

f (<https://www.facebook.com/fullstacktutorials>) **t** (<https://twitter.com/fullstacktuts>)

in (<https://www.linkedin.com/company/fullstacktutorials>)

g+ (<https://plus.google.com/117327870733441845636>)

YouTube (https://www.youtube.com/channel/UCcvYO7gu-NpA5i6-UZPc_eA)

🔗 **Ask a Question** (/contact.html) **📈** **Advertise** (/advertise.html)

🔗 **Follow** (<https://www.facebook.com/fullstacktutorials>)

🔒 **Login** (<https://www.fullstacktutorials.com/login>)

👤 **Register** (<https://www.fullstacktutorials.com/register>)

FullStackTutorials (/)

📘 Ads by Google

Nosql Database

Question Answer

Node Express

What are you looking for? search here...

🔍 SEARCH

Menu



ABOUT US ▾

LATEST ARTICLES

TUTORIALS

INTERVIEWS Q & A

PROJECTS ▾

DOWNLOADS

📄
url=http://

TRAINING

GOVT. JOBS

CONTACT

mini=true

10-

express-

js-

interview-

questions

answers-

30.html)



(http://ww

u=http://v

10-

express-

js-

interview-

questions

Interview Question Answers



Home (/) >> Interviews Q & A (/interviews.html) >> Top 10 Express Js Interview Questions and Answers

(https://pl

Ads by Google

[Answer Quiz](#)
[React JS](#)
[Node App](#)

10-

Top 10 Express Js Interview Questions and Answers

interview-

questions

answers-



30.html)

<https://api.whatsapp.com/send?text=Top 10 Express Js Interview Questions and Answers https://www.fullstacktutorials.com/interviews/top-10-express-js-interview-questions-answers-30.html>



(http://tw

Author - Full Stack Tutorials

Last Updated - October 20, 2018

express-

js-

Top 10 Express Js Interview Questions and Answers, Express Js Interview Questions, Express.js Interview Questions

interview-

questions

Q.1) What is Express Js?

answers-

30.html)

1. Express js is a free open-source, light-weight node js based web application framework.
2. It is designed for building (single-page, multi-page, and hybrid) web applications and APIs.
3. It has been developed by TJ Holowaychuk in 2010 and written in JavaScript.

(http://ww

Express Js Features:

Following are some of the core features of Express framework:

1. **Middlewares:** Set up middlewares in order to respond to HTTP/RESTful Requests.
2. **Routing:** It is possible to defines a routing table in order to perform different HTTP operations.
3. **Template Engine:** Express dynamically renders HTML Pages based on passing arguments to templates.
4. **High Performance:** Express prepare a thin layer, therefore, the performance is adequate.
5. **Database Support:** Express supports RDBMS as well as NoSQL databases.
6. **MVC Support:** Organize the web application into an MVC architecture.
7. **Manages everything** from routes to rendering view and performing HTTP request.

Q.2) How can we allow CORS in Express JS? Explain with an Example?

In order to allow CORS in Express.js, add the following code in app.js:

```
code source
1. app.all('*', function(req, res, next) {
2.   res.setHeader('Access-Control-Allow-Origin', '*');
3.   res.setHeader('Access-Control-Allow-Methods', 'GET, POST, DELETE, PUT');
4.   res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With, Content-Type');
5.   if ('OPTIONS' == req.method) return res.send(200);
6.   next();
7. });
```

🔖 You may also like - **Node Js Interview Questions Answers**
<https://www.fullstacktutorials.com/interviews/node-js-interview-questions-29.html>

Q.3) What is Scaffolding in Express JS?

Scaffolding is creating the skeleton structure of application

There are 2 way to do this:

1. Express application generator
2. Yeoman

1. Express application generator:

Use express-generator to quickly create an application skeleton.

```
code source
1. npm install express-generator -g
2. express myTestExpressApp
```

The above command will create your project named - myTestExpressApp with following the files/folders in project.

1. **bin:** The bin folder have one file called **www** is the main configuration file of our app.
2. **public:** Public folder contains JavaScript, CSS, images etc.

3. **routes:** Routes folder contains routing files.
4. **views:** View folder contains the view files of the application.
5. **app.js:** The app.js file is the main file of the application.
6. **package.json:** package.json file is the manifest file. It contains all metadata of the project such as the packages used in the app (called dependencies) etc.

Install all the dependencies mentioned in the package.json file:

```
code source
1. cd myTestExpressApp
2. npm install
```

Q.4) How to enable debugging in express app?

