
1 Mudanças no Contrato

Por se basear nos mecanismos de transmissão de mensagem HTTP, uma API REST é composta de vários elementos que podem ser alterados causando impactos aos seus consumidores. Por via de regra, alterações de APIs que exigem que os consumidores alterem os seus sistemas para poder continuar utilizando a funcionalidade devem necessariamente ser feitas com um versionamento do tipo *major*. Mudanças que não necessitem desenvolvimento por parte dos consumidores poderão ser tratadas como *minor*.

No *request* de uma API, são pontos de alteração relevantes: verbo, *schema*, recurso, *pathParams*, *queryParams*, *headers* e o formato do *body*. No *response*, as alterações podem ocorrer no *status code*, *headers* e no formato do *body*. Sendo que dependendo da API nem todos os elementos estão presentes.

Existe ainda o cenário em que APIs definem *callbacks (webhooks)*. Neste caso alterações em *pathParams*, *queryParams*, *headers* e *body*, além do próprio uso do *webhook* podem impactar o consumidor.

Estrutura de um *request*

```
VERBO schema://host:porta/open-banking/api/versão/recurso/pathParam?queryParam=ipsum
Header: ipsum lorem

{JSON BODY}
```

Estrutura de um *response*

```
HTTP Status
Header: ipsum lorem

{JSON BODY}
```

Estrutura de um *callback*

```
VERBO schema://callbackUri/recurso/pathParam?queryParam=ipsum
Header: ipsum lorem

{JSON BODY}
```

Exemplo:

```
GET https://api.banco.com.br/open-banking/customers/v1/customers/12345678900/accounts?type=CONTA_CORRENTE
Authorization: Bearer token
```

```
200 Success
```

```
Content-Type: application/json
```

```
{
  "data": {
    "accounts": [
      "branch": "0001",
      "account": "123456"
    ]
  }
  "meta": {
    "totalRecords": 1,
    "totalPages": 1
  }
}
```

API que lista as contas do cliente de id "12345678900" filtrando apenas as contas do tipo "CONTA_CORRENTE". Neste exemplo o verbo utilizado foi o *GET*, o *schema* o HTTPS, o caminho a API é a "customers", a versão "v1", o recurso "customers", existe um *pathParam* que identifica o cliente, um *queryParams* que filtra os tipos de contas retornados, um *header* informando o *token* de autorização e a requisição não possui um *body*.

A resposta desta API contém um *status* de sucesso, um *header* identificando o tipo de conteúdo retornado e um *body* contendo a lista de contas do cliente.

1.1 Lista de Mudanças disruptivas:

-
- BC1. Remover um recurso;
 - BC2. Remover um verbo de um recurso (operação);
 - BC3. Alterar o verbo de um recurso (operação);
 - BC4. Remover um *path*;
 - BC5. Remover um parâmetro do *request* (*pathParam* e/ou *queryParams*);
 - BC6. Renomear um parâmetro (*queryParams* e/ou *body*)
 - BC7. Adicionar um parâmetro obrigatório no *request* (*pathParam*, *queryParams* e/ou *body*)
 - BC8. Alterar local onde um parâmetro é recebido
 - BC9. Alterar um parâmetro do tipo *enum* (adicionar ou remover valores)
 - BC10. Remover o suporte a um tipo de conteúdo (*content-type*) previamente aceito no *request*;
 - BC11. Remover o suporte a um tipo de conteúdo (*content-type*) previamente aceito no *response*;
 - BC12. Adicionar um *header* obrigatório no *request*;
 - BC13. Remover um *header* do *response*;
 - BC14. Alterar a estrutura do *body* (*request*, *response* e/ou *webhook*);
 - BC15. Alterar o tipo de um parâmetro (*pathParam*, *queryParams*, *header* e/ou *body*);
 - BC16. Alterar o formato de um parâmetro (*pathParam*, *queryParams*, *header* e/ou *body*);

- BC17. Aumentar as restrições de um parâmetro no *request* (*pathParam*, *queryParams*, *header* e/ou *body*);
- BC18. Diminuir as restrições de um parâmetro no *response* (*body*);
- BC19. Alterar o valor padrão de um parâmetro opcional;
- BC20. Alterar a forma de representar *arrays* nos parâmetros (*queryParams*)
- BC21. Adicionar um novo código HTTP de resposta;
- BC22. Remover um código HTTP de resposta;
- BC23. Modificar o código HTTP de uma resposta existente;
- BC24. Adicionar/Remover o uso de *callbacks* (*webhooks*) assíncrono

1.2 Lista Mudanças não disruptivas:

- NBC1. Adicionar um novo recurso na API;
- NBC2. Adicionar um novo verbo em um recurso existente (operação);
- NBC3. Adicionar um novo *path*;
- NBC4. Tornar um parâmetro obrigatório do *request* (*pathParam* e/ou *queryParams*) em opcional;
- NBC5. Adicionar novos parâmetros opcionais no *request* (*header* ou *body*)
- NBC6. Adicionar novos parâmetros no *response* (*header* ou *body*)

BC1. Remover um recurso

Remover um recurso é uma mudança disruptiva pois indisponibiliza um conjunto de funcionalidades utilizadas pelos consumidores atuais da API, exigindo que eles adequem o seu negócio e implementação para se adequar a ausência do recurso removido.

