### 1) What is Next.js?

Next.js is an open-source, lightweight React.js framework that facilitates developers to build static and server-side rendering web applications. It was created by Zeit. Next.js framework is based on React, Webpack, and Babel and allows us to write server-rendered React apps easily. It doesn't require any webpack configuration and only needs npm run dev start building your next feature-filled web application.

### 2) Why is Next.js used for? / Why do world's leading companies prefer Next.js?

If you want to build a complete web application with React from scratch, you have to fulfill the following points:

* Your code has to be bundled using a bundler like webpack and transformed using a compiler like Babel.
* You have to do production optimizations such as code splitting.
* You have to pre-render some pages for performance and SEO statically. You might also want to use server-side rendering or client-side rendering.
* You might have to write some server-side code to connect your React app to your data store.

Next.js fulfills the above all requirements.

65.2M

1.2K

C++ vs Java

### Reasons why the world's leading companies prefer Next.js:

****Zero Setup:**** Next.js provides automatic code-splitting, filesystem-based routing, hot code reloading, and universal rendering; that's why the world's leading companies prefer it.

****Fully Extensible:**** Next.js is fully extensible and has complete control over Babel and Webpack. It provides a customizable server, routing, and next plugins.

****Ready for Production:**** Next.js is optimized for smaller build sizes, faster dev compilation, and many other improvements, making it a popular choice.

****Next.js can Deploy Anywhere:**** Next.js is an open-source, lightweight React.js framework that facilitates developers to build static and server-side rendering web applications.

### 3) What is the process to install Next.js? / How to install Next.js?

Before installing Next.js, you must have installed Node.js on your system. Learn how to install Node.js on your system: <https://www.javatpoint.com/install-nodejs>

It requires NPM to start installing Next.js with all its dependencies.

****Follow the steps given below to install Next.js:****

* First, create a directory to keep the Next.js project and go into it:

1. mkdir my-portfolio-site
2. cd my-portfolio-site

* Now, initialize this with a package.json file.
* Use the y flag by npm init -y
* Use the following syntax to install Next.js

1. npm install react react-dom next

* Update package.json with run script languages to start the initialization of the Next.js application.
* Now, find the package.json file on the root folder and add the below mentioned script

1. {
2. "scripts": {
3. "dev": "next",
4. "build": "next build",
5. "start": "next start"
6. }
7. }

* After that, the filesystem is the main API. Every ".js" file becomes a route that gets automatically processed and rendered.
* Now, the process is completed, and Next.js is installed on your system.

### 4) What are the most prominent features of Next.js?

Following is a list of the most prominent features of Next.js that excite the developers most:

* js provides the by default and easy server rendering.
* js supports static exporting.
* It provides a Webpack-based dev environment which supports Hot Module Replacement (HMR)
* It seaports automatic code-splitting for faster page loads.
* It supports simple client-side routing (page-based) or file system-based routing.
* It provides complete Webpack and Babel control.
* It provides a faster and optimized development compilation.
* It can be implemented with Express or any other Node.js HTTP server.
* You can easily customize it with your own Babel and Webpack configurations.
* It supports hot code reloading.

****Besides this, Next.js also has some awesome features such as:****

* Dynamic styles and themes support
* Built-in CSS vendor prefixing
* CSS Preprocessing via Plugins
* Full CSS support, no tradeoffs in power
* Its runtime size is very small. It is just 3kb (zipped from 12kb)
* It provides source maps support

### 5) Which types of websites most popularly use Next.js?

Next.js is a popular framework of React.js that is most popularly used for building the following types of apps and websites:

* Static Websites
* Desktop Websites
* SEO Friendly Websites
* Server Rendered Apps
* Progressive Web Apps (PWA) etc.

### 6) Is it possible to use Next.js with Redux?

Yes. You can easily use Next.js with Redux.

### 7) What is the recommended method to fetch data in Next.js?

There are multiple ways to fetch data in Next.js, but Next.js itself recommends ****getInitialProps,**** an async function to retrieve data from anywhere. When we use ****getInitialProps**** to retrieve data, it receives a context object which has the following properties:

* ****pathname-**** It specifies the path section of the URL.
* ****query-**** It is used to specify the query string section of URL parsed as an object.
* ****asPath-**** It specifies the string of the actual path (including the query) shows in the browser.
* ****req-**** It is used to specify the HTTP request object (server only).
* ****res-**** It is used to specify the HTTP response object (server only).
* ****err-**** It is used to specify the error object if any error is encountered during the rendering.

### 8) Give an example to demonstrate how do you set up CDN in Next.js?

To setup CDN in Next.js, the developers have to follow the steps given below:

To start, we have to first set up the "assetPrefix" setting and configure our CDN origin to support and resolve the domain that our Next.js is hosted on.

1. const isProd = process.env.NODE\_ENV === 'production';
2. module.exports = {
3. // You may only need to add assetPrefix in the production.
4. assetPrefix: isProd ? 'https://cdn.mydomain.com' : ''
5. };

If the CDN is present on a separate domain, we have to set a configuration option as following:

1. // next.config.js
2. odule.exports = {
3. crossOrigin: 'anonymous'
4. ;

### 9) Are Create-React-App and Next.js used for the same thing?

The Create-React-App is basically React with an integrated build system. It acts like a good boilerplate, so we don't need to set up Webpack, Babel, and other dependent packages to run React. Other than that, if you require extra functionalities such as routing, server-side rendering, and so on, you just need to add packages on top of Create-React-App. On the other hand, The Next.js is an open-source, lightweight full-stack React framework that comes bundled with an efficient build system, server-side rendering, routing, API routing, and many other awesome features that make the production environment easy.

