

Iniciando com Framework Next.js

Introdução

Bem-vindos à jornada de exploração do Next.js em nossa disciplina de Programação II. Neste segmento, mergulharemos nos conceitos e práticas que definem o Next.js, um framework de ponta para desenvolvimento front-end baseado em Node.js. Ao longo desta aula, desvendaremos o funcionamento, a estrutura e as peculiaridades do Next.js, começando com uma visão geral sobre sua arquitetura e seguindo para a iniciação de um projeto real. Exploraremos as características que fazem do Next.js uma escolha eficaz e popular entre desenvolvedores, como a renderização do lado do servidor e a geração estática de páginas, culminando na aplicação prática desses conceitos. Preparados para adentrar o mundo do Next.js, estabeleceremos uma sólida compreensão inicial para embasar nossos futuros mergulhos práticos no desenvolvimento de aplicações eficientes e modernas.

Apresentação do Next.js

Este momento é especialmente importante para aqueles ansiosos pelo aprendizado em front-end. Dividiremos nosso estudo sobre o Next.js em quatro partes principais. Começaremos com uma introdução ao Next.js, explicando sua estrutura e funcionalidades. Em seguida, aprenderemos como iniciar um projeto Next, explorando os comandos necessários e suas aplicações. Posteriormente, avançaremos para compreender o funcionamento do Next.js em nossa aplicação, identificando e resolvendo possíveis erros.

Continuando, discutiremos a estrutura de um projeto Next.js, aplicando o conhecimento adquirido em prática. O Next.js é um framework baseado em Node.js, direcionado para a construção de interfaces de usuário, ou front-end. Desde seu lançamento em 2016, o Next.js ganhou popularidade devido à sua eficiência e velocidade, competindo diretamente com o React. Essa ferramenta se tornou uma escolha popular para desenvolvedores que desejam aprimorar suas habilidades e portfólios.

Uma das principais características do Next.js é a capacidade de renderizar conteúdo no lado do servidor, oferecendo uma experiência de usuário otimizada através das Single Page Applications (SPA). Essa abordagem permite que o servidor processe e devolva uma página web completa ao navegador, melhorando significativamente o desempenho e a experiência do usuário.

Além disso, o Next.js é apreciado por sua geração estática de sites, configuração inicial simplificada, sistema automático de roteamento e suporte nativo ao TypeScript. Essas características fazem do Next.js uma opção valiosa para desenvolvedores que buscam eficiência, rapidez e atualização tecnológica em seus projetos de front-end. A seguir, exploraremos mais sobre a estrutura, instalação e outros aspectos fundamentais do Next.js, preparando-os para implementar efetivamente esse framework em seus projetos.

Iniciando um projeto Next.js

Vamos avançar no estudo do framework Next.js, focando em como iniciar um projeto Next.js. Este segmento explicará teoricamente o processo de inicialização de um projeto nesse framework, crucial para o desenvolvimento front-end. Iniciaremos discutindo o uso do comando `npx create-next-app` para criar um novo projeto. Este comando faz parte do Node Package Executor (npx), que, em conjunto com o Node Package Manager (npm), facilita a execução de pacotes Node.js.

Antes de prosseguir, é essencial garantir que o Next.js esteja instalado como uma dependência no projeto, utilizando `npm install next`. Esse passo é fundamental para integrar o framework ao nosso ambiente de desenvolvimento. Ao executar o comando de criação do projeto, serão solicitadas informações básicas como o nome da aplicação e a linguagem de programação preferida. Optaremos pelas configurações padrão, que já fornecem uma estrutura inicial robusta para o projeto.

