

API-Endpoints

SQL

Suppose our database 'testing' has table 'emp' with 'staff_number' as primary key and has the following description:

```
mysql> desc emp;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| staff_number   | int           | NO   | PRI | NULL    |       |
| fname         | varchar(20)   | YES  |     | NULL    |       |
| lname         | varchar(30)   | YES  |     | NULL    |       |
| gender         | char(1)       | YES  |     | NULL    |       |
| joining        | date          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)
```

And it contains the following data:

```
mysql> select * from emp;
+-----+-----+-----+-----+-----+
| staff_number | fname | lname | gender | joining   |
+-----+-----+-----+-----+-----+
| 1           | Bill  | Gates | M      | 1980-10-28 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

SELECT

Query	Endpoint	Response
SELECT * FROM emp;	GET /api/v1/testing/emp/	{ "schema": ["*"], "data": { "0": [1, "Bill", "Gates", "M", "1980-10-28"] } }
SELECT * FROM emp WHERE fname='Bill' and lname='Gates' and gender='M';	GET /api/v1/testing/emp/? row_filter='fname=Bill,lname=Gates, gender=M'	{ "schema": ["*"], "data": { "0": [1, "Bill", "Gates", "M", "1980-10-28"] } }
SELECT fname,lname,staff_number FROM emp	GET /api/v1/testing/emp/? col_filter='fname,lname,staff_nu mber'	{ "schema": ["fname", "lname", "staff_number"], "data": { "0": ["Bill", "Gates", 1] } }

<pre>SELECT fname,lname,staff_number FROM emp WHERE fname='Bill' and lname='Gates' and gender='M'</pre>	<pre>GET /api/v1/testing/emp/? row_filter='fname=Bill,lname=Gates,gender=M'& col_filter=col_filter='fname,lname,staff_number'</pre>	<pre>{ "schema": ["fname", "lname", "staff_number"], "data": { "0": ["Bill", "Gates", 1] } }</pre>
---	---	--

INSERT

Query	Endpoint	Data
INSERT INTO emp (staff_number, fname, lname) VALUES (2, 'John', 'doe');	POST /api/v1/testing/emp/	{ "column": ["staff_number", "fname", "lname"], "values": [2, "John", "doe"] }

```
mysql> select * from emp;
```

```
+-----+-----+-----+-----+-----+
| staff_number | fname | lname | gender | joining |
+-----+-----+-----+-----+-----+
|          1 | Bill  | Gates | M       | 1980-10-28 |
|          2 | John  | doe   | NULL    | NULL       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Query	Endpoint	Data
INSERT INTO emp VALUES (3, 'John', 'doe', 'M', '1980-10-28');	POST /api/v1/testing/emp/	{ "column": [], "values": [3, "John", "doe", "M", "1980-10-28"] }

```
mysql> select * from emp;
```

```
+-----+-----+-----+-----+-----+
| staff_number | fname | lname | gender | joining |
+-----+-----+-----+-----+-----+
|          1 | Bill  | Gates | M       | 1980-10-28 |
|          2 | John  | doe   | NULL    | NULL       |
|          3 | John  | doe   | M       | 1980-10-28 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

UPDATE

Query	Endpoint	Data
UPDATE emp SET fname = "Joe", lname = "Mama" WHERE staff_number=3;	PUT /api/v1/testing/emp/primary_key/ Ex: /api/v1/testing/emp/3/	{ "column": ["fname", "lname"], "values": ["Joe", "Mama"], }

```
mysql> select * from emp;
```

```
+-----+-----+-----+-----+-----+
| staff_number | fname | lname | gender | joining |
+-----+-----+-----+-----+-----+
|          1 | Bill  | Gates | M       | 1980-10-28 |
|          2 | John  | doe   | NULL    | NULL       |
|          3 | Joe   | Mama  | M       | 1980-10-28 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Query	Endpoint	Data
UPDATE emp SET fname = "Tim", lname = "Cosby" WHERE fname = "Joe" and lname="Mama" ;	PUT /api/v1/testing/emp/	{ "column": ["fname", "lname"], "values": ["Tim", "Cosby"], "filter": { "column": ["fname", "lname"], "values": ["Joe", "Mama"] } }

```
mysql> select * from emp;  
+-----+-----+-----+-----+-----+  
| staff_number | fname | lname | gender | joining |  
+-----+-----+-----+-----+-----+  
|          1 | Bill  | Gates | M      | 1980-10-28 |  
|          2 | John  | doe   | NULL   | NULL      |  
|          3 | Tim   | Cosby | M      | 1980-10-28 |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

DELETE

Query	Endpoint	Data
DELETE FROM emp WHERE staff_number=3;	DELETE /api/v1/testing/emp/primary_key/ Ex: /api/v1/testing/emp/3/	{ }

