



Teste prático para QA - API Panda Vídeo

Thomás Sousa Causin Alves

16/07/2025

Futuro analista QA Jr

## Sumário

- Introdução
- Ambiente e Ferramentas
- Endpoints testados
- Casos de Teste
- Explicando Console do código Robot
- Conclusões e Recomendações

# Introdução

Este relatório documenta os testes manuais realizados sobre o endpoint `/folders` da API da Panda Vídeo, com foco na validação de funcionalidades de criação, leitura, atualização e exclusão de pastas (CRUD), bem como o comportamento da API em cenários positivos e negativos. Também foi adicionada uma parte no final como um extra que explica a saída dos resultados no console do projeto criado através do framework “Robot Framework”.

## Descrição do Desafio

### 1. Ferramenta de Automação:

- Utilize o Robot Framework para realizar a automação dos testes.

## 2. Fluxo a ser Automatizado:

- Realizar login no site.
- Validar o fluxo de adição de itens no carrinho.
- Validar todo o fluxo de compra até a finalização.

## 3. Site para Teste:

- Utilize o site [Sauce Demo](#).

## 4. Tarefas a Realizar:

- Automação do Login: Desenvolver um script para realizar o login no site.
- Validação de fluxo de adição de itens no carrinho: Validar a adição de itens no carrinho.
- Validação do Fluxo de Compra: Validar todo o fluxo de compra

# Ambiente e Ferramentas

Plataforma: Panda Vídeo

URL base: <https://api-v2.pandavideo.com.br>

Ferramentas:

- Postman (testes de API) mas poderia ser Insomnia também

- VSCode (Para a parte 2)
- Criar conta na Plataforma da panda vídeo para conseguir acesso a chave de api que será utilizada no postman a seguir.

Frameworks:

- Robot Framework
- SeleniumLibrary

Autenticação: Header `Authorization: panda-<token>`

## Endpoints Testados

Tabela com os endpoints utilizados no postman:

Endpoint original: <https://api-v2.pandavideo.com.br/folders/> usaremos ele para todas as requisições do CRUD.

Método	Endpoint	Descrição
GET	/folders	Listar todas as pastas
POST	/folders	Criar nova pasta
PUT	/folders/{id}	Atualizar nome da pasta criada (que seria a pasta mãe no caso)
DELETE	/folders/{id}	Excluir a pasta pelo Id que conseguimos ao criar a pasta

## Casos de Teste

Aqui temos diversos casos de testes para mostrar, desde os bem sucedidos até os que apresentaram erros por algum motivo, neste tópico vamos discutir cada um desses casos, explicando os erros e também mostrando estratégias de correção para que cheguemos no melhor resultado possível.

### List-folders

Através do link da documentação da [Panda Video](#) temos os seguintes “responses” abaixo:

## ■ Responses

- **200** : Success. The list of folders was retrieved successfully.
- **400** : Bad request. Check the request parameters or syntax.
- **401** : Unauthorized. The request lacks valid authentication credentials.
- **404** : Not found. The requested resource (folders) was not found.
- **500** : Internal server error. Please try again later.

Meu objetivo principal aqui é descrever o que significa cada um no cenário de teste e assim, poder demonstrar como chegar em cada uma desses resultados (tanto os de “success” quanto os de “erros”)

### **200: Success (Sucesso)**

Significado: A requisição foi processada com êxito e o servidor retornou os dados solicitados.

Quando ocorre:

Operação bem-sucedida (ex: listagem de pastas, criação de recursos, consulta válida).

### **Exemplo geral:**

GET /api/folders → Retorna a lista de pastas do usuário.

Causas comuns:

- Parâmetros corretos
- Autenticação válida

- Recurso disponível

## **400: Bad Request (Requisição Inválida)**

Significado: O servidor não entendeu a requisição devido a erros do cliente (sintaxe/parâmetros inválidos).

Quando ocorre:

- Parâmetros ausentes ou mal formatados
- Tipos de dados incorretos
- Validação falhou (ex: campo numérico recebe texto)

### **Exemplo geral:**

GET /api/folders?limit=abc → limit deve ser um número.

Como solução, devemos verificar:

- Parâmetros da URL
- Corpo da requisição (JSON/XML)
- Cabeçalhos obrigatórios

## **401: Unauthorized (Não Autorizado)**

Significado: Falta de credenciais de autenticação válidas para acessar o recurso.

Quando ocorre:

- Token de acesso ausente/expirado
- Credenciais incorretas (usuário/senha)

- Sessão inválida

### **Exemplo geral:**

GET /api/folders sem token de autenticação.

Solução:

- Incluir cabeçalho Authorization: Bearer <token>
- Realizar login novamente

### **404: Not Found (Não Encontrado)**

Significado: O recurso solicitado não existe no servidor.

Quando ocorre:

- Endpoint com URL incorreta
- Recurso deletado ou nunca existiu
- Rota mal configurada

### **Exemplo geral:**

GET /api/folder (o correto é /api/folders no caso está faltando o caractere “s”).

Solução:

- Verificar a URL e ortografia
- Consultar documentação da API
- Validar se o recurso existe (ex: ID da pasta válido)

### **500: Internal Server Error (Erro Interno do Servidor)**



Significado: Falha inesperada no servidor durante o processamento da requisição. Como esse erro é gerado por problemas internos, não há uma maneira confiável de reproduzi-lo intencionalmente, mas aqui estão cenários que podem desencadear essa resposta.

Quando ocorre:

- Exceções não tratadas no código
- Erros de conexão com banco de dados
- Limites de recursos excedidos (ex: memória, timeout)
- Bugs em bibliotecas externas

### **Exemplo geral:**

Falha na conexão com o banco de dados durante GET /api/folders.

