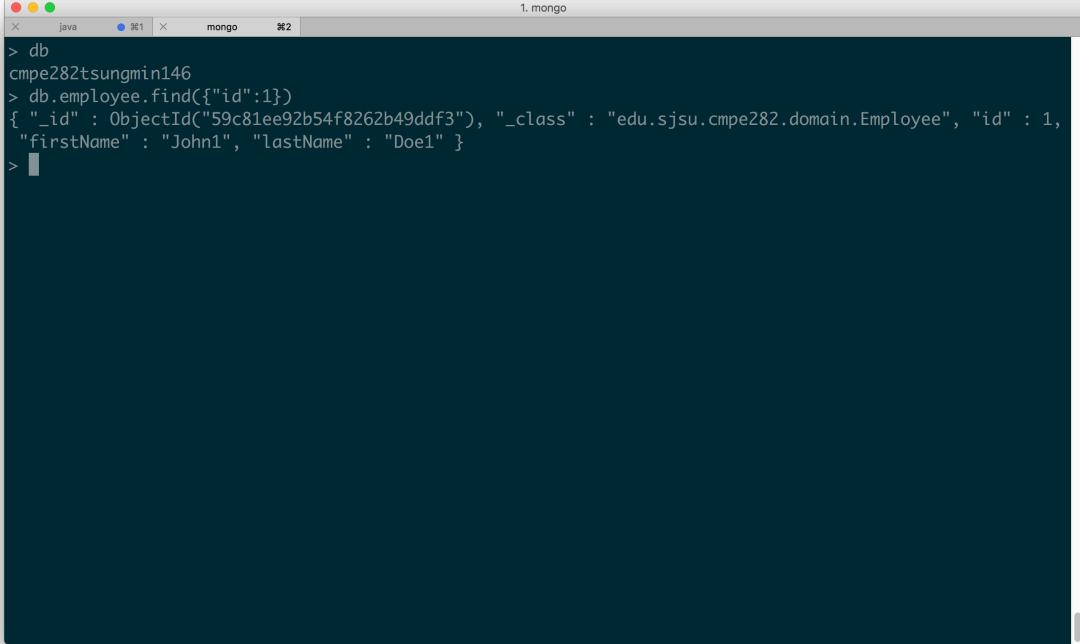
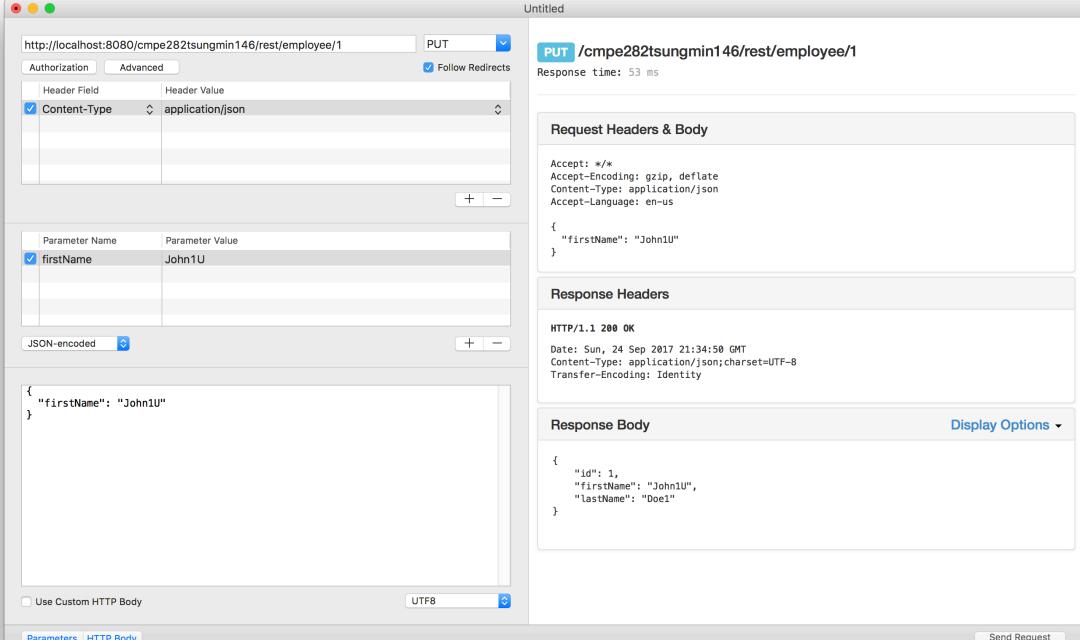
MongoDB



```
1. mongo
> db
cmpe282tsungmin146
> db.project.find({"id":1})
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project1", "budget" : 100 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project1", "budget" : 100 }
>
```

PUT .../rest/employee/n

Success: Update employee Id=1 field

	Screen Shot
MongoDB	 <pre>> db.cmpe282tsungmin146 > db.employee.find({"_id":1}) { "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "John1", "lastName" : "Doe1" } ></pre>
Client	 <p>The screenshot shows a REST client interface with the following details:</p> <ul style="list-style-type: none">Request URL: http://localhost:8080/cmpe282tsungmin146/rest/employee/1Method: PUTHeaders: Content-Type: application/jsonParameters: firstName: John1URequest Body: {"firstName": "John1U"}Response Headers: HTTP/1.1 200 OK, Date: Sun, 24 Sep 2017 21:34:50 GMT, Content-Type: application/json; charset=UTF-8, Transfer-Encoding: IdentityResponse Body: {"_id": 1, "firstName": "John1U", "lastName": "Doe1"}

MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1", "lastName" : "Doe1" }
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1U", "lastName" : "Doe1" }
> 
```

Client

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

- Request URL:** `http://localhost:8080/cmpe282tsungmin146/rest/employee/1`
- Method:** PUT
- Headers:** Content-Type: application/json
- Parameters:** lastName: Doe1U
- Request Body:** `{ "lastName": "Doe1U" }`
- Response Headers:** HTTP/1.1 200 OK, Date: Sun, 24 Sep 2017 21:44:55 GMT, Content-Type: application/json; charset=UTF-8, Transfer-Encoding: Identity
- Response Body:** `{ "id": 1, "firstName": "John1U", "lastName": "Doe1U" }`

MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1", "lastName" : "Doe1" }
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1U", "lastName" : "Doe1" }
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1U", "lastName" : "Doe1U" }
>
```

Client

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

Request Headers & Body

```
PUT /cmpe282tsungmin146/rest/employee/1
Response time: 59 ms
```

Request Headers & Body

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

Request Headers & Body

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

Response Headers

```
HTTP/1.1 200 OK
Date: Sun, 24 Sep 2017 21:41:25 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: Identity
```

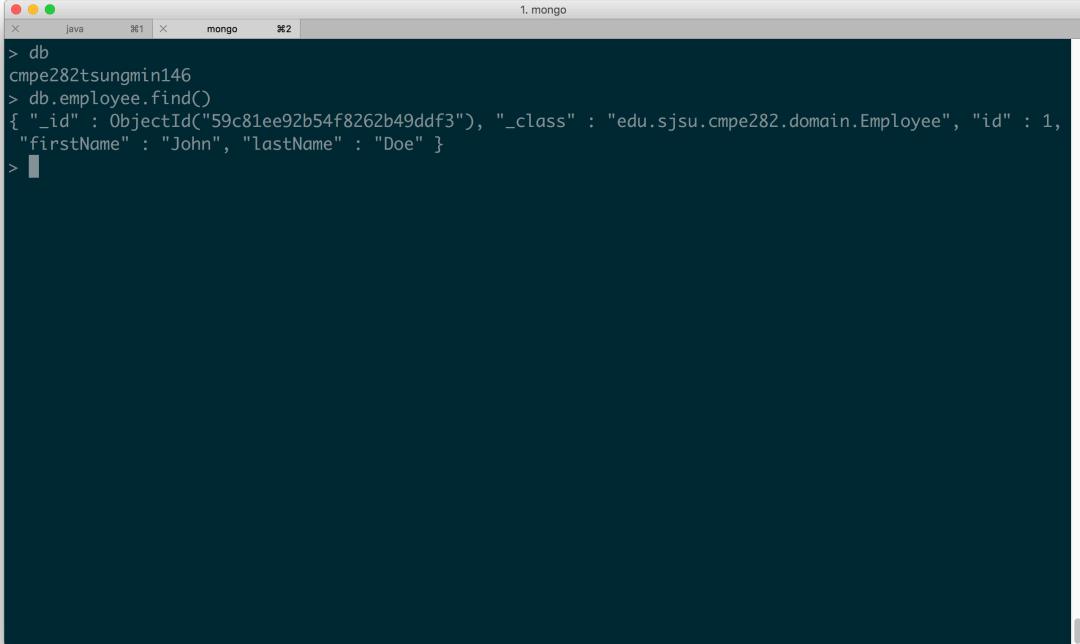
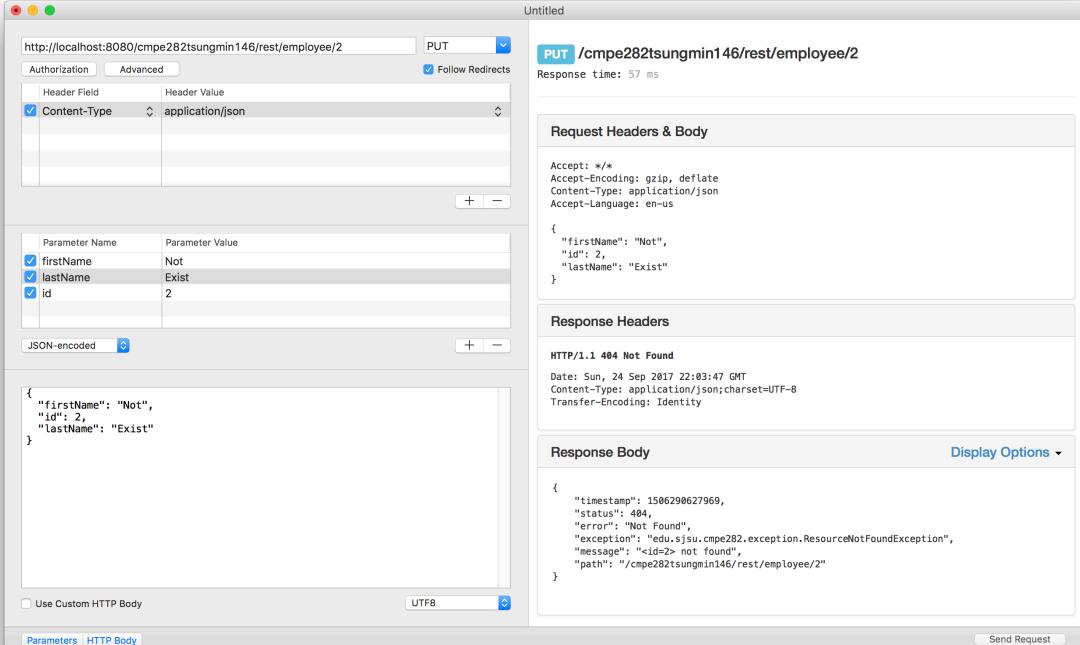
Response Body