```
mysql> select * from emp;
+-----+-----+-----+-----+-----+
| staff_number | fname | lname | gender | joining |
+-----+-----+-----+-----+-----+
|          1 | Bill  | Gates | M       | 1980-10-28 |
|          2 | John  | doe   | NULL    | NULL       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Query	Endpoint	Data
DELETE FROM emp WHERE fname = 'John' and lname = 'Doe' ;	DELETE /api/v1/testing/emp/	{ "filter": { "column": ["fname", "lname"], "values": ["John", "Doe"] } }

```
mysql> select * from emp;  
+-----+-----+-----+-----+-----+  
| staff_number | fname | lname | gender | joining |  
+-----+-----+-----+-----+-----+  
|          1 | Bill  | Gates | M      | 1980-10-28 |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```


NOSQL

Suppose our database 'testing-techfest' has a collection 'collection1' and it contains the following data:

```
_id: ObjectId('6394c19c2243e5ccb40d283c')
author: "Mike"
text: "New text"
tags: Array
  0: "mongodb"
  1: "python"
  2: "pymongo"
```

```
_id: ObjectId('6394c19c2243e5ccb40d283d')
author: "Mike"
text: "My first blog post!"
tags: Array
  0: "mongodb"
  1: "python"
  2: "pymongo"
```

SELECT

Log	Endpoint	Response
find testing-techfest.collection1 filter: { }	GET /api/v1/testing-techfest/collection1/	{ "0": { "_id": { "\$oid": "6394c19c2243e5ccb40d283c" }, "author": "Mike", "text": "New text", "tags": ["mongodb", "python", "pymongo"] }, "1": { "_id": { "\$oid": "6394c19c2243e5ccb40d283d" }, "author": "Mike", "text": "My first blog post!", "tags": ["mongodb", "python", "pymongo"] } }
find testing-techfest.collection1 filter: {"author": "Mike", "text": "My first blog post!" } }	GET /api/v1/testing-techfest/collection1/?filters='author=Mike,text=My first blog post!'	{ "0": { "_id": { "\$oid": "6394c19c2243e5ccb40d283d" }, "author": "Mike", "text": "My first blog post!", "tags": ["mongodb", "python", "pymongo"] } }

INSERT

Log	Endpoint	Data
<pre>insert testing-techfest.collection1 query: { "name": "MikeNew", "location": ""New text" }</pre> <pre>insert testing-techfest.collection1 query: { "name": ""MikeNew", "location": ""New text2" }</pre>	POST /api/v1/testing-techfest/collection1/	[{ "author": "MikeNew", "text": "New text" }, { "author": "MikeNew", "text": "New text2" }]

```
> tags: Array
```

```
_id: ObjectId('639a79a513b4d9fad2d7fc0d')
author: "MikeNew"
text: "New text"
```

```
_id: ObjectId('639a79a513b4d9fad2d7fc0e')
author: "MikeNew"
text: "New text2"
```

UPDATE

Query	Endpoint	Data
update testing-techfest.collection1 query: { "name": "James" } update: { \$set: { "name": "JamesNew" } }	PUT /api/v1/testing-techfest/collection1/	{ "data": { "author": "Mike Updated", "text": "Something new" }, "filter": { "author": "Mike", "text": "New text" } }

```
_id: ObjectId('6394c19c2243e5ccb40d283c')  
author: "Mike Updated"  
text: "Something new"  
> tags: Array
```

```
_id: ObjectId('6394c19c2243e5ccb40d283d')  
author: "Mike"  
text: "My first blog post!"  
> tags: Array
```

DELETE

Query	Endpoint	Data
delete testing-techfest.collection1 query: { "author": "Mike Updated", "text": "Something new" }	DELETE /api/v1/testing-techfest/collection1/	{ "filter": { "author": "Mike Updated", "text": "Something new" } }

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('6394c19c2243e5ccb40d283d')  
author: "Mike"  
text: "My first blog post!"  
> tags: Array
```

```
_id: ObjectId('639a79a513b4d9fad2d7fc0d')  
author: "MikeNew"  
text: "New text"
```

```
id: ObjectId('639a79a513b4d9fad2d7fc0d')
```