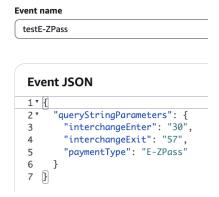## Get Toll Function w/ API Gateway endpoint + Lambda Layer for MySQL database in EC2 instance and also RDS service

```javascript
exports.handler = async (event) => {

    console.log(event);    // Let's take a quick look at what is passed in "event"

    //
    // Create the string that the Lambda function will return.
    var MySQL = require('sync-mysql');
    var connection = new MySQL ({
        host: 'tup84910.cdim6m6mwkbu.us-east-1.rds.amazonaws.com',
        user: 'admin',
        password: 'opensesame',
        database: 'tollschedule'
    });

    //var eventBodyInJSONFormat = Object.fromEntries(new URLSearchParams(event.body));
    //var paymentType = eventBodyInJSONFormat.paymentType;
    //var interchangeEnter = eventBodyInJSONFormat.interchangeEnter;
    //var interchangeExit = eventBodyInJSONFormat.interchangeExit;

    const interchangeEnter = event.queryStringParameters.interchangeEnter;
    const interchangeExit = event.queryStringParameters.interchangeExit;
    const paymentType = event.queryStringParameters.paymentType;

    if (paymentType == "E-ZPass") {
        var query = `
            SELECT toll
            FROM ezpasstollschedule
            WHERE interchangeEnter = ?
            AND interchangeExit = ?
        `;
        var result = connection.query(query, [interchangeEnter, interchangeExit]);
        var toll = result[0].toll.toFixed(2);

        var valueToReturn = {Toll : ` $${toll} ` };
        console.log(` User chose ${paymentType} as their payment choice, and their toll is ${toll} `);
    }

    else {
        var query = `
            SELECT toll
            FROM tollbyplatetollschedule
            WHERE interchangeEnter = ?
            AND interchangeExit = ?
        `;
        var result = connection.query(query, [interchangeEnter, interchangeExit]);
        var toll = result[0].toll.toFixed(2);

        var valueToReturn = {Toll : ` $${toll} ` };
        console.log(` User chose Toll by Plate as their payment choice, and their toll is ${toll} `);
    }


    //
    // Create the response object that will inculde the status code, the headers needed by CORS, and
    // the string to be returned formatted as a JSON data structure.
    //
    const response = {
        statusCode: 200,
        headers: {                                  // Required headers for CORS (cross-origin resource sharing).
            'Access-Control-Allow-Origin': '*',     // This just means content on a web page can come from a
            'Access-Control-Allow-Credentials': true // domain other than the domain of that page.
        },
        body: JSON.stringify(valueToReturn),
    };

    //
    // Return the response object
    //
    return response;
};
```
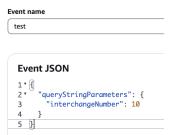
Example test event

Event name

```
testE-ZPass
```

Event JSON

```json
1 ▾ {
2 ▾   "queryStringParameters": {
3         "interchangeEnter": "30",
4         "interchangeExit": "57",
5         "paymentType": "E-ZPass"
6     }
7 }
```

, Toll by Plate is another test event

GetInterchange Function w/ API Gateway endpoint + Lambda layer

```javascript
exports.handler = async (event) => {

    console.log(event);    // Let's take a quick look at what is passed in "event"

    //
    // Create the string that the Lambda function will return.
    var MySQL = require('sync-mysql');
    var connection = new MySQL ({
        host: 'tup84910.cdim6m6mwkbu.us-east-1.rds.amazonaws.com',
        user: 'admin',
        password: 'opensesame',
        database: 'tollschedule'
    });

    var interchangeNumber = event.queryStringParameters.interchangeNumber

    var query = `
        SELECT latitude, longitude
        FROM interchangeinfo
        WHERE interchange = ?
    `;
    var result = connection.query(query, [interchangeNumber]);
    var latitude = result[0].latitude
    var longitude = result[0].longitude

    const valueToReturn = {
        Latitude: latitude,
        Longitude: longitude,
    };
    //
    // Create the response object that will inculde the status code, the headers needed by CORS, and
    // the string to be returned formatted as a JSON data structure.
    //
    const response = {
        statusCode: 200,
        headers: {                               // Required headers for CORS (cross-origin resource sharing).
            'Access-Control-Allow-Origin': '*',      // This just means content on a web page can come from a
            'Access-Control-Allow-Credentials': true  // domain other than the domain of that page.
```

Example Test event

Event name

```
test
```

Event JSON

```json
1 ▾ {
2 ▾   "queryStringParameters": {
3         "interchangeNumber": 10
4     }
5 }
```

(only 1 test event since we invoke it twice for getting on and getting off in the html)

for HTML, see TollCalculator2.html