Para iniciar um novo projeto Next.js, utilizamos o comando `'npx create-next-app meu-projeto-next'`, substituindo `'meu-projeto-next'` pelo nome desejado para o seu projeto. Após a execução do comando, um diretório com o nome do projeto será criado e preenchido com a estrutura inicial do Next.js. Para testar, entramos no diretório do projeto usando `'cd meu-projeto-next'` e iniciamos o servidor de desenvolvimento com `'npm run dev'`. Agora, podemos acessar <http://localhost:3000> no navegador para ver a página inicial do Next.js, confirmando que nosso ambiente de desenvolvimento está configurado corretamente. É importante lembrar de verificar a disponibilidade da porta, ajustando conforme necessário para evitar conflitos, especialmente se outros serviços estiverem sendo executados na mesma porta.

Esse processo destaca a integração e a facilidade de uso do Next.js, enfatizando sua eficácia na criação de projetos front-end. Na próxima parte, exploraremos mais sobre a implementação e teste dos elementos iniciais do projeto Next.js, aprofundando nosso entendimento sobre a configuração e a estruturação necessárias para um desenvolvimento eficiente.

Entendendo o funcionamento do Next.js

Vamos mergulhar no funcionamento do Next.js e como implementá-lo em nosso projeto. A compreensão do seu fluxo de trabalho é crucial para integrá-lo de maneira eficiente em nossas aplicações.

No Next.js, o processo começa com o usuário fazendo uma requisição, similar ao que ocorre em outros frameworks JavaScript como Express. Essa requisição é processada pelo servidor Next.js, que, através do Server-Side Rendering (SSR), processa a requisição e envia de volta ao navegador do usuário a página completa, já renderizada. Essa abordagem de SSR é um dos principais atrativos do Next, permitindo uma experiência de usuário mais rápida e fluida, como observamos em aplicações dinâmicas e populares como Uber, TikTok e Twitch, onde a velocidade e eficiência de carregamento são essenciais.

Para ilustrar o SSR no Next.js, consideremos uma página que exibe postagens de um blog. Quando um usuário acessa a página do blog, o servidor Next.js processa a requisição e retorna o HTML completo da página, incluindo as postagens do blog já renderizadas. Isso elimina a necessidade de carregar os dados após o carregamento da página, melhorando a performance e a experiência do usuário. O código para isso pode incluir uma função `'getServerSideProps'` no arquivo da página do blog, que busca as postagens do blog no servidor antes de renderizar a página.

Para iniciar a implementação do Next.js em nosso projeto, devemos primeiro assegurar que o ambiente de desenvolvimento está preparado e funcionando corretamente. Isso envolve, entre outras coisas, verificar se todas as dependências necessárias estão instaladas e atualizadas. Utilizamos o comando `npx create-next-app` para criar uma nova aplicação Next.js, escolhendo configurações iniciais e definindo as características do projeto, como o uso de TypeScript e configurações específicas de roteamento e estilo.

Durante a criação do projeto, enfrentaremos decisões críticas, como escolher entre as configurações padrão ou customizar conforme as necessidades específicas do projeto. Essas escolhas influenciarão diretamente na facilidade de manutenção e na capacidade de expandir a aplicação no futuro.

Concluindo, a implementação do Next.js envolve não apenas a configuração técnica, mas também uma compreensão detalhada de como o framework processa e renderiza as páginas, otimizando a experiência do usuário. Na próxima parte, vamos explorar mais sobre a estrutura de projetos Next.js e como configurar rotas e demais elementos essenciais para nosso aplicativo.

Estrutura de um projeto Next.js

Aqui, comentaremos sobre a estrutura de um projeto Next.js, elemento fundamental para o desenvolvimento front-end. Após instalar o Next.js, você encontrará três pastas principais que formam a espinha dorsal do projeto.

A primeira pasta é `node_modules`, que contém todos os módulos necessários para o projeto. Essa pasta é um componente padrão em projetos JavaScript e pode ser reconstruída usando o comando `npm install`, se necessário.

A segunda pasta importante é `public`. Aqui, você armazenará recursos visuais como imagens, estilos e scripts que são diretamente acessíveis no navegador. Essa pasta é crucial para a gestão dos ativos estáticos do seu site ou aplicação web.