In different Operating Systems, we have following commands:

On Linux:

```
code source
1. DEBUG=express:*
2. node app.js
```

On Windows:

```
code source
1. set DEBUG=express:*
2. node app.js
```

Q.5) Serving static files in Express?

```
code source
1. #Example:
2. app.use(express.static('public'))
3. app.use('/', static, express.static(path.join(__dirname, 'public')))
```

Q.6) What is routing and how routing works in Express.js?

Routing refers to determining how an application responds to a client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, and so on).

Each route can have one or more handler functions, which are executed when the route is matched.

Route Syntax

```
code source
1. app.METHOD(PATH, HANDLER)
```

Where:

- **app** is an instance of **express**.
- **METHOD** is an HTTP request method, in lowercase.
- **PATH** is a path on the server.
- **HANDLER** is the function executed when the route is matched.

#Example:

```
code source
1. app.get('/', function (req, res) {
2.   res.send('Express Js Interview Questions')
3. })
```

Dynamic routing and how it works in Express.js?

When someone pass parameters in URL i.e. Parametrized URL, this routing phenomenon is called dynamic routing.

```
code source
1. var express = require('express'),
2.   app = express();
3.
4. app.get('/article/:id', function(req , res){
5.   res.render('article' + req.params.id);
6. })
```

In above example, `id` is a parameters, which can be different for different calls.

Q.7) What is middleware in Express Js?

"A function that is invoked by the Express routing layer before the final request handler."

Middleware functions can perform the following tasks:

1. Execute any code.
2. Make changes to the request and the response objects.
3. End the request-response cycle.
4. Call the next middleware function in the stack.

If the current middleware function does not end the request-response cycle, it must call `next()` to pass control to the next middleware function. Otherwise, the request will be left hanging.

An Express application can use the following types of middleware:

1. Application-level middleware
2. Router-level middleware
3. Error-handling middleware
4. Built-in middleware
5. Third-party middleware

You may also like - **React Js Interview Questions**
 (https://www.fullstacktutorials.com/interviews/react-js-interview-questions-answers-31.html)

1. Application-level middleware:

This kind of middleware method is bind to the app Object using app.use() method.

code source

```
1. //This middleware will execute for each route.
2. app.use(function (req, res, next) {
3.   console.log('Current Time:', Date.now())
4.   next()
5. })
```

2. Router-level middleware:

Router-level middleware works in the same way as application-level middleware, except it is bound to an instance of express.Router()

3. Built-in middleware:

Starting with version 4.x, Express no longer depends on **Connect**
 (https://github.com/senchalabs/connect).

Express has the following built-in middleware functions:

- **express.static** serves static assets such as HTML files, images, and so on.
- **express.json** parses incoming requests with JSON payloads. **NOTE: Available with Express 4.16.0+**
- **express.urlencoded** parses incoming requests with URL-encoded payloads. **NOTE: Available with Express 4.16.0+**

4. Third-party middleware:

There are a number of third party middleware, such as body-parser cookie-parser, mongoose and so on.

To handle HTTP POST request in Express.js version 4 and above, you need to install middleware module called body-parser, body-parser extract the entire body portion of an incoming request stream and exposes it on req.body, The middleware was a part of Express.js earlier but now you have to install it separately.

These can be installed by using command:

code source

```
1. >> npm install MODULE_NAME
```

And they can be loaded using requires and used later.

#Example: (http://www

code source

```

1. var bodyParser = require('body-parser');
2. app.use(bodyParser.json());
3. app.use(bodyParser.urlencoded({ extended: false }));

```

Q.8) Database integration in Express Js?

Adding the ability to connect databases to Express apps is just a matter of loading an appropriate node.js driver for the database in your app

Express Js supports many RDBMS & NoSQL Database like

- MongoDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server
- SQLite

#Example: Install MongoDB

code source

```

1. >>npm install mongodb
2.
3. var MongoClient = require('mongodb').MongoClient
4.
5. MongoClient.connect('mongodb://localhost:27017/test_db', function (err, db) {
6.   if (err) throw err
7.
8.   db.collection('mammals').find().toArray(function (err, result) {
9.     if (err) throw err
10.
11.     console.log(result)
12.   })
13. })

```

Q.9) Error handling in Express Js and How to redirect 404 errors to a page in Express Js?

code source

```

1. var express = require('express'),
2. app = express();
3.
4. app.use(function (err, req, res, next) {
5.   console.log(err.stack)
6.   res.status(500).send('Something went wrong, Express Js Interview Questions')
7. })

```

code source

30.html)

in
(http://wv

```

1. //To redirect 404 errors
2. var express = require('express'),
3. app = express();
4.
5. app.use(function(req, res, next) {
6.   res.status(404).json({
7.     errorCode: 404,
8.     errorMsg: "route not found"
9.   });
10. });

```

Q.10) How to implement JWT authentication in Express app ? Explain with an Example?