Solução:

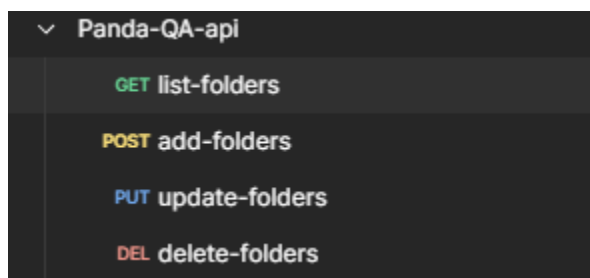
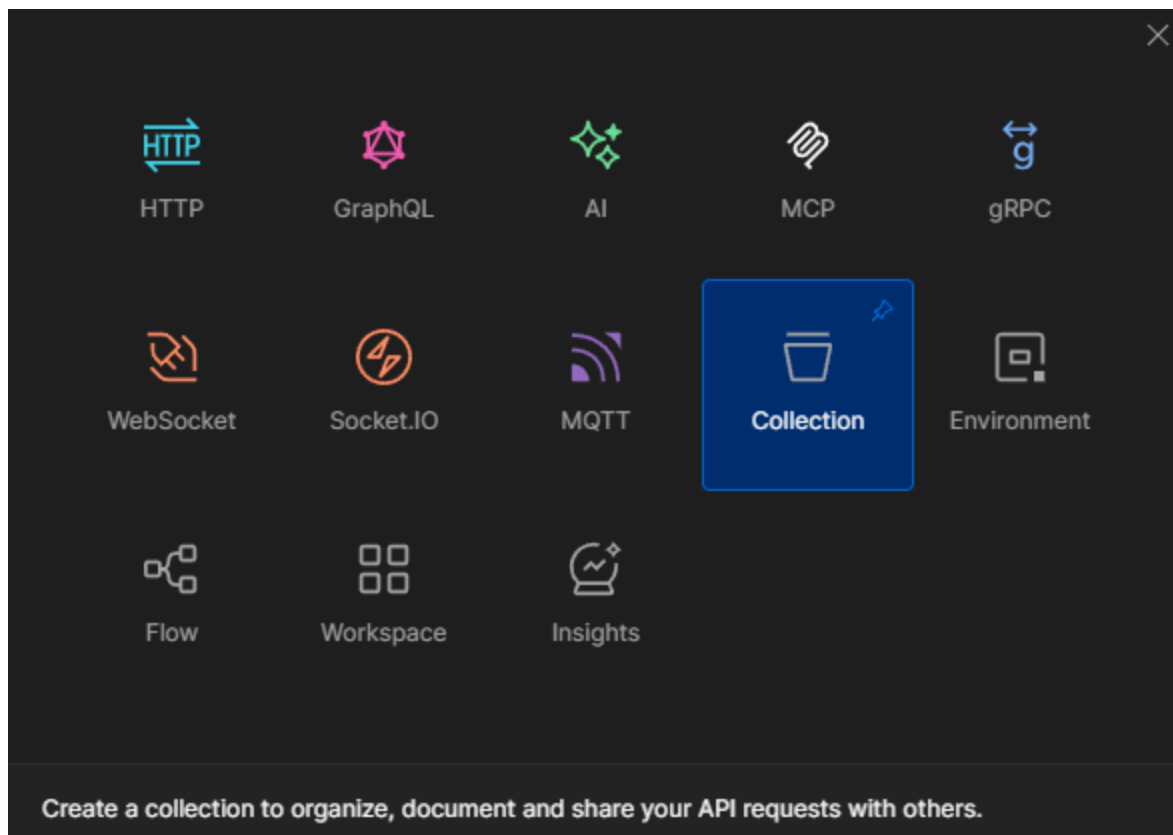
- Por parte do cliente: Tentar novamente mais tarde
- Por parte do Administrador: Verificar logs do servidor, monitorar sistemas

## Resumo Visual:

Código	Categoria	Origem do Erro	Ação Recomendada
200	Sucesso	—	Resposta processada normalmente
400	Erro do Cliente	Parâmetros inválidos	Corrigir Requisição
401	Erro do Cliente	Autenticação falha	Incluir Credenciais válidas
404	Erro do Cliente	Recurso não existe	Verificar a URL por exemplo
500	Erro do Servidor	Falha interna	Tentar novamente ou dar report ao Admin

## Cenário de testes da API da Panda:

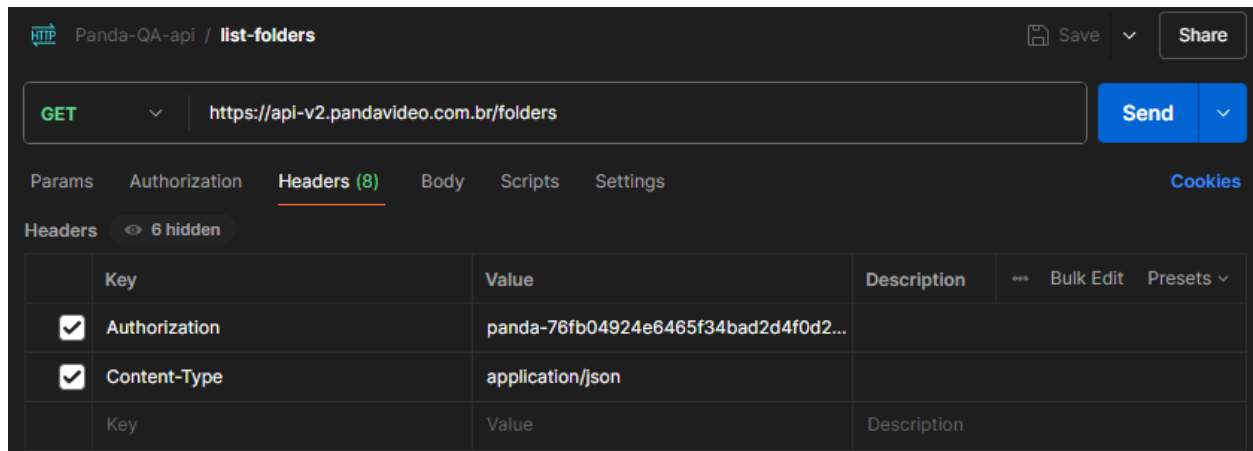
No primeiro momento foi criada uma Collection para reunir todos os endpoints que serão testados a seguir:



Entendo que fazer dessa forma é uma boa prática pois fica organizado e você também pode exportar também depois para outros usuários. Deixo claro também que sei que não é legal mostrar a chave da api, eu poderia ter declarado o valor dela em uma variável e feito o chamado dela depois, porém nesse caso fiz dessa maneira para mostrar realmente que coloquei a chave certa de API, mas tenho conhecimento dos riscos que podem existir em projetos.