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

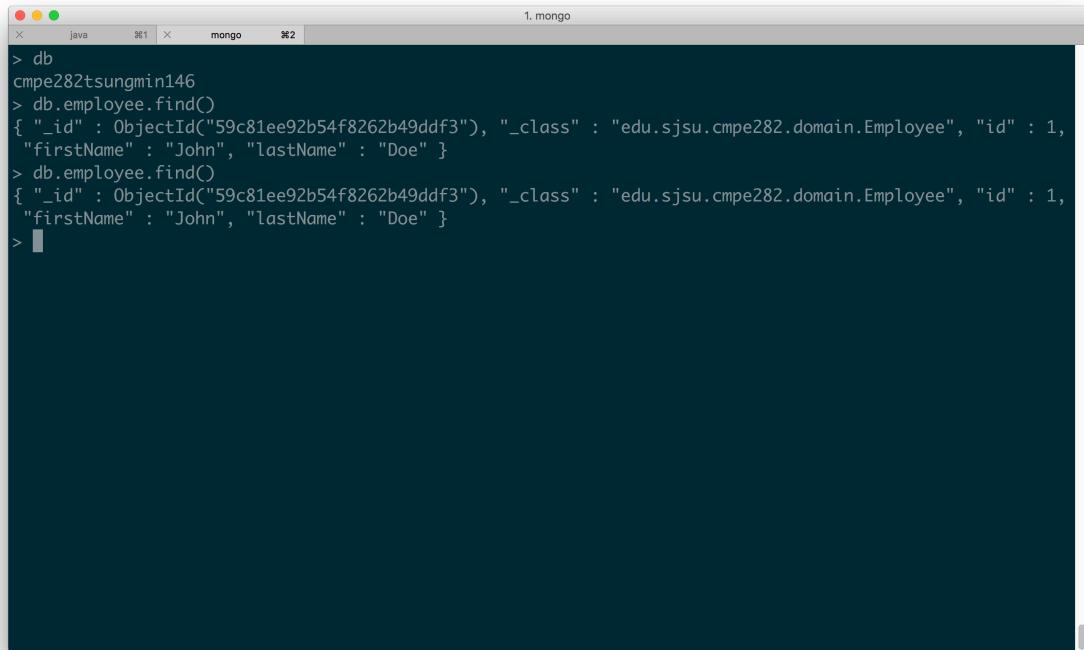
MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1", "lastName" : "Doe1" }
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1U", "lastName" : "Doe1" }
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John1U", "lastName" : "Doe1U" }
> db.employee.find({"id":1})
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John", "lastName" : "Doe" }
> █
```

Error: Updte employee Id=2, employee Id does not match Id in JSON content

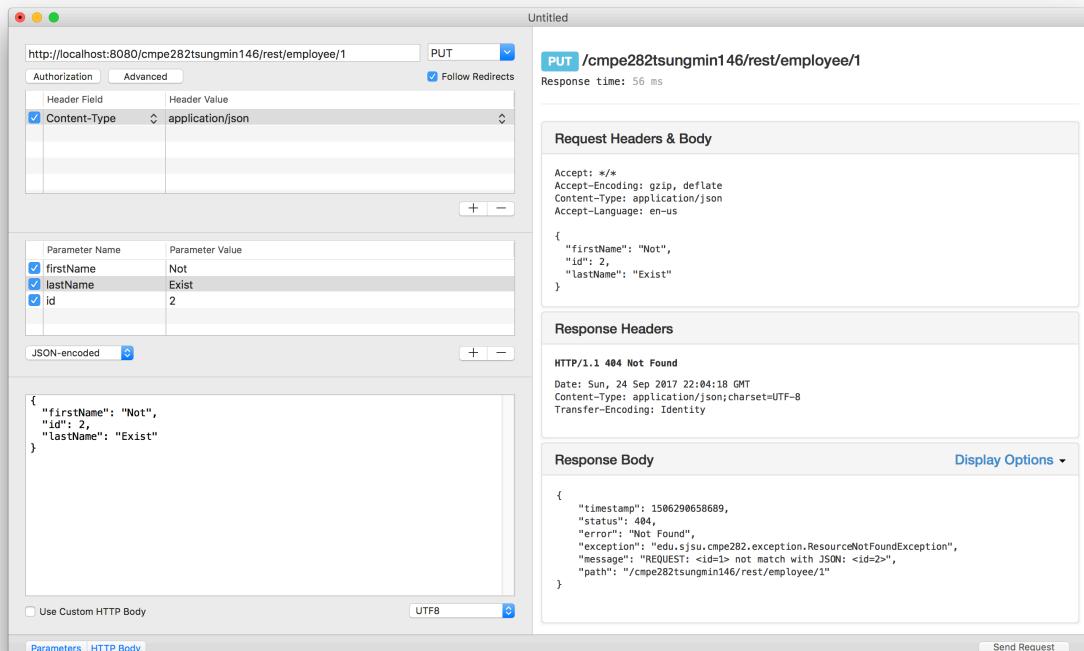
	<h3>Screen Shot</h3>  <pre>> db.cmpe282tsungmin146 > db.employee.find() { "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "John", "lastName" : "Doe" } ></pre>
MongoDB	
Client	 <p>PUT /cmpe282tsungmin146/rest/employee/2</p> <p>Request Headers & Body</p> <pre>Accept: */* Accept-Encoding: gzip, deflate Content-Type: application/json Accept-Language: en-us { "firstName": "Not", "id": 2, "lastName": "Exist" }</pre> <p>Response Headers</p> <pre>HTTP/1.1 404 Not Found Date: Sun, 24 Sep 2017 22:03:47 GMT Content-Type: application/json;charset=UTF-8 Transfer-Encoding: Identity</pre> <p>Response Body</p> <pre>{ "timestamp": 1506290627969, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id=2> not found", "path": "/cmpe282tsungmin146/rest/employee/2" }</pre>

MongoDB



```
> db
cmpe282tsungmin146
> db.employee.find()
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John", "lastName" : "Doe" }
> 
```

Client



PUT /cmpe282tsungmin146/rest/employee/1

Response time: 56 ms

Request Headers & Body

Accept: */*

Accept-Encoding: gzip, deflate

Content-Type: application/json

Accept-Language: en-us

{ "firstName": "Not",
 "id": 2,
 "lastName": "Exist" }
}

Response Headers

HTTP/1.1 404 Not Found

Date: Sun, 24 Sep 2017 22:04:18 GMT

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

Transfer-Encoding: Identity

Response Body

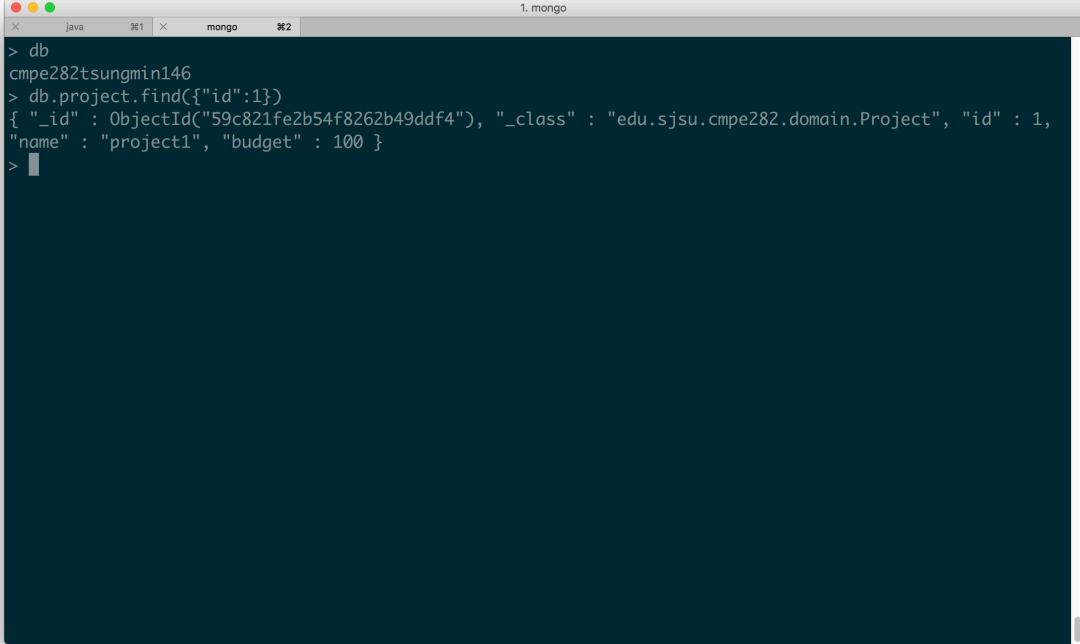
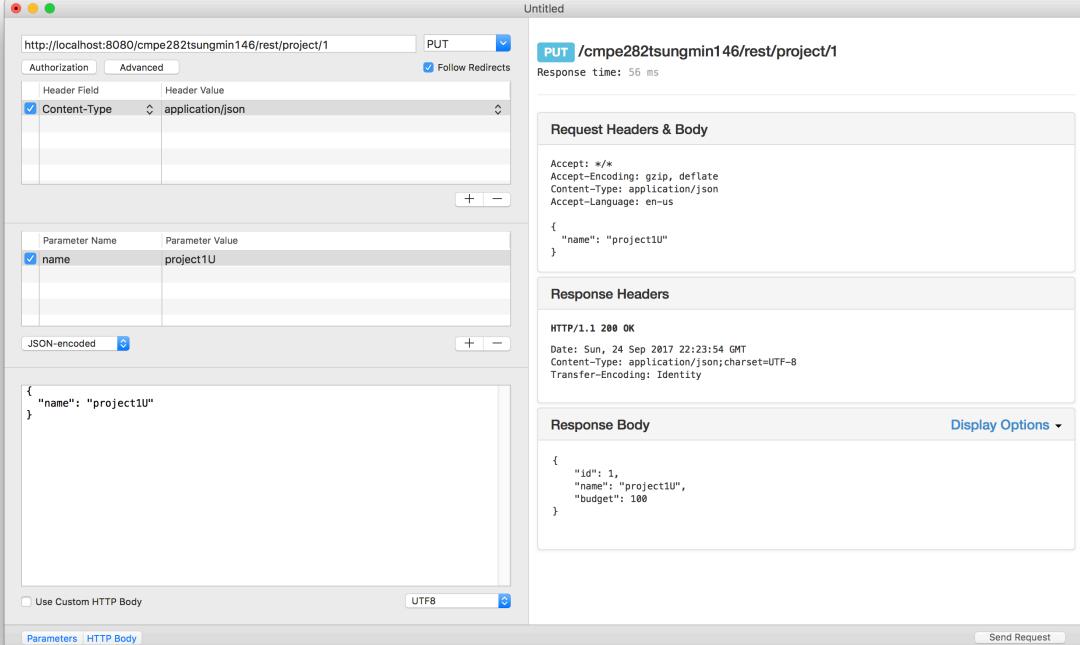
{
 "timestamp": 1506290659689,
 "status": 404,
 "error": "Not Found",
 "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException",
 "message": "REQUEST: <id=> not match with JSON: <id=2>",
 "path": "/cmpe282tsungmin146/rest/employee/1"
}

MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.employee.find()
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John", "lastName" : "Doe" }
> db.employee.find()
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John", "lastName" : "Doe" }
> █
```

