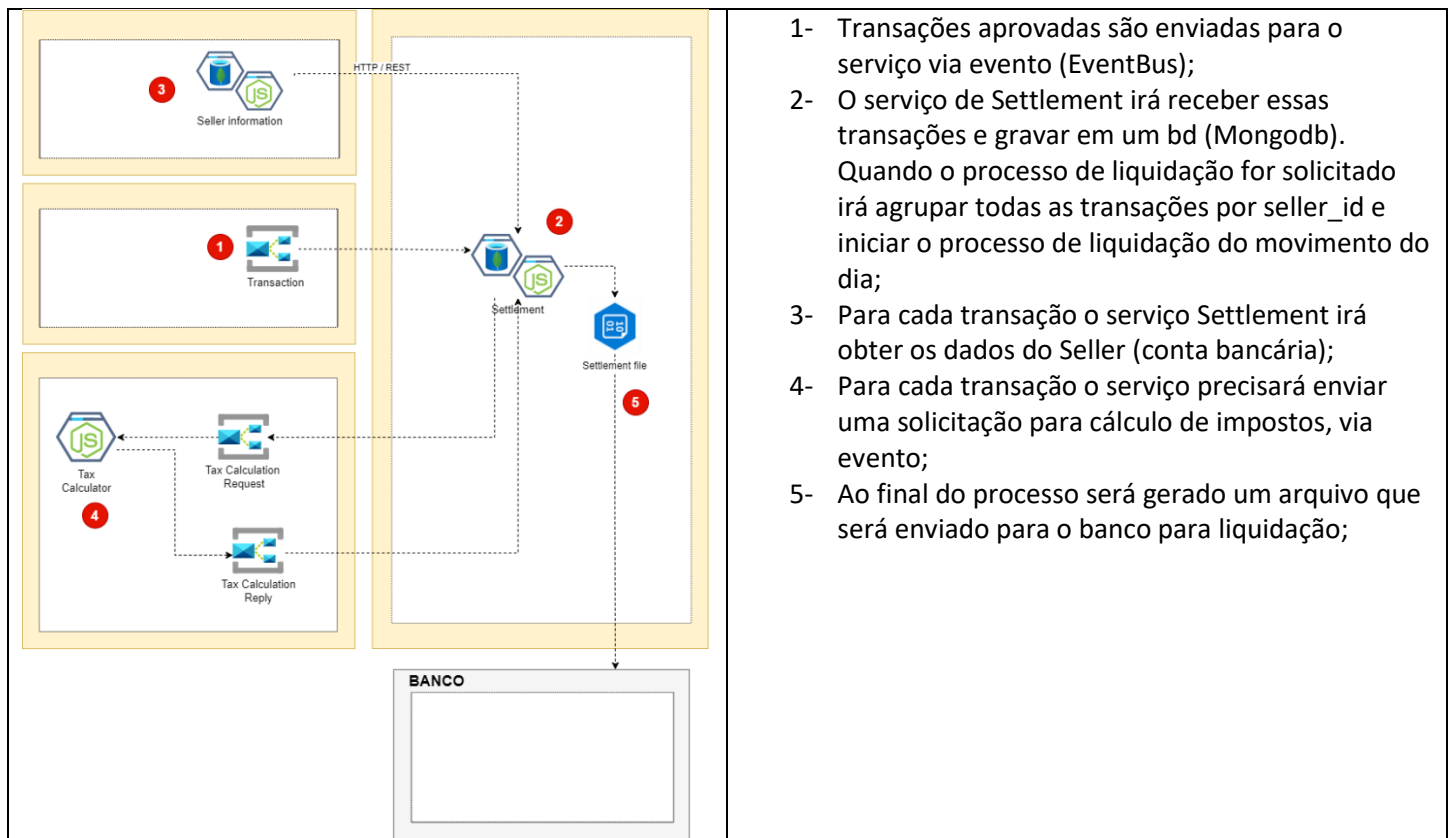


O objetivo agora é complementar a nossa API Seller Information com algumas validações

Cenário:

Ainda estamos falando do mesmo cenário e da mesma API (seller_information)



Fase 2:

São duas tarefas:

1- Adicionar validações nos endpoints

POST /v1/sellers

Inclui um novo Seller

request:

```
{
  "sellerId": 1,
  "name": "McDonalds",
  "cnpj": "90891366000190",
  "bankCode": 33,
  "bankAccount": 1000,
  "notes": ""
}
```

- Se as regras de validação não forem atendidas, deve retornar um erro HTTP 400 (Bad request).
- Todos os campos são obrigatórios, exceto o campo “notes”
- CNPJ deve ser um cnpj válido
- Se o sellerId já existir, retornar erro HTTP 409 (Conflict)

GET /v1/sellers/1

Obtém seller por id

response:

```
{
  "sellerId": 1,
  "name": "McDonalds",
  "cnpj": "90891366000190",
  "bankCode": 033,
  "bankAccount": 1000,
  "notes": ""
}
```

- Se não receber o seller_id, retornar HTTP 400 (Bad request)
- Se não encontrar o seller_id cadastrado, retornar HTTP 404 (Not found)

PATH /v1/sellers/1

Altera informações de um seller

request:

```
{
  "notes": "anotações 1"
}
```

- Se não receber o seller_id, retornar HTTP 400 (Bad request)
- Se não encontrar o seller_id cadastrado, retornar HTTP 404 (Not found)

Aqui tem uma referência sobre boas práticas de API Restfull: <https://opensource.zalando.com/restful-api-guidelines/>

2- Escrever testes automatizados

Para escrever os testes vamos utilizar Mocha e Chai, que são o padrão de testes na PagoNxt.

Referência: <https://medium.com/@rafaelvicio/testando-api-rest-com-mocha-e-chai-bf3764ac2797>

Para cada endpoint escrever testes que validem o retorno de sucesso, ex: cadastro HTTP 201 e os cenários de retorno de erro, ex: Enviou um cnpj inválido ou não enviou cnpj.

Ao executar “npm test” os testes devem ser executados.

DESAFIO:

Se tiver um tempo disponível, tente resolver o desafio abaixo:

Verificar se uma palavra é um "palindromo". Palíndromos são palavras ou frases que podem ser lidas da esquerda para a direita ou da direita para a esquerda.

```
function palindrome(str){
  return false;
}

console.log('>> Shoud return true:')
palindrome("a") //true
palindrome("pop") //true
palindrome("deed") //true
palindrome("kayak") //true
palindrome("racecar") //true

console.log('>> Shoud return false:')
palindrome("rad") //false
palindrome("dodo") //false
palindrome("polo") //false
```