

Esse código em JavaScript implementa uma API utilizando o framework Express para fornecer informações sobre criptomoedas. Ele possui duas principais funcionalidades: a busca do preço de uma criptomoeda por ID e a busca por nome. A aplicação se conecta a um banco de dados (via Sequelize), utiliza um cache Redis para otimizar o desempenho, e consome dados da API externa do CoinGecko para obter os preços das criptomoedas em tempo real.

## 1. Configuração e Conexão

- A aplicação carrega as variáveis de ambiente com `dotenv` e define a porta de execução.
- Conecta-se ao Redis para caching e ao banco de dados SQL utilizando o Sequelize.

## 2. Cache Redis:

- Redis é usado para armazenar temporariamente (por 5 minutos) os dados das consultas realizadas, evitando chamadas repetidas à API do CoinGecko e melhorando a performance da aplicação.

## 3. Rota `/crypto/:id` :

- Busca o preço da criptomoeda utilizando um **ID**.
- Verifica primeiro o cache Redis; se encontrado, retorna os dados diretamente.
- Caso não esteja em cache, faz uma requisição à API do CoinGecko, obtém o nome, símbolo e preço, e armazena os dados no banco de dados e no cache.

## 4. Rota `/preco/:nome` :

- Similar à rota anterior, mas a busca é realizada pelo **nome** da criptomoeda.
- Caso o dado não esteja em cache, busca as informações na API do CoinGecko, valida se a criptomoeda existe, e armazena no banco e cache.
- Retorna os dados com a fonte indicando se vieram do cache ou da API externa.

## 5. Banco de Dados:

- Os dados das consultas são persistidos no banco de dados para armazenar um histórico das consultas realizadas.

## 6. Inicialização:

- O servidor é iniciado e ouve na porta especificada, permitindo que a API receba requisições e forneça os dados requisitados.

Essas funcionalidades fazem da aplicação uma solução eficiente para consultas de preços de criptomoedas, otimizando a performance com uso de cache e persistindo dados para análises futuras.