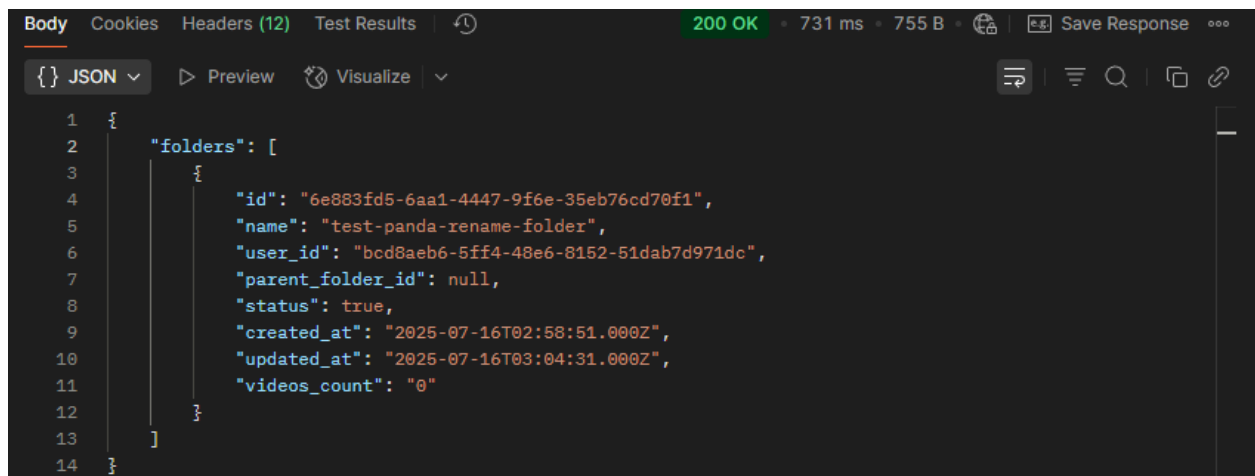
## GET list-folders

### 200: Success (Sucesso)



Aqui definimos as keys que serão usadas nessa requisição. Mas para que elas servem?

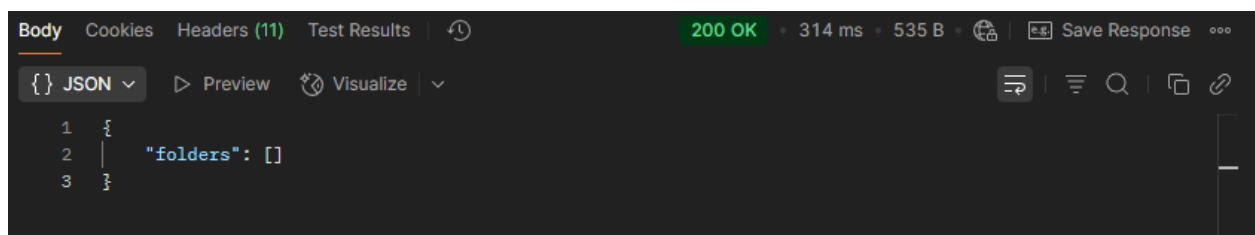
- Authorization: autenticar o cliente (usuário ou aplicação) no servidor, provando que ele tem permissão para acessar um recurso.
- Ela leva as credenciais na seguinte formação <tipo> <credenciais>.
- Ela é usada em toda requisição a recursos protegidos. Se ausente ou inválido → Erro 401 Unauthorized.



```
1 {
2   "folders": [
3     {
4       "id": "6e883fd5-6aa1-4447-9f6e-35eb76cd70f1",
5       "name": "test-panda-rename-folder",
6       "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
7       "parent_folder_id": null,
8       "status": true,
9       "created_at": "2025-07-16T02:58:51.000Z",
10      "updated_at": "2025-07-16T03:04:31.000Z",
11      "videos_count": "0"
12    }
13  ]
14 }
```

Aqui foi retornado o id da minha pasta, o nome que eu coloquei para ela, meu user\_id, parent\_folder\_id (nesse caso é null pq não tem nenhuma pasta mãe), o status da verificação (status) , a hora que foi criada e quando foi alterada (created\_at e updated\_at) e videos\_count, que seriam os números de vídeos na pasta.

ATENÇÃO! Aqui está explicado detalhadamente cada campo porque eu precisei criar uma pasta, caso eu não tivesse criado, seria um retorno vazio como no exemplo abaixo.



