Indian Institute of Technology Kanpur

Undertaking from the Students as per the provisions of anti - ragging verdict by the Hon'ble Supreme Court

I, Mr./Ms. ...., Program: .... Dept. ...., student of Indian Institute of Technology, Kanpur do hereby undertake on this day.....................month.....................year ...., the following with respect to above subject and Office Order No: Dir-IITK/2007/00/54, June 21, 2007.

1) That I have read and understood the directives of the Hon'ble Supreme Court of India on anti-ragging and the measures proposed to be taken in the above references. (available at http://www.iitk.ac.in/dosa/)

2) That I understand the meaning of Ragging and know that the ragging in any form is a punishable offence and the same is banned by the Court of Law.

3) That I have not been found or charged for my involvement in any kind of ragging in the past. However, I undertake to face disciplinary action/legal proceedings including expulsion from the Institute if the above statement is found to be untrue or the facts are concealed, at any stage in future.

4) That I shall not resort to ragging in any form at any place and shall abide by the rules/laws prescribed by the Courts, Govt. of India and the Institute authorities for the purpose from time to time.

Signature of Student

I hereby fully endorse the undertaking made by my child / ward.

Signature of Mother / Father and or Guardian

Witness:

Express

Serving static files in Express

To serve static files such as images, CSS files, and JavaScript files, use the `express.static` built-in middleware function in Express.

The function signature is:

```
express.static(root, [options])
```

The `root` argument specifies the root directory from which to serve static assets. For more information on the `options` argument, see express.static.

For example, use the following code to serve images, CSS files, and JavaScript files in a directory named `public`:

```
app.use(express.static('public'))
```

Now, you can load the files that are in the `public` directory:

```
http://localhost:3000/images/kitten.jpg
http://localhost:3000/css/style.css
http://localhost:3000/js/app.js
http://localhost:3000/images/bg.png
http://localhost:3000/hello.html
```

> Express looks up the files relative to the static directory, so the name of the static directory is not part of the URL.

To use multiple static assets directories, call the `express.static` middleware function multiple times:

```
app.use(express.static('public'))
app.use(express.static('files'))
```

Express looks up the files in the order in which you set the static directories with the `express.static` middleware function.

> NOTE: For best results, use a reverse proxy cache to improve performance of serving static assets.

To create a virtual path prefix (where the path does not actually exist in the file system) for files that are served by the `express.static` function, specify a mount path for the static directory, as shown below:

```
app.use('/static', express.static('public'))
```

Now, you can load the files that are in the `public` directory from the `/static` path prefix.

```
http://localhost:3000/static/images/kitten.jpg
http://localhost:3000/static/css/style.css
http://localhost:3000/static/js/app.js
http://localhost:3000/static/images/bg.png
http://localhost:3000/static/hello.html
```

However, the path that you provide to the `express.static` function is relative to the directory from where you launch your `node` process. If you run the express app from another directory, it's safer to use the absolute path of the directory that you want to serve:

```
const path = require('path')
app.use('/static', express.static(path.join(__dirname, 'public')))
```

For more details about the `serve-static` function and its options, see serve-static.

Previous: Basic Routing   Next: More examples