PUT .../rest/project/n

Success: Update project Id=1 field

	Screen Shot
MongoDB	 A screenshot of a Mac OS X terminal window titled "mongo". The session shows a query to find a document where the id is 1. The output is a single document: { "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "project1", "budget" : 100 }. This indicates that the update operation was successful.
Client	 A screenshot of the Postman application interface. The request URL is "http://localhost:8080/cmpe282tsungmin146/rest/project/1". The method is set to "PUT". In the "Request Headers & Body" section, the Content-Type is set to "application/json". The body contains a JSON object with a "name" parameter set to "project1U". The response status is "HTTP/1.1 200 OK" and the response body is a JSON object with the updated fields: { "id": 1, "name": "project1U", "budget": 100 }.

MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "project1", "budget" : 100 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "project1U", "budget" : 100 }
> 
```

Client

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`
- Request Body (JSON-encoded):** `{ "budget": 10 }`
- Response Headers:** `HTTP/1.1 200 OK`, `Date: Sun, 24 Sep 2017 22:25:23 GMT`, `Content-Type: application/json; charset=UTF-8`, `Transfer-Encoding: Identity`
- Response Body:** `{ "id": 1, "name": "project1U", "budget": 10 }`

MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "project1", "budget" : 100 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "project1U", "budget" : 100 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1,
  "name" : "project1U", "budget" : 10 }
>
```

Client

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

Request Headers & Body

```
PUT /cmpe282tsungmin146/rest/project/1
Response time: 63 ms
```

Request Headers & Body

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

Request Body

```
{
  "budget": 100000,
  "name": "project"
}
```

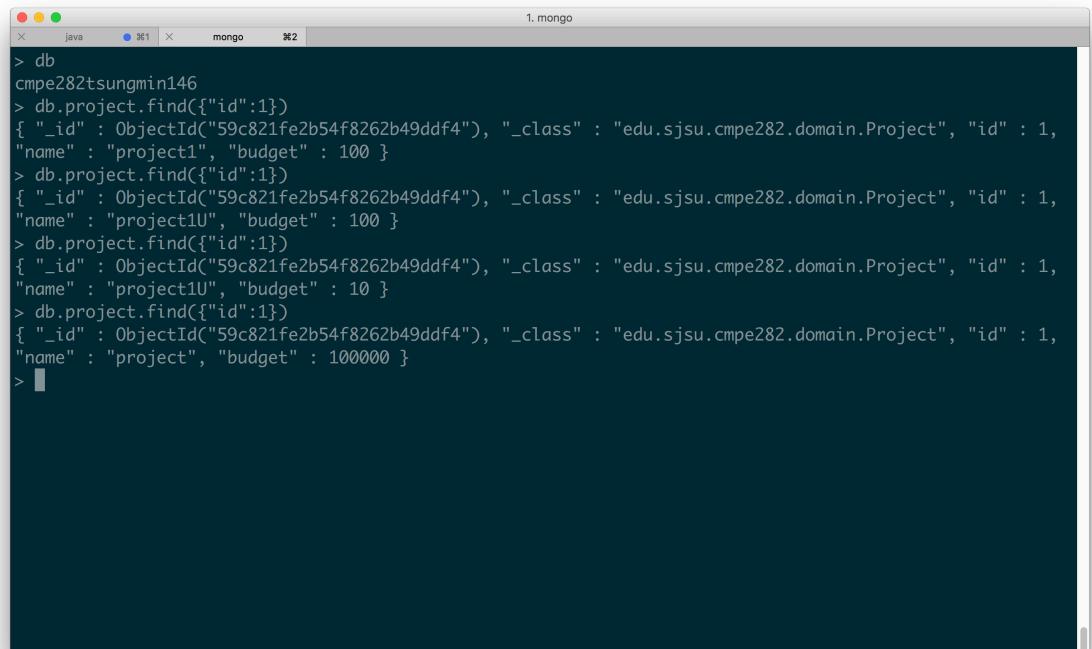
Response Headers

```
HTTP/1.1 200 OK
Date: Sun, 24 Sep 2017 22:27:54 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: Identity
```

Response Body

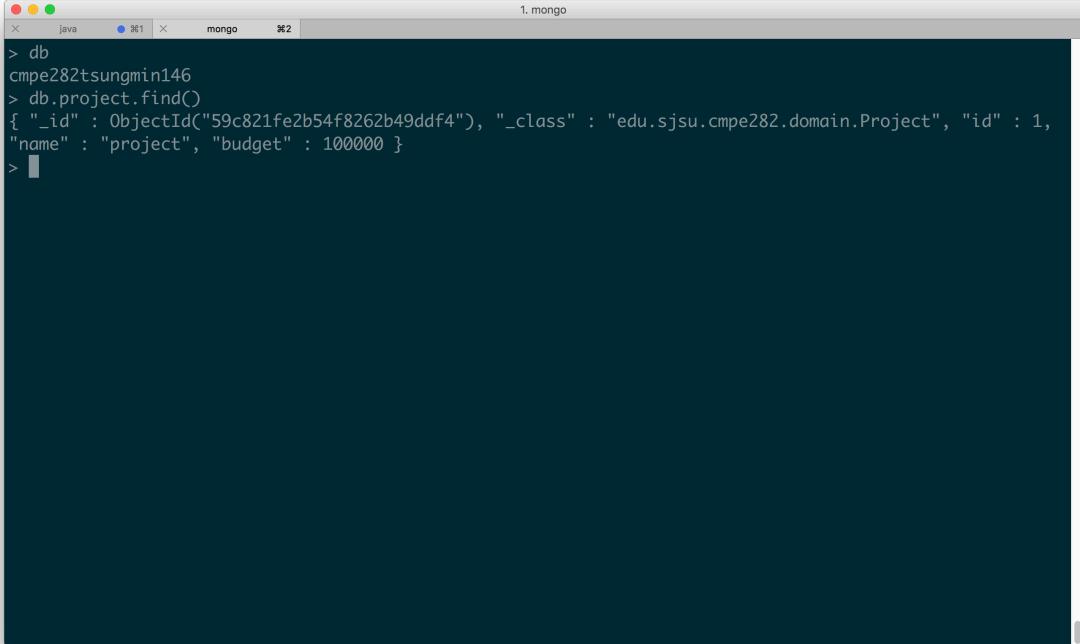
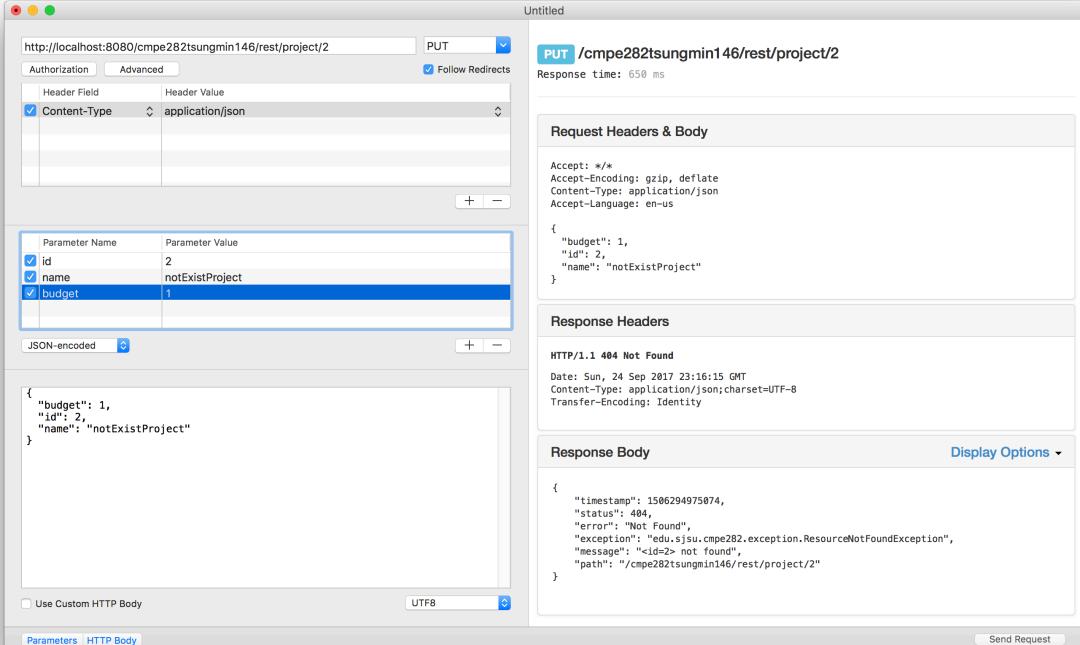
```
{
  "id": 1,
  "name": "project",
  "budget": 100000
}
```

