**GitHub:**

https://github.com/fraigo/node-express-rest-api-examplenode-express-rest-api-example

# node-express-rest-api-example

A Basic Node.js/Express REST API implementation example.

Full Tutorial at <https://developerhowto.com/2018/12/29/build-a-rest-api-with-node-js-and-express-js/>

**Prerequisites**

For Windows

* Python 2.7 (for microsoft build tools)
* Install Microsoft build tools (to build sqlite using node-gyp)
  + Instructions here <https://github.com/nodejs/node-gyp#on-windows>
  + Or install using npm (npm install --global windows-build-tools)
* Node-gyp (npm install --global node-gyp)

**Usage**

* Run npm install to installl dependencies
* Run npm run start to start the local server
* Load http://localhost:8000 to test the endpoint. It will display a json result {"message":"Ok"}

**API Endpoints**

**GET /api/users**

Get a list of users

{

"message": "success",

"data": [

{

"id": 1,

"name": "admin",

"email": "admin@example.com",

"password": "a66abb5684c45962d887564f08346e8d"

},

{

"id": 2,

"name": "user",

"email": "user@example.com",

"password": "4da49c16db42ca04538d629ef0533fe8"

}

]

}

**GET /api/user/{id}**

Get user information by user id

{

"message": "success",

"data": {

"id": 1,

"name": "admin",

"email": "admin@example.com",

"password": "a66abb5684c45962d887564f08346e8d"

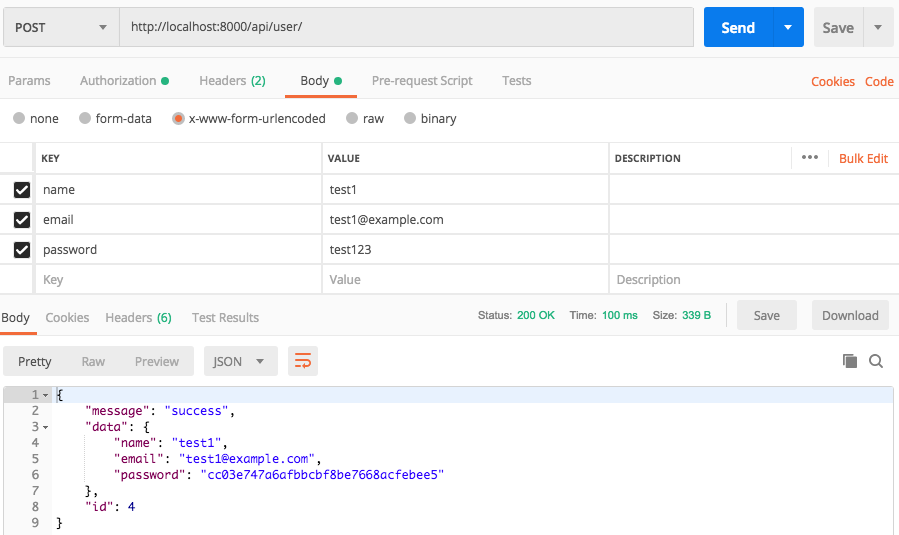
}

}

**POST /api/user/**

To create a new user based on POST data (x-www-form-url-encoded)

* name: User name
* email: User email
* password: User password

[](https://camo.githubusercontent.com/2a2c74c8fbdbe6b5ba2c65cc65e9c94234dfde0915478e6fa9de3a488da1af19/68747470733a2f2f646576656c6f706572686f77746f2e636f6d2f77702d636f6e74656e742f75706c6f6164732f323031382f31322f506f73744d616e2d504f53542d726571756573742e706e67)

**PATCH /api/user/{id}**

To update user data by id, based on POST data (x-www-form-url-encoded)

* name: User name
* email: User email
* password: User password

You can send only one attribute to update, the rest of the info remains the same.

In this example, using CURL you can update the user email:

curl -X PATCH -d "email=user@example1.com" http://localhost:8000/api/user/2

**DELETE /api/user/{id}**

To remove a user from the database by user id.

This example is using the curl command line

curl -X "DELETE" http://localhost:8000/api/user/2

The result is:

{"message":"deleted","rows":1}