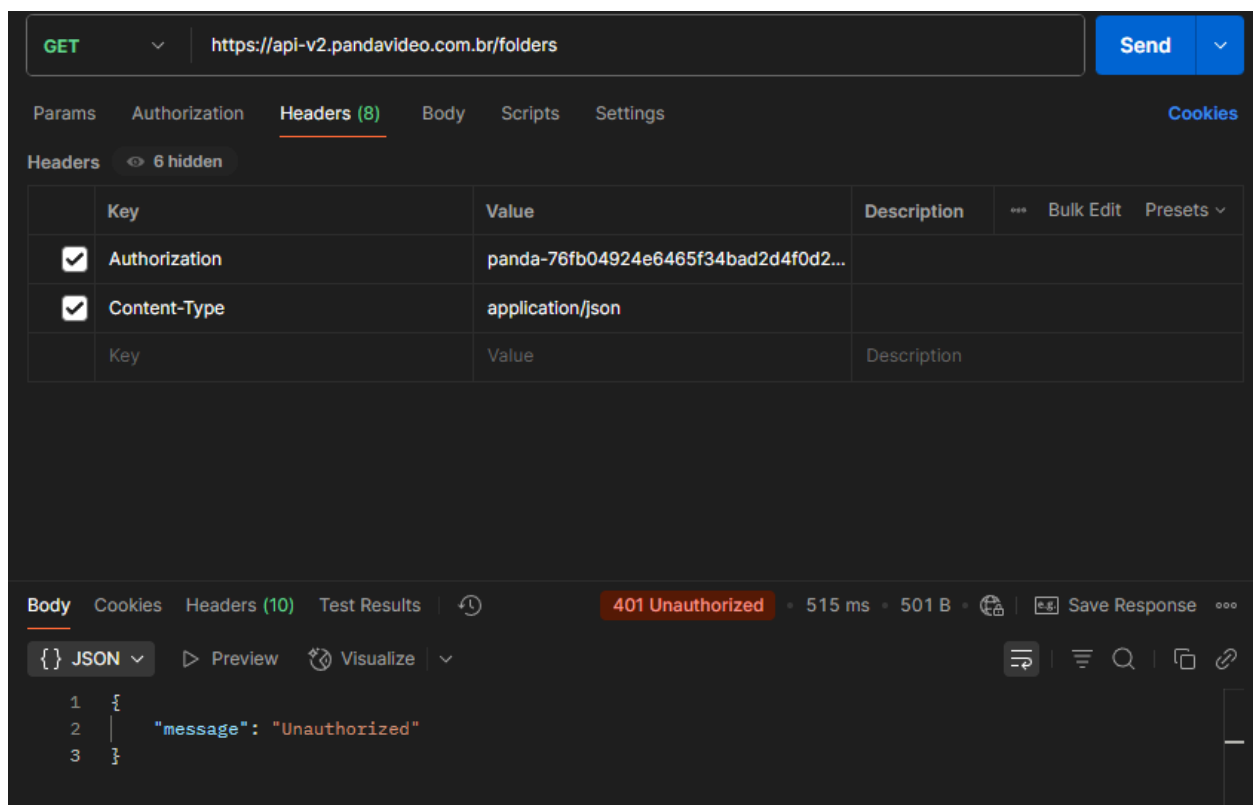
```
1 {
2   "folders": []
3 }
```

Aqui eu deletei a pasta que havia sido criada para poder mostrar que a listagem caso eu não crie nenhuma é vazia (seria esse o retorno caso eu só fizesse um get sem criar nenhuma pasta).

## Forçar erros:

Nessa parte vou tentar forçar erros na requisição de GET, para poder explicar o porque eles acontecem e como podemos resolver eles. Apesar de já ter explicado na introdução do capítulo, reforço que é necessário explicar no caso aplicado, já que usamos um caso geral o início.

## 401: Unauthorized (Não Autorizado)



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `https://api-v2.pandavideo.com.br/folders`
- Headers (8):**

Key	Value	Description
Authorization	panda-76fb04924e6465f34bad2d4f0d2...	
Content-Type	application/json	
- Body:** JSON format showing the response:

```
1 {  
2   "message": "Unauthorized"  
3 }
```
- Status:** 401 Unauthorized (515 ms, 501 B)

Aqui eu tentei passar uma chave de API inválida e ele deu acesso não autorizado justamente por isso.

### **403: Forbidden**

Esse error ele não falava na página da documentação, porém ele acontece quando por exemplo colocamos a URL errada, nesse caso pode-se perceber que eu tirei o “s” de folders e assim apresentou tal error.

HTTP Panda-QA-api / list-folders Save Share

**GET** ▼  Send ▼

Params Authorization **Headers (8)** Body Scripts Settings Cookies

Headers 6 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	panda-76fb04924e6465f34bad2d4f0d2...				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

**Body** Cookies Headers (11) Test Results 🔄 403 Forbidden 510 ms 725 B 🌐 📄 Save Response ...

**{ } JSON** ▼ ▶ Preview 🔗 Visualize ▼ 🔄 ☰ 🔍 📄 🔗

```
1 {
2   "message": "Authorization header requires 'Credential' parameter. Authorization header requires
              'Signature' parameter. Authorization header requires 'SignedHeaders' parameter. Authorization
              header requires existence of either a 'X-Amz-Date' or a 'Date' header. (Hashed with SHA-256
              and encoded with Base64) Authorization=8G+0Z4X7d6TzSaw1MmvrSfyKqETdTYn0aiWUyNSM9Sg="
3 }
```

## 400: Bad Request (Requisição Inválida)



Aqui a lógica é bem simples, eu tenho a URL da minha API e estou tentando acessar um endpoint que não existe.

The screenshot shows a REST client interface with the following details:

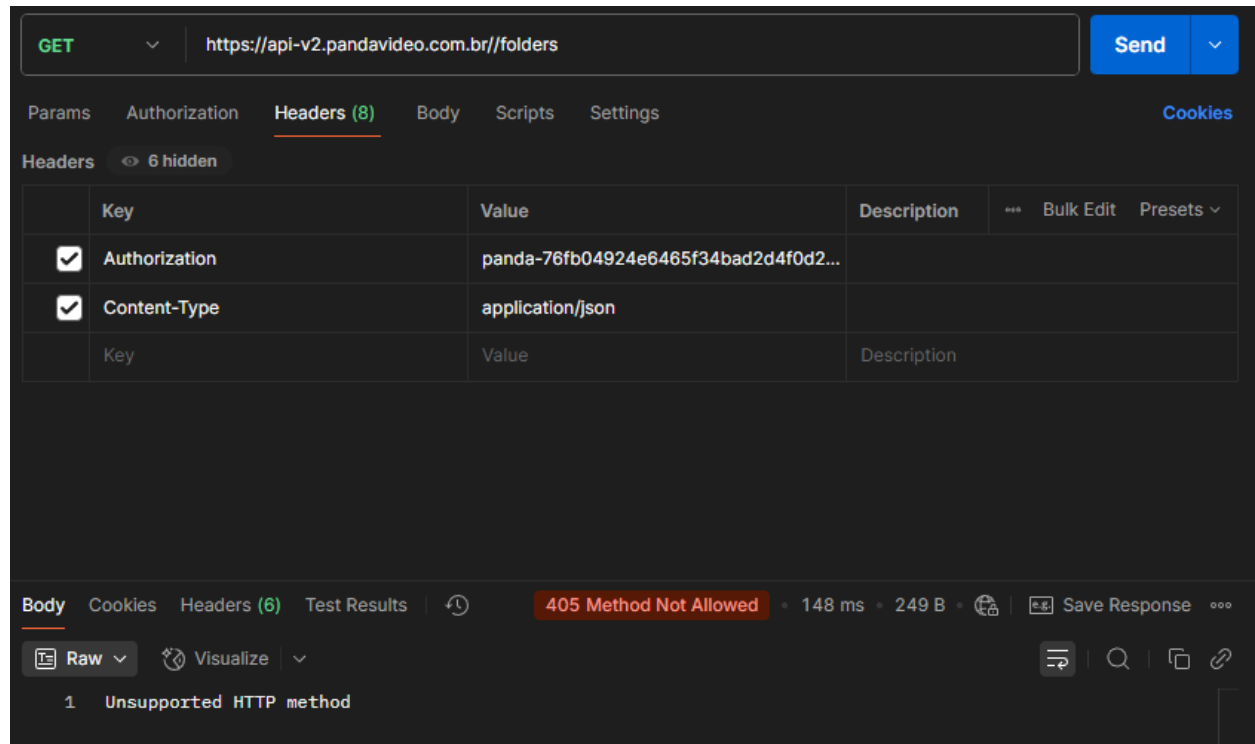
- Method:** GET
- URL:** `https://api-v2.pandavideo.com.br/folders?limit=abc`
- Headers (8):**