### 10) How can you install and use Next.js?

There are mainly two ways to install and run Next.js on your system. If you're new to Next.js, we recommend that you make sure that your development environment is ready. Next.js is a React framework, and it requires Node.js to be installed on your system. If you don't have Node.js installed, you can install it from here: <https://www.javatpoint.com/install-nodejs>. Next.js requires Node.js version 10.13 or later. You should also have your text editor and terminal.

****System Requirement for Next.js****

* You must have installed Node.js version 12.0 or later.
* js supports Windows, MacOS, and Linux also.

****The simplest way to install Next.js****

* The simplest way to install and run Next.js on your system is by creating a new Next.js app using create-next-app. It sets up everything automatically for you. To create a project, run the following command on the Node.js console.

1. npx create-next-app
2. # or
3. yarn create next-app

* If you want to start with a TypeScript project, you can use the following command:

1. npx create-next-app --typescript
2. # or
3. yarn create next-app -typescript

* After completing the installation, follow the instructions to start the development server. You can also try to edit the pages/index.js and see the result on your browser, which by default runs on http://localhost:3000

****Manual Installation and Setup of Next.js****

* First, install the next react and react-dom in your project by running the following command.

1. npm install next react react-dom
2. or
3. yarn add next react react-dom

* Now, open the package.json file and add the following scripts:

1. "scripts": {
2. "dev": "next dev",
3. "build": "next build",
4. "start": "next start",
5. "lint": "next lint"
6. }

****The above script specifies the different stages of developing an application:****

* ****dev -**** It runs the next dev, which starts Next.js in development mode.
* ****build -**** It runs "next build", which builds the application for production usage.
* ****start -**** It runs "next start", which starts a Next.js production server.
* ****lint -**** It runs "next lint", which sets up Next.js' built-in ESLint configuration.

Next.js is built around the concept of pages. A page is a React Component exported from a .js, .jsx, .ts, or .tsx file in the pages directory. Pages are associated with a route based on their filename. For example, pages/about.js is mapped to /about. You can even add dynamic route parameters with the filename.

### 11) How can you disable the etag generation in Next.js?

Generally, we use the app.disable('etag') syntax to disable the etag generation in Next.js. But, this may not work for all static contents. So, we should use the following syntax to disable the etag for all static contents.

****Syntax:****

1. app.use(express.static(path.join(\_\_dirname, 'public'), {
2. etag: false
3. }));

### 12) How can you configure the build-id in Next.js?

To configure the build-id in Next.js, we must configure a static ID between our builds. So, we have to provide the "generateBuildId" function with the following configuration.

****Syntax:****

1. // next.config.js
2. module.exports = {
3. generateBuildId: async () =**>** {
4. // For example get the latest git commit hash here
5. return 'my-build-id';
6. }
7. };

### 13) How can you create a page directory inside your project?

To create a page directory inside our project we have to populate the ./pages/index.js with the following contents:

1. function HomePage() {
2. return **<div>**Welcome to Next.js!**</div>**
3. }

To start developing our application, we have to run the npm run dev or yarn dev command. This will start the development server on http://localhost:3000. Now we can visit the localhost: http://localhost:3000 to view our application.

### 14) Give an example to demonstrate how to create a custom error page in Next.js?

We can create our custom error page by defining a \_error.js in the pages folder. See the following example:

1. **import** React from "react";
2. **class** Error **extends** React.Component {
3. **static** getInitialProps({ res, err }) {
4. **const** statusCode = res ? res.statusCode : err ? err.statusCode : **null**;
5. **return** { statusCode };
6. }
7. render() {
8. **return** (
10. {**this**.props.statusCode
11. ? `An error ${**this**.props.statusCode} has occurred on the server`
12. : "An error occurred on client-side"}
14. );
15. }
16. }
17. export **default** Error;

### 15) What do you understand by code splitting in Next.js?

Generally, code splitting is one of the most compelling features of webpack. This feature facilitates us to split our code into various bundles, which can be loaded only on-demand or in parallel. This is mainly used to achieve the smaller bundles and facilitates us to control resource load prioritization which finally has a great impact on the load time.

****There are mainly three approaches to code splitting:****

* ****Entry Points:**** It is used to split code using entry configuration manually.
* ****Prevent Duplication:**** It uses Entry dependencies or SplitChunksPlugin to dedupe and split chunks.
* ****Dynamic Imports:**** It splits the code via inline function calls within modules

It is mainly used to enable pages that can never load unnecessary code.

### 16) How can you enable AMP in Next.js?

This is an important question and is asked in many Next.js interview questions. There are two ways to enable AMP in Next.js.

* AMP-First Pages
* Hybrid AMP Pages

****AMP-First Pages:**** The AMP-First Pages are served to the primary traffic of the website as well as traffic generated from the search engine. Use the following syntax to implement AMP-first pages.

****Example:****

1. // pages/index.js
2. import { withAmp } from 'next/amp'
3. function HomePage() {
4. return **<p>** Welcome to AMP + Next.js.**</p>**
5. }
6. export default withAmp(HomePage)

****Hybrid AMP Pages:**** The Hybrid AMP pages allow the users to have coexisted AMP version of a traditional page so that the search engines can display the AMP version or the page in different mobile search results.

See the following example to understand how to implement the Hybrid AMP to pages:

****Example:****

1. // pages/index.js
2. import { withAmp } from 'next/amp'
3. function HomePage() {
4. return **<p>** Welcome to AMP + Next.js.**</p>**
5. }
6. export default withAmp(HomePage)

### 17) Is it possible to host Next.js in a web server like Nginx?

Next.js is not as simple as static html files. It requires an application server that runs Node.js to deploy and run a Next.js application. Here, we get requests that have to be processed on the server.

