

Ian Page
B00748023
CSCI 4145
A1C
May 7, 2020

JS CODE

```
const express023 = require('express');
const mysql023 = require('mysql');
const app023 = express023();
app023.use(express023.json());

const db023 = mysql023.createConnection({
  host: '127.0.0.1',
  user: 'test',
  password: 'password',
  port: '3306',
  database: 'A1C'
});

db023.connect((err023) => {
  if(err023) {
    throw err023;
  }
  console.log('Mysql connected');
});

app023.get('/', (req023, res023) => {
  let sql023 = 'SELECT * FROM new_table';
  let query023 = db023.query(sql023, (err023, records023) => {
    if(err023) {
      res023.statusCode = 400;
      res023.send('Error: Could not retrieve record info');
    }
    res023.send(JSON.stringify(records023));
  });
});

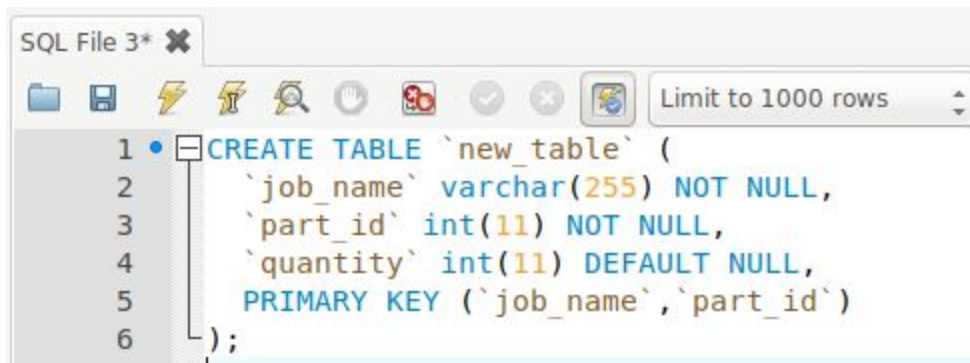
app023.get('/:jobName023/:partId023', (req023, res023) => {
  var sql023 = "SELECT * FROM new_table WHERE job_name = '" + req023.params.jobName023 + "' AND part_id = '" + req023.params.partId023 + "' LIMIT 1";
  let query023 = db023.query(sql023, (err023, record023) => {
    if(err023 || record023.length === 0) {
      res023.statusCode = 400;
      res023.send('Error: Could not retrieve qty info');
    } else {
      res023.send(JSON.stringify(record023));
    }
  });
});

app023.post('/', (req023, res023) => {
  var sql023 = "INSERT INTO new_table VALUES('" + req023.body.jobName023 + "', " + req023.body.partId023 + ", " + req023.body.qty023 + ")";
  let query023 = db023.query(sql023, (err023, result023) => {
    if(err023) {
      res023.statusCode = 400;
      res023.send("ERROR: " + err023.sqlMessage);
    } else {
      res023.send("Success");
    }
  });
});

app023.put('/', (req023, res023) => {
  var sql023 = "UPDATE new_table SET quantity = " + req023.body.qty023 + " WHERE job_name = '" + req023.body.jobName023 + "' AND part_id = '" + req023.body.partId023;
  let query = db023.query(sql023, (err, result) => {
    if(err) {
      res023.statusCode = 400;
      res023.send("ERROR: " + err.sqlMessage);
    } else if(result.affectedRows === 0) {
      res023.statusCode = 400;
      res023.send("ERROR: record not found");
    } else {
      res023.send("Success");
    }
  });
});

app023.listen(3000, () => console.log('Listening on port 3000...'));
```