Exemplo: a API originalmente define três recursos e na nova versão exclui o recurso 2.

Versão 1

```
/api/v1/recurso1  
/api/v1/recurso2  
/api/v1/recurso3
```

Versão 2

```
/api/v2/recurso1  
/api/v2/recurso3
```

BC2. Remover um verbo de um recurso (operação)

Remover um verbo de um recurso (operação) é uma mudança disruptiva pois indisponibiliza uma funcionalidade utilizada pelos consumidores atuais da API, exigindo que eles adequem o seu negócio e implementação para se adequar a ausência da operação removida.

Exemplo: a API originalmente define um recurso com três operações e na nova versão uma das operações foi removida.

Versão 1

```
GET /api/v1/recurso1
GET /api/v1/recurso1/{id}
PUT /api/v1/recurso1/{id}
```

Versão 2

```
GET /api/v2/recurso1
GET /api/v2/recurso1/{id}
PUT /api/v2/recurso1/{id}
```

BC3. Alterar o verbo de um recurso (operação)

Alterar um verbo de um recurso (operação) é uma mudança disruptiva pois exige que os consumidores atuais da API alterem a sua implementação para continuar consumindo um recurso. Geralmente, uma mudança no verbo da operação é acompanhada de uma mudança na regra de negócio.

Exemplo: a API originalmente define um recurso com o verbo PUT e na nova versão ele é alterado para PATCH.

Versão 1

```
PUT /api/v1/recurso1/{id}
```

Versão 2

```
PATCH /api/v2/recurso1/{id}
```

BC4. Remover um *path*

Em uma API um path (recurso + subrecurso) pode ser utilizado como uma forma de facilitar o acesso de consulta e/ou edição a um subrecurso. Removê-lo implica na necessidade de alteração na implementação dos consumidores que precisarão modificar a forma de acesso a informação.

Exemplo: a API originalmente define um path para acessar apenas os dados do subrecurso2 com facilidade. A nova versão remove este path.

Versão 1

```
GET /api/v1/recurso1/{id}
GET /api/v1/recurso1/{id}/subrecurso1
GET /api/v1/recurso1/{id}/subrecurso2
```

Versão 2

```
GET /api/v2/recurso1/{id}
GET /api/v2/recurso1/{id}/subrecurso1
```

BC5. Remover um parâmetro do *request* (*PathParam* e/ou *QueryParam*)

Remover um parâmetro do *request* de uma API, seja ele um *PathParam* ou *QueryParam*, faz com a implementação dos consumidores precise ser alterada ou até revista dado que uma condição para o funcionamento do seu negócio pode ter sido removida.

Exemplo 1: a API originalmente define um *PathParam* que identifica o recurso sendo. A nova versão não possui mais este parâmetro, não sendo mais possível identificar o recurso.

Versão 1

```
GET /api/v1/recurso1/{id}/subrecurso1
```

Versão 2

```
GET /api/v2/recurso1/subrecurso1
```

Exemplo 2: a API originalmente define como *QueryParam* um filtro. A nova versão não possui mais este filtro.

Versão 1

```
GET /api/v1/recurso1?filter=lorem
```

Versão 2

```
GET /api/v2/recurso1
```

BC6. Renomear um parâmetro (*queryParam* e/ou *body*)

Renomear um parâmetro, seja ele um *queryParam* ou um parâmetro no *body*, exige que os consumidores alterem a sua implementação para continuar utilizando a API.

BC7. Adicionar um parâmetro obrigatório no *request* (*pathParam*, *queryParam* e/ou *body*)

Adicionar um parâmetro, seja ele um *pathParam*, *queryParam* ou um parâmetro no *body*, exige que os consumidores alterem a sua implementação para continuar utilizando a API.

BC8. Alterar local onde um parâmetro é recebido

Alterar o local onde um parâmetro é recebido exige que os consumidores alterem a sua implementação para continuar utilizando a API.

BC9. Alterar um parâmetro do tipo *enum* (adicionar ou remover valores)

Um *enum* define os valores válidos para um parâmetro e alterar a lista exige que os consumidores alterem a sua implementação.

BC10. Remover o suporte a um tipo de conteúdo (*content-type*) previamente aceito no request

BC11. Remover o suporte a um tipo de conteúdo (*content-type*) previamente aceito no response

BC12. Adicionar um *header* obrigatório no request;

BC13. Remover um *header* do response

BC14. Alterar a estrutura do *body* (*request*, *response* e/ou *webhook*);

BC15. Alterar o tipo de um parâmetro (*pathParam*, *queryParam*, *header* e/ou *body*);

BC16. Alterar o formato de um parâmetro (*pathParam*, *queryParam*, *header* e/ou *body*)

BC17. Aumentar as restrições de um parâmetro no request (*pathParam*, *queryParam*, *header* e/ou *body*)

BC18. Diminuir as restrições de um parâmetro no response (*body*);

BC19. Alterar o valor padrão de um parâmetro opcional;

BC20. Alterar a forma de representar *arrays* nos parâmetros (*queryParam*)

BC21. Adicionar um novo código HTTP de resposta;

BC22. Remover um código HTTP de resposta;

BC23. Modificar o código HTTP de uma resposta existente;

BC24. Adicionar/Remover o uso de *callbacks* (*webhooks*) assíncrono