

# Tylar and the Boyz Presents

This is a quick tutorial on how to throw json data at our server and database to get a response.

## Prerequisite

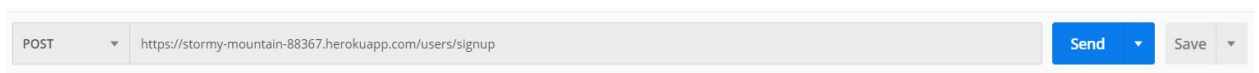
- Websites URL: <https://still-spire-82139.herokuapp.com/> (returns not found for now)
- Install Postman it makes dealing with json requests real simple.

<https://www.postman.com/downloads/>

## Explaining Postman

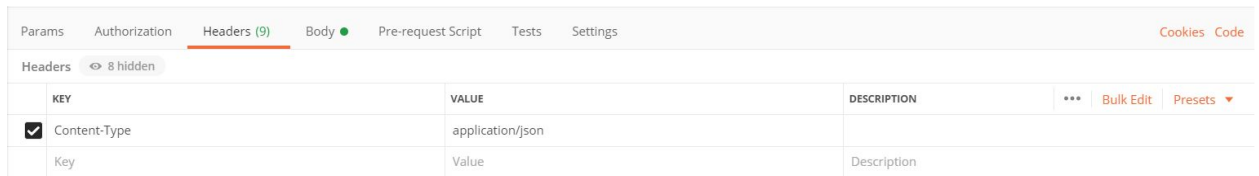
In postman we will only ever reference the

- URL bar



The image shows the top section of the Postman interface. On the left, a dropdown menu is set to 'POST'. Next to it is the URL bar containing 'https://stormy-mountain-88367.herokuapp.com/users/signup'. To the right of the URL bar are two buttons: a blue 'Send' button and a grey 'Save' button with a dropdown arrow.

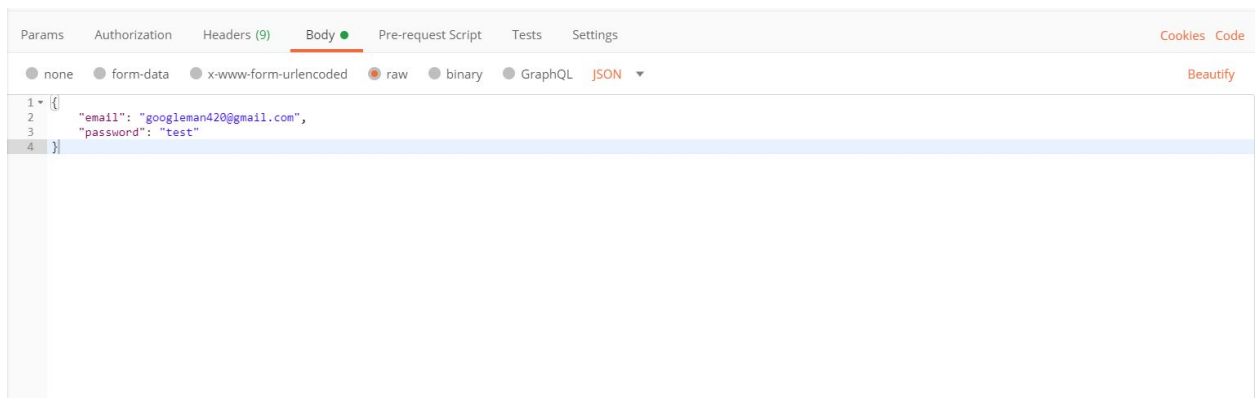
- Headers section



The image shows the 'Headers' tab selected in the Postman interface. The tab bar at the top includes 'Params', 'Authorization', 'Headers (9)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Headers' section is expanded, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. There are also buttons for 'Bulk Edit' and 'Presets'. The table contains one header: 'Content-Type' with the value 'application/json'.

