Contents

Routing:

Routing refers to how an application's endpoints (URIs) respond to client requests. For an introduction to routing, see [Basic routing](#).

You define routing using methods of the Express `app` object that correspond to HTTP methods; for example, `app.get()` to handle GET requests and `app.post` to handle POST requests. For a full list, see [app.METHOD](#). You can also use [app.all()](#) to handle all HTTP methods and [app.use()](#) to specify middleware as the callback function (See [Using middleware](#) for details).

Route paths

Route paths, in combination with a request method, define the endpoints at which requests can be made. Route paths can be strings, string patterns, or regular expressions.

Check for more info : https://expressjs.com/en/guide/routing.html

This route path will match requests to / newsletter.

app.get() Function:
The **app.get()** function returns the value *name* app setting.
The **app.set()** function is used to assigns the setting name to value. This function is used to get that values which are assigned.

Syntax:
```
app.get(name)
```

Installation of express module:
1. You can visit the link to Install express module. You can install this package by using this command.
   ```
   npm install express
   ```

2. After that, you can just create a folder and add a file for example, index.js. To run this file you need to run the following command.
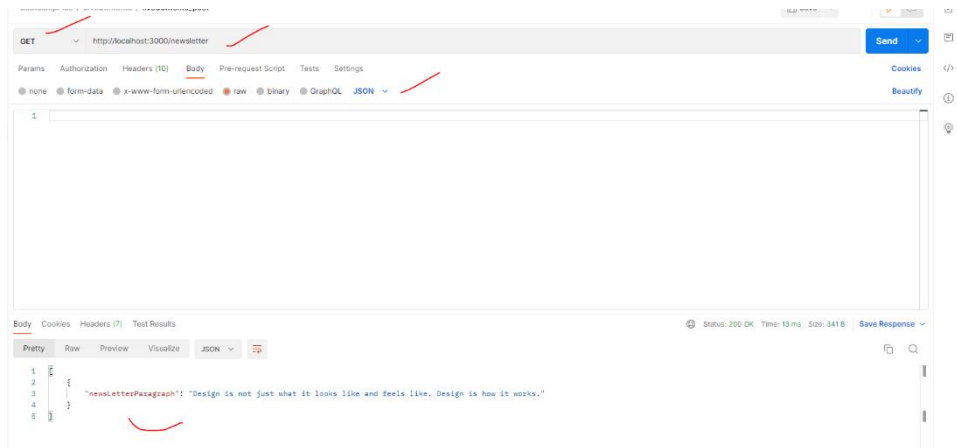   ```
   node server.js
   ```

*More info: https://www.geeksforgeeks.org/express-js-app-get-function/*

**Code:**

```
app.get('/newsletter', function (req, res) {
  res.json(  //it will send the json
    [{
      "newsLetterParagraph": "Design is not just what it looks like and feels
like. Design is how it works.",
    }]


  )
});
```

Output:

GET: http://localhost:3000/newsletter

GET   http://localhost:3000/newsletter   Send   Cookies

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

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

1

Body Cookies Headers (7) Test Results   Status: 200 OK Time: 13 ms Size: 341 B   Save Response

Pretty Raw Preview Visualize JSON

1
2   {
3   "newsLetterParagraph": "Design is not just what it looks like and feels like. Design is how it works."
4   }
5

app.post() Function

The **app.post() function** routes the HTTP POST requests to the specified path with the specified callback functions.

Syntax:
`app.post(path, callback [, callback ...])`

Arguments:

1. **Path:** The path for which the middleware function is invoked and can be any of:
   - A string representing a path.
   - A path pattern.
   - A regular expression pattern to match paths.
   - An array of combinations of any of the above.

2. **Callback:** Callback functions can be:
   - A middleware function.
   - A series of middleware functions (separated by commas).
   - An array of middleware functions.
   - A combination of all of the above.

Installation of express module:
1. You can visit the link to *Install express module*. You can install this package by using this command.
   `npm install express`

2. After that, you can just create a folder and add a file for example, index.js. To run this file you need to run the following command.
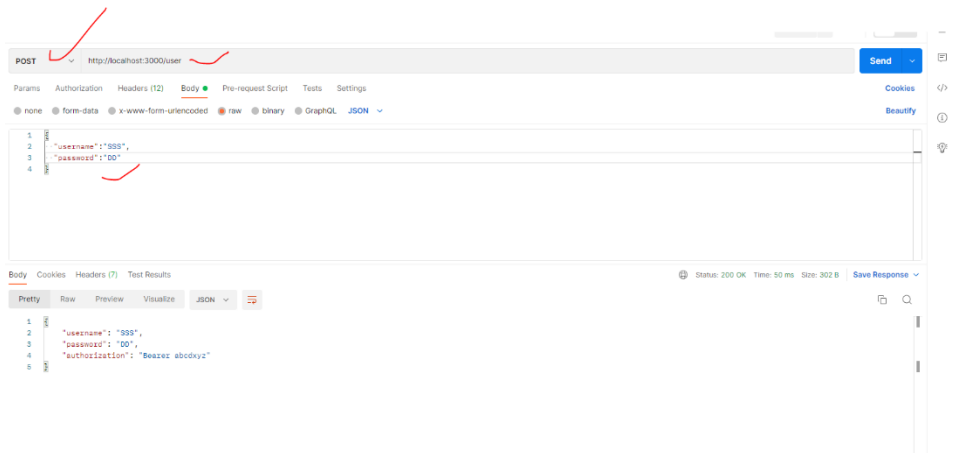   `node server.js`

More Info: *https://www.geeksforgeeks.org/express-js-app-post-function/*

Code:

```
//************POST*************** */
app.post('/user', (req, res) => {
  const { username, password } = req.body;  // get the request from UI in
json  format
  const { authorization } = req.headers;  //passing the headers with
authorization token
  res.send({
    username,
    password,
    authorization,
  });
  console.log(req.body); // see the results in node terminal

});
```

Output:

POST: *http://localhost:3000/user*

app.put() Function:

The **app.put() function** routes the HTTP PUT requests to the specified path with the specified callback functions.
Syntax:
```
app.put(path, callback [, callback ...])
```

Arguments:
1. **Path:** The path for which the middleware function is invoked and can be any of:
   - A string representing a path.
   - A path pattern.
   - A regular expression pattern to match paths.
   - An array of combinations of any of the above.

2. **Callback:** Callback functions can be:
   - A middleware function.
   - A series of middleware functions (separated by commas).
   - An array of middleware functions.
   - A combination of all of the above.

Installation of express module:
1. You can visit the link to *Install express module*. You can install this package by using this command.
   ```
   npm install express
   ```

2. After that, you can just create a folder and add a file for example, index.js. To run this file you need to run the following command.
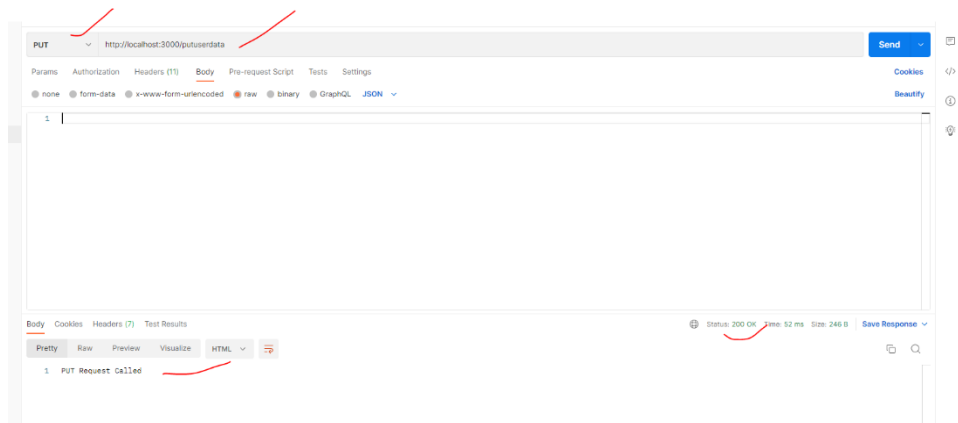   ```
   node node server.js
   ```

More info: https://www.geeksforgeeks.org/express-js-app-put-function/

Code:

```
//************PUT************** */
app.put('/putuserdata:', (req, res) => {
  res.send("PUT Request Called")
})
```

Output:

PUT: http://localhost:3000/putuserdata

app.delete() Function:

The **app.delete()** function is used to route the HTTP DELETE requests to the path which is specified as parameter with the callback functions being passed as parameter.
Syntax:

```
app.delete(path, callback)
```

Parameters:

1. **path:** It is the path for which the middleware function is being called.
2. **callback:** It is a middleware function or a series/array of middleware functions.

Installation of express module:

1. You can visit the link to Install express module. You can install this package by using this command.

```
npm install express
```

2. After that, you can just create a folder and add a file for example, index.js. To run this file you need to run the following command.

```
node server.js
```

More info: https://www.geeksforgeeks.org/express-js-app-delete-function/

Code:

```javascript
//************PUT*************** */
app.put('/putuserdata:', (req, res) => {
  res.send("PUT Request Called")
})
```

Output:
DELETE:

http://localhost:3000/user/1