### **1. Mostly for which type of websites is Next JS used?**

Several types of websites and applications can be built. But the most popular are-

* Desktop websites
* Static websites
* Server rendered applications
* SEO friendly websites
* Progressive web applications (PWA)

## **2. Can you use Next JS with Redux?**

Yes, it is possible to use Next JS with Redux.

### **3. How can a custom error page be created in Next JS?**

In order to create a custom error page, we need to follow the given steps-

* An "\_error.js" has to be defined in the page folder.
* Then "\_error" component of our own has to be imported rather than "next/error" for using our custom error page.

### **4. Mention the benefits of implementing a serverless model.**

It helps in splitting the app into smaller parts called lambdas. Hence, it helps in improving the readability as well as scalability of any app. The affordability is also enhanced due to the model- "pay for what you use."

### **5. What do you mean by SSR?**

This is server-side rendering. This enables rendering on the server a client-side page app, and then we can send that rendered page to that client. These pages get loaded faster as the browser gets access to them sooner.

### **6. How can serverless mode be implemented?**

For implementing server-less mode, the ‘serverless’ target has to be added in ‘next.config.js’.

// next.config.js

module.exports = {

target: ‘serverless’

}

### **7. What is meant by Styled JSX in Next JS?**

We use this CSS-in-JS library for writing encapsulated and scoped CSS for styling Next JS components. No other component gets affected by introducing the styles to a component using Styled JSX. This allows adding, changing, and deleting the styles without any complications.

|  |
| --- |
| Looking forward to become a master in ****Programming & Frameworks Courses?**** Check out the ****"**[Node JS Training](https://mindmajix.com/node-js-training" \o "Node JS Training" \t "https://mindmajix.com/_blank)**"****and get certified today. |

### **8. What do you mean by automatic code-splitting?**

Each page bundles as well as serves every import declared by us. And therefore, unnecessary code is never loaded on the pages.

### **9. Is Next JS backend, frontend, or full-stack?**

Next JS is full-stack. This means that it allows rendering client-side as well as server-side content. And this is the reason it is highly valued on React, as React allows only frontend development without it.

### **10. What is DOM?**

DOM that is, Document Object Model, is HTML elements' object representation. The user interface and our code are kind of bridged by it. And it contains a tree-like structure having relationships such as child and parent.

### **11. Differentiate between imperative and declarative programming. And what kind is used in React?**

In Imperative programming, we have to specify each step of building anything, such as a user interface. Whereas in Declarative programming, we just need to describe the end product, and the software will create it for us. It takes comparatively fewer efforts and time.

React allows its users to do declarative programming.

[](https://bit.ly/3if9dmk)

### **12. What types of pre-rendering are available in Next JS?**

Next JS provides two types of pre-rendering- Server-side rendering and Static rendering.

### **13. Differentiate between the pre-rendering types available in Next JS.**

The difference lies in the generation of HTML for a page.

Static Generation- It generates the HTML at build time, and we can reuse it on each request. It is the recommended one of the two. For using Static generation, either the page component has to be exported or 'getStaticProps'

Server-side rendering- It generates the HTML on each request. For using Server-side rendering, 'getServerSideProps' has to be exported.

### **14. What is the by default pre-render in Next JS? Give an example.**

By default, Static Generation is the pre-render available in Next JS without any data fetched. An example is given below-

function About()

{

return <div>About</div>

}

export default About

### **15. What are the main scripts in Next JS?**

The main scripts which are listed in the 'package.json' file are-

* Build- It creates a ready-to-deploy app.
* Dev- A development server is run on localhost: 3000.
* Start- Our built Next application is started
* Lint- Our Next project will be 'lint' with the help of dev dependency. ESLint warns if the written code needs fixing.

## **Next JS Interview Questions and Answers for the Experienced:**

### **1. Which method does Next JS recommend for fetching data?**

Next JS provides several methods for fetching data. But the one that it recommends is getInitialProps. It is an async function using which data can be retrieved from anywhere.

### **2. What are the properties available in a context object that arises on using getInitialProps?**

The following properties are there-

1. Pathname- The path section of the URL is specified.
2. asPath- A string of the actual path that is shown in the browser is specified.
3. Query- It specifies the query string section that is parsed as an object.
4. Req- It specifies the HTTP request object (server only).
5. Res- It specifies the HTTP response object (only server)
6. Err- It specifies the error object in case any error is found while rendering.

### **3. Differentiate between Next JS and Create-React-App.**

|  |  |
| --- | --- |
| ****Create-React-App**** | ****Next JS**** |
| The Create-React-App is nothing but React with some integrations. There is no need for setting up Babel, Webpack, and other such packages for running React as it is a good boilerplate.  But packages need to be added on top of it if we want extra functions, including server-side rendering and routing. | On the other hand, Next JS is a full-stack and open-source React framework. It comes with important in-built features like routing, server-side rendering, API routing, and a lot of others. Hence, it is far more convenient. |

### **5. What are the most important features introduced in Next.js 12.1?**

The most important features are given below-

* New Rust-based compiler
* Faster image optimization
* On-demand incremental static regeneration (Beta)
* Self-hosted Next JS improvements
* Zero-configuration Jest plugin
* React 18 support

### **6. Which key features are provided by Next JS in terms of SEO?**

The following perks are there-

1. Jamstack compatibility
2. Increased flexibility in designing the UX of our website
3. Automatic static optimization
4. Improved data security
5. Fast static websites
6. Responsiveness and adaptability

### **7. How can build-id be configured in Next JS?**

For configuring build-id in Next JS, a static ID must be configured between our build. Therefore, the ‘generateBuildId’ function must be provided with the given configuration-