- Create a folder called 'keys' inside project folder.
- Install some dependencies as following: `npm install jsonwebtoken --save`
- Add the login router routes/index.js

```

code source
1. router.post('/login, function(req, res) {
2.   // find the user
3.   User.findOne({
4.     name: req.body.username
5.   }, function(err, res) {
6.     if (err) throw err;
7.     if (!res) {
8.       res.json({ success: false, message: Login failed.' });
9.     } else if (res) {
10.      // check if password matches
11.      if (res.password != req.body.password) {
12.        res.json({ success: false, message: Login failed. Wrong password.' });
13.      } else {
14.        var token = jwt.sign(res, app.get('superSecret'), {
15.          expiresInMinutes: 1600
16.        });
17.        // return the information including token as JSON
18.        res.json({
19.          success: true,
20.          message: 'Valid token!',
21.          token: token
22.        });
23.      }
24.    }
25.  });
26. }

```

- Use the token in application

```

code source
1. // find the user
2. User.findOne({
3.   name: req.body.username
4. }, function(err, res) {
5.   if (err) throw err;
6.   if (!res) {
7.     res.json({ success: false, message: Login failed.' });
8.   } else if (res) {
9.     // check if password matches
10.    if (res.password != req.body.password) {
11.      res.json({ success: false, message: Login failed. Wrong password.' });
12.    } else {
13.      var token = jwt.sign(res, app.get('superSecret'), {
14.        expiresInMinutes: 1600
15.      });
16.      // return the information including token as JSON
17.      res.json({
18.        success: true,
19.        message: 'Valid token!',
20.        token: token
21.      });
22.    }
23.  }
24. }

```



```

1. mini=true
   jwt = require("express-jwt");
2. app.use(function(req, res, next) {
3.   var token = req.body.token || req.query.token || req.headers['x-access-token'];
4.   if (token) {
5.     jwt.verify(token, app.get('superSecret'), function(err, decoded) {
6.       if (err) {
7.         return res.json({ success: false, message: 'Invalid token.' });
8.       } else {
9.         req.decoded = decoded;
10.        next();
11.      }
12.    });
13.  } else {
14.    return res.status(403).send({
15.      success: false,
16.      message: 'No token given.'
17.    });
18.  }
19. });

```

questions

👉 You may also like - **Angularjs Interview Questions**
 (https://www.fullstacktutorials.com/interviews/angularjs-tricky-interview-questions-1.html)

G

(https://pl

url=http://

10-

express-



f

Qs-

(https://api.whatsapp.com/send?text=Top 10 Express Js Interview Questions and Answers https://www.fullstacktutorials.com/interviews/top-10-express-js-interview-questions-answers-30.html)

answers-

30.html)

FullStack
Tutorials

Author | **interview** @FullStackTutorials | **View all articles**

questions
 (https://www.fullstacktutorials.com/blog.html)

answers-

30.html)

in

(http://ww



mini=true f
 10-
 सोशल मीडिया पर हमारे साथ जुड़ें - (https://www.facebook.com/fullstacktutorials)
 express-
 js- in
 (https://twitter.com/fullstacktuts) (https://www.linkedin.com/company/fullstacktutorials/)
 questions
 8+ Share
 answers-
 (https://plus.google.com/117327870733441845636)
 30.11.19

(http://ww
 v=https/

You may also like:

👍 **NodeJS Interview Questions (/interviews/node-js-interview-questions-29.html)**

👍 **Web Services Interview Questions (/interviews/restful-web-services-interview-questions-26.html)**

👍 **JavaScript Interview Questions (/interviews/top-50-basic-javascript-interview-questions-and-answers-freshers-and-experienced-10.html)**

👍 **AngularJS Interview Questions (/interviews/angularjs-tricky-interview-questions-1.html)**

👍 **Java Interview Questions (/interviews/core-java-interview-questions-answers-freshers-7.html)**

👍 **PHP Interview Questions (/interviews/php-developer-interview-questions-and-answers-for-experienced-4.html)**

express-
 js-
 interview-
 questions
 answers-
 Interview Question Answers

👍 **BIGDATA** Interview QA - **(1)** (/interviews/bigdata.html)

👍 **C** Interview QA - **(2)** (/interviews/c.html)

👍 **DATABASE** Interview QA - **(4)** (/interviews/database.html)

👍 **DOTNET** Interview QA - **(1)** (/interviews/dotnet.html)






👍 **FIREBASE** Interview QA - **(1)** (/interviews/firebase.html)

👍 **JAVA** Interview QA - **(4)** (/interviews/java.html)

👍 **JAVASCRIPT** Interview QA - **(14)** (/interviews/javascript.html)

👍 **MICROSERVICES** Interview QA - **(1)** (/interviews/microservices.html)

👍 **OOPS** Interview QA - **(1)** (/interviews/oops.html)

-  **PHP Interview QA - (17)** (/interviews/php.html)
-  **PYTHON Interview QA - (1)** (/interviews/python.html)
-  **SERVER Interview QA - (1)** (/interviews/server.html)
-  **SOFTSKILLS Interview QA - (1)** (/interviews/softskills.html)
-  **WEB Interview QA - (5)** (/interviews/web.html)


 (http://wv
 u=http://v
 10-

Subscribe our free NewsLetter and receive all latest notifications in your Inbox!

