How to drive rws.js with Postman

Here rws is executing JavaScript under an asynchronous Node runtime using an Express router.

Download as follows:

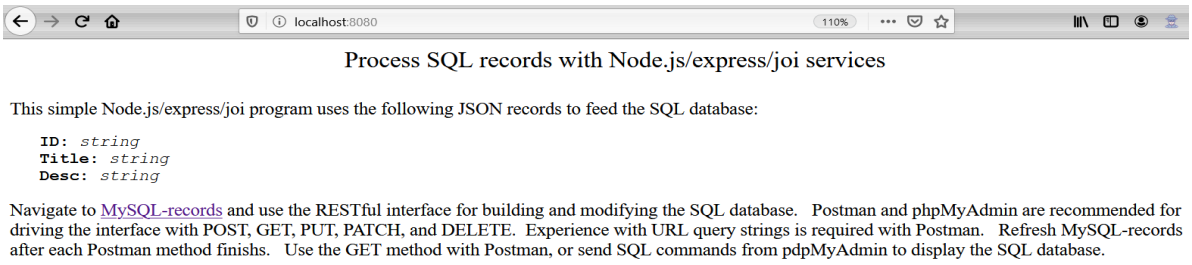
```
$ svn export https://github.com/rory911/misc/tree/master/js-express
$ cd js-express
```

In an attempt to simplify, `package.json` and `node_modules` are included, but there is no promise that it will be compatible in an unconfigured environment. If these additional packages do not operate, then you will need to configure as you normally would for node.js, for example:

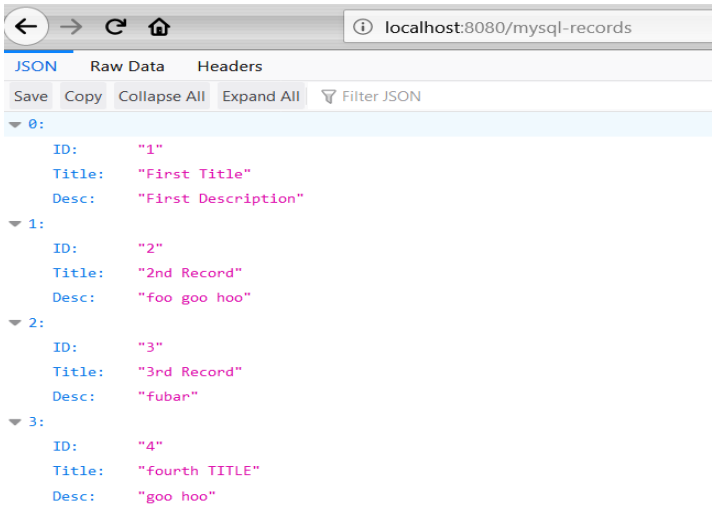
```
$ npm init
$ npm install express
```

You might want to install nodemon, but it's not required. If you would like to modify this code, with say Joi for validation with Node.js and Express, it's another optional package to consider. In all cases, it is assumed you are family with the Node.js environment and how to configure it.

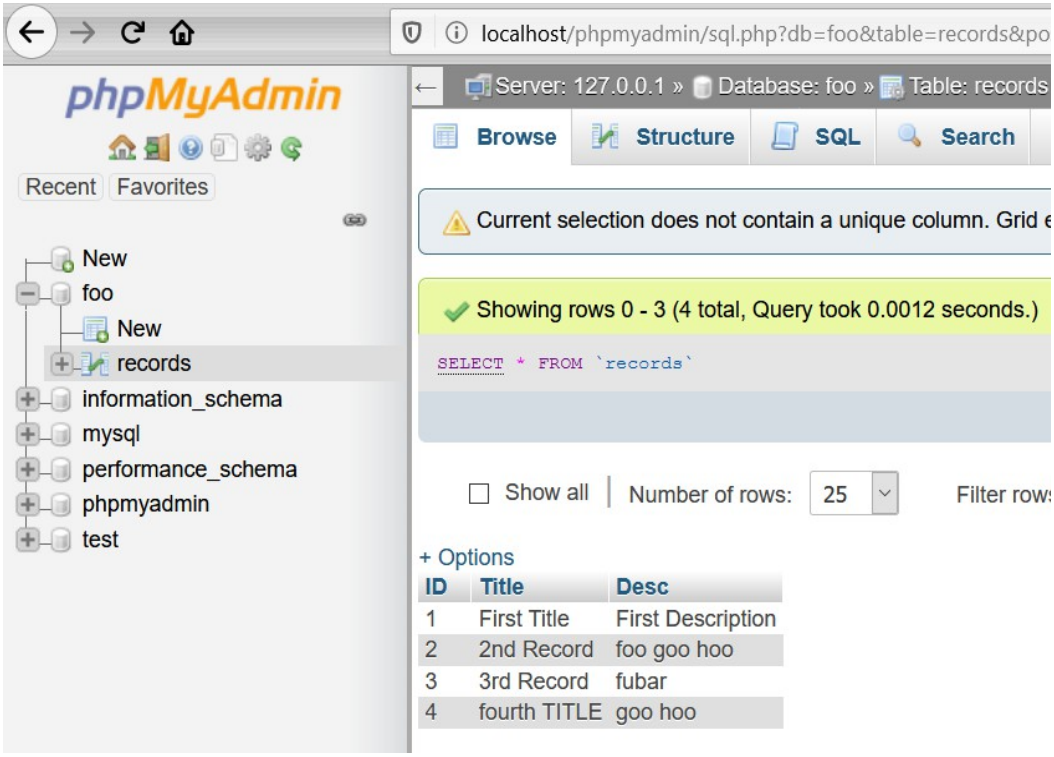
Once rws.js is running under Node, navigate to `localhost:8080` using a web brouser, but note that you might need to grant permission to start the session.