KEY	VALUE	DESCRIPTION
Content-Type	application/json	

- Body section

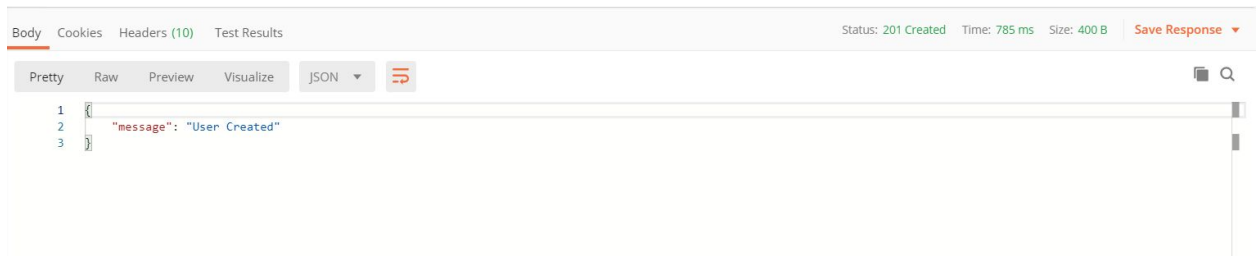


The image shows the 'Body' tab selected in the Postman interface. The tab bar at the top includes 'Params', 'Authorization', 'Headers (9)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' section is expanded, showing a list of body types: 'none', 'form-data', 'x-www-form-urlencoded', 'raw', 'binary', 'GraphQL', and 'JSON'. The 'JSON' option is selected. The body content is a JSON object: 

```
{ "email": "googleman420@gmail.com", "password": "test" }
```

. There is a 'Beautify' button on the right.

- Response section



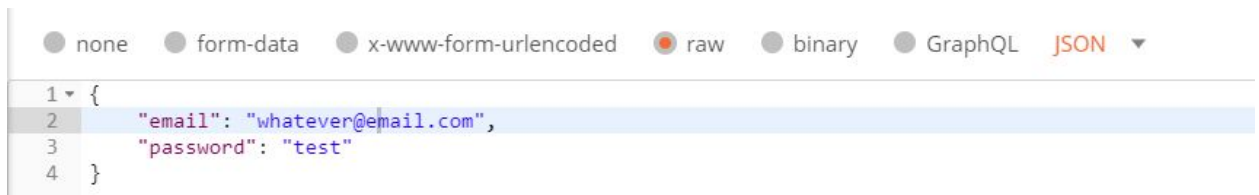
## Step 1: Sign-up and Login

Our first test will be a post request to signup for authorization through an email and password.

1. In the URL bar select POST then enter <https://still-spire-82139.herokuapp.com/users/signup>
2. In the Header section enter

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/json

3. In the Body section enter (make sure raw and JSON are selected)



4. Press the send button to the right of the URL bar
5. Then you should get this response

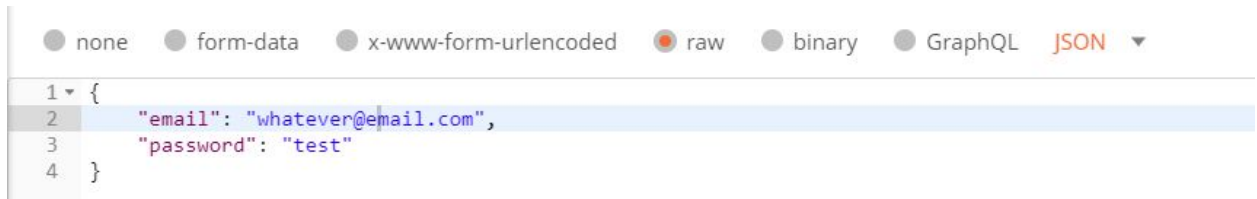


Next we will login

1. In the URL bar keep POST selected then enter <https://still-spire-82139.herokuapp.com/users/login>
2. In the Header section enter