// next.config.js

module.exports = {

generateBuildId: async () =>{

//For instance get the latest git commit hash

return ‘my-build-id’;

}

};

### **8. Explain the importance of code splitting in Next JS.**

Using this feature, we can split the code into several bundles. These will be loaded only in parallel or on-demand. Smaller bundles allow us to control the prioritization of resource load. And it majorly impacts the load time.

|  |
| --- |
| Would you like to ace ****Node.JS job interviews?**** Top **[Node.JS Interview Questions And Answers](https://mindmajix.com/node-js-interview-questions-answers" \o "Node.JS Interview Questions And Answers" \t "https://mindmajix.com/_blank)** from MindMajix are exclusively for you! |

### **9. Mention the three main ways to split code.**

Three main ways to split code are given below-

1. Entry Points- In this method, configurations need to be entered manually.
2. Prevent Duplication- SplitChinksPlugin or Entry dependencies are used here for splitting chunks.
3. Dynamic Imports- The code is split using inline function calls within the modules.

### **10. How can a page directory be created inside a project?**

We need to populate ‘./pages/index.js’ for creating a page directory within a project with the given contents:

function HomePage() {

return<div>Hello there!</div>

}

### **11. Explain the AMP-First Pages method to enable AMP in Next JS.**

These can be served to the website’s primary traffic and search engine-generated traffic as well. The following syntax can be used-

// pages/index.js

import { withAmp } from ‘next/amp’

function HomePage() {

return <p> Welcome to AMP+Next.js.</p>

}

export default withAMP (HomePage)

### **12. Explain the Hybrid AMP Pages method to enable AMP in Next JS.**

The coexisting AMP version of a traditional page is allowed by the Hybrid AMP pages. This enables the search engine to display the AMP page or the version in different search results on mobile.

### **13. How can CDN be set up in Next JS?**

The 'assetPrefix' setting needs to be set up first. And then, the CDN origin has to be configured for supporting and resolving the domain on which Next JS is hosted.

Const isProd = process.env.Node\_ENV === ‘production’;

Module.exports = {

// We might only need to add assetPrefix in production.

assetPrefix: isProd? https://cdn.mydomain.com’:”

};

### **14. What configuration option will you set if the CDN is present on a separate domain?**

// next.config.js

Odule.exports = {

crossOrigin: 'anonymous'

;

### **15. Can Next JS be hosted in a web server such as Nginx?**

Next JS is relatively more complex than the static HTML files. Here, an application server is required to run Node JS for deploying and running a Next JS application. Requests have received that need to be processed on the server.

## **Frequently Asked Nest JS Interview Questions and Answers:**

### **1. What are some of the major companies that use Next JS?**

Some of the major companies that use Next JS are given below-

* Tik Tok
* Nike
* Netflix Jobs
* GitHub Copilot
* Target
* Twitch
* Hulu

### **2. List some competitors and alternatives to Next JS.**

Some alternatives to Next JS are listed below-

* Gatsby
* React
* Create React App
* Hexo
* Hugo
* LoopBack
* Angular Universal
* React Router

### **3. Mention some features of Next JS.**

Next JS has a lot of features. Some of them are given below-

1. Static export is supported by it.
2. Easy server rendering by default is available.
3. Automatic code-splitting can be sea-ported.
4. Full Babel and Webpack control is provided.
5. The Hot Module Replacement is supported by its environment.
6. We can implement it with any of the HTTP servers of Node JS.
7. A simple routine of client-side content is supported by it.
8. Reloading of hot code is also supported by it.
9. Easily customizable.

### **4. How can different digital products be built using Next JS?**

Several digital products can be built, such as-

* Jamstack websites
* Single web pages
* Minimum Viable Product (MVP)
* SaaS products
* Web portals
* Dashboards
* Retail and e-commerce websites
* User interfaces
* Complex web applications
* Static websites

### **5. In which languages is Next JS written?**

Next JS is written in JavaScript, Rust, TypeScript, and React languages.

### **6. Why is there a built-in router in Next JS?**

It has its own router because of the following reasons-

1. A router based on a file system is used in reducing the configuration.
2. Shallow routing is supported by it. Hence, it doesn't need the methods for data fetching to change the URL.
3. Routers are lazy-loadable.

### **7. How can the data be fetched in Next JS?**

We can use multiple methods for fetching data, such as-

1. Server-side renders using getServerSideProps.
2. Client-side renders using SWR or useEffect inside the components of React.
3. Static-side renders using getStaticProps.
4. Regenerate Incremental Static using the 'revalidate' prop inside getStaticProps.

### **8. What are the requirements for building a web app from scratch using React?**

The requirements are given below-

1. A bundler such as webpack should be there to bundle the code. Also, a compiler such as a babel should be there to transform the code.
2. We have to optimize the productions like the splitting of code.
3. The pages have to be statistically pre-rendered for Search Engine Optimization. We might need to render from client-side and server-side as well.
4. Server-side code might also have to be written to connect the data store and React.

### **9. Why is Next JS preferred by most of the major companies?**

The main reasons for this are-

1. It is fully extensible.
2. No setup is required.
3. It is ready for production.
4. It enables server-side rendering as well.

### **10. What is the process of installing Next JS?**

1. Node JS should be already installed in the system.
2. A directory for keeping Next JS is created as follows-

mkdir my-portfolio-site

cd my-portfolio-site

1. It is now initialized using the package.json file.
2. The y flag is used by npm init- y. And the given syntax is used for installing Next.js-

npm install react-dom next

1. Next.js app is initialized by updating the package.json by using run script languages.he given script is then added after we find the package.json file.

