

You

TOTAL INTERVIEW QNAs FOR SPFX WITH REACT JS DEVELOPMENT, CUSTOMIZATION, WORKFLOWS, PLUGINS, HTML,CSS,XML JS & UNIFIED INTERFACE INTEGRATION IN A TABULAR REPRESENTATION WITH ALL FEATURES MENTIONED POINT WISE, BEST PRACTICES, HOW TO OPTIMISE IT, PROS & CONS, WHAT'S LATEST ADVANCEMENTS !!

Copilot

Certainly! SharePoint Framework (SPFx) combined with React is a powerful combination for building dynamic and responsive web elements within SharePoint. Let's dive into some key points and interview questions related to SPFx with React:

#### 1. Why Choose React and SPFx?

- **SharePoint Framework (SPFx)** provides a modern and flexible framework for building solutions within SharePoint.
- **React** simplifies UI development with its component-based structure.
- Together, they offer a powerful toolkit for developers, promoting code reusability and enhancing the user experience.

#### 2. Setting Up Your Development Environment:

- Ensure **Node.js** and **Yeoman** are installed.
- Use Yeoman to scaffold a new SPFx project with React framework.
- Trust the self-signed developer certificate for secure connection between development environment and SharePoint.

#### 3. Building React Components in SPFx:

- Create React components within the "src" directory.
- Organize components in a "web parts" folder based on specific web parts.
- Customize or create new components to suit project requirements.

#### 4. Common Interview Questions:

- **Q1:** What are the new delegate controls in SharePoint?
  - **A:** Three new delegate controls are introduced for displaying the top suite bar (with links to SkyDrive, NewsFeed, Sync, and follow): SuiteBarBrandingDelegate, SuiteLinksDelegate, and PromotedActionsDelegate.
- **Q2:** What changes were made in CSOM and REST-based APIs?
  - **A:** Microsoft improved both CSOM and REST APIs by adding support for Search, User Profiles, Taxonomies, and Publishing Object Model.
- **Q3:** How does an app for SharePoint request permissions to access SharePoint resources?
  - **A:** An app requests permissions during installation, and the developer specifies required permissions in the app manifest file.
- **Q4:** What is the new SPSecurityEventReceiver?
  - **A:** SharePoint introduces a Cloud App Model for creating apps that extend SharePoint capabilities.
- **Q5:** What are the differences between the two base classes (WebPart vs. WebPartPages.WebPart)?
  - **A:** The WebPart base class supports cross-page connections and client-side connections, while WebPartPages.WebPart is for backward compatibility.

Remember, preparing for interviews involves understanding these concepts thoroughly, practicing coding exercises, and being ready to discuss your experiences. Good luck! 🍀

You

You

INCLUDE REACTJS TOP INTERVIEW QNAS EXPLAINED IN TABULAR BASED COMPARION WITH ALL FEATURES MENTIONED, PROS & CONS, WHAT'S LATEST ADVANCEMENTS, REAL TIME APPLICATIONS USED !!

Copilot

Certainly! ReactJS is a powerful JavaScript library used for building reusable components in the view layer of an MVC architecture. Let's explore some key React interview questions, along with their features, pros, cons, and real-world applications:

Question	Description	Features	Pros	Cons	Latest Advancements	Real-Time Applications
What is ReactJS?	ReactJS is a JavaScript library for creating reusable components in the view layer. It uses a virtual DOM for efficient rendering.	- Virtual DOM	- Reusable components	- Learning curve	- Concurrent Mode	- Facebook
Explain the MVC architecture.	MVC separates an app into Model, View, and Controller. React handles the view layer.	- Model-View-Controller	- Isolates concerns	- Boilerplate code	- Server Components	- Instagram
Building blocks of React	Components, JSX, Props, State, Context, Virtual DOM	- Components: Reusable code	- JSX: HTML in React	- State management	- Suspense	- WhatsApp
Props vs. State	Props pass data between components. State is local to a component.	- Props: Function parameters	- State: Local data	- Prop drilling	- Concurrent Rendering	- Dropbox
Lifecycle methods	Methods like componentDidMount, componentDidUpdate, componentWillUnmount	- componentDidMount	- Lifecycle control	- Deprecated methods	- React Server	- Atlassian

Remember, these questions cover both basic and advanced React concepts. Good luck with your interviews! 🍀

You

What are some common React performance optimizations?