STARTING SQL

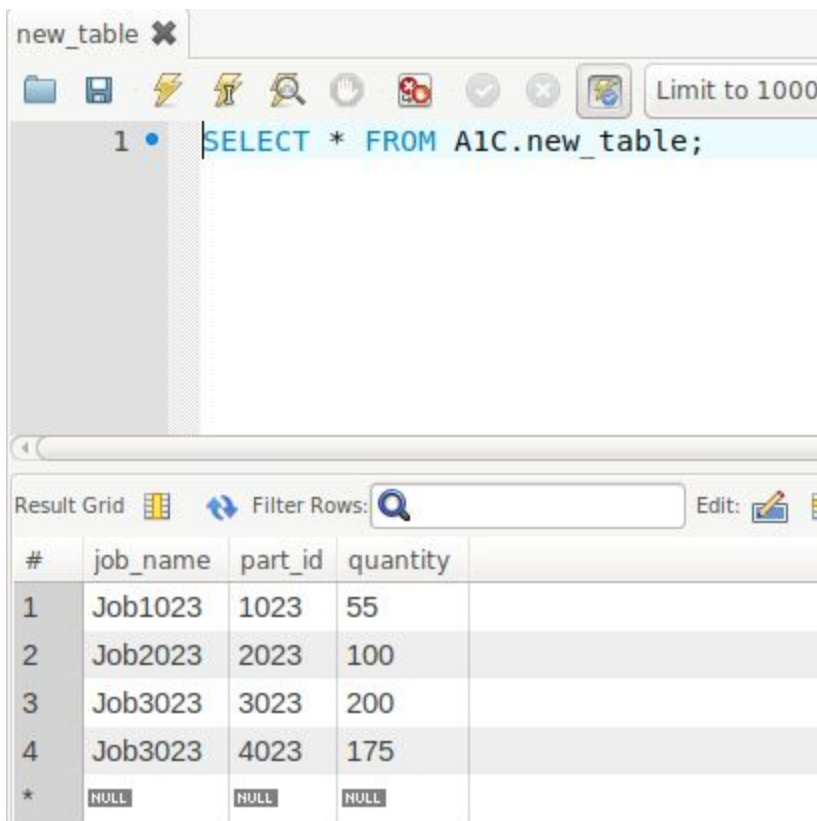


The screenshot shows a SQL editor window titled "SQL File 3*". The editor contains the following SQL code:

```
1 CREATE TABLE `new_table` (  
2     `job_name` varchar(255) NOT NULL,  
3     `part_id` int(11) NOT NULL,  
4     `quantity` int(11) DEFAULT NULL,  
5     PRIMARY KEY (`job_name`, `part_id`)  
6 );
```

The code is syntax-highlighted, with keywords in blue and identifiers in black. The window has a toolbar with various icons and a "Limit to 1000 rows" dropdown menu.

STARTING INFORMATION



The screenshot shows the same SQL editor window, now titled "new_table". The code in the editor is:

```
1 SELECT * FROM A1C.new_table;
```

Below the editor, the "Result Grid" is displayed, showing the results of the query. The grid has columns for row number, job_name, part_id, and quantity. The data is as follows:

#	job_name	part_id	quantity
1	Job1023	1023	55
2	Job2023	2023	100
3	Job3023	3023	200
4	Job3023	4023	175
*	NULL	NULL	NULL

The Result Grid also includes a "Filter Rows" search bar and an "Edit" button.

GET CALLS

GET - All

GET

localhost:3000/

Send

Save

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookiesCode

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

BodyCookiesHeaders (6)Test ResultsStatus: 200 OKTime: 5 msSize: 418 BSave Response

Pretty

Raw

Preview

Visualize

HTML

1 [{"job_name":"Job1023","part_id":1023,"quantity":55},{ "job_name":"Job2023","part_id":2023,"quantity":100}, {"job_name":"Job3023","part_id":3023,"quantity":200}, {"job_name":"Job3023","part_id":4023,"quantity":175}]

GET - JobId/PartId

GET

localhost:3000/Job1023/1023

Send

Save

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookiesCode

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

BodyCookiesHeaders (6)Test ResultsStatus: 200 OKTime: 5 msSize: 258 BSave Response

Pretty

Raw

Preview

Visualize

HTML

1 [{"job_name":"Job1023","part_id":1023,"quantity":55}]

GET - JobId/PartId; Incorrect Arguments

GET

localhost:3000/Job1023/2023

Send

Save

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookiesCode

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

BodyCookiesHeaders (6)Test ResultsStatus: 400 Bad RequestTime: 5 msSize: 248 BSave Response

Pretty

Raw

Preview

Visualize

HTML

1 Error: Could not retrieve qty info

POST/PUT CALLS

POST - Add Record

POST localhost:3000/ Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON Beautify

```
1 {  
2   "jobName023": "TEST",  
3   "partId023": 15,  
4   "qty023": 1111  
5 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 105 ms Size: 210 B Save Response

Pretty Raw Preview Visualize HTML

1 Success

POST - Add Record; Duplicate Record

POST localhost:3000/ Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON Beautify

```
1 {  
2   "jobName023": "TEST",  
3   "partId023": 15,  
4   "qty023": 1111  
5 }
```

Body Cookies Headers (6) Test Results Status: 400 Bad Request Time: 16 ms Size: 264 B Save Response

Pretty Raw Preview Visualize HTML

1 ERROR: Duplicate entry 'TEST-15' for key 'PRIMARY'

PUT - Update Quantity

PUT

localhost:3000/

Send

Save

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Code

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {  
2   "jobName023": "TEST",  
3   "partId023": 15,  
4   "qty023": 2222  
5 }
```

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 41 ms

Size: 210 B

Save Response

Pretty

Raw

Preview

Visualize

HTML

1

Success

PUT - Update Quantity; Non-Existant Record

PUT

localhost:3000/

Send

Save

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Code

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {  
2   "jobName023": "TEST",  
3   "partId023": 16,  
4   "qty023": 2222  
5 }
```

Body

Cookies

Headers (6)

Test Results

Status: 400 Bad Request

Time: 17 ms

Size: 237 B

Save Response

Pretty

Raw

Preview

Visualize

HTML

1

ERROR: record not found

RESULTING INFORMATION

new_table ✕

Limit to 1000 rows

```
1 • SELECT * FROM A1C.new_table;
```

Result Grid

#	job_name	part_id	quantity
1	Job1023	1023	55
2	Job2023	2023	100
3	Job3023	3023	200
4	Job3023	4023	175
5	TEST	15	2222
*	NULL	NULL	NULL