Key	Value	Description
Authorization	panda-76fb04924e6465f34bad2d4f0d2...	
Content-Type	application/json	
- Response:** 400 Bad Request (518 ms, 608 B)
- Body (JSON):**

```
{  "errCode": "BadRequest",  "errMsg": "must NOT have additional properties: limit"}
```

**405: Method not Allowed**

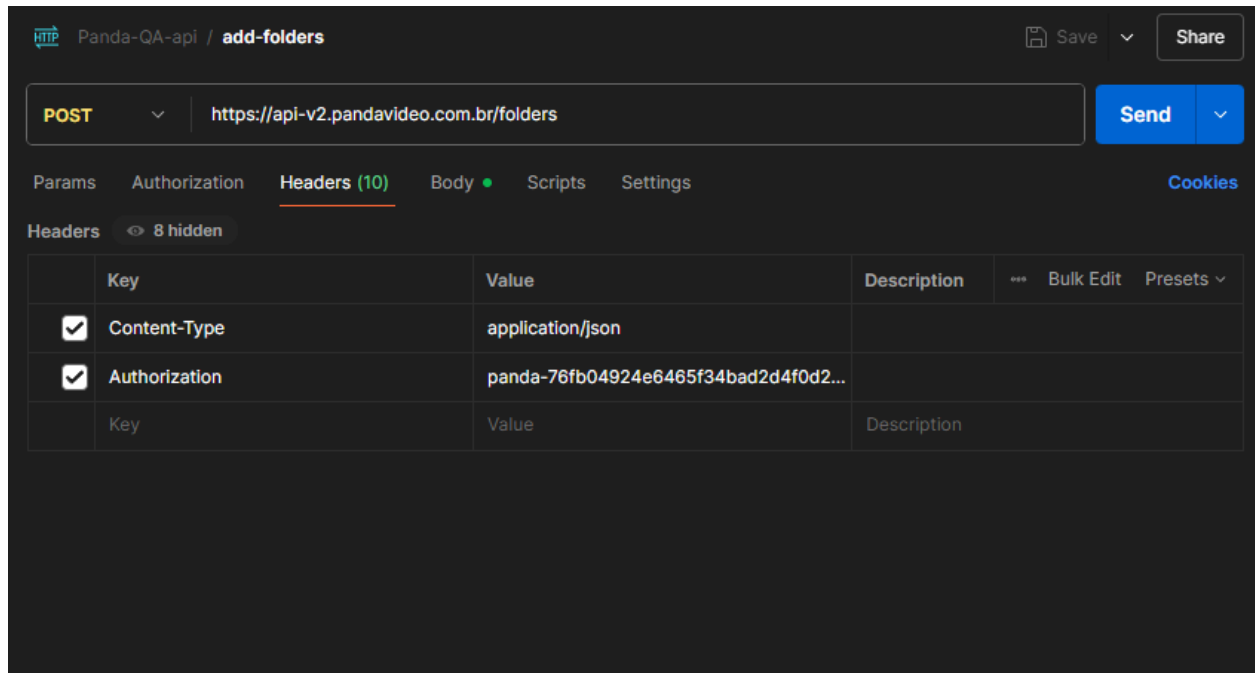
Aqui foi colocado uma “/” a mais na URL e isso alterou o método, sendo assim foi apontado como não suportado.



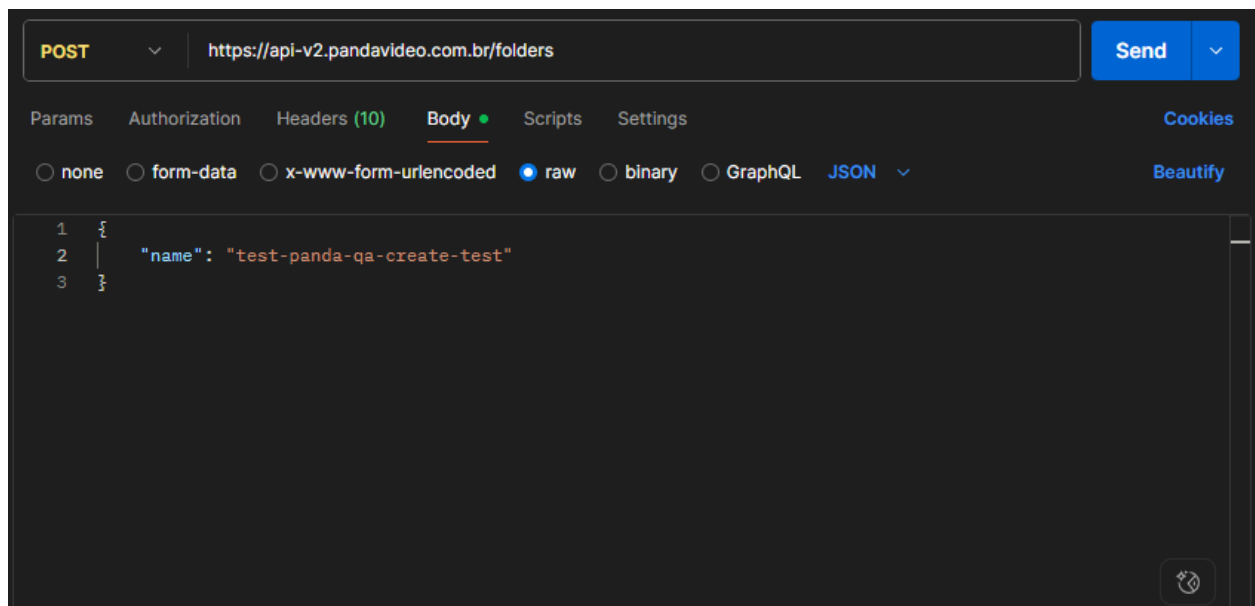
**Os demais erros serão tratados nas próximas requisições!**

