

How to drive rws using Postman

The rws module should download with the following “vendor” dependencies:

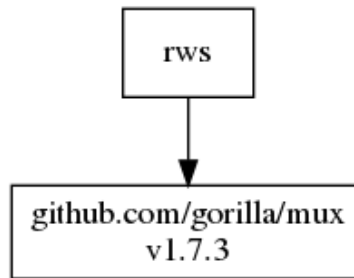
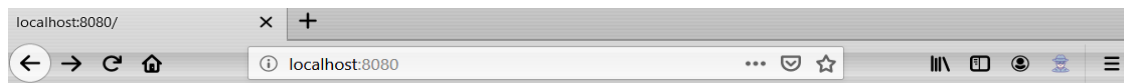


Figure 1 Dependency output from dep

Download and build as follows:

```
$ svn export https://github.com/rory911/misc/trunk/rws
$ cd rws
$ go build rws
$ rws
```

You might have to give firewall permission in order to continue. Now, using a web browser, navigate to the home page at `localhost:8080` as follows:



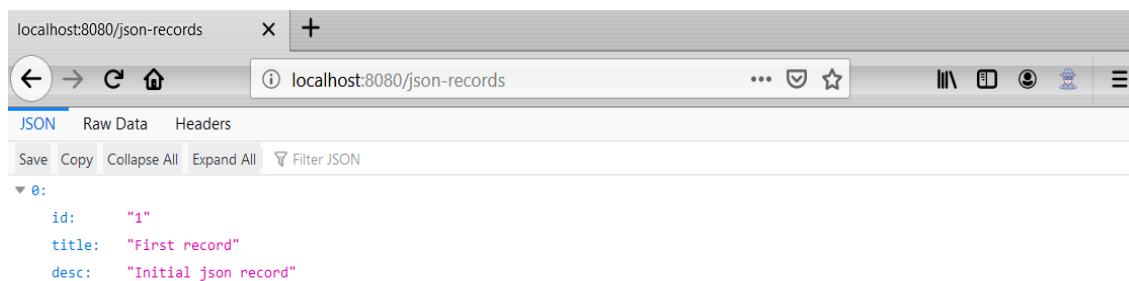
Process JavaScript Object Notation (JSON) Records with Gorilla MUX

This simple Golang program uses the following JSON records:

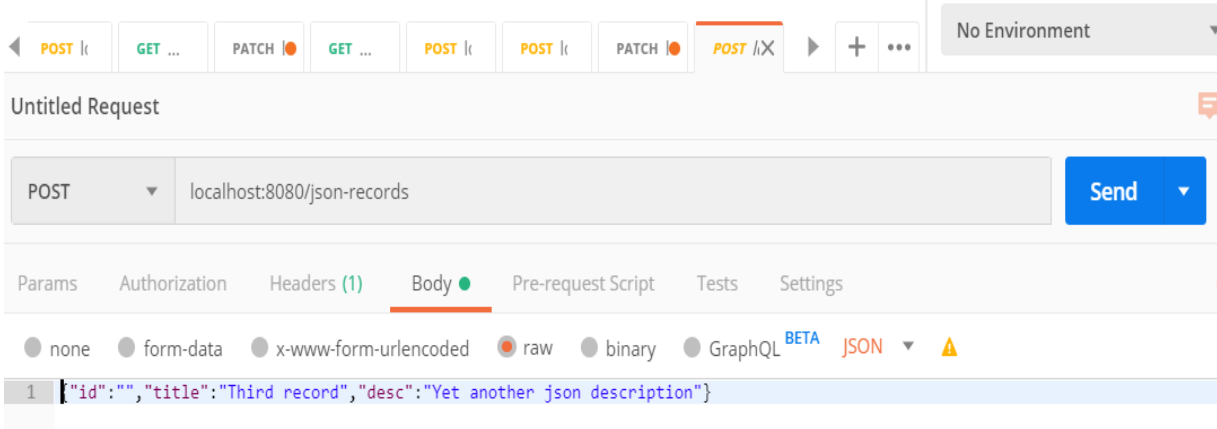
```
ID: string
Title: string
Desc: string
```

Navigate to [json-records](#) and use the RESTful interface for building and modifying the JSON records database. Postman is recommended for driving the interface with POST, GET, PUT, PATCH, and DELETE. Refresh json-records after each Postman method finishes, or use the GET method in order to view the current state of the database.