MongoDB

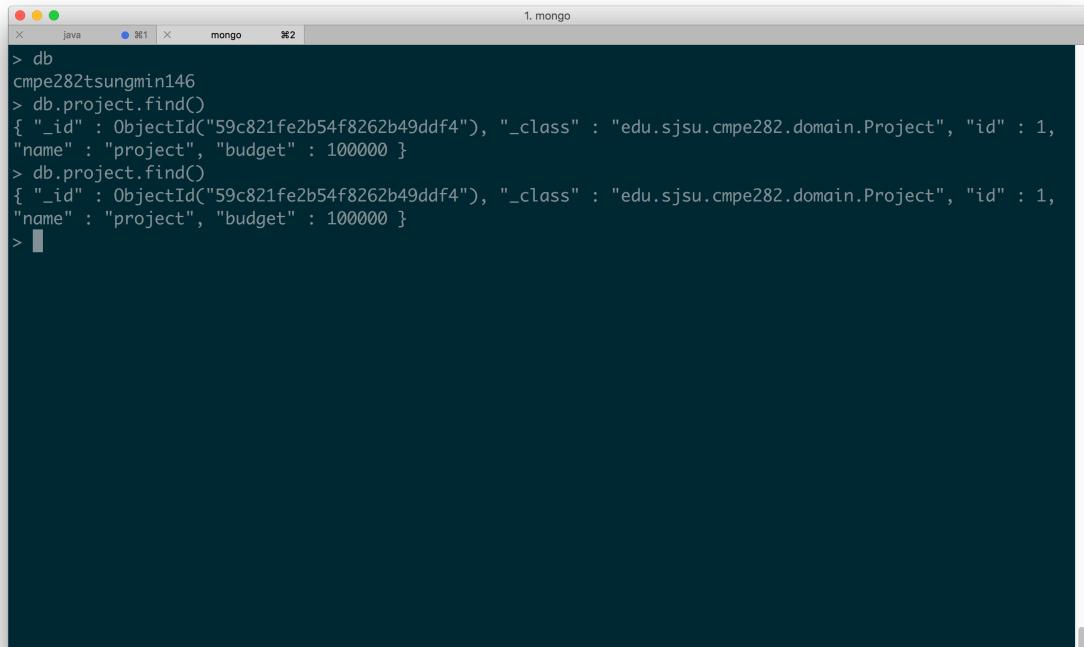


```
1. mongo
> db
cmpe282tsungmin146
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project1", "budget" : 100 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project1", "budget" : 100 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project1U", "budget" : 10 }
> db.project.find({"id":1})
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project", "budget" : 100000 }
> 
```

Error: Update project Id=2, project Id does not match Id in JSON content

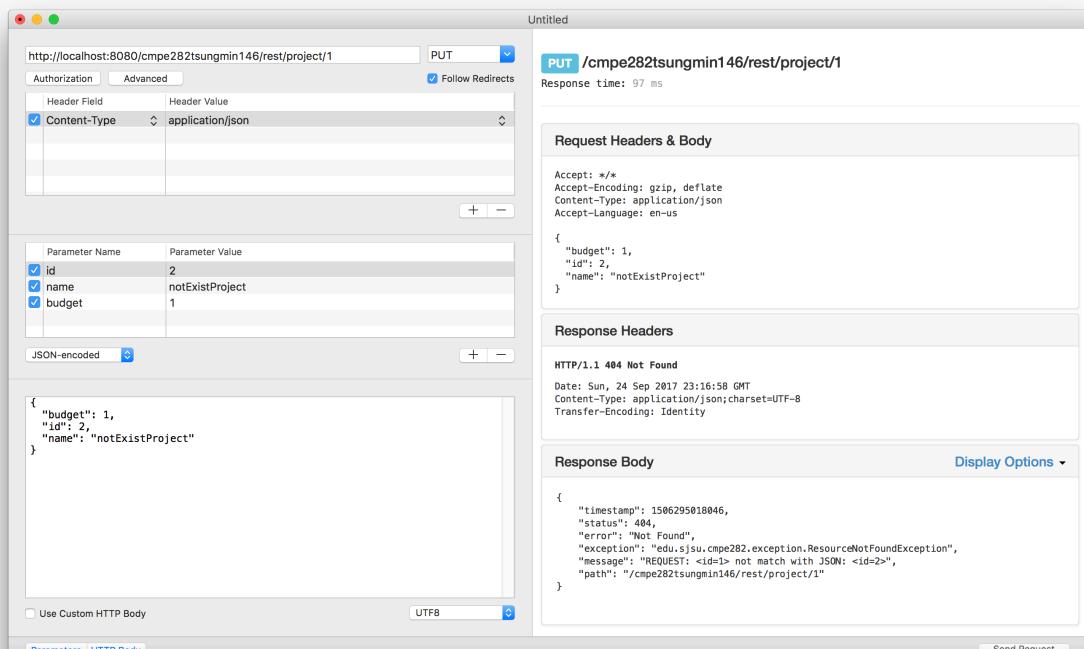
	<h3>Screen Shot</h3>  <pre>> db cmpe282tsungmin146 > db.project.find() { "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "_class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "project", "budget" : 100000 } ></pre>
MongoDB	
Client	 <p>PUT /cmpe282tsungmin146/rest/project/2</p> <p>Response time: 650 ms</p> <p>Request Headers & Body</p> <pre>Accept: */* Accept-Encoding: gzip, deflate Content-Type: application/json Accept-Language: en-us { "budget": 1, "id": 2, "name": "notExistProject" }</pre> <p>Response Headers</p> <pre>HTTP/1.1 404 Not Found Date: Sun, 24 Sep 2017 23:16:15 GMT Content-Type: application/json;charset=UTF-8 Transfer-Encoding: Identity</pre> <p>Response Body</p> <pre>{ "timestamp": 1506294975074, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id=2> not found", "path": "/cmpe282tsungmin146/rest/project/2" }</pre>

MongoDB



```
1. mongo
> db
cmpe282tsungmin146
> db.project.find()
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "project", "budget" : 100000 }
> db.project.find()
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 1, "name" : "project", "budget" : 100000 }
> 
```

Client



POST /cmpe282tsungmin146/rest/project/1

Request Headers & Body

Request Headers:

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

Body (JSON-encoded):

```
{
  "budget": 1,
  "id": 2,
  "name": "notExistProject"
}
```

Response Headers:

HTTP/1.1 404 Not Found

Date: Sun, 24 Sep 2017 23:16:58 GMT

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

Transfer-Encoding: Identity

Response Body:

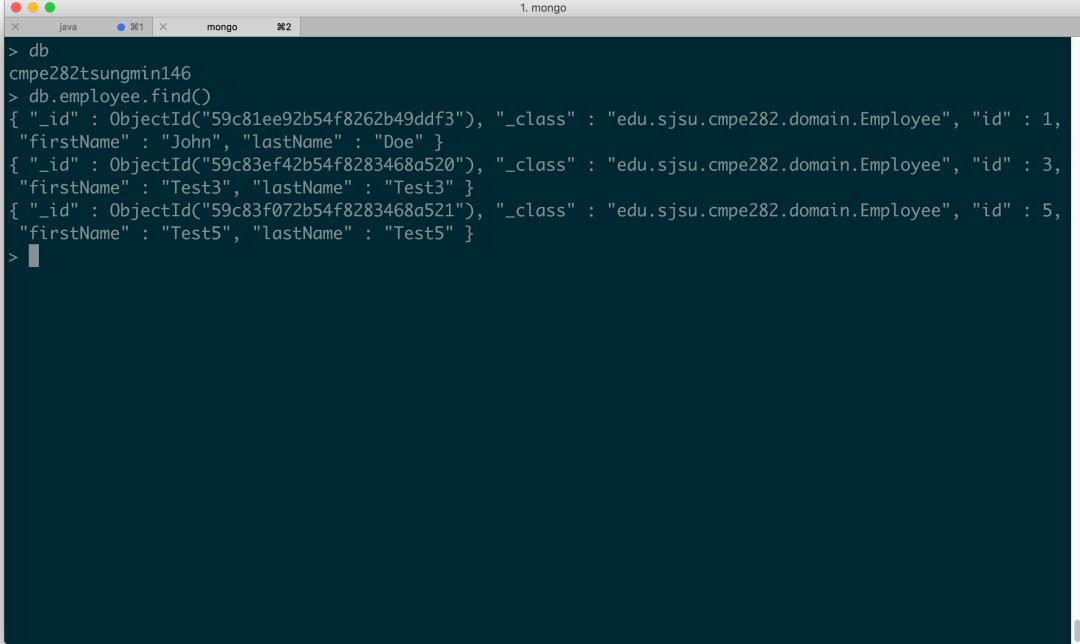
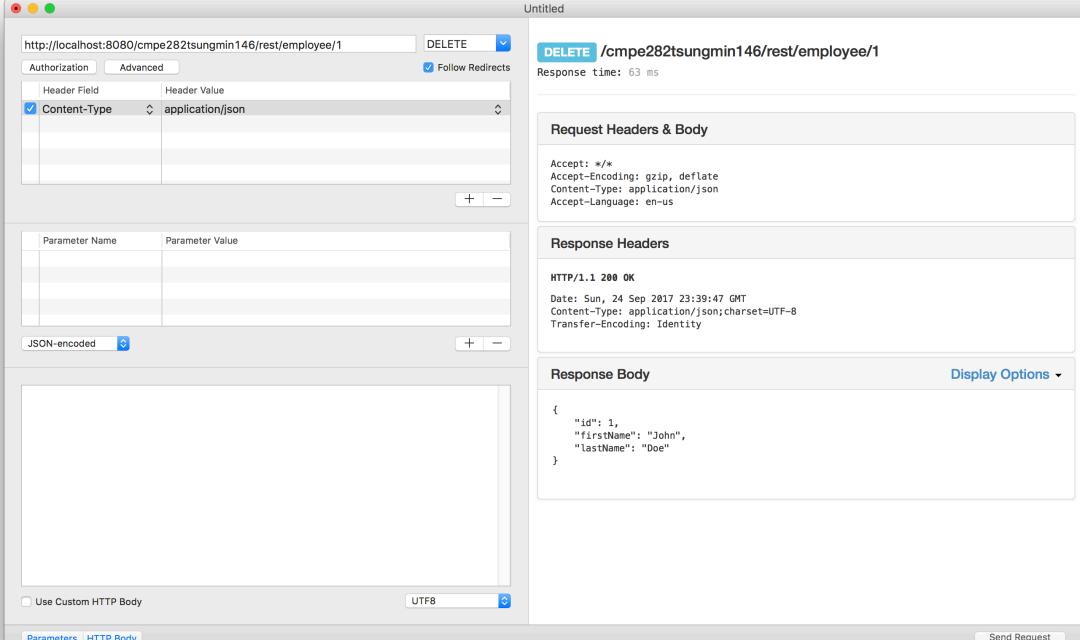
```
{
  "timestamp": 1506295018046,
  "status": 404,
  "error": "Not Found",
  "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException",
  "message": "REQUEST: <id=> not match with JSON: <id=2>",
  "path": "/cmpe282tsungmin146/rest/project/1"
}
```

MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.project.find()
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project", "budget" : 100000 }
> db.project.find()
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project", "budget" : 100000 }
> db.project.find()
{ "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project", "budget" : 100000 }
>
```

DELETE /.../rest/employee/n

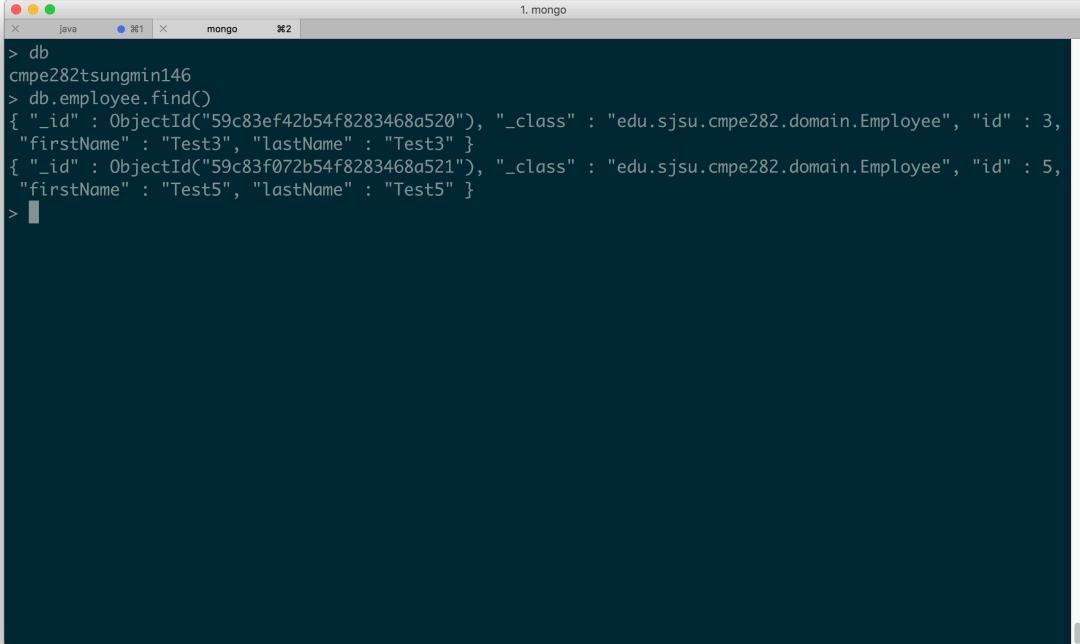
Success: Delete employee Id=1

	Screen Shot
MongoDB	 <p>A screenshot of the MongoDB shell interface. The command `db.employee.find()` is run, displaying five documents with IDs 1, 3, 5, and Test3. The document with ID 1 has first name 'John' and last name 'Doe'. The document with ID 5 has first name 'Test5' and last name 'Test5'. The document with ID Test3 has first name 'Test3' and last name 'Test3'. The command `db.employee.remove({ '_id': ObjectId('59c81ee92b54f8262b49ddf3') })` is run to delete the document with ID 1.</p>
Client	 <p>A screenshot of a REST client tool (like Postman) showing a successful DELETE request. The URL is <code>http://localhost:8080/cmpe282tsungmin146/rest/employee/1</code>. The response status is <code>HTTP/1.1 200 OK</code> with a response time of 63 ms. The response body shows the deleted document: { "id": 1, "firstName": "John", "lastName": "Doe" }.</p>

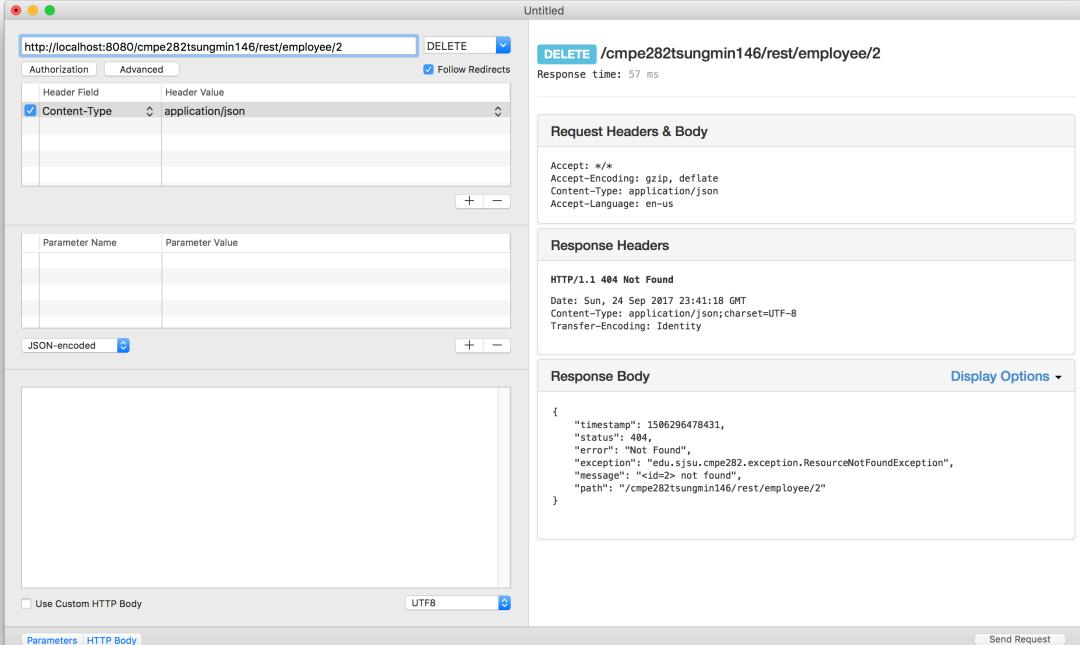
MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.employee.find()
{ "_id" : ObjectId("59c81ee92b54f8262b49ddf3"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1,
  "firstName" : "John", "lastName" : "Doe" }
{ "_id" : ObjectId("59c83ef42b54f8283468a520"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 3,
  "firstName" : "Test3", "lastName" : "Test3" }
{ "_id" : ObjectId("59c83f072b54f8283468a521"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 5,
  "firstName" : "Test5", "lastName" : "Test5" }
> db.employee.find()
{ "_id" : ObjectId("59c83ef42b54f8283468a520"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 3,
  "firstName" : "Test3", "lastName" : "Test3" }
{ "_id" : ObjectId("59c83f072b54f8283468a521"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 5,
  "firstName" : "Test5", "lastName" : "Test5" }
> █
```

Error: Delete employee Id=2

	<h3>Screen Shot</h3>  <p>The terminal window shows the following MongoDB session:</p> <pre>> db cmpe282tsungmin146 > db.employee.find() { "_id" : ObjectId("59c83ef42b54f8283468a520"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 3, "firstName" : "Test3", "lastName" : "Test3" } { "_id" : ObjectId("59c83f072b54f8283468a521"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 5, "firstName" : "Test5", "lastName" : "Test5" } ></pre>
--	---

MongoDB

	 <p>The screenshot shows a cURL interface with the following configuration:</p> <ul style="list-style-type: none">Method: DELETEURL: <code>http://localhost:8080/cmpe282tsungmin146/rest/employee/2</code>Headers:<ul style="list-style-type: none">Content-Type: application/jsonRequest Headers & Body:<ul style="list-style-type: none">Accept: */*Accept-Encoding: gzip, deflateContent-Type: application/jsonAccept-Language: en-usResponse Headers:<ul style="list-style-type: none">HTTP/1.1 404 Not FoundDate: Sun, 24 Sep 2017 23:41:18 GMTContent-Type: application/json; charset=UTF-8Transfer-Encoding: IdentityResponse Body:<pre>{ "timestamp": 1506296478431, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "id=2> not found", "path": "/cmpe282tsungmin146/rest/employee/2" }</pre>
--	--