## POST add-folders

Aqui segue a mesma lógica que a anterior, precisando de um Authorization e um Content-Type (no caso na anterior não precisaria desse último, porém eu sempre aplico em todas que faço, porque acredito que se não causa nenhum problema, é melhor seguir um padrão sempre).

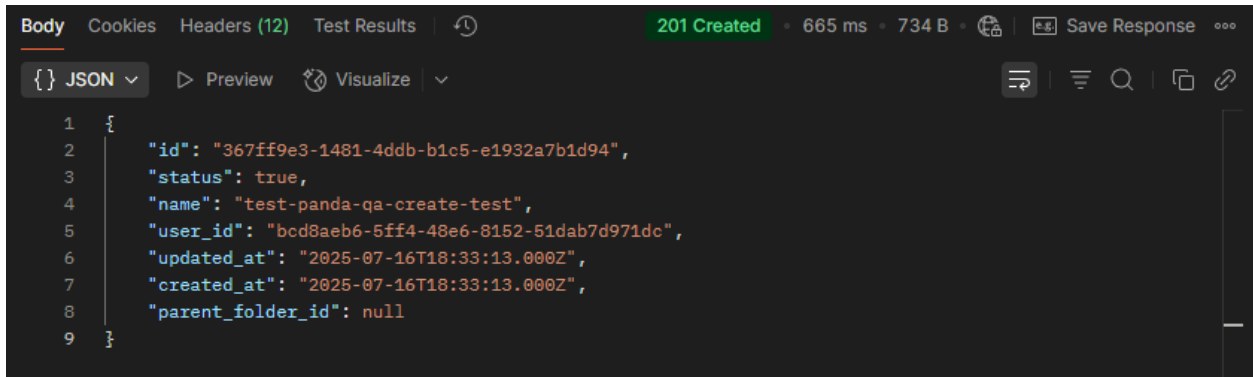


O método POST eu estou enviando “alguma coisa” para o servidor, no caso, estou enviando um “name” que será o nome da pasta que estamos criando.



## 201: Created

O próprio nome já diz, “criado”. Significa que a requisição foi concluída e a pasta foi criada com o nome desejado. Tem todos aqueles campos que já foram explicados no primeiro GET que vimos.



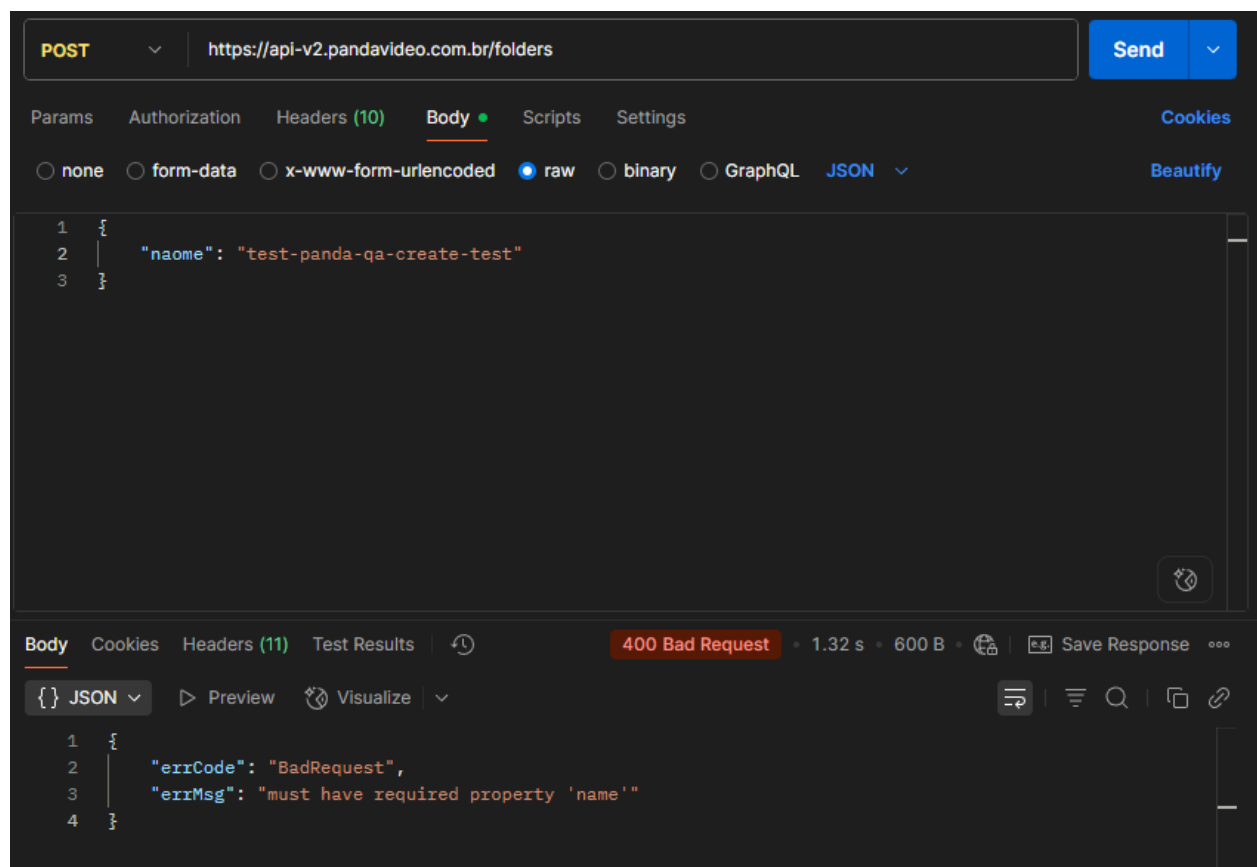
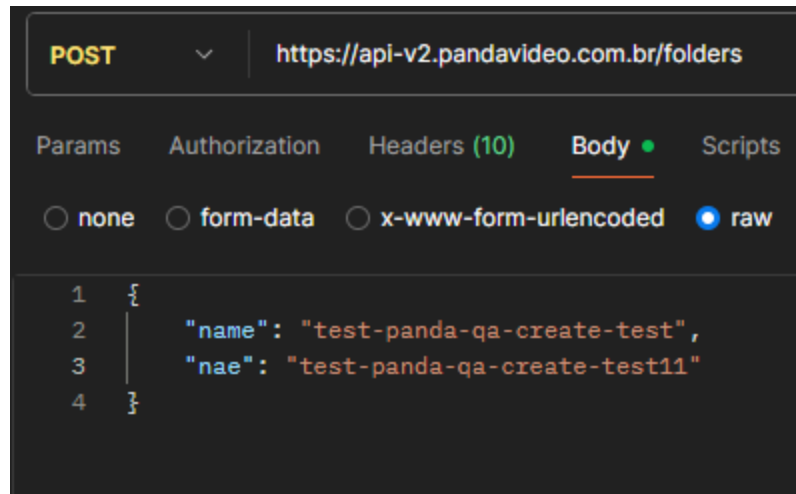
The screenshot shows a REST client interface with the following details:

- Status:** 201 Created
- Time:** 665 ms
- Size:** 734 B
- Body:** JSON
- JSON Content:**

```
1 {
2   "id": "367ff9e3-1481-4ddb-b1c5-e1932a7b1d94",
3   "status": true,
4   "name": "test-panda-qa-create-test",
5   "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
6   "updated_at": "2025-07-16T18:33:13.000Z",
7   "created_at": "2025-07-16T18:33:13.000Z",
8   "parent_folder_id": null
9 }
```

## 400: Bad Request

Nesse caso aqui, eu mudei o campo de “name” para “naome” como podemos ver no print abaixo, isso indica que o nome da propriedade está errado, sendo assim, um badrequest. Porém observe que ele já explica o que você pode estar errando. Pode ocorrer também se você colocar um campo inexistente também, porém por algum motivo o body aceita e eu mandar 2 names e cria pastas duplicadas, deixo aqui minha sugestão para que seja revisado isso no futuro.



## 403: Forbidden

Esse caso é um caso em que alteramos a url para uma que não existe, sendo assim aponta a mesma questão apontada no get, que não existe esse endpoint que estamos tentando acessar, além de indicar possíveis locais que devemos checar para ver se existem falhas ou não.

The screenshot shows a REST client interface with the following details:

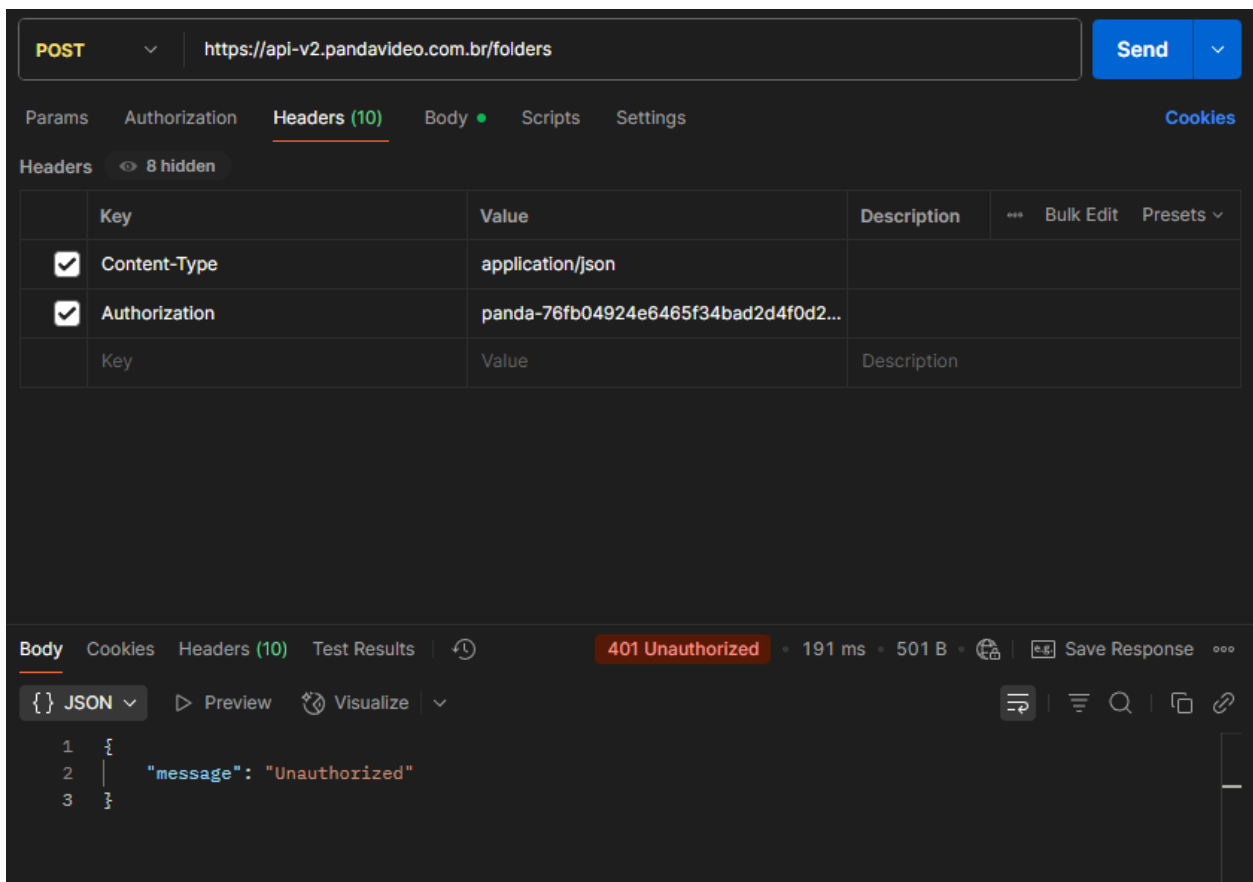
- URL:** `https://api-v2.pandavideo.com.br/folder`
- Method:** `POST`
- Body:**

```
{
  "name": "test-panda-qa-create-test"
}
```
- Response:** `403 Forbidden` (476 ms, 725 B)
- Response Body:**

```
{
  "message": "Authorization header requires 'Credential' parameter. Authorization header requires 'Signature' parameter. Authorization header requires 'SignedHeaders' parameter. Authorization header requires existence of either a 'X-Amz-Date' or a 'Date' header. (Hashed with SHA-256 and encoded with Base64) Authorization=8G+0Z4X7d6TzSaW1Mmvr5fyKqeTdTYn0aiWUyNSM9Sg="
}
```

