

Teorihandbok React

Del 1.

1. Ramverket React (en översikt, vad som är unikt för ramverket, fördelar, ev nackdelar, osv)

React är ett javascript-baserat ramverk som framförallt är känt för sitt sätt att hantera komponentbaserad utveckling. En av anledningarna till att de blivit så populärt är av den anledning att användargränssnittet delas in i återanvändbara och självständiga komponenter, där det är till stor fördel för utvecklare att kunna dra nytta av dessa återanvändbara komponenter för att effektivisera utvecklingen av produkt markant.

Fördelar:

Komponentbaserad utveckling: Strukturen i React med komponenter gör att det lätt går att bygga och återanvända dem. Det effektiviserar som sagt utveckling.

Virtuell DOM: React använder en virtuell DOM vilket möjliggör att det kan snabbt uppdatera gränssnitt vilket resulterar i snabbare prestanda.

Enkel integration: React kan enkelt integreras med andra ramverk och bibliotek, vilket gör att det blir väldigt flexibelt för utvecklare att använda.

Stort ekosystem: I och med att React är så pass stort som det är och används av många utvecklare har det byggts upp en stark community, samt mycket resurser och verktyg för att ytterligare underlätta utvecklingen.

Nackdelar:

Inlärningskurva: Som med alla andra ramverk och bibliotek tar det en stund att lära sig och behärska ramverkets koncept.

Komplexitet: När man utvecklar större och mer komplexa applikationer kan Reacts komponentsystem bli överväldigande och svårt att hålla strukturerat.

Stor verktygslåda: I React finns det massor med verktyg, bibliotek och tillägg vilket kan göra det svårt att välja det rätta till specifika behov.

2. Vad innebär Rendering och Virtual DOM?

DOM står för "Document Object Model" och representerar strukturen av HTML-element som bildar en webbsida. En virtuell DOM är en kopia av DOM-trädet som React använder sig av för att höja prestandan, då det möjliggör att enkelt manipulera och uppdatera webbsidans användargränssnitt. Med hjälp av kopian (virtuella DOM:en) av den verkliga DOM:en, jämför React skillnaden mellan de bägge och uppdaterar endast de relevanta delarna vilket leder till ökad prestanda.

Rendering är processen att ta data och generera en visuell representation av den på en webbsida eller i en applikation. Med hjälp av ramverk som React och dess komponentstruktur kan man definiera hur olika delar av användargränssnittet ska se ut baserat på data och tillstånd. Det smidiga är att när data eller tillstånd ändras, uppdateras komponenterna och den visuella representationen renderas om för att återspegla de nya värdena. Detta gör det möjligt att dynamiskt visa förändringar i användargränssnittet i realtid.

3. Vad är JSX? Vad används det till?

JSX är en syntaktisk utökning av Javascript som används i React för att definiera och skapa användargränssnittet. Det tillåter utvecklare att skriva HTML-liknande kod direkt inom JavaScript-filer. JSX gör det enklare och mer intuitivt att beskriva strukturen och utseendet på React-komponenter genom att kombinera JavaScript och HTML-kod i samma fil. Det underlättar också dynamisk generering och visning av data inom komponenterna. JSX-koden översätts sedan till vanlig JavaScript när applikationen byggs.

4. Vad är ett undantag inom programmering? Eng. Exception

Ett undantag inom programmering är en ovanlig eller felaktig händelse som kan inträffa under körningen av en programvara. Undantagshantering används för att fånga och hantera dessa situationer för att hantera fel och upprätthålla programflödet.

5. Vad innebär autentisering inom webbapplikationer? Vad används det till?

Autentisering inom webbapplikationer innebär att verifiera och fastställa identiteten hos en användare eller en enhet som försöker få åtkomst till en webbplats eller en applikation. Det används för att bekräfta att den aktuella användaren är den de påstår sig vara, vanligtvis genom att ange ett användarnamn och lösenord eller genom att använda sig av andra autentiseringsmetoder som biometriska uppgifter eller säkerhetsnycklar.

Autentisering är en viktig del av säkerheten inom webbapplikationer eftersom den hjälper till att förhindra obehörig åtkomst och skyddar användares personliga information. Genom autentisering kan användare få tillgång till sina egna konton och data medan obehöriga personer hindras från att få åtkomst till känslig information eller utföra icke-godkända åtgärder.

Vanliga metoder för autentisering inom webbapplikationer inkluderar användarnamn och lösenord, tvåfaktoraутентisering (2FA), autentisering via OAuth-protokollet, biometrisk autentisering, (t.ex fingeravtryck eller ansiktsigenkänning) och säkerhetsnycklar. Genom att tillämpa autentisering kan webbapplikationer säkerställa att endast auktoriserade användare får åtkomst till sina konton och skydda användarnas data och integritet.

Del 2.

När jag renderar en komponent för min portföljsida hamnar den först i den virtuella DOM:en. Därefter jämförs den med den befintliga DOM:en för att identifiera eventuella skillnader mellan dem. När skillnaderna har upptäckts appliceras ändringarna i den befintliga DOM:en.

Det som finns inuti return-satsen inom en komponent kallas JSX (JavaScript XML). JSX är en syntaxförkortning som används i React för att skapa och beskriva användargränssnittet i form av HTML-liknande element och komponenter. Det tillåter oss att skriva HTML-liknande strukturer och inkludera JavaScript-uttryck inuti komponenter. JSX-element kan representera både inbyggda HTML-element och egna komponenter. När komponenten renderas kommer JSX att omvandlas till vanligt JavaScript och visas i DOM:en.