A terceira pasta, `app`, é onde o cerne do seu código front-end reside. Essa pasta contém o código que define a estrutura, lógica e estilo da aplicação. Pense nela como o back-end do seu front-end, onde a mágica do processamento e renderização das páginas acontece.

Na pasta `app`, podemos encontrar subdiretórios como `pages`, onde cada arquivo representa uma rota específica no aplicativo. Por exemplo, um arquivo `index.js` dentro de `pages` corresponderá à rota raiz (`/`), enquanto um arquivo `about.js` representará a rota `/about`. Essa organização de arquivos facilita o roteamento automático do Next.js, onde a estrutura de arquivos define a estrutura de navegação do site.

Além dessas, você pode criar pastas adicionais para organizar melhor o projeto conforme sua necessidade. No entanto, lembre-se de que no Next.js, a estrutura de pastas desempenha um papel crucial no roteamento e na organização dos componentes do seu aplicativo.

Para ilustrar, considere a seguinte estrutura de código simplificada de um projeto Next.js:

```
meu-projeto-next/
```

```
├─ node_modules/
```

```
├─ public/
```

```
|   └─ favicon.ico
```

```
|   └─ style.css
```

```
└─ app/
```

```
    └─ pages/
```

```
        └─ index.js
```

```
        └─ about.js
```

```
    └─ components/
```

```
        └─ header.js
```

```
        └─ footer.js
```

Prosseguindo, ajustaremos a estrutura do nosso projeto para facilitar a manutenção e compreensão. Isso envolve separar claramente as partes do front-end e do back-end, mesmo que estejam no mesmo repositório. Por

exemplo, movendo arquivos não relacionados ao front-end para uma pasta src (source), ajudamos a clarificar a separação de responsabilidades dentro do projeto.

Essa organização não só facilita a manutenção como também prepara você para as práticas do mercado de trabalho, onde frequentemente o front-end e o back-end são gerenciados como projetos separados, embora possam estar no mesmo repositório para fins de desenvolvimento.

Concluindo, compreender a estrutura de um projeto Next.js e saber como adaptá-la às necessidades específicas do seu projeto é crucial para o sucesso do desenvolvimento front-end. Nas próximas aulas, aprofundaremos no roteamento do Next.js e como ele facilita a construção de aplicativos web eficientes. Até a próxima!

GitHub da Disciplina

Você pode acessar o repositório da disciplina no GitHub a partir do seguinte link: <https://github.com/FaculdadeDescomplica/ProgramacaoII>. Esse repositório tem como principal objetivo guardar os códigos das aulas práticas da disciplina para aprimorar suas habilidades em vários tópicos, incluindo a criação e consumo de APIs com controle de autenticação utilizando Node.js e utilizando boas práticas de programação e mercado.

Conteúdo Bônus

Para expandir seu conhecimento em Next.js e entender sua aplicabilidade prática, recomendo o vídeo “O que é Next.js, na prática, em 15 minutos!”. Este material é apresentado por Luiz Duarte no canal LuizTools no YouTube. A apresentação oferece uma visão concisa e efetiva do funcionamento do Next.js, ilustrando através de exemplos práticos como iniciar e manipular projetos com este framework.

Referências Bibliográficas

Bibliografia Básica:

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 7.ed. Pearson: 2018.

MEDEIROS, L. F. de. **Banco de dados**: princípios e prática. Intersaberes: 2013.

VICCI, C. (Org.). **Banco de dados**. Pearson: 2014.

Bibliografia Complementar:

CARDOSO, L. da C. **Design de aplicativos**. Intersaberes: 2022.

JOÃO, B. do N. **Usabilidade e interface homem-máquina**. Pearson: 2017.

LEAL, G. C. L. **Linguagem, programação e banco de dados**: guia prático de aprendizagem. Intersaberes: 2015.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de dados**: implementação em SQL, PL/SQL e Oracle 11g. Pearson: 2013.

SETZER, V. W.; SILVA, F. S. C. **Bancos de dados**. Blucher: 2005.

Ir para exercício