## 401: Unauthorized

Como já havíamos dito anteriormente, aqui é quando tem algum token errado por exemplo.



The screenshot shows a REST client interface with a POST request to `https://api-v2.pandavideo.com.br/folders`. The request headers are visible, including `Content-Type: application/json` and `Authorization: panda-76fb04924e6465f34bad2d4f0d2...`. The response status is `401 Unauthorized`, with a response time of 191 ms and a body size of 501 B. The response body is shown in JSON format:

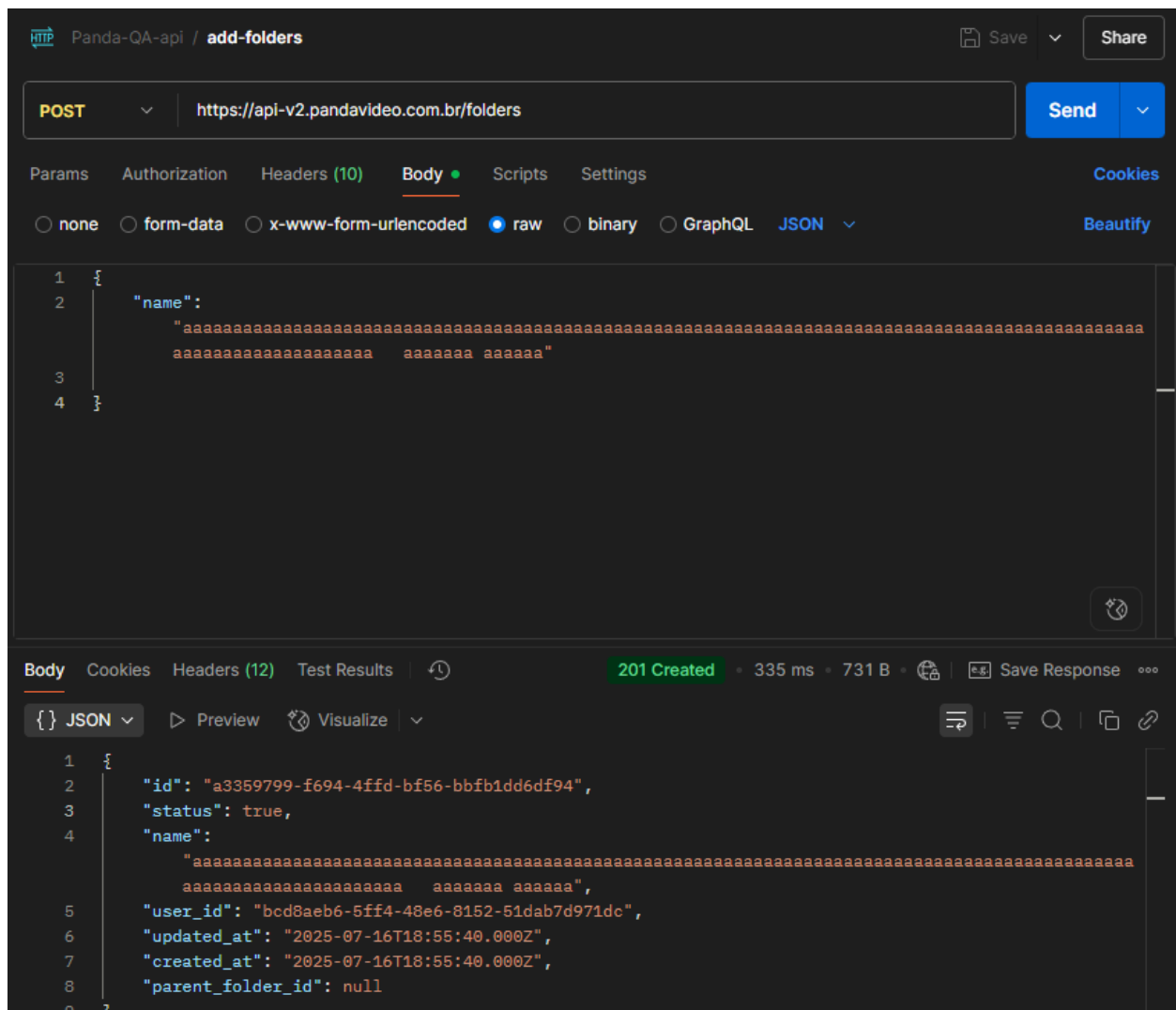
```
1 {
2   "message": "Unauthorized"
3 }
```

## EXTRA:

Tentei forçar um error de payload (413), e também ver se conseguia falha em caracteres vazios, mas não consegui porque não sei quais são os limites do campo “name”, mas reconheço a existência dele

também caso seja enviado nos campos muitos caracteres. Esse erro de payload é um problema grande porque quando você envia um payload muito grande:

