

Configurando um Projeto TypeScript com GitHub e Vercel

Por Escola Dnc

Introdução

Neste guia, vamos explorar o processo de criação e configuração de um projeto TypeScript, integrando-o com o GitHub para controle de versão e com a Vercel para implantação contínua. Este processo é fundamental para desenvolvedores que desejam estabelecer um fluxo de trabalho eficiente e moderno para seus projetos web.

Criando o Projeto no GitHub

Passos Iniciais

1. Acesse sua conta no GitHub
2. Clique em "New" para criar um novo repositório
3. Nomeie o repositório de acordo com seu projeto local (ex: dnc-sales-board-gsd)
4. Deixe a descrição em branco por enquanto
5. Selecione a opção "Public" para o repositório
6. Clique em "Create repository"

Configuração Inicial

Após criar o repositório, você verá instruções semelhantes às usadas em projetos React. Porém, adaptaremos essas instruções para nosso projeto TypeScript.

Importante: Mantenha o nome do repositório consistente entre seu projeto local e o GitHub para evitar confusões futuras.

Conectando o Projeto Local ao GitHub

Comandos Git Essenciais

Para conectar seu projeto local ao repositório GitHub recém-criado, siga estes passos no terminal do seu VSCode:

1. Inicialize o repositório Git local:

```
git init
```

2. Adicione todos os arquivos ao staging:

```
git add .
```

3. Faça o primeiro commit:

```
git commit -m "first commit"
```

4. Renomeie a branch principal para 'main':

```
git branch -m main
```

5. Adicione a origem remota (copie o URL do seu repositório GitHub):

```
git remote add origin [URL_DO_SEU_REPOSITÓRIO]
```

6. Envie os arquivos para o GitHub:

```
git push -u origin main
```

Dica Importante

Sempre verifique se está na branch correta antes de fazer push para o repositório remoto.

Integrando o Projeto com a Vercel

Configuração na Plataforma Vercel

1. Acesse sua conta na Vercel
2. Clique em "Add New" e selecione "Project"
3. Localize e selecione o repositório que você acabou de criar no GitHub
4. Clique em "Import"

Processo de Deploy

A Vercel iniciará automaticamente o processo de deploy do seu projeto. Este processo inclui:

- Construção (build) do projeto
- Verificação de erros
- Criação de uma URL única para seu projeto

Benefício: A Vercel oferece previews automáticos para pull requests, facilitando a revisão de código e testes antes da integração.

Vantagens da Integração GitHub-Vercel

1. **Deployment Contínuo:** Cada push para a branch principal resulta em um novo deploy automático.
2. **Previews de Pull Requests:** A Vercel cria links de preview para cada PR, permitindo testes em ambiente isolado.
3. **Verificação de Build:** A plataforma verifica se o código está buildando corretamente antes de cada deploy.
4. **CICD Integrado:** O processo de Integração Contínua e Entrega Contínua é gerenciado automaticamente.
5. **Facilidade de Colaboração:** Equipes podem revisar mudanças em ambientes de staging antes de mesclar com a branch principal.

Configurações Adicionais do Projeto

Padrões de Código

- **ESLint:** Configurado para manter consistência no estilo de código.
- **Prettier:** Utilizado para formatação automática do código.

Estrutura Inicial do Projeto

- Configuração de rotas básicas
- Criação da estrutura inicial de páginas

Lembre-se: Manter uma estrutura de projeto organizada desde o início facilita o desenvolvimento e a manutenção futura.

Conclusão

Ao seguir este guia, você terá configurado com sucesso um projeto TypeScript integrado ao GitHub e à Vercel. Esta configuração proporciona um ambiente de desenvolvimento robusto, com controle de versão eficiente e deployment automatizado.

As práticas aqui descritas estabelecem uma base sólida para o desenvolvimento contínuo do seu projeto, permitindo um fluxo de trabalho ágil e colaborativo. À medida que você avança no desenvolvimento, lembre-se de aproveitar os recursos de CI/CD oferecidos pela integração GitHub-Vercel para manter a qualidade e eficiência do seu processo de desenvolvimento.