Please enter your email address



Get helpful tutorials, interviews QA and special offers! straight into your Inbox.

(https://pl
 url=http://
 10-
 express-
 Most Popular | Full Stack Tutorials

- ☒ JavaScript Interview Questions (/interviews/advanced-javascript-interview-questions-and-answers-for-experienced-12.html)
- ☒ React Js Interview Questions (/interviews/react-js-interview-questions-answers-31.html)
- ☒ Node Js Interview Questions (/interviews/node-js-interview-questions-29.html)
- ☒ Laravel Interview Questions (/interviews/top-50-laravel-interview-questions-and-answers-for-experienced-13.html)
- ☒ OOPS Interview Questions (/interviews/oops-interview-questions-and-answers-for-experienced-in-php-5.html)
- ☒ Core PHP Interview Questions (/interviews/core-php-interview-questions-and-answers-30.html)
- ☒ Core Java Interview Questions (/interviews/core-java-interview-questions-and-answers-for-freshers-7.html)

- ☑ MongoDB Interview Questions (/interviews/mongodb-interview-questions-for-beginners-and-professionals-17.html)
- ☑ Python Interview Questions (/interviews/python-interview-questions-25.html)
- ☑ RESTful Web Services Questions (/interviews/restful-web-services-interview-questions-26.html)
- ☑ Drupal 8 Interview Questions (/interviews/drupal-8-interview-questions-and-answers-for-experienced-developers-15.html)
- ☑ Top 10 HR Round Questions (/interviews/top-10-hr-interview-questions-and-answers-for-freshers-and-experienced-18.html)

[View All \(/interviews.html\)](/interviews.html)

< >

Featured Reading



FullStack

Tutorials

FullStack

Tutorials

PHP

(/interviews/php-developer-interview-questions-and-answers-for-experienced-4.html)

(http://www.fullstacktutorials.com/interviews/core-php-interview-questions-and-answers-36.html)

BY FULL STACK TUTORIALS TEAM () | MAY 12, 2018

interview-
questions
answers-
30.html)

JAVA

(/interviews/core-java-interview-questions-and-answers-for-freshers-7.html)

Interviews Question Answer - JAVA
(/Interviews/Core-Java-Interview-Questions-And-Answers-For-Freshers-7.Html)

BY FULL STACK TUTORIALS TEAM () | MAY 12, 2018

About Us in
(http://www.fullstacktutorials.com/interviews/top-10-express-js-interview-questions-answers-30.html)



Full Stack Tutorials - Provides Tutorials, Interviews Questions and Answers and Video Tutorials, Programming Solutions, Code Snippet, SEO & Blogging Tips etc.

Start learning thousands of online tutorials today on **FullStackTutorials (/)**

 (https://www.facebook.com/fullstacktutorials)
  (https://twitter.com/fullstacktuts)
  (https://www.linkedin.com/company/fullstacktutorials/)
  (https://api.whatsapp.com/send?text=Full+Stack+Tutorials https://www.fullstacktutorials.com/)





Important Topics

INTERVIEWS Q & A (/INTERVIEWS.HTML) TUTORIALS (/TUTORIALS.HTML)
 LATEST ARTICLES & BLOG (/BLOG.HTML) QUIZ (/QUIZ.HTML)

Interview QA

- React Js Interview Questions (/interviews/react-js-interview-questions-answers-31.html)
- Node js Interview Questions (/interviews/node-js-interview-questions-29.html)
- Laravel Interview Questions (/interviews/top-50-laravel-interview-questions-and-answers-for-experienced-13.html)
- CorePHP Interview Questions (/interviews/core-php-interview-questions-and-answers-36.html)

Help & Support

-  About us (/about.html)
-  Write for us (/contact.html)
-  Ask a Question (/contact.html)
-  Write Feedback (/advertise.html)

-
- Privacy Policy (/privacy-policy.html) • About Us (/about.html) • FAQ (/faq.html)

Copyright ©2019 fullstacktutorials.com. All rights reserved | Powered by - **Full Stack Tutorials** (https://www.fullstacktutorials.com/)