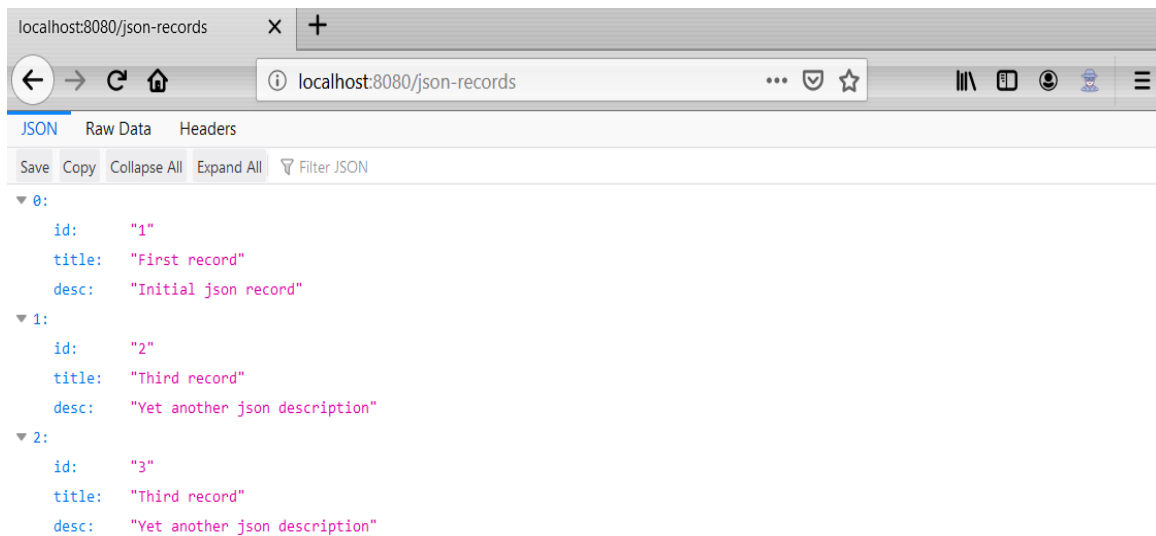
From the home page, you can click on the hyperlink (json-records), or you can navigate to the following URL: `localhost:8080/json-records`



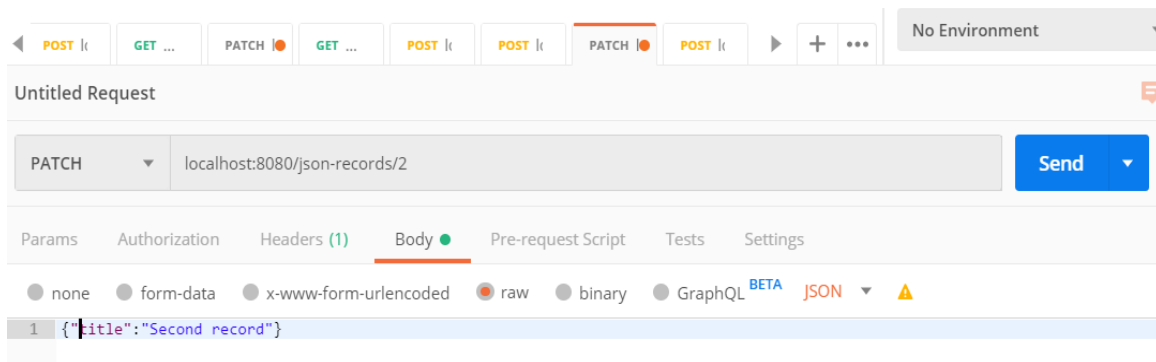
Next, start postman and add some records.



