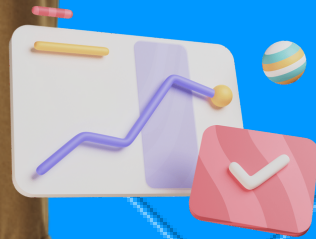




Express

CheatSheet



CODE HELP

In this Cheatsheet, we will cover the basics of Express.js. We will provide examples to help you understand how Express.js works and how to use them in your own web development projects. Whether you are a beginner or an experienced developer, this PDF can serve as a useful reference guide.



Express.js is a popular open-source web application framework for Node.js, a JavaScript runtime that allows developers to build scalable, high-performance server-side applications. It provides a wide range of features and utilities for web application development, such as routing, middleware, and template rendering.

With Express.js, developers can quickly create server-side applications that handle HTTP requests, process data, and respond with dynamic content or data from a database. It also supports a wide range of middleware that can be used to add functionality, such as authentication, compression, logging, and more.

Express.js is designed to be flexible and easy to use, allowing developers to customize and extend it to meet the specific needs of their applications. Its modular structure and extensive community support make it one of the most popular frameworks for building web applications with Node.js.

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

INSTALLATION

```
npm install express
```

HELLO WORLD SERVER

```
const express = require('express');
const app = express();
app.set('port', (process.env.PORT || config.port));
app.get('/', (req, res) => res.send('Hello World!'));
app.listen(app.get('port'), () =>
  console.log(`Server started on ${app.get('port')} port`))
```

BASIC ROUTING:

```
// GET
app.get('/', function (req, res) {
  res.send('Hello World!')
})

// POST
app.post('/', function (req, res) {
  res.send('POST request. body:', req.body)
})

// DELETE
app.delete('/:id', function (req, res) {
  res.send('DELETE request for id:'. req.params.id)
})
```

STATIC FILE SERVING:

```
app.use(express.static(__dirname + '/public'));
```

LOGGING ALL ROUTES:

```
app._router.stack.forEach(function(r) {  
  if (r.route && r.route.path) {  
    console.log(r.route.path)  
  }  
});
```

DEFINING ROUTES IN A DIFFERENT FILE:

File: /routes/users.js

```
// File Path: /routes/users.js  
  
var express = require('express');  
var router = express.Router();  
router.get('/', (req, res) => {  
  const users = []; // get from db  
  res.send(users);  
});  
router.get('/:id', (req, res) => {  
  const user = {}; // get from db  
  res.send(user);  
});  
router.post('/', (req, res) => {  
  const user = req.body; // save user to db  
  res.send({status: 'success'});  
});  
module.exports = router;
```

ADDING ROUTES FROM: /routes/users.js

```
app.use('/user', require('./routes/user'));
```

REDIRECTS:

```
router.get('/old-path', function(req, res) {  
  res.redirect('/temp-new-path'); // sends a 302  
});
```

```
router.get('/old-path', function(req, res) {  
  res.redirect(301, '/permanent-new-path'); // sends a 301  
});
```