{

“scripts”: {

“dev”: “next”,

“build”: “next build”,

“start”: “next start”

}

}

1. What is Next Js and why it is used for?

#### **Answer**

Next, JS is an open-source, JavaScript framework that lets developers build static and server-side rendering web applications. Created by Zeit, Next JS doesn’t require any Webpack configuration and only needs npm run dev start building your next feature filled web application.

Is this helpful?  Yes  No

 27  5

2. How to install Next js?

#### **Answer**

##### **Developers will need NPM to start installing Next JS with all its dependencies. Here are the steps to follow:**

* Create a directory to keep the Next JS project and go into it:  
  mkdir my-portfolio-site  
  cd my-portfolio-site
* Now initialize this with a package.json file.
* Use the y flag by npm init –y
* Use the below-mentioned syntax to install Next JS  
  npm install react react-dom next
* Update package.json with run script languages to start the initialization of Next JS application.
* Please find the package.json file on root folder and add the below mentioned script  
  "dev": "next",  
  "build": "next build",  
  "start": "next start"

Now, we are finished with the process.

Is this helpful?  Yes  No

 7  2

3. What are the features of next js?

#### **Answer**

**Here is a list of most developer-exciting Next JS features:**

* Default and easy server rendering
* Static exporting
* Hot code reloading
* Automatic code splitting
* Complete Webpack and Babel control
* Filesystem based routing
* Faster and optimized development compilation

Is this helpful?  Yes  No

 7  0

4. How to disable etag generation in next js?

#### **Answer**

To disable etag generation in Next JS, we have to use the app.disable('etag') syntax. But, this may not work for all static contents. The below mentioned syntax will disable etag for all static contents.

#### **Example**

app.use(express.static(path.join(\_\_dirname, 'public'), {

etag: false

}));

Is this helpful?  Yes  No

 0  6

5. How to create pages in next js?

#### **Answer**

Is this helpful?  Yes  No

 13  22

6. How to create a custom error page in next js?

#### **Answer**

* To create a custom error page in Next JS, we have to define a “\_error.js” in the page folder with this given syntax.
* We have to import our own “\_ error” component instead of “next/error” further to use our custom error page.

#### **Example**

import React from 'react';

class Error extends React.Component {

     static getInitialProps({ res, err }) {

        const statusCode = res ? res.statusCode : err ? err.statusCode : null;

      return { statusCode };

}

render() {

return (

<p>

   {this.props.statusCode

   ? `An error ${this.props.statusCode} occurred on server`

   : 'An error occurred on client'}

</p>

    );

  }

}

export default Error;

Is this helpful?  Yes  No

 0  0

7. How to setup CDN in next js?

#### **Answer**

Developers have to follow these steps to setup CDN in Next JS.

#### **Example**

* To start, we have to first set up the “assetPrefix” setting and configure our CDN origin to support resolve to the domain that our Next JS is hosted on.

            const isProd = process.env.NODE\_ENV === 'production';

            module.exports = {

               // You may only need to add assetPrefix in the production.

         assetPrefix: isProd ? 'https://cdn.mydomain.com' : ''

};

* For a CDN present on a separate domain that you may like assets to be requested with use of CORS aware request, we have to set a configuration option as following.

         // next.config.js

         module.exports = {

           crossOrigin: 'anonymous'

};

Is this helpful?  Yes  No

 1  0

8. How to configure build id in Next JS?

#### **Answer**

To configure a static ID between our builds, we have to provide “generateBuildId” function with this given configuration.

#### **Example**

// next.config.js

module.exports = {

   generateBuildId: async () => {

  // For example get the latest git commit hash here

  return 'my-build-id';

  }

};

Is this helpful?  Yes  No

 0  0

9. How to write inline CSS in next js?

#### **Answer**

We have to use the further mentioned syntax configuration to write inline CSS in Next JS.

#### **Example**

function HiThere() {

  return <p style={{ color: 'red' }}>hi there</p>;

  }

export default HiThere;

Is this helpful?  Yes  No

 1  0

10. What is AMP in Next JS?

#### **Answer**

This is a Next JS standard used to build high-performance websites rendering overhead. AMP implemented websites are indexed faster in modern and popular search engines with enhanced promoting behavior. AMP web pages are loaded directly to Google's mobile search results with a lightning icon, better performance, fewer restrictions, and better scalability.

Is this helpful?  Yes  No

 0  1

11. How to validate AMP in the next JS?

#### **Answer**

To validate your AMP pages, ‘**amphtml-validator**’ is used during the development. Warnings and fatal errors will be displayed in the terminal where the Next JS is started. AMP pages also get validated during ‘**next export**’ and issues will be printed in the terminal, and the ‘**next export**’ will fail due to the absence of proper AMP validation.

Is this helpful?  Yes  No

 0  0

12. How to enable AMP in Next JS?

#### **Answer**

This one is crucial. **Next JS interview question** to practice and remember all its aspects. There are two processes to enable AMP in Next JS. The thing to remember here is, AMP is a crucial part of many **Next JS interview questions**, so we would advise it to practice well.

* AMP-First Pages

           These are served to the primary traffic of the website as well as traffic generated from the search engine. We have to use the following syntax to implement AMP-first pages.

* Hybrid AMP Pages

           Hybrid AMP pages allow users to have a coexist AMP version of a traditional page so that search engines can easily display the AMP version or the page in different mobile search results. To               implement Hybrid AMP to pages, we have to use the following syntax.