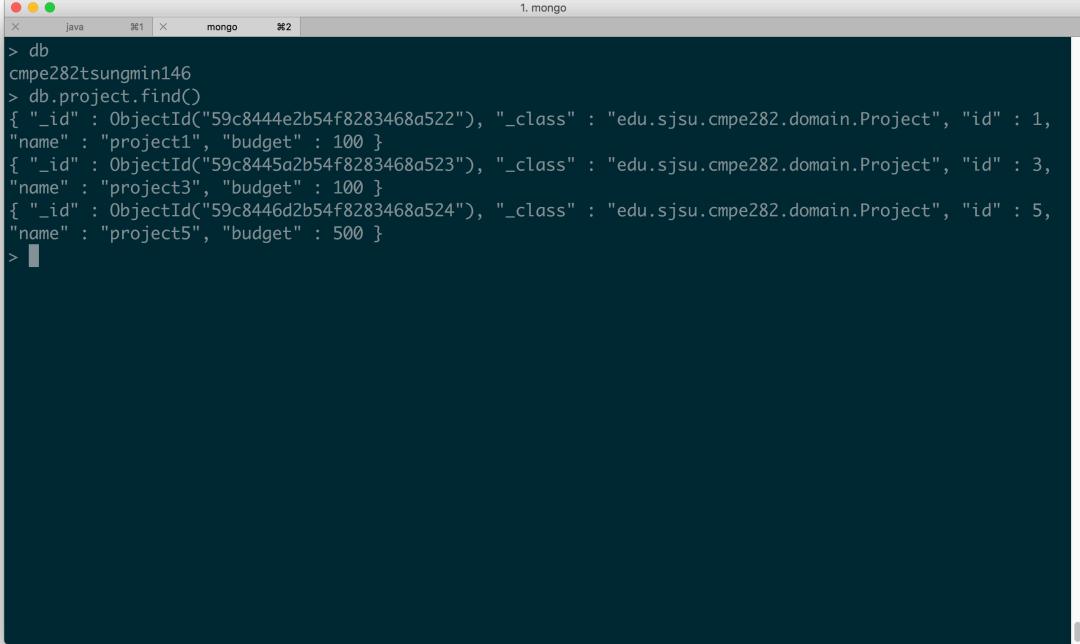
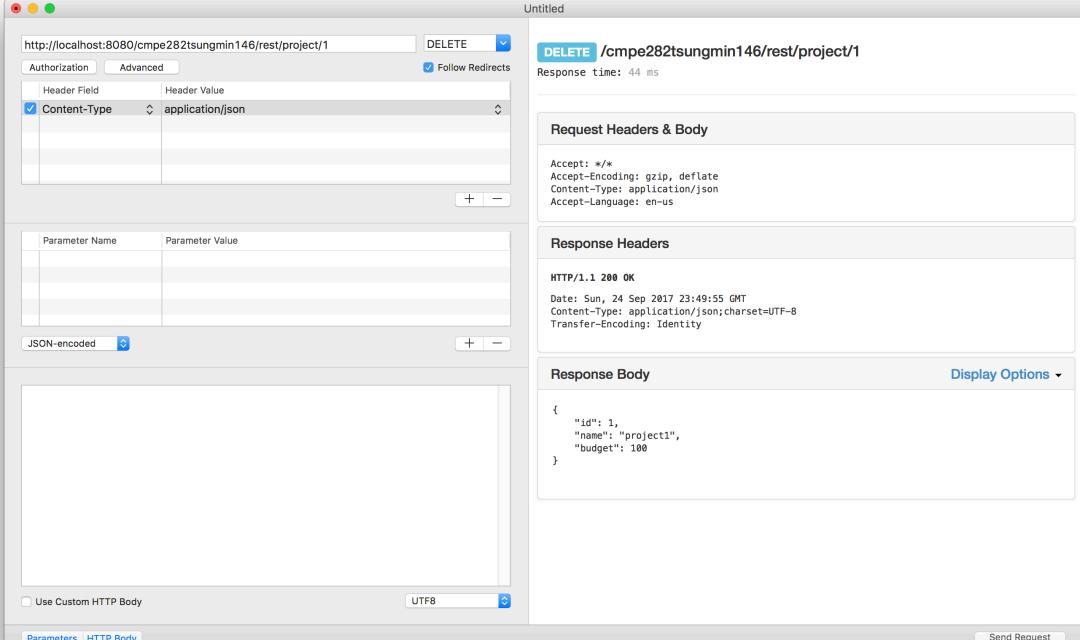
Client

MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.employee.find()
{ "_id" : ObjectId("59c83ef42b54f8283468a520"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 3,
  "firstName" : "Test3", "lastName" : "Test3" }
{ "_id" : ObjectId("59c83f072b54f8283468a521"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 5,
  "firstName" : "Test5", "lastName" : "Test5" }
> db.employee.find()
{ "_id" : ObjectId("59c83ef42b54f8283468a520"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 3,
  "firstName" : "Test3", "lastName" : "Test3" }
{ "_id" : ObjectId("59c83f072b54f8283468a521"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 5,
  "firstName" : "Test5", "lastName" : "Test5" }
> 
```

DELETE /.../rest/project/n

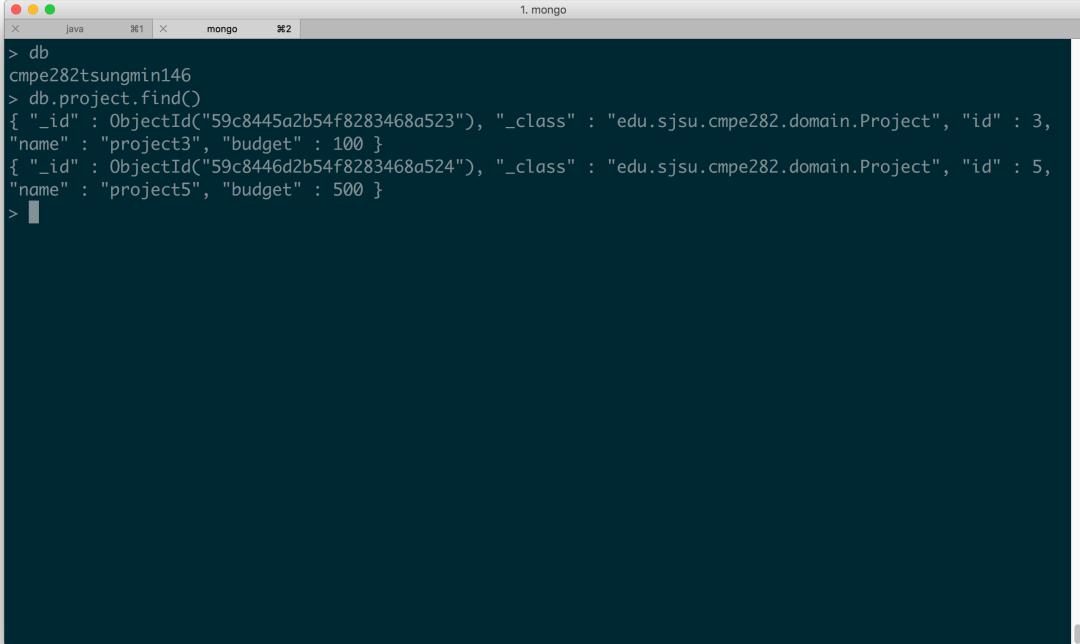
Success: Delete project Id=1

	Screen Shot
MongoDB	 <p>A screenshot of the MongoDB shell window titled "1. mongo". It shows the command > db.project.find() and its output, which lists five projects with IDs 1, 3, and 5 removed. The removed documents are represented by empty square boxes.</p> <pre>> db cmpe282tsungmin146 > db.project.find() { "_id" : ObjectId("59c8444e2b54f8283468a522"), "name" : "project1", "budget" : 100 } { "_id" : ObjectId("59c8445a2b54f8283468a523"), "name" : "project3", "budget" : 100 } { "_id" : ObjectId("59c8446d2b54f8283468a524"), "name" : "project5", "budget" : 500 } > ■</pre>
Client	 <p>A screenshot of a REST client tool window titled "Untitled". It shows a DELETE request to http://localhost:8080/cmpe282tsungmin146/rest/project/1. The response status is 200 OK, and the response body contains the JSON object { "id": 1, "name": "project1", "budget": 100 }.</p> <p>Request Headers & Body</p> <pre>Accept: */* Accept-Encoding: gzip, deflate Content-Type: application/json Accept-Language: en-us</pre> <p>Response Headers</p> <pre>HTTP/1.1 200 OK Date: Sun, 24 Sep 2017 23:49:55 GMT Content-Type: application/json; charset=UTF-8 Transfer-Encoding: Identity</pre> <p>Response Body</p> <pre>{ "id": 1, "name": "project1", "budget": 100 }</pre>

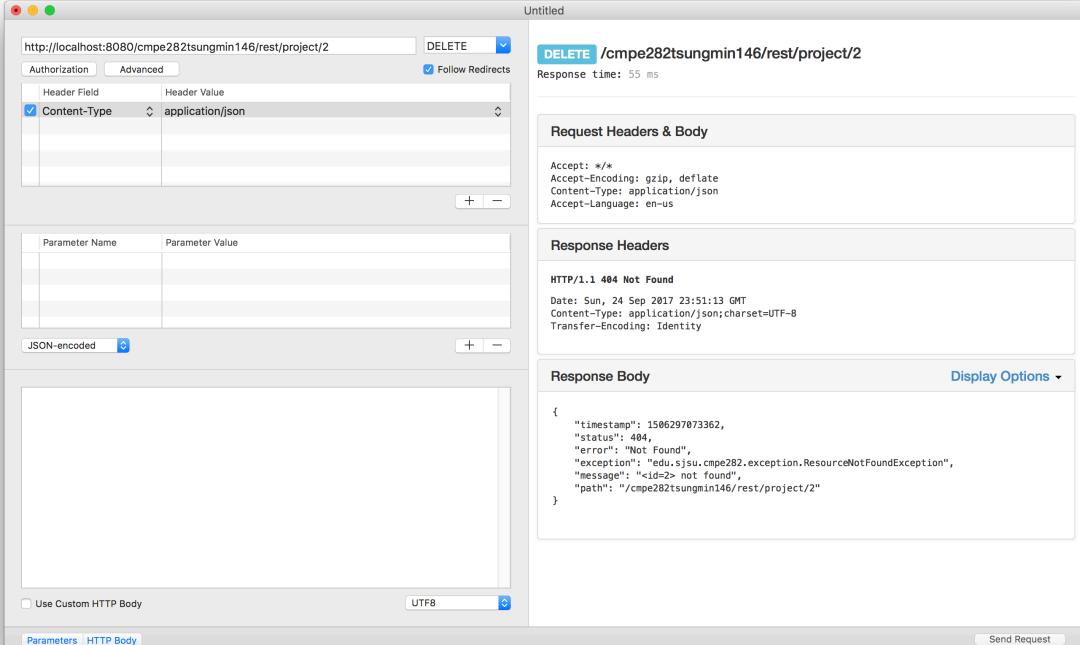
MongoDB

