

Criando Componentes Compartilhados em React: O Componente Banner

Por Escola Dnc

Introdução

Neste ebook, exploraremos o processo de criação de componentes compartilhados em React, com foco especial no desenvolvimento de um componente de banner. Abordaremos desde a estruturação básica do componente até técnicas avançadas de estilização e passagem de propriedades. Este guia é ideal para desenvolvedores React que desejam aprimorar suas habilidades na criação de componentes reutilizáveis e eficientes.

Estrutura Básica do Componente Banner

Criação dos Arquivos

Para iniciar o desenvolvimento do componente Banner, seguimos os seguintes passos:

1. Criamos uma pasta chamada "banner" dentro do diretório de componentes
2. Dentro desta pasta, criamos dois arquivos:
 - `Banner.jsx` : para a lógica e estrutura do componente
 - `Banner.css` : para os estilos específicos do banner

Estrutura do Componente em JSX

No arquivo `Banner.jsx`, definimos a estrutura básica do componente:

```
import React from 'react';import './Banner.css';const Banner =  
(props) => { return ( <div className="banner displayFlex  
alignEnd" style={{backgroundImage: `url(${props.image})`} >  
<div className="container" > <div className="titlePanel  
displayFlex alignCenter jcCenter" > <h1>{props.title}</h1>  
</div> </div> </div> );};export default Banner;
```

Pontos importantes:

- Utilizamos `props` para receber propriedades de forma flexível
- A imagem de fundo é definida através de um estilo inline
- O título é passado como uma prop e renderizado dentro de um `h1`

Estilização do Componente

Estilos CSS Básicos

No arquivo `Banner.css`, definimos os estilos básicos para o componente:

```
.banner { background-position: center; background-size: cover; height: 356px; width: 100%; background-color: #ccc; /* Fallback color */}.titlePanel { background-color: white; border-radius: 10px 10px 0 0; font-size: 50px; height: 150px; margin: 0 auto; max-width: 500px;}
```

Destaques:

- Utilizamos `background-size: cover` para garantir que a imagem cubra todo o banner
- Definimos uma cor de fundo fallback para quando a imagem ainda não foi carregada
- O `titlePanel` tem bordas arredondadas apenas no topo

Uso de Classes Utilitárias

Aproveitamos classes utilitárias definidas em um arquivo CSS global:

- `displayFlex` : para usar flexbox
- `alignEnd` : para alinhar itens ao final do container
- `alignCenter` : para centralizar itens verticalmente
- `jcCenter` : para centralizar itens horizontalmente

Estas classes ajudam a manter o código mais limpo e reutilizável.

Passagem de Propriedades e Renderização Dinâmica

Uso de Props

O componente Banner recebe duas props principais:

1. `image` : URL da imagem de fundo
2. `title` : Texto a ser exibido no banner

Exemplo de uso:

```
<Banner image="/about.jpg" title="Sobre Nós" />
```

Renderização Dinâmica da Imagem de Fundo

A imagem de fundo é renderizada dinamicamente através de um estilo inline:

```
style={{backgroundImage: `url(${props.image})`}}
```

Isso permite que diferentes imagens sejam usadas para cada instância do banner.

Considerações Finais e Boas Práticas

Reutilização e Flexibilidade

- O componente Banner foi projetado para ser altamente reutilizável
- Pode ser facilmente adaptado para diferentes páginas e contextos
- A passagem de props permite personalização sem alterar o componente base

Performance e Carregamento

- Considere a otimização das imagens para melhorar o tempo de carregamento
- O uso de uma cor de fundo fallback melhora a experiência do usuário durante o carregamento

Manutenção e Escalabilidade

- A estrutura modular facilita a manutenção e atualização do componente
- Considere adicionar PropTypes para validação de props em projetos maiores

Conclusão

A criação de um componente de Banner compartilhado demonstra princípios importantes do desenvolvimento React:

- Modularidade e reutilização de código
- Uso eficiente de props para personalização
- Integração de estilos CSS com lógica JSX
- Considerações de performance e experiência do usuário

Ao dominar essas técnicas, você estará bem equipado para criar componentes React robustos e flexíveis, melhorando a eficiência e a manutenibilidade de seus projetos.

