

Documentação Completa - Conversor de Moedas

Projeto Conversor de Moedas

=====

Descrição Geral

Este projeto é um conversor de moedas simples, com backend em Python usando FastAPI e frontend em HTML, CSS e JavaScript puro.

O backend consome a API pública 'exchangerate.host' usando uma chave de API fornecida.

O projeto permite:

- Selecionar moedas de origem e destino através de dropdowns.
- Inserir um valor e receber a conversão em tempo real.
- Tratar erros de conexão, entrada inválida e falta de dados.

Estrutura do Backend - main.py

1. Importações:

- FastAPI: criar a API.
- Query: validar parâmetros de URL.
- CORSMiddleware: permitir que o frontend acesse o backend.
- requests: realizar requisições HTTP à API externa.

2. app = FastAPI():

Cria a instância da aplicação.

3. Middleware CORS:

Permite que qualquer origem acesse o backend, permitindo GET, POST, e outros métodos.

4. ACCESS_KEY:

Chave de API usada para conversão de moedas.

5. Rota "/":

Retorna mensagem simples confirmando que a API está rodando.

6. Rota "/moedas":

Retorna uma lista fixa de moedas com código e descrição.

Evita falha caso a API externa não responda.

7. Rota "/converter":

Recebe parâmetros: de, para, valor.

Monta URL da API externa com a chave.

Faz a requisição e retorna JSON com valor convertido ou mensagem de erro.

8. Tratamento de exceções:

Captura erros de conexão e retorna mensagem adequada.

Estrutura do Frontend - script.js

1. Seleção de elementos HTML:

Dropdowns, input de valor, botão e área de resultado.

2. Função carregarMoedas():

- Faz fetch para /moedas do backend.
- Preenche dropdowns com código e descrição.
- Define valores padrão USD -> BRL.
- Trata erros de conexão.

3. Função converterMoeda():

- Pega valores do frontend.
- Valida valor digitado.
- Faz fetch para /converter do backend.
- Mostra resultado ou mensagem de erro.

4. Eventos:

- Ao carregar página, chama carregarMoedas.
- Ao clicar no botão, chama converterMoeda.

Erros Corrigidos

1. Lista de moedas não carregava: resolvido criando lista fixa no backend.
2. Erro de conversão: resolvido adicionando chave de API.
3. Problemas de CORS: resolvido adicionando middleware CORS.
4. ModuleNotFoundError: requests: resolvido instalando a biblioteca.
5. Dropdown undefined: resolvido validando dados recebidos do backend.

Conclusão

O projeto está finalizado, funcional e seguro para consumo de API pública.
Frontend e backend estão conectados corretamente e tratam erros de forma robusta.