```
1. mongo
> db
cmpe282tsungmin146
> db.project.find()
[{"_id": ObjectId("59c8444e2b54f8283468a522"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 1, "name": "project1", "budget": 100}, {"_id": ObjectId("59c8445a2b54f8283468a523"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 3, "name": "project3", "budget": 100}, {"_id": ObjectId("59c8446d2b54f8283468a524"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 5, "name": "project5", "budget": 500}, {"_id": ObjectId("59c8445a2b54f8283468a523"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 3, "name": "project3", "budget": 100}, {"_id": ObjectId("59c8446d2b54f8283468a524"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 5, "name": "project5", "budget": 500}]
> █
```

Error: Delete project Id=2

	<h3>Screen Shot</h3>  <pre>1. mongo > db cmpe282tsungmin146 > db.project.find() { "_id" : ObjectId("59c8445a2b54f8283468a523"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 3, "name" : "project3", "budget" : 100 } { "_id" : ObjectId("59c8446d2b54f8283468a524"), "class" : "edu.sjsu.cmpe282.domain.Project", "id" : 5, "name" : "project5", "budget" : 500 } ></pre>
--	---

MongoDB

	 <p>Client</p> <p>The screenshot shows a cURL interface with the following details:</p> <ul style="list-style-type: none">Method: DELETEURL: <code>/cmpe282tsungmin146/rest/project/2</code>Headers:<ul style="list-style-type: none">Content-Type: application/jsonRequest Headers & Body:<ul style="list-style-type: none">Accept: */*Accept-Encoding: gzip, deflateContent-Type: application/jsonAccept-Language: en-usResponse Headers:<ul style="list-style-type: none">HTTP/1.1 404 Not FoundDate: Sun, 24 Sep 2017 23:51:13 GMTContent-Type: application/json; charset=UTF-8Transfer-Encoding: IdentityResponse Body:<pre>{ "timestamp": 1506297073362, "status": 404, "error": "Not Found", "exception": "edu.sjsu.cmpe282.exception.ResourceNotFoundException", "message": "<id>2 not found", "path": "/cmpe282tsungmin146/rest/project/2" }</pre>
--	--

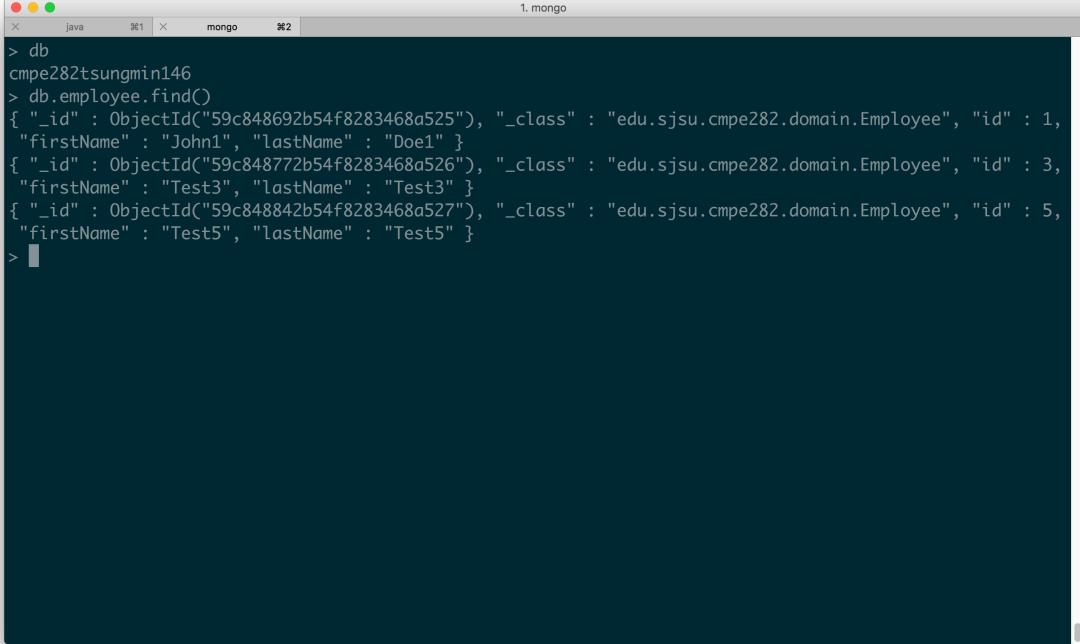
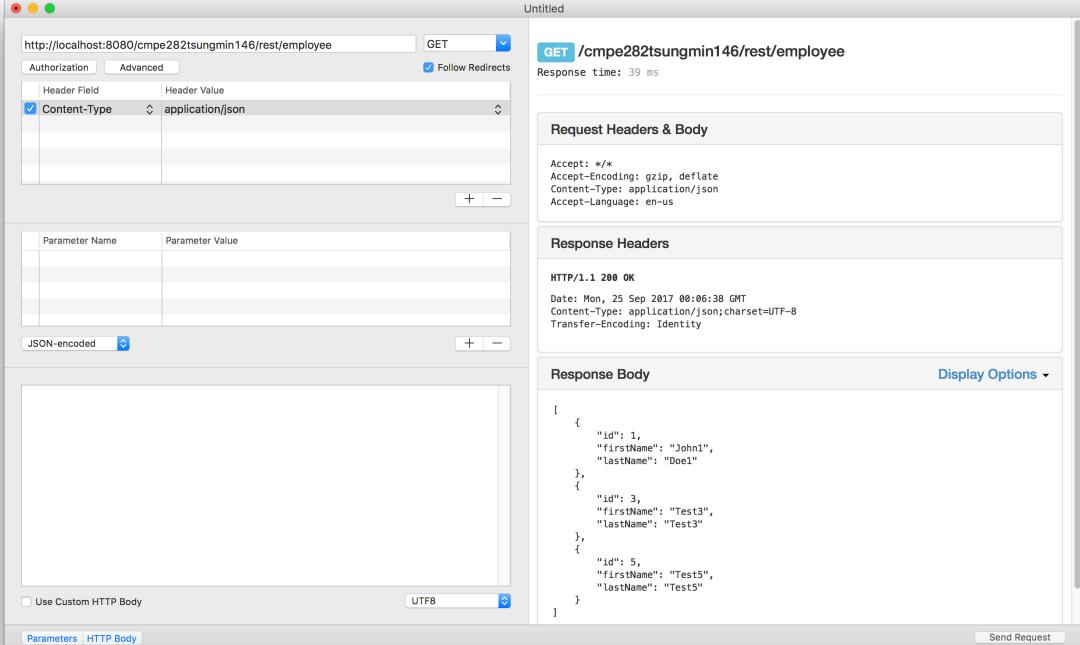
Parameters HTTP Body Send Request

MongoDB

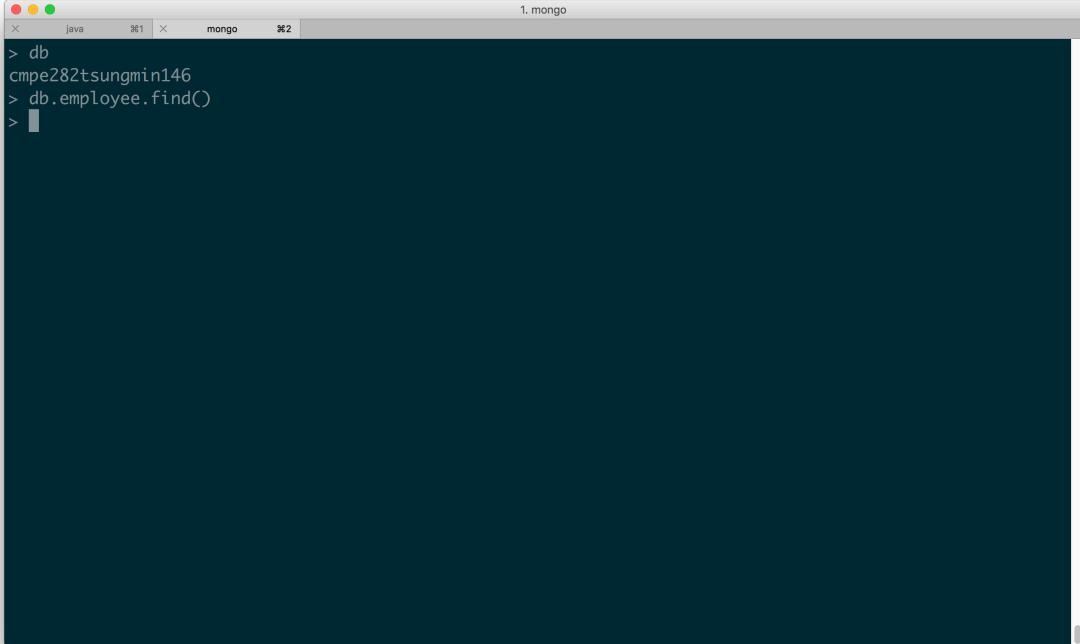
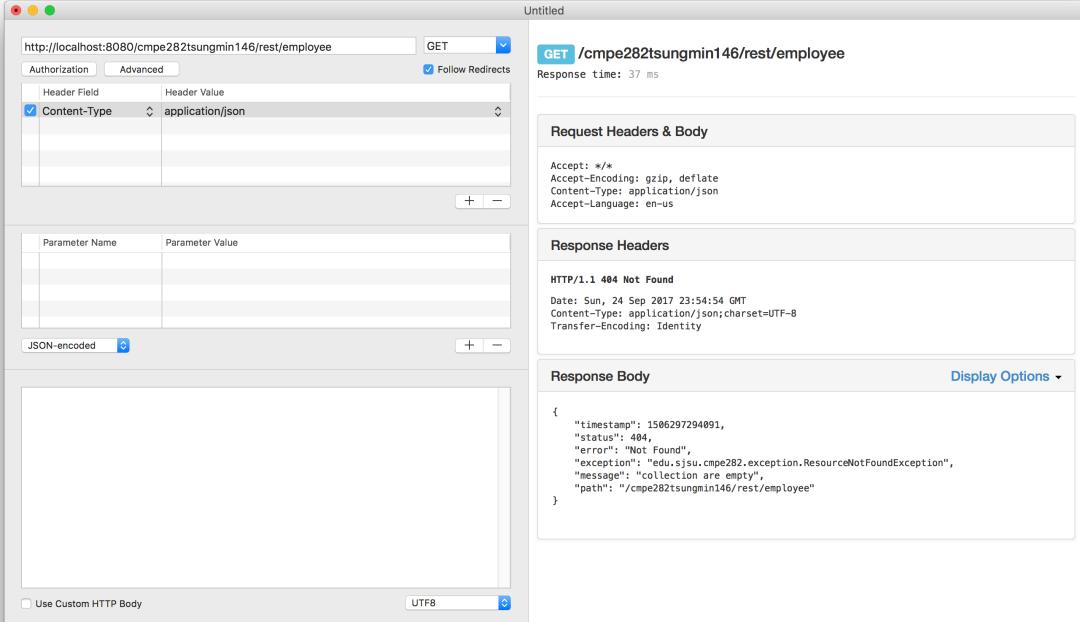
```
1. mongo
> db
cmpe282tsungmin146
> db.project.find()
[{"_id": ObjectId("59c8445a2b54f8283468a523"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 3, "name": "project3", "budget": 100}, {"_id": ObjectId("59c8446d2b54f8283468a524"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 5, "name": "project5", "budget": 500}]
> db.project.find()
[{"_id": ObjectId("59c8445a2b54f8283468a523"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 3, "name": "project3", "budget": 100}, {"_id": ObjectId("59c8446d2b54f8283468a524"), "class": "edu.sjsu.cmpe282.domain.Project", "id": 5, "name": "project5", "budget": 500}]
>
```

GET /.../rest/employee

Success: Get all employee

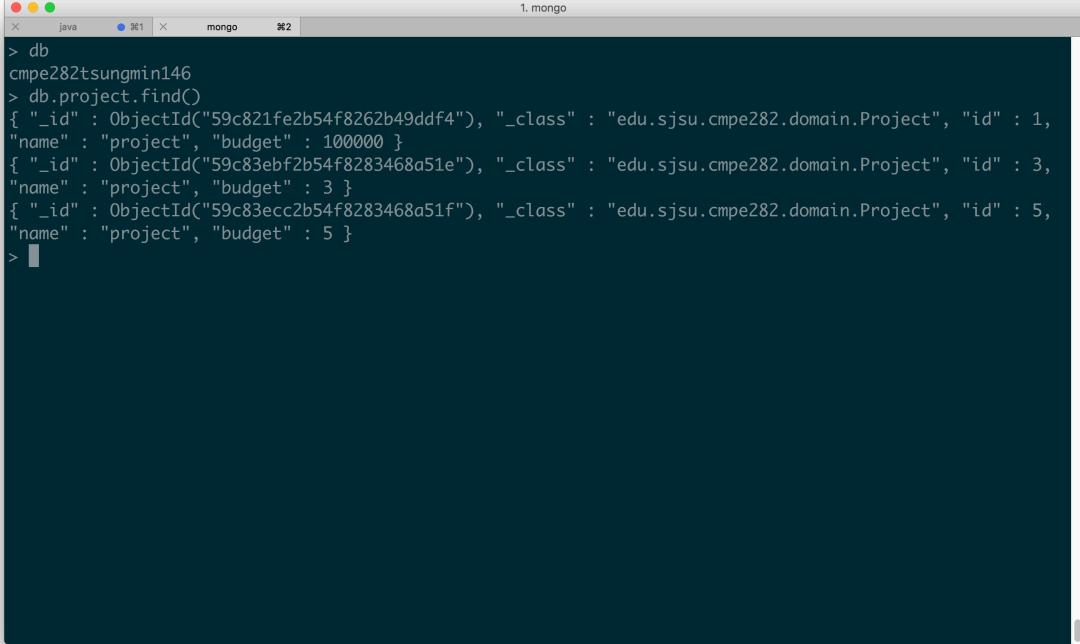
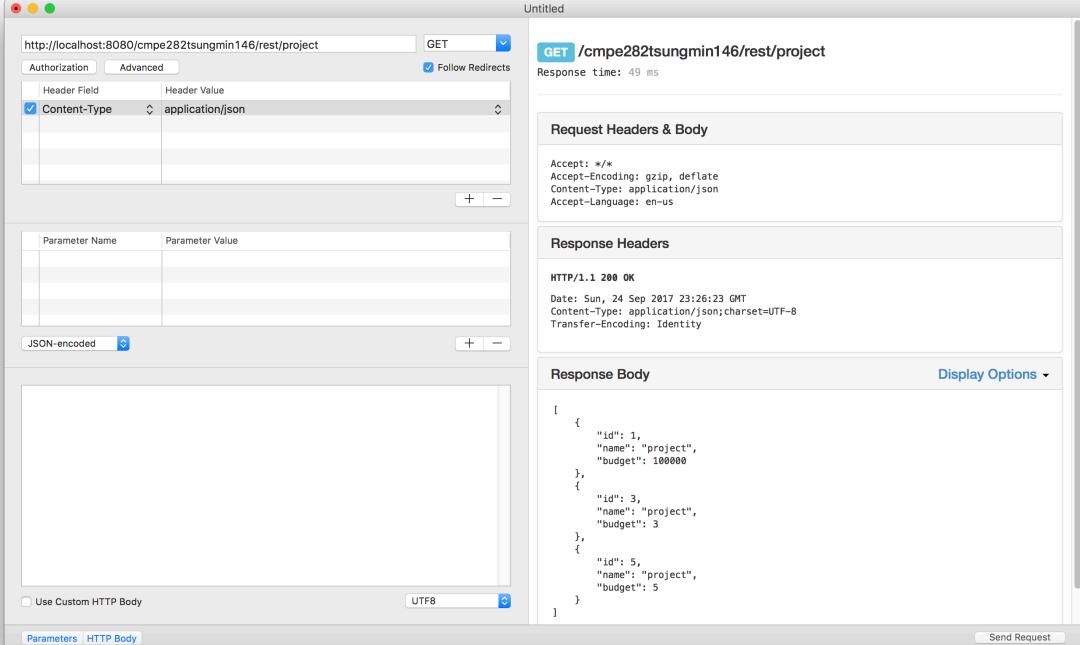
	Screen Shot
MongoDB	 A screenshot of a Mac OS X terminal window titled "mongo". It contains the following MongoDB shell session: <pre>> db cmpe282tsungmin146 > db.employee.find() { "_id" : ObjectId("59c848692b54f8283468a525"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 1, "firstName" : "John1", "lastName" : "Doe1" } { "_id" : ObjectId("59c848772b54f8283468a526"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 3, "firstName" : "Test3", "lastName" : "Test3" } { "_id" : ObjectId("59c848842b54f8283468a527"), "class" : "edu.sjsu.cmpe282.domain.Employee", "id" : 5, "firstName" : "Test5", "lastName" : "Test5" } ></pre>