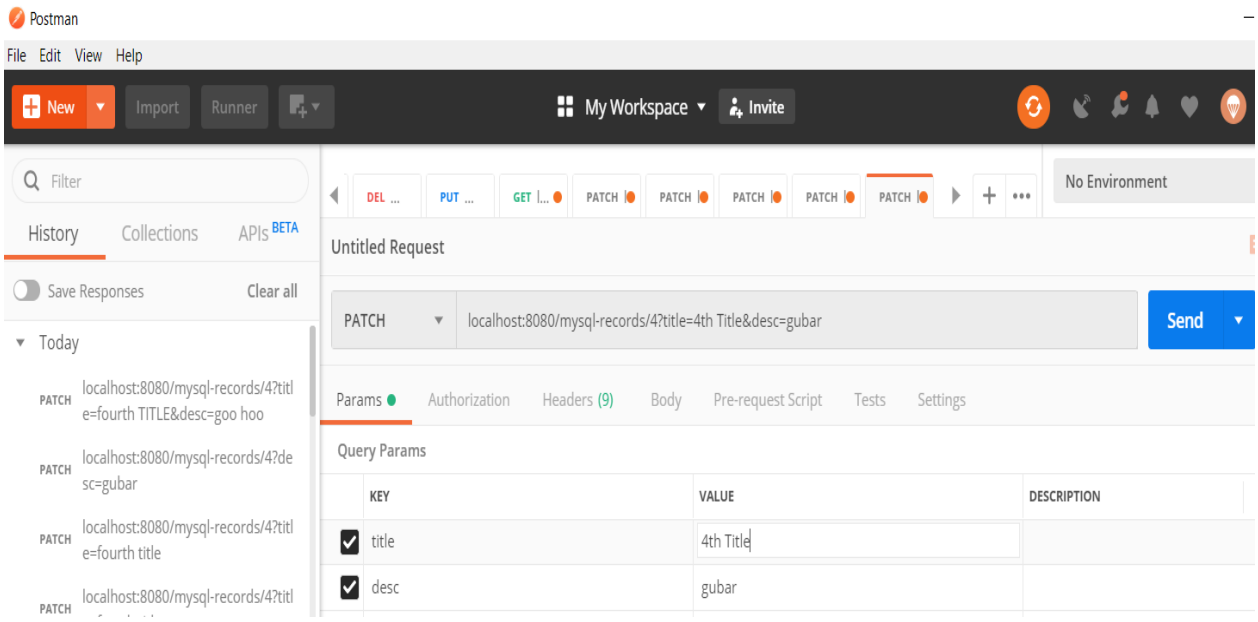
As in the Golang examples, this code opens a MySQL database named "'foo" with entries name "records." Once the MySQL database is initiated with at least one entry, just click the hyperlink to dump the entire database:



Compare this to the phpMyAdmin display of the database:



Now, you can drive the RESTful API through Postman, as follows:



with the following result in phpMyAdmin:

ID	Title	Desc
1	First Title	First Description
2	2nd Record	foo goo hoo
3	3rd Record	fubar
4	4th Title	gubar

The other methods `GET, GET/<id>, POST, PATCH/<id>` operate as expected. As in previous coding examples, this is very simple-minded code that illustrates the only the basic fundamentals of Node.js, Express, MySQL, JSON, Postman, and phpMyAdmin conforming to a RESTful API convention that's marked more by what is *doesn't do* than what it *does do*.