

Dear New Entrant to IIT Kanpur

The computer centre would like to extend a warm welcome to you. The centre provides facilities to support and enhance the educational mission of the institute; for enriching the educational experience of the students, faculty and the staff support their research and academic activities; and for providing automation support to the institute. This facility is needed for your day to day research as well as courses and assignments. **Your computer centre account will be closed after completion of your degree or taking no-dues or on instructions from a competent authority.**

A computer account is created for every individual on admission to the institute. The account allows you to browse the web, send and receive e-mail, use the computational and interactive facilities of the centre. You can use the state of the art services of the computer centre with freedom. This freedom, you will find, is unsurpassed, when compared with elsewhere. Hence it is expected that you have to use the service in a responsible manner. Your conduct and use should be tempered with propriety and a spirit that respects the right of everyone. Any abuse of the facility may have serious implications on the reputation and the standing of the institute as well as can bring the facility of the institute to dispute. Such misuse of the facility will not be tolerated.

Certain guidelines that should be followed in the use of the computer centre and its facilities have been formulated. The compliance with the guidelines and the do's and don'ts are necessary for the continued use of your freedom. Not only should users stick to the use of the facility in accordance with the guidelines, but also report any use that is against these guidelines or the spirit of the guidelines. Failure to report infractions of the guidelines shall also be construed as an act of omission in following the guidelines.

Following activities amount to unethical use of computing facilities and will attract punishment from the Institute authorities. You should read the following list of activities that amount to unethical use of computing facilities and the previous discussion carefully and then sign your declaration of acceptance of the policies.

Blocking Resources:

1. Playing games on the machines in the computer centre
2. Locking the screen of machines in the computer centre
3. Sending junk mail to all the users
4. Sending chain letters

Minimum punishment is suspension of computer access facilities for two weeks. Additionally financial fine may also be imposed.

Wasting Resources:

1. Unnecessary download from the Internet

Minimum punishment is suspension of computer access facilities for two weeks. In addition (s)he will have to compensate the institute for the resources wasted.

Misusing Facilities:

1. Giving accounts to other persons , sometimes outsiders
2. Storing pornographic material on the disk

3. Viewing pornographic materials on terminals
4. Using personal account to do outside (non-institute) work for which the individual is paid.

Minimum punishment is suspension of computer access facilities for six months and cases being sent to concerned authorities for disciplinary action.

Security related Misuse:

1. Breaking security of the system
2. Trying to capture password of other users
3. Damaging/gaining access to the data of other users

This kind of abuse is taken most seriously. Anyone found involved in these activities will have access being denied for one year. The cases will be sent to concerned authorities for necessary disciplinary action.

Software related Misuse:

1. Any kind of licensed software's use that does not comply with software's Terms and Conditions

You should always use licensed software. License software have their own Terms and Conditions that should be followed by users.

We cannot enumerate all these cases. However, a thumb rule is that any activity which inconveniences users, depletes the computer centre resources, or jeopardizes the security of the systems, amounts to unethical use. Moreover, it should be noted that the punishment set out for various cases can change and can be very severe depending on the view that the institute takes of the offence.

The Institute, whenever possible, gives all legitimate users the discretion to determine how to best use the computing resources and facilities within the guidelines of this policy. Users are responsible for their actions, the consequences of those actions, and the consequences of negligent inaction. As such, users whose judgement leads to activities inconsistent with the guidelines of this policy risk disciplinary action and possible imposition of restrictions to enforce the guidelines of this policy. Also the guidelines should be met with letter and the spirit of the guidelines as no set of ethical/unethical guidelines can be exhaustive.

Thanking you and wish you a very happy stay at IIT Kanpur

Computer Centre, IIT Kanpur.

I, _____, hereby declare that I have read the above document and promise to abide by the terms mentioned above.

JEE (Advanced) Roll No.

Department:

Signature:

Date:

Black Lives Matter.
Support the Equal Justice Initiative.

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**

Documentation translations provided by [StrongLoop/IBM](#): [French](#), [German](#), [Spanish](#), [Italian](#), [Japanese](#), [Russian](#), [Chinese](#), [Traditional Chinese](#), [Korean](#), [Portuguese](#).

Community translation available for: [Slovak](#), [Ukrainian](#), [Uzbek](#), [Turkish](#) and [Thai](#).



Star

Express is a project of the [OpenJS Foundation](#).

[Edit this page on GitHub](#).

Copyright © 2017 StrongLoop, IBM, and other [expressjs.com](#) contributors.



This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 United States License](#).



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:

Black Lives Matter.
Support the Equal Justice Initiative.

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**

Documentation translations provided by [StrongLoop/IBM](#): [French](#), [German](#), [Spanish](#), [Italian](#), [Japanese](#), [Russian](#), [Chinese](#), [Traditional Chinese](#), [Korean](#), [Portuguese](#).

Community translation available for: [Slovak](#), [Ukrainian](#), [Uzbek](#), [Turkish](#) and [Thai](#).



Star

Express is a project of the [OpenJS Foundation](#).

[Edit this page on GitHub](#).

Copyright © 2017 StrongLoop, IBM, and other [expressjs.com](#) contributors.



This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 United States License](#).