

Podcast

Disciplina: Frameworks front end

Título do tema: React: desenvolvimento de componentes com o Framework do Facebook

Autoria: Anderson da Silva Marcolino

Leitura crítica: Paulo Henrique Santini

Abertura:

Olá, ouvinte! No *podcast* de hoje vamos falar um pouco sobre boas práticas quando vamos trabalhar com o *framework* React.

Escrever código simples e padronizado é essencial para melhorar a qualidade do código e dos projetos, como um todo. Adicionalmente, seguir uma padronização também auxilia na execução de testes com melhor qualidade. Neste contexto, a padronização também deve ser mantida no uso de frameworks, como o React. Podemos achar que por se utilizar um framework, este já traz toda a padronização necessária para se desenvolver com melhor qualidade, porém isso é uma falácia, visto que há diversos padrões boas práticas que podem ser integradas, vejamos algumas delas ao adotarmos o React. Utilize nomes que indiquem do que se trata cada um dos itens do código, como os manipuladores de eventos. Um único manipulador é fácil de gerenciar, mas vários, torna difícil identificar e dar manutenção. Uma outra dica que sempre escutamos ao programarmos é: utilize nomes de variáveis que indiquem facilmente o que esta irá manter ou manipular - é importante destacar que tais nomes não sejam tão extensos. Não é necessário utilizar o termo `this` para realizar o binding, ou seja, referenciar a fonte de dados ou atributos, já que no React as referências são manipuladas automaticamente, graças ao Babel. O Babel é um transcompilador de JavaScript gratuito, de código aberto, usado principalmente para converter o código ECMAScript 2015+ em uma versão compatível com versões anteriores de JavaScript, que podem ser executadas por mecanismos JavaScript mais antigos, ou seja, ele transcreve códigos de versões mais recentes do JavaScript em versões mais antigas, possibilitando que o comportamento da aplicação ocorra normalmente, tal como ocorreria se fosse executado em um navegador atual e atualizado. Utilize o conceito de React Hooks, que é uma espécie de função que permite criar um gancho entre os dados da aplicação e os transferir para os componentes e classes. É necessário importa-lo em todos os componentes que o usarão. Adicionalmente a tal recurso, é interessante utilizar o `useMemo` Hook que cria um cache de informações processadas e que garante uma redução no processamento, se a mesma informação for solicitada novamente. Um recurso de cache é sempre interessante, principalmente para aplicações mobile, visto que reduzem o consumo de pacote de dados, já que somente quando os dados forem atualizados é que uma nova requisição será direcionada ao servidor e, deste modo, fazer o download do pacote - uma vez solicitados os dados ficam em um registro local - o cache, de fato - e tais dados são sempre retornados desta origem. Uma outra boa prática é o desacoplamento de funções em funções puras, para melhorar o código. Mas

o que isso quer dizer? Tal prática busca remover as funções que não dependem do seu componente, ou seja, não dependem de nenhum estado ou ganchos do React. Em outras palavras, isolam-se as funções de acordo com suas responsabilidades, o que resulta em código mais limpo e com maior qualidade, favorecendo manutenções posteriores. Considerando essas boas práticas, certamente seus projetos React obterão maior qualidade e sucesso! É sempre válido lembrar que estas e outras boas dicas podem ser obtidas na documentação oficial do React, que sempre está atualizada e sintonizadas nas novas funcionalidades, quando atualizado!

Fechamento:

Este foi nosso *podcast* de hoje! Até uma próxima oportunidade e bons estudos!