Client	 A screenshot of a REST client application window titled "Untitled". It shows a successful GET request to "http://localhost:8080/cmpe282tsungmin146/rest/employee". The request details include: <ul style="list-style-type: none">Method: GETHeader Field: Content-Type, Value: application/jsonParameter Name: Parameter ValueContent Type: JSON-encoded The response details show: <ul style="list-style-type: none">Status: 200 OKDate: Mon, 25 Sep 2017 08:06:38 GMTContent-Type: application/json; charset=UTF-8Transfer-Encoding: Identity The Response Body pane displays the JSON array of employee documents: <pre>[{ "id": 1, "firstName": "John1", "lastName": "Doe1" }, { "id": 3, "firstName": "Test3", "lastName": "Test3" }, { "id": 5, "firstName": "Test5", "lastName": "Test5" }]</pre>

Error: Not any employee data

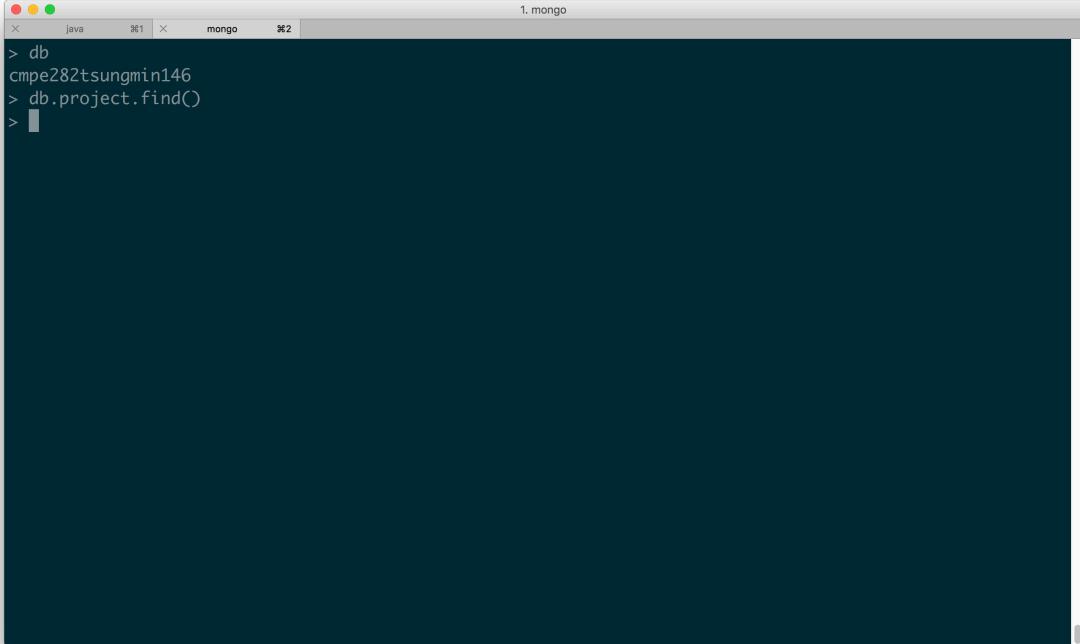
MongoDB	<p style="text-align: center;">Screen Shot</p>  <pre>1. mongo > db cmpe282tsungmin146 > db.employee.find() > </pre>
Client	 <p>The screenshot shows a cURL interface with the following details:</p> <ul style="list-style-type: none">Method: GETURL: <code>http://localhost:8080/cmpe282tsungmin146/rest/employee</code>Headers:<ul style="list-style-type: none">Content-Type: application/jsonRequest Headers & Body:<ul style="list-style-type: none">Accept: */*Accept-Encoding: gzip, deflateContent-Type: application/jsonAccept-Language: en-usResponse Headers:<ul style="list-style-type: none">HTTP/1.1 404 Not FoundDate: Sun, 24 Sep 2017 23:54:54 GMTContent-Type: application/json; charset=UTF-8Transfer-Encoding: IdentityResponse Body:<pre>{ "timestamp": 1506297294091, "status": 404, "error": "Not Found", "exception": "edu.sisu.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 with the command <code>db.project.find()</code> and its results. The results are three documents, each representing a project with fields <code>_id</code>, <code>name</code>, and <code>budget</code>.</p> <pre>> db.cmpe282tsungmin146 > db.project.find() { "_id" : ObjectId("59c821fe2b54f8262b49ddf4"), "name" : "project", "budget" : 100000 } { "_id" : ObjectId("59c83ebf2b54f8283468d51e"), "name" : "project", "budget" : 3 } { "_id" : ObjectId("59c83ecc2b54f8283468a51f"), "name" : "project", "budget" : 5 }</pre>
Client	 <p>The screenshot shows a REST client interface with the following details:</p> <ul style="list-style-type: none">Method: GETURL: <code>http://localhost:8080/cmpe282tsungmin146/rest/project</code>Headers:<ul style="list-style-type: none">Content-Type: application/jsonAccept: */*Accept-Encoding: gzip, deflateContent-Type: application/json; charset=UTF-8Accept-Language: en-usResponse Headers:<ul style="list-style-type: none">HTTP/1.1 200 OKDate: Sun, 24 Sep 2017 23:26:23 GMTContent-Type: application/json; charset=UTF-8Transfer-Encoding: IdentityResponse Body:<pre>[{"id": 1, "name": "project", "budget": 100000}, {"id": 3, "name": "project", "budget": 3}, {"id": 5, "name": "project", "budget": 5}]</pre>

Error: Not any project data

MongoDB	<p style="text-align: center;">Screen Shot</p> 
Client	