

Necessidade

Será necessário desenvolver uma API REST para ser integrada nas aplicações web e mobiles, possibilitando simulações de empréstimos. A API deve considerar que será informado o valor desejado, quantidade de parcelas, instituições e convênios.

Requisitos

1. O Backend da aplicação deve ser desenvolvido utilizando a linguagem PHP com o framework Laravel
2. Serão disponibilizados três arquivos no formato json contendo as informações de instituições, convênios e taxas por instituição. Com base nessas informações desenvolva as rotas dos serviços necessários para o funcionamento da aplicação.
3. Não utilizar banco de dados.
4. Criar uma rota para retornar as instituições disponíveis
 - a. O method deverá ser do tipo GET
 - b. O response deve ser no formato **chave** e **valor**. Onde a chave será o identificador da instituição e o valor o nome da instituição.
 - c. Exemplo de retorno:

```
[
  {
    "chave": "PAN",
    "valor": "Pan"
  },
  {
    "chave": "OLE",
    "valor": "Ole"
  },
  {
    "chave": "BMG",
    "valor": "Bmg"
  }
]
```

5. Criar uma rota para retornar os convênios disponíveis
 - a. O method deverá ser do tipo GET
 - b. O response deve ser no formato **chave** e **valor**. Onde a chave será o identificador do convênio e o valor o nome do convênio
 - c. Exemplo de retorno:

```
[
  {
    "chave": "INSS",
    "valor": "INSS"
  },
  {
    "chave": "FEDERAL",
    "valor": "Federal"
  },
  {
    "chave": "SIAPE",
    "valor": "Siape"
  }
]
```

6. Criar uma rota para realizar a simulação de crédito disponível para o cliente
 - a. O method deverá ser do tipo POST
 - b. Payload (JSON)
 - i. valor_emprestimo
 1. Atributo do tipo float
 2. Parâmetro obrigatório
 - ii. instituicoes
 1. Atributo do tipo Array
 2. Parâmetro opcional
 - iii. convenios
 1. Atributo do tipo Array
 2. Parâmetro opcional
 - iv. parcela
 1. Atributo do tipo numérico
 2. Parâmetro opcional
 - c. Reponse
 - i. Para cada instituição deverá ser retornado as informações de parcelas, valor da parcela, taxas e convênio
 - ii. Exemplo:

```

"BMG": [
  {
    "taxa": 2.05,
    "parcelas": 72,
    "valor_parcela": 305.97,
    "convenio": "INSS"
  },
  {
    "taxa": 2.05,
    "parcelas": 60,
    "valor_parcela": 354.26,
    "convenio": "INSS"
  },
  {
    "taxa": 2.05,
    "parcelas": 48,
    "valor_parcela": 414.66,
    "convenio": "INSS"
  },
  {
    "taxa": 2.05,
    "parcelas": 36,
    "valor_parcela": 554.48,
    "convenio": "INSS"
  },
  {
    "taxa": 1.9,
    "parcelas": 84,
    "valor_parcela": 286.51,
    "convenio": "INSS"
  }
],

```

7. Para realizar o calculo do valor da parcela deverá ser considerada a seguinte fórmula
 - a. Valor solicitado multiplicado pelo o valor do coeficiente
8. Ao finalizar o desenvolvimento da API a mesma deverá ser disponibilizada no github e o link do repositório deverá ser enviado por e-mail para a avaliação, juntamente com a collection do postman ou insomnia para importação e realização dos testes.

Observação

A arquitetura do projeto e a interpretação da prova também são critérios de avaliação.