



## Welcome to Bexs Bank challenge!

O Bexs é um banco de câmbio digital, onde conectamos o Brasil e o mundo via soluções de câmbio e pagamentos cross-border.

Com o Bexs Banco, oferecemos soluções de câmbio customizadas e via API, atendendo empresas de atuação internacional, de comércio exterior, Fintechs e AgTechs.

Com a Bexs Pay, oferecemos soluções de pagamentos cross-border para que empresas globais digitais possam oferecer seus produtos, receber e pagar em moeda local, atuando no Brasil sem uma operação local.

Nós fortemente acreditamos que todo mundo pode ter acesso aos serviços de câmbio e pagamentos digitais, se conectando em todo o mundo. Para resolver isso, utilizamos tecnologia :)

## Backend challenge

No Bexs fazemos de tudo para oferecer os melhores serviços aos nossos consumidores, fornecendo os melhores e mais rápidos serviços digitais. Para alcançar esse objetivo, usamos APIs.

### Case

Hoje nós temos uma empresa parceira que está criando um e-commerce que vende produtos internacionais para Brasileiros, sendo assim, eles precisam utilizar um serviço de pagamentos para a efetivação dessas vendas.

E você está aqui para nos ajudar com essa solução :)

Quando estamos desenvolvendo uma solução, tentamos seguir algumas boas práticas de desenvolvimento e padrões (como os mencionados em Clean Code, Clean Architecture, The Pragmatic Programmer, Domain-Driven Design, Microservice Patterns).

Para nós, escrever bom código é um dever e esperamos que pessoas que queiram entrar para o nosso time pensem igual.

### 1. O que queremos fazer

Nós esperamos que você desenvolva serviços que forneça API Rest e que implemente as features a seguir, respeitando os requisitos técnicos:

#### 1.1. Cadastro de parceiro(partners)

Crie uma API que receba e salve no banco de dados um parceiro com os campos representados no JSON a seguir:



```
{  
  "id": "1",  
  "trading_name": "International Ecommerce",  
  "document": "1284498339812/0001",  
  "currency": "USD"  
}
```

1. Id deve ser único, mas não necessariamente um número.
2. Document deve ser único.
3. Currency devemos aceitar apenas os códigos ISO de Moeda abaixo, qualquer outro devera ser retornado uma mensagem de erro

### ISO Moedas

|  
ISO code  
|  
Currency

USD	United States dollar
EUR	Euro
GBP	Pound sterling

### 1.2. Registro dos pagamentos

Crie uma API que receba informações de pagamentos e salve no banco de dados com os campos representados no JSON a seguir:

```
{  
  "partner_id": "1",  
  "amount": "99.05",  
  "consumer": {  
    "name": "Oliver Tsubasa",  
    "national_id": "30243434597"  
  }  
}
```

1. Partner id, precisa existir, caso contrário, negar o pagamento e retornar uma mensagem de erro.
2. Amount deve ser representado em duas casas decimais.
3. Consumer national id deve conter 11 caracteres.
4. Todos os campos são obrigatórios.
5. Gerar um id único para o pagamento e retorná-lo na resposta
6. Calcular o valor da moeda estrangeira, de acordo com a moeda cadastrada no partner, seguindo as quotasções a seguir, deve ser adicionado no retorno



o campo “foreign\_amount” com o valor calculado (amount x quote = foreign\_amount):

### Quotes

|  
ISO code  
|  
Quote

USD	4.75
EUR	5.50
GBP	6.35

### 1.3 Consultar pagamentos

Crie uma API que receba o id do pagamento e retorne as informações do pagamento.

### 1.4 Consultar pagamentos com paginação.

Crie uma API que receba o parâmetro offset e limit e faça consulta paginada.

### 1.5 Bloqueio de pagamentos duplicados

No momento de um pagamento, podem ocorrer problemas e isso pode gerar alguns re-envios de transações, para fornecer uma boa experiência ao nosso cliente, precisamos evitar a duplicação de pagamentos. Crie uma funcionalidade que não aceite registrar pagamentos duplicados por X segundos.

### 1.6 Requisitos Técnicos

- A linguagem de programação e banco de dados que serão utilizados ficam a seu critério.
- Deve ser utilizado o github como versionador de código.
- Crie uma documentação (README.md) explicando como executar localmente a aplicação (foque na simplicidade e não esqueça que vamos testar sua aplicação sem a sua ajuda)

### Método de avaliação

- Objetivo: seu código seguiu todos os requisitos apresentados no item 1;
- Código: seu código está simples, organizado e fácil de evoluir;
- Documentação: sua documentação está simples, clara e objetiva;



### **Como entregar meu desafio**

- Faça a organização, design, teste e documentação do seu código como se estivesse indo para produção
- Faça o push do seu código a um repositório privado do Github e adicione como colaborador o usuário ArturMartiniBexs
- Ao finalizar, entre em contato com o pessoal do RH

**Good Luck! =)**