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/json

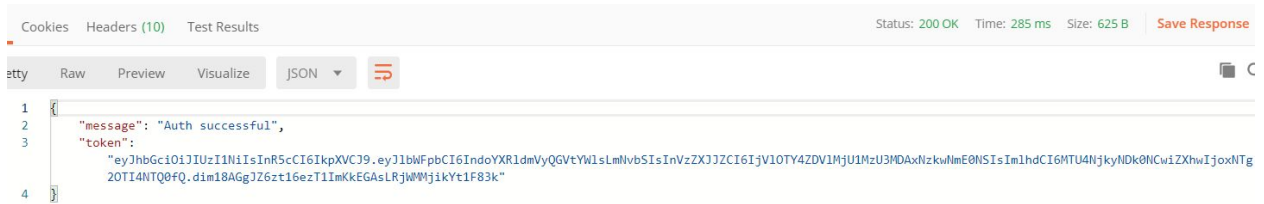
3. In the Body section enter (make sure raw and JSON are selected)



The screenshot shows a REST client interface with a dropdown menu at the top containing options: none, form-data, x-www-form-urlencoded, raw (selected), binary, GraphQL, and JSON. Below the menu, a JSON body is entered in a text area:

```
1 {  
2   "email": "whatever@email.com",  
3   "password": "test"  
4 }
```

4. Press the send button to the right of the URL bar
5. Then you should get this response



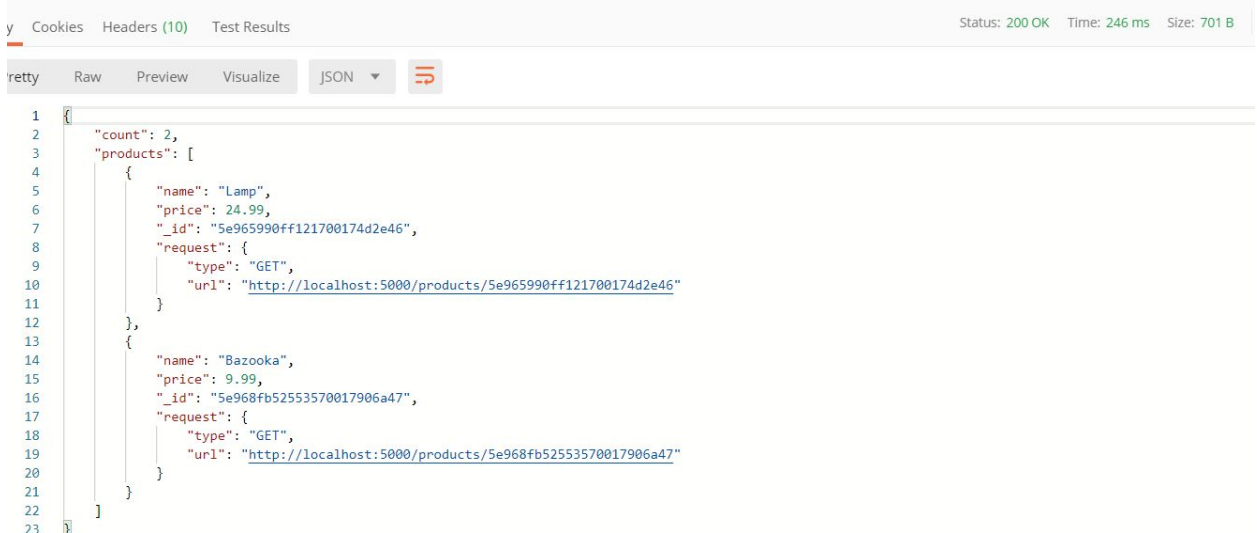
The screenshot shows the response of a REST client. The status bar at the top indicates "Status: 200 OK", "Time: 285 ms", and "Size: 625 B". The response is displayed in a JSON format:

```
1 {  
2   "message": "Auth successful",  
3   "token":  
4     "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmFpbCI6ImdoYXR1dmVycG9tZW50IiwiaXN0IjoiZjV1OTY4ZDV1MjU1MzU3MDAxNzkwNmE0NSIsIm1hdCI6MTU4NjkyNDk0NCwiZXhwIjoxNDg2OTI4NTQ0fQ.dim18AGgJ26zt16ezT1ImKKEGAsLRjWjYt1F83k"
```

6. That token is a json web token and is a valid authorization value that lasts for one hour.  
Copy it and save it until we finish testing.