<

#### **Example**

**AMP-First Pages :-**

// pages/index.js

import { withAmp } from 'next/amp'

function HomePage() {

return <p> Welcome to AMP + Next.js.</p>

}

export default withAmp(HomePage)

**Hybrid AMP Pages :-**

// pages/index.js

function HomePage() {

return <p> Welcome to AMP + Next.js.</p>

}

export default withAmp(HomePage, { hybrid: true })

Is this helpful?  Yes  No

 0  0

13. What is Styled JSX in Next JS?

#### **Answer**

It’s a CSS-in-JS library used by developers to write scoped and encapsulated CSS to style Next JS components. The styles introduced to one component with Styled JSX will not affect other components, allowing developers to add, delete, and change styles without worrying about any side effects.

Is this helpful?  Yes  No

 1  0

14. What are the benefits of implementing Serverless mode and how to implement it?

#### **Answer**

Implementing Serverless mode excellently improves scalability and readability of an application by splitting it into smaller parts known as lambdas. It also promotes affordability with a "pay for what you use" model.

To enable Serverless mode in Next JS, we have to add ‘serverless’ build target in **next.config.js**.

#### **Example**

// next.config.js

module.exports = {

target: 'serverless'

}

Is this helpful?  Yes  No

 0  0

15. Does Next JS support static CDN?

#### **Answer**

Yes, Next JS 5 and above supports static CDN. With the introduction of assetPrefix, Next.JS automatically loads assets from CDN.

Is this helpful?  Yes  No

 0  0

Submit Your Question   Download Next js Interview Questions

Here is a list of advance Next JS interview questions with most appropriate answers suitable best for both freshers as well as experienced developers of this niche to practice.

### Q1) What exactly is Next JS?

**Ans:**An open-source, compact React.js framework called Next.js makes it easier for programmers to create static and server-side rendering web applications. Zeit came up with it. We can quickly create server-rendered React apps thanks to the Next.js framework, which is built on React, Webpack, and Babel. Your next feature-rich web application can be built with just npm run dev start and without any webpack settings.

### Q2) Mention a few features of Next JS

**Ans:**

* JS offers simple server rendering by default.
* Static exporting is supported in JS.
* It offers a Webpack-based development environment that facilitates the use of hot module replacement (HMR)
* Automatic code-splitting is supported for quicker page loading.
* It supports straightforward file system-based routing as well as client-side routing (page-based).
* It offers total control over Webpack and Babel.
* It offers a more efficient and quick development compilation.
* You can use Express or another Node.js HTTP server to implement it.
* With your own Babel and Webpack configurations, you may quickly change it.
* It allows for instant code reloading.

### Q3) What categories of websites utilize Next JS the most?

**Ans:** The most common uses for Next JS, a React JS framework, are to create the following apps and websites:

* Static Websites
* PC-based websites
* SEO-friendly websites
* Server Rendered Apps

|  |
| --- |
| Want to acquire industry skills and gain complete knowledge of React JS? Enroll in Instructor-Led live [ReactJS Training](https://tekslate.com/reactjs-training" \t "https://tekslate.com/_blank) to become Job Ready! |

### Q4) Why should anyone choose Next JS?

**Ans:**

* Zero Setup - Routing based on filesystems, hot code reloading, and universal rendering are all automatic.
* Completely customizable - Total command over Webpack and Babel. Server, routing, and next-plugins customization options.
* Production-Ready - Reduced build size optimization, quicker dev compilation, and other enhancements.

### Q5) Does Create React App do the same function as Next JS?

**Ans:**In essence, Create-React-App is React with a built-in build system. So it serves as a useful boilerplate. Just to run React, we don't need to worry about setting up Webpack, Babel, or other dependent packages. Other than that, we must build packages on top of Create-React-App if we need additional functions like routing, server-side rendering, and so forth.

A whole stack React framework is Next.js. It is pre-loaded with useful features including effective routing, build system, API routing, server-side rendering, and many others that perform well in a production context.

### Q6) How can AMP be verified in the next JS?

**Ans:**'amp HTML-validator' is used during development to validate your AMP pages. When the Next JS is launched, a terminal will show any warnings and fatal errors. The "next export" process validates AMP pages as well, and if improper AMP validation is present, errors will be reported in the terminal and the "next export" will fail.

### Q7) How may AMP be enabled in Next JS?

**Ans:**There are two methods for turning on AMP in Next JS.

* **AMP – First Pages:** Both the website's regular visitors and search engine visitors are given access to these. To implement AMP-first pages, the following syntax must be used.
* **Hybrid AMP Pages:** In order for search engines to quickly display the AMP version of the page in various mobile search results, hybrid AMP pages let users have an AMP version of a standard page alongside it.

### Q8) What is the suggested approach for Next JS data fetching?

**Ans:**There are many ways to fetch data in Next JS but the framework itself suggests using the async function getInitialProps to fetch data from any location. The context object that getInitialProps obtains when we use it to retrieve data has the features shown below:

* **Pathname:** This identifies the URL's path portion.
* **Query:** It is used to define the URL query string portion of an object-parsed URL.
* **asPath:** This defines the string that represents the actual path that the browser displays, along with the query.
* **Req:** The HTTP request object (server only) is specified using the word "req.”
* **Res:** The HTTP response object (server only) is specified by the prefix res.
* **Err:** If a rendering error occurs, the word "err" is used to specify the error object.

### Q9) What does Next JS's code splitting mean, in your opinion?

**Ans:**In general, code splitting is one of webpack's most alluring features. This feature makes it easier for us to divide our code into different bundles that can only be loaded in parallel or on-demand. This primarily serves to create smaller bundles and makes it easier to manage resource load prioritization, both of which have a significant impact on load time.

Code splitting primarily uses three methods:

* **Entry Points:** Manual entry configuration is used to separate code.
* **Eliminate Duplication:** Deduplication and chunk splitting are done using Entry dependencies or SplitChunksPlugin.
* **Adaptive Imports:** Using inline function calls within modules, it divides the code using dynamic imports.

