

Tarefa para Processo Seletivo de Golang

Criar uma API utilizando Golang com autenticação via JWT. A API deve expor endpoints para as seguintes funcionalidades: cadastro, login, recuperação de senha e listagem de usuários. Todos os endpoints devem ser implementados seguindo as melhores práticas, com validações e tratamento adequado de erros.

Requisitos:

1. **Endpoint de Cadastro:**
 - Deve receber os seguintes campos: nome completo, e-mail, senha, confirmar senha e endereço (Rua, Bairro, Número, Cidade, Estado, CEP).
 - Deve haver validação do CEP e a utilização de uma API externa para buscar os dados do endereço através do CEP.
2. **Endpoint de Login:**
 - Deve receber e-mail e senha.
 - Deve gerar e retornar um token JWT para autenticação nas demais requisições.
3. **Endpoint de Recuperação de Senha:**
 - Deve receber o e-mail do usuário.
 - Deve gerar um token de recuperação e enviar por e-mail (o envio do e-mail pode ser mockado).
4. **Endpoint de Listagem de Usuários:**
 - Deve retornar uma lista com todos os usuários cadastrados no sistema.
 - Este endpoint deve estar protegido por autenticação JWT.

Detalhes:

1. **Autenticação:**
 - Utilizar JWT para autenticação nos endpoints que requerem usuário autenticado.
 - Implementar middleware de validação do token JWT.
2. **Estrutura do Projeto:**
 - Seguir os princípios de design SOLID.
 - O ideal é utilizar um layout estruturado com services, controllers, repositories e models.
 - Utilizar boas práticas de Go, como contextos para controle de requests.
3. **Validações e Boas Práticas:**
 - Implementar validações para todos os campos obrigatórios.
 - Realizar tratamento de erros adequado, retornando mensagens e códigos HTTP apropriados.
4. **Utilização de API Externa para CEP:**
 - Integração com uma API externa para buscar os dados de endereço através do CEP fornecido.
 - Implementar o serviço de integração com essa API de forma desacoplada para fácil manutenção.

Implementação

1. **Configuração Inicial:**
 - Criar um novo projeto Go com uma estrutura organizada.
 - Configurar dependências necessárias, como pacotes de JWT, validação e integração com APIs externas.
2. **Endpoint de Cadastro:**
 - Criar o endpoint para cadastrar novos usuários.
 - Implementar validação de senha e verificação de endereço via API externa.
3. **Endpoint de Login:**
 - Criar o endpoint para login e geração de token JWT.
4. **Endpoint de Recuperação de Senha:**
 - Criar o endpoint para recuperação de senha, gerando um token de recuperação.
5. **Endpoint de Listagem de Usuários:**
 - Criar o endpoint para listar todos os usuários, protegido com autenticação JWT.

Certifique-se de que o código esteja bem documentado e inclua instruções claras para a configuração e execução do projeto, incluindo dependências e rotinas de inicialização.

Entrega

- O código deve ser versionado no GitHub ou GitLab, e deve conter um README com as instruções para rodar o projeto.
- Será avaliado o uso de boas práticas, organização do código, e aderência aos princípios SOLID.
- Deve-se utilizar o Docker
- Deve-se utilizar ou o mongodb ou o sqlite como base de dados.