## Step 2: Get All Products

1. In the URL bar select GET then enter <https://still-spire-82139.herokuapp.com/products>
2. The Header and Body section can be blank.
3. Press the send button to the right of the URL bar
4. Then you should get a response like this



The screenshot shows the response of a REST client for a GET request. The status bar at the top indicates "Status: 200 OK", "Time: 246 ms", and "Size: 701 B". The response is displayed in a JSON format:

```
1 {  
2   "count": 2,  
3   "products": [  
4     {  
5       "name": "Lamp",  
6       "price": 24.99,  
7       "_id": "5e965990ff121700174d2e46",  
8       "request": {  
9         "type": "GET",  
10        "url": "http://localhost:5000/products/5e965990ff121700174d2e46"  
11      }  
12    },  
13    {  
14      "name": "Bazooka",  
15      "price": 9.99,  
16      "_id": "5e968fb52553570017906a47",  
17      "request": {  
18        "type": "GET",  
19        "url": "http://localhost:5000/products/5e968fb52553570017906a47"  
20      }  
21    }  
22  ]  
23 }
```

## Step 3: Create a Product

1. In the URL bar select POST then enter <https://still-spire-82139.herokuapp.com/products>
2. In the Header section enter

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/json
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFnZWpCI6IndoYi...

For the Authorization key you will need to enter “Bearer <replace with json web token>” as a value.

3. In the Body section enter (make sure raw and JSON are selected)

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

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

```

1 {
2   "name": "Bazooka",
3   "price": "9.99"
4 }
```

Replace Bazooka and 9.99 with whatever you like.

4. Then you should get a response like this

Cookies Headers (10) Test Results Status: 201 Created Time: 197 ms Size: 584 B

pretty Raw Preview Visualize JSON

```

1 {
2   "msg": "Created product successfully",
3   "createdProduct": {
4     "name": "Bazooka",
5     "price": 9.99,
6     "_id": "5e968fb52553570017906a47",
7     "request": {
8       "type": "GET",
9       "url": "http://localhost:5000/products/5e968fb52553570017906a47"
10    }
11  }
12 }
```

5. Copy the “\_id” value given.

## Step 4: Get a single Product

1. In the URL bar select GET then enter <https://still-spire-82139.herokuapp.com/products/<replace with product id>>

GET <https://stormy-mountain-88367.herokuapp.com/products/5e968fb52553570017906a47>

2. In the Header section enter

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFnZWpCI6IndoYi...

you use the same web token throughout the entire demo.

3. Leave the Body section empty
4. Then you should get a response like this

The screenshot shows a REST client interface with tabs for Cookies, Headers (10), and Test Results. The status bar indicates a 200 OK response with a time of 1106 ms and a size of 538 B. The response body is displayed in JSON format, showing a product object with an ID, name, and price, and a request object with type, description, and URL.

```

1 {
2   "product": {
3     "id": "5e968fb52553570017906a47",
4     "name": "Bazooka",
5     "price": 9.99
6   },
7   "request": {
8     "type": "GET",
9     "description": "GET_ALL_PRODUCTS",
10    "url": "http://localhost/products"
11  }
12 }

```

of the product you created

## Step 5: Update a Product

1. In the URL bar select PATCH then enter `https://still-spire-82139.herokuapp.com/products/<replace with product id>`
2. In the Header section enter

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFnZW50IjY7...