It is primarily used to allow unneeded code to load on pages that can never do so.

### Q10) List a few advantages of Next JS

**Ans:**

* Simple project creation, modification, and needed package discovery.
* Optimum application performance because automatic code-splitting is available.
* With the use of prefetching, Next JS enables efficient code bundles to be loaded slowly in the background.
* It enables SSR, or server-side rendering, in application code, providing SEO-friendly flexibility, an initial render to application view, and doing away with the need to download code.
* Hot-Module Replacement that works with robust error reporting.

### Q11) What does Next JS's Styled JSX mean?

**Ans:**It is a library for writing scoped and encapsulated CSS that is used by developers to style Next JS components. Developers can add, remove, and modify styles without being concerned about unintended consequences thanks to Styled JSX, which prevents styles from being introduced to one component from affecting other components.

### Q12) What are the advantages of utilizing Serverless mode, and how do you utilize it?

**Ans:**By dividing an application into smaller, more manageable pieces known as lambdas, Serverless mode implementation greatly enhances a program's scalability and readability. A "pay for what you use" philosophy, also encourages affordability.

Next JS's Serverless mode must be enabled by adding the "serverless" build target to next.config.js.

### Q13) Is static CDN supported by Next JS?

**Ans:**Yes, static CDN is supported by Next JS 5 and higher. Next.JS now seamlessly loads assets from CDN thanks to the assets prefix.

### Q14) Is JS more difficult than React?

**Ans:**It's simple to code; Next. js takes less code than React and other React-compatible frameworks. Less code, greater readability, and better project management are all benefits of this approach, which only requires developers to create the page and link to the component in the header.

### Q15) What kind of framework—frontend or backend—is Next JS?

**Ans:**By going one step further, Next JS eliminates the necessity for a backend framework. You can make React applications that are entirely server-rendered thanks to it. This means that a whole HTML page with all required resources, such as JavaScript files, CSS files, and graphics, will be downloaded to the user's browser.

### Q16) Is it possible to learn Next JS without React?

**Ans:**Studying React is a prerequisite for learning Next JS because Next JS is based on React. React is a JavaScript package that enables the creation of reusable components and aids with coding structure.

**1 .**

**[What is Next.js?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?What-is-Next.js?&id=6857" \t "https://www.freetimelearning.com/interview-questions/_blank)**

****Next.js**** is an open-source, ****lightweight**React.js**framework**** that gives you building blocks to create web applications. ****Next.js**** was first released as an open-source project on GitHub on ****October 25, 2016****. It was created by ****Zeit****. By this framework, mean ****Next.js**** handles the tooling and configuration needed for React, and provides additional structure, features, and optimizations for your web application.

**2 .**

**[What are the Features of NextJS?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?What-are-the-Features-of-NextJS?&id=6858" \t "https://www.freetimelearning.com/interview-questions/_blank)**

For React developers, ****NextJS**** has a lot of advantages. Let's go over all of ****NextJS's**** features :

****Automatic Routing :**** There's no need to set up URLs for routing. The files should be saved in the pages folder. The file system will be mapped to all URLs. Customization is possible.

****Dynamic Components :**** Next.js allows us to import JavaScript modules and React Components.

****Component-specific styles :**** Global and component-specific styles are supported by styled-jsx.

****Server-side rendering :**** As React components are prerendered on the server, they load more quickly on the client.

****Node Ecosystem :**** As Next.js is React-based, it fits well with the Node ecosystem.

****Automatic code split :**** Next.js renders pages with all of the required libraries. Next.js creates multiple resources rather than a single large javascript file. Only the required javascript pages are loaded when a page is loaded.

****Prefetch :**** Next.js is a framework for developing web applications. The prefetch property of the Link component, which is used to link multiple components, can be used to prefetch page resources in the background.

****Export Static Site :**** With Next.js, we can export our entire static site from our web application.

****Hot Code Reload :**** The Next.js server detects modified files and automatically reloads them.

****Built-in Typescript Support :**** Next.js is a Typescript-based framework with excellent Typescript support.

**3 .**

**[Why is Next.js used for? and Why do world's leading companies prefer Next.js?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?Why-is-Next.js-used-for?-and-Why-do-worlds-leading-companies-prefer-Next.js?&id=6859" \t "https://www.freetimelearning.com/interview-questions/_blank)**

If you want to build a complete web application with****React**** from scratch, you have to fulfill the following points :

****\***** Your code has to be bundled using a bundler like ****webpack**** and ****transformed**** using a compiler like Babel.

****\***** You have to do production optimizations such as code splitting.

****\***** You have to ****pre-render**** some pages for performance and SEO statically. You might also want to use ****server-side rendering**** or ****client-side rendering****.

****\***** You might have to write some server-side code to connect your React app to your data store.

****Next.js**** fulfills the above all requirements.

****Reasons why the world's leading companies prefer**Next.js**:****

****\* Zero Setup :**** ****Next.js**** provides automatic ****code-splitting****, ****filesystem-based routing****, ****hot code reloading****, and universal rendering; that's why the world's leading companies prefer it.

****\* Fully Extensible :**** Next.js is fully extensible and has complete control over Babel and Webpack. It provides a customizable ****server****, ****routing****, and next plugins.

****\* Ready for Production :**** ****Next.js**** is optimized for ****smaller build sizes****, ****faster dev compilation****, and many other improvements, making it a popular choice.

****\***Next.js**can Deploy Anywhere :**** ****Next.js**** is an ****open-source****, lightweight ****React.js**** framework that facilitates developers to ****build static**** and s****erver-side**** rendering ****web applications****.

**4 .**

**[What is Building Blocks of a Web Application in Next.js?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?What-is-Building-Blocks-of-a-Web-Application-in-Next.js?&id=6860" \t "https://www.freetimelearning.com/interview-questions/_blank)**

There are a few things you need to consider when ****building modern applications****. Such as:

****\* User Interface -**** how users will consume and interact with your application.

****\* Routing -**** how users navigate between different parts of your application.

****\* Data Fetching -**** where your data lives and how to get it.

****\* Rendering -**** when and where you render static or dynamic content.

****\* Integrations -**** what third-party services you use (****CMS, auth, payments, etc****) and how you connect to them.

****\* Infrastructure -**** where you deploy, store, and run your application code (Serverless, CDN, Edge, etc).

****\* Performance -**** how to optimize your application for end-users.

****\* Scalability -**** how your application adapts as your team, data, and traffic grow.

****\* Developer Experience -**** your team’s experience building and maintaining your application.

For each part of your application, you will need to decide whether you will build a solution yourself or use other tools such as libraries and frameworks.

**5 .**

**[How to Install / Setup in Next.js?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?How-to-Install-/-Setup-in-Next.js?&id=6861" \t "https://www.freetimelearning.com/interview-questions/_blank)**

****System Requirements****

****\***** ****Node.js 12.22.0**** or later

****\***** ****MacOS****, ****Windows**** (including WSL), and ****Linux**** are supported

****Automatic Setup :****We recommend creating a new ****Next.js**** app using ****create-next-app****, which sets up everything automatically for you. To create a project, run:

npx create-next-app@latest

# or

yarn create next-app

# or

pnpm create next-app

If you want to start with a TypeScript project you can use the ****--typescript**** flag :

npx create-next-app@latest --typescript

# or

yarn create next-app --typescript

# or

pnpm create next-app --typescript

After the installation is complete :

****\***** Run ****npm run dev**** or ****yarn dev**** or ****pnpm dev**** to start the development server on ****http://localhost:3000****

****\***** Visit ****http://localhost:3000**** to view your application

****\***** Edit ****pages/index.js**** and see the updated result in your browser

****Manual Setup :****Install ****next****, ****react**** and ****react-dom**** in your project:

npm install next react react-dom

# or

yarn add next react react-dom

# or

pnpm add next react react-dom

Open ****package.json**** and add the following scripts :

"scripts": {

"dev": "next dev",

"build": "next build",

"start": "next start",

"lint": "next lint"}

These scripts refer to the different stages of developing an application:

****\***dev**-**** Runs ****next dev**** to start Next.js in development mode

****\***build**-**** Runs ****next build**** to build the application for production usage

****\***start**-**** Runs ****next start**** to start a Next.js production server

****\***lint**-**** Runs ****next lint**** to set up Next.js' built-in ESLint configuration

Create two directories ****pages**** and ****public**** at the root of your application:

****\***pages**-**** Associated with a route based on their file name. For example ****pages/about.js**** is mapped to ****/about****

****\***public**-**** Stores static assets such as images, fonts, etc. Files inside public directory can then be referenced by your code starting from the base URL (/).

****Next.js**** is built around the concept of pages. A page is a React Component exported from a ****.js****, ****.jsx****, ****.ts****, or ****.tsx**** file in the pages directory. You can even add dynamic route parameters with the filename.

Inside the pages directory add the ****index.js**** file to get started. This is the page that is rendered when the user visits the root of your application

Populate **pages/index.js** with the following contents :

function HomePage() {

return <div>Welcome to Next.js!</div>}

export default HomePage

After the set up is complete :

****\* Run :**** ****npm run dev**** or ****yarn dev**** or ****pnpm dev**** to start the development server on ****http://localhost:3000****

****\* Visit :**** ****http://localhost:3000**** to view your application

****\* Edit :**** ****pages/index.js**** and see the updated result in your browser

**6 .**

**[Which types of websites most popularly use Next.js?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?Which-types-of-websites-most-popularly-use-Next.js?&id=6862" \t "https://www.freetimelearning.com/interview-questions/_blank)**

****Next.js**** is a popular framework of ****React.js**** that is most popularly used for building the following types of apps and websites :

****\***** Static Websites

****\***** Desktop Websites

****\***** SEO Friendly Websites

****\***** Server Rendered Apps

****\***** Progressive Web Apps (PWA) etc.

**7 .**

**[What is the recommended method to fetch data in Next.js?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?What-is-the-recommended-method-to-fetch-data-in-Next.js?&id=6863" \t "https://www.freetimelearning.com/interview-questions/_blank)**

There are multiple ways to fetch data in ****Next.js****, but ****Next.js**** itself recommends ****getInitialProps****, an async function to retrieve data from anywhere. When we use ****getInitialProps**** to retrieve data, it receives a context object which has the following properties :

****\* pathname :**** It specifies the path section of the URL.

****\* query :**** It is used to specify the query string section of URL parsed as an object.

****\* asPath :**** It specifies the string of the actual path (including the query) shows in the browser.

****\* req :**** It is used to specify the HTTP request object (server only).

****\* res :**** It is used to specify the HTTP response object (server only).

****\* err :**** It is used to specify the error object if any error is encountered during the rendering.

**8 .**

**[How can you disable the etag generation in Next.js?](https://www.freetimelearning.com/software-interview-questions-and-answers.php?How-can-you-disable-the-etag-generation-in-Next.js?&id=6864" \t "https://www.freetimelearning.com/interview-questions/_blank)**

Generally, we use the ****app.disable('etag')**** syntax to disable the etag generation in ****Next.js****. But, this may not work for all static contents. So, we should use the following syntax to disable the etag for all static contents.

****Syntax :****

app.use(express.static(path.join(\_\_dirname, 'public'), {

etag: false })); ​