Copilot

Certainly! Optimizing the performance of your React applications is crucial for delivering a better user experience and ensuring your app runs smoothly. Here are some common React performance optimization techniques:

### 1. List Virtualization (Windowing):

- Render only the items currently visible on the screen, especially for large lists or tabular data.
- Dynamically replace rendered items as users scroll, keeping the visible portion of the list responsive and conserving resources.

## 2. Memoization:

- o Cache the results of expensive computations or function calls.
- o Use `React.memo`, `useMemo`, or `useCallback` to prevent unnecessary re-renders of components.

### 3. Throttling and Debouncing Events:

- Limit the frequency of function invocations by setting time intervals or periods of inactivity.
- Helps prevent excessive updates and improves performance.

#### 4. Code Splitting:

- Split a large JavaScript bundle into smaller chunks that load selectively.
- Enhances initial load times and reduces unnecessary code execution.

### 5. React Fragments:

- Group multiple elements without adding an extra DOM node.
- Useful for cleaner component structures and better performance.

## 6. Web Workers:

- Run scripts in the background on a separate thread from the main JavaScript thread.
- Ideal for computationally intensive tasks without blocking the UI.

### 7. UseTransition Hook:

- Update state without blocking the UI, specifying some state updates as less important.
- Helps maintain responsiveness during state changes.

Remember, applying these techniques can significantly enhance your React app's performance and create a smoother user experience! 😊🚀

## Copilot

## React performance optimizations

<https://www.bing.com/search?>

IG=8E00AF4B261B406184A33E5E94949C0C&IID=SERP.5016&cw=1280&ch=551&&dissrchswrite=1&lightschemeovr=1&kseed=19500&SFX=26&partnerId=bingchat&tone=Balance1a5-460d-8517-66bff7806c45

## You

## What are some tools for profiling React apps?

## Copilot

Certainly! Profiling React applications is essential for optimizing performance. Here are some tools and techniques you can use:

### 1. React DevTools Extension:

- Visual profiling tool bundled with React.
- Provides insights into component renders, re-renders, and performance.
- Helps identify unnecessary renders and optimize performance.

## 2. Profiler API:

- o Built-in React feature for measuring component performance.
- o Records timing and duration of renders.
- o Use it via the React DevTools extension or the Profiler component.

### 3. List Virtualization (Windowing):

- Render only visible items in large lists or tabular data.
- Dynamically replace rendered items as users scroll.
- Improves efficiency and conserves resources.

#### 4. Memoization:

- o Cache results of expensive computations using `React.memo`, `useMemo`, or `useCallback`.
- o Prevents unnecessary re-renders and optimizes components.

### 5. Code Splitting:

- Split large JavaScript bundles into smaller chunks.
- Load only what's needed, enhancing initial load times.

Remember, profiling tools empower you to make informed decisions and enhance your React app's performance! 😊🚀

## Copilot

## React performance optimizations

## You

Tell me more about React performance audits.

## Copilot

Certainly! When it comes to measuring and improving the performance of React applications, there are several tools and techniques you can use:

### 1. Lighthouse:

- An open-source tool from Google that audits web pages for performance, accessibility, and more.
- Generates a detailed report with suggestions for improving your application's performance.

### 2. Chrome DevTools Audits:

- Integrated directly into Chrome DevTools.
- Provides insights into metrics like first contentful paint, time to interactive, and more.
- Helps identify bottlenecks and areas for optimization.

### 3. React DevTools:

- Allows you to dig deep into your component hierarchy.
- Identify rendering issues, expensive components, and methods.
- Essential for fine-tuning your app's performance.

Remember, regularly auditing and optimizing your React app ensures a smoother user experience and better overall performance! 😊🚀