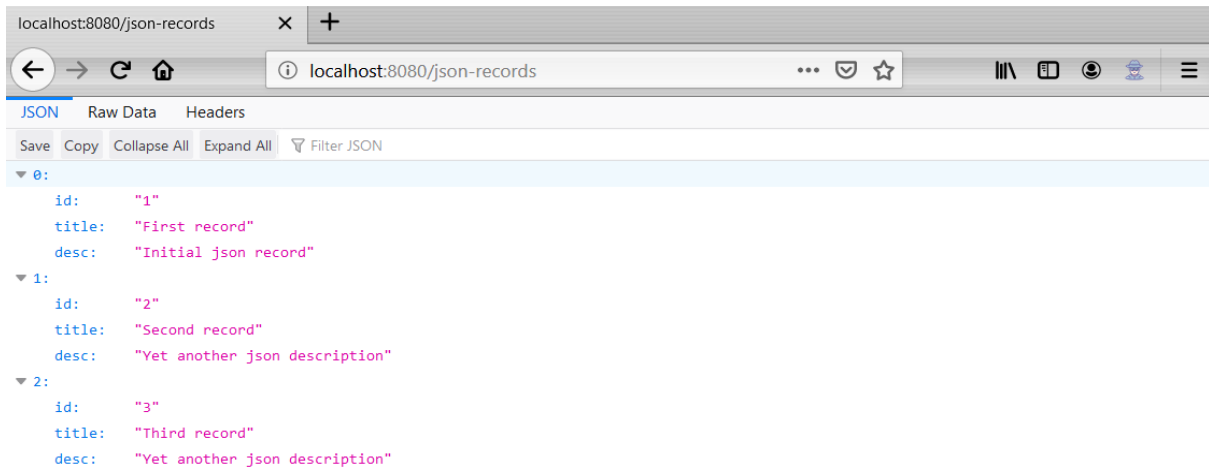
And refresh the browser:



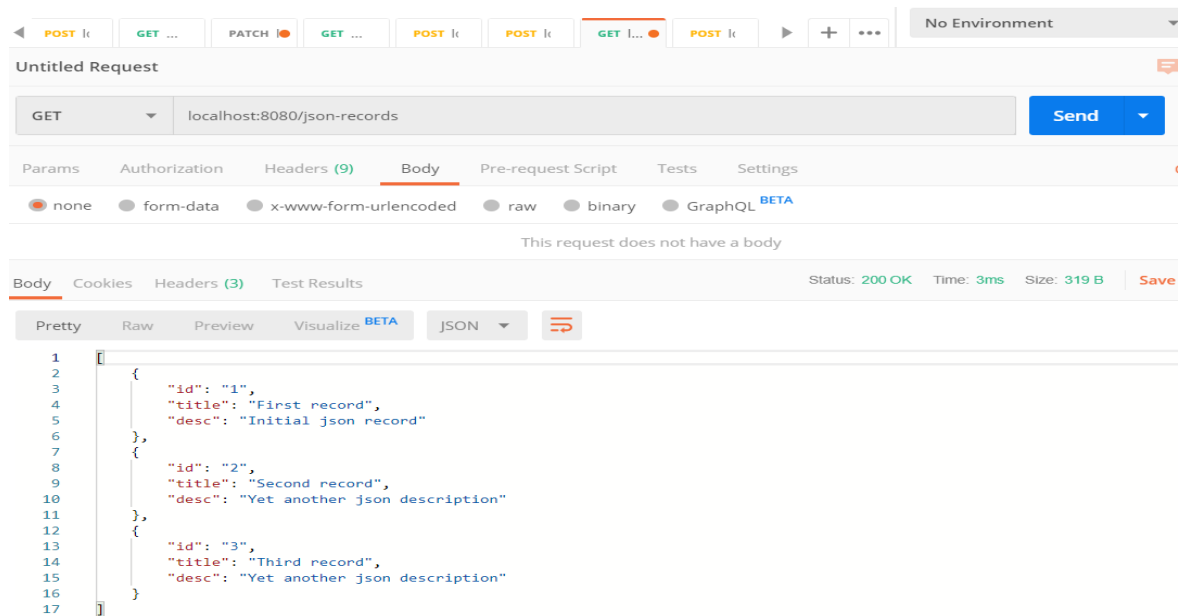
You can see the second record doesn't look right. We can overwrite it with a PUSH method, but let's use the PATCH method for the sake of brevity.



Again, refresh the browser.



We can also use Postman to send a GET all message.



Yeah, it's a pretty simple-minded database, but it illustrates the basics of MUX, JSON, and Postman.