Rafael B. Castilhos
Email: rafaelbcastilhos@gmail.com
Github: rafaelbcastilhos/brytecnologia-desafio

Desafio prático BRy Tecnologia - Estágio desenvolvedor back-end Java

Geral: Antes de iniciar o desenvolvimento, reservei um espaço de tempo para ler, assistir e executar aplicações voltadas para API REST com Spring, também busquei entender de maneira simples e superficial o que é uma assinatura e um certificado digital e ler a documentação do BouncyCastle, por fim tentei praticar pequenos exemplos integrando todos os itens citados acima.

Como possuo entendimento de APIs REST e suas aplicações, não foi necessário abordar esse tópico, passando diretamente para o desenvolvimento do núcleo e regras definidas no desafio.

Para realizar as requisições HTTP e demais testes, foi utilizado o software Postman, e para gerenciar as dependências foi utilizado o Maven, ambos escolhidos por preferência pessoal.

Para iniciar o servidor é necessário entrar no diretório *initial* e executar "./mvnw spring-boot:run"

Organização do projeto: A organização do projeto e demais classes, possui como objetivo delimitar as responsabilidades de cada classe, separando-as em camadas.

No diretório de aplicação, é dada a inicialização e execução da ferramenta Spring. Já o *ApiController* é a porta de entrada das requisições HTTP, definindo os endpoints e métodos disponibilizados.

Nos manipuladores, a ideia é que possa haver regras específicas ao determinado endpoint, realizando manipulações nos dados de entrada e saída, aqui também poderia haver a autenticação do cliente que realizou a requisição, caso necessário.

Nos serviços, o intuito é que possa ter métodos mais genéricos possíveis, na qual outros manipuladores podem fazer uso, sendo assim reutilizáveis para diversas aplicações. Dado isso, utilizando Singleton otimizaria a execução, uma vez que pode ter o uso da instância compartilhada.

Por fim, existe uma classe ajudante, cujo objetivo é lidar com a entrada e criação de arquivos.

Etapa 1 - Obtenção do resumo criptográfico: Não obtive dificuldades, foi resolvido de forma facilitada utilizando um conversor disponibilizado na web, que realiza o hash em sha256 e escreve a saída em hexadecimal no arquivo sha256_hexadecimal.txt, localizado na pasta initial/resources/arquivos/.

Etapa 2 - Realizar uma assinatura digital: Acredito que tenha sido a etapa mais demorada e mais complexa para entender e iniciar o desenvolvimento. Não tinha conhecimento prévio na biblioteca BouncyCastle, o que levou certo tempo até realizar a leitura da documentação e maiores detalhes para ter um entendimento básico. Felizmente já havia me deparado com os tipos de dados *File* e *InputStream* e seus derivados, então a manipulação dos arquivos foi tranquila.

Etapa 3 - Verificar a assinatura gerada: Tive um leve problema de interpretação para identificar o que e como estava sendo pedida essa verificação. Fiz uso de exemplos disponíveis em fóruns na web para facilitar e ter melhor entendimento do contexto, adaptando-os para o objetivo do desafio.

Etapa 4 - API REST: Essa etapa consegui realizar de forma facilitada e tranquila, com a etapa 2 e 3 elaboradas, apenas foi necessário definir os endpoints, implementar os parâmetros dos atributos necessários e ajustar o local de captura de algumas exceções.

Extra - Documentação da API REST: Com objetivo de facilitar e documentar as APIs desenvolvidas, foi elaborado uma documentação com a ferramenta Swagger, que pode ser acessada no endereço: http://brytecnologia-api.s3-website-sa-east-1.amazonaws.com/

Com isso, agradeço a oportunidade de participar do desafio!