1. Introdução ao Angular

O que é o Angular?

O Angular é um framework de desenvolvimento web criado pelo Google que permite construir aplicativos web modernos e dinâmicos. Ele utiliza a linguagem TypeScript e oferece um conjunto de ferramentas poderosas para criar interfaces de usuário eficientes.

Principais recursos do Angular

Componentização

Data binding

Roteamento

Injeção de dependência

Testabilidade

Diferenças entre Angular e AngularJS

AngularJS vs. Angular

Motivos para a migração

Instalação e configuração do Angular

Para começar a desenvolver com Angular, você precisa configurar o ambiente de desenvolvimento. Isso inclui a instalação do Node.js, do Angular CLI e a criação de um novo projeto Angular.

2. Componentes

Criando componentes

Componentes são a estrutura básica de um aplicativo Angular. Vamos aprender como criar e usar componentes.

Ciclo de vida dos componentes

Os componentes têm um ciclo de vida que permite que você execute código em diferentes estágios. Explore os eventos do ciclo de vida.

Comunicação entre componentes

Como os componentes podem se comunicar uns com os outros e compartilhar dados?

Diretivas

As diretivas são instruções no HTML que permitem adicionar funcionalidades dinâmicas aos elementos da página.

Serviços e injeção de dependência

Os serviços são classes reutilizáveis que podem ser injetadas em componentes para executar tarefas comuns, como fazer solicitações HTTP.

3. Templates e Data Binding

Criando templates

Aprenda a criar templates em Angular usando HTML e adicione diretivas para torná-los dinâmicos.

Data binding unidirecional e bidirecional

Entenda como funciona o data binding e como ligar dados de componentes aos templates.

Interpolação

Uma técnica para exibir variáveis no template.

Diretivas estruturais

Diretivas que afetam a estrutura do DOM, como \*ngIf e \*ngFor.

Event binding

Como responder a eventos do usuário, como cliques de botão.

4. Roteamento

Configuração de rotas

Configure o sistema de roteamento para criar uma navegação em várias páginas em seu aplicativo.

Roteamento aninhado

Como criar rotas filhas para páginas aninhadas.

Passagem de parâmetros

Passe dados entre páginas usando parâmetros de rota.

Guardas de rota

Proteja rotas com guardas para controlar o acesso.

E assim por diante para cada seção do índice. Lembre-se de fornecer exemplos de código, práticas recomendadas, exercícios práticos e referências adicionais para ajudar os leitores a aprofundar seus conhecimentos em cada tópico. Crie exemplos de projeto para que os leitores possam praticar o que aprenderam ao longo da apostila.