3. In the Body section enter

The screenshot shows a REST client interface with tabs for none, form-data, x-www-form-urlencoded, raw, binary, GraphQL, and JSON. The response body is displayed in JSON format, showing a request object with type, description, and URL.

```

1 [
2   { "propName": "name", "value": "akoozaB" },
3   { "propName": "price", "value": "20.01" }
4 ]

```

We use square brackets first, then inside use curly braces, this format allows you to change both name and price or just one of the two in a single response.

4. Then you should get a response like this

The screenshot shows a REST client interface with tabs for Cookies, Headers (10), and Test Results. The status bar indicates a 200 OK response with a time of 555 ms and a size of 488 B. The response body is displayed in JSON format, showing a message and a request object with type, description, and URL.

```

1 {
2   "message": "Product updated",
3   "request": {
4     "type": "GET",
5     "description": "GET_ALL_PRODUCTS",
6     "url": "http://localhost:5000/products/5e968fb52553570017906a47"
7   }
8 }

```

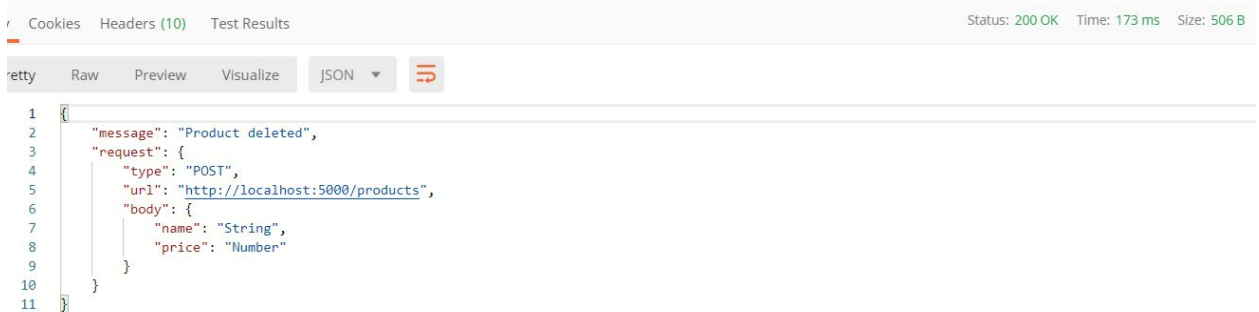
5. Feel free to perform the get single product request to check on your changes.

## Step 6: Delete a Product

1. In the URL bar select DELETE then enter `https://still-spire-82139.herokuapp.com/products/<replace with product id>`
2. In the Header section enter

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImdoY...

3. Leave the Body section empty
4. Then you should get a response like this



5. Feel free to perform the get single product request to check on your changes.

## Step 7: Delete a User

1. Copy your json web token again and visit here <https://jwt.io/>
2. Scroll down and paste your token in the encode section

Encoded

PASTE A TOKEN HERE

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFPbCI6IndoYXRldmVpQGVTYWwzLmNvbSIsInVzZXJJZCI6IjVlOTY4ZDV1MjU1MzUzMDAxNzkwNmE0NSIsIm1hdCI6MTU0NjkyNDk0NCwiZXhwIjoxNTg2OTI4NTQ0fQ.dim18AGgJZ6zt16ezT1ImKkEGAsLRjWMMjikYt1F83k

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "email": "whatever@email.com",
  "userId": "5e968d5e2553570017906a45",
  "iat": 1586924944,
  "exp": 1586928544
}
```

- Under Decoded in the PAYLOAD section copy the userId
3. In the URL bar select DELETE then enter `https://still-spire-82139.herokuapp.com/users/<replace with user id>`

4. In the Header section enter

	KEY	VALUE
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImdoYV...

5. Leave the Body section empty
6. Then you should get a response like this

Cookies Headers (10) Test Results Status: 200 OK Time: 1171 ms Size: 395 B

ttty Raw Preview Visualize JSON 

```
1 {
2   "message": "User deleted"
3 }
```

7. Praise the lord since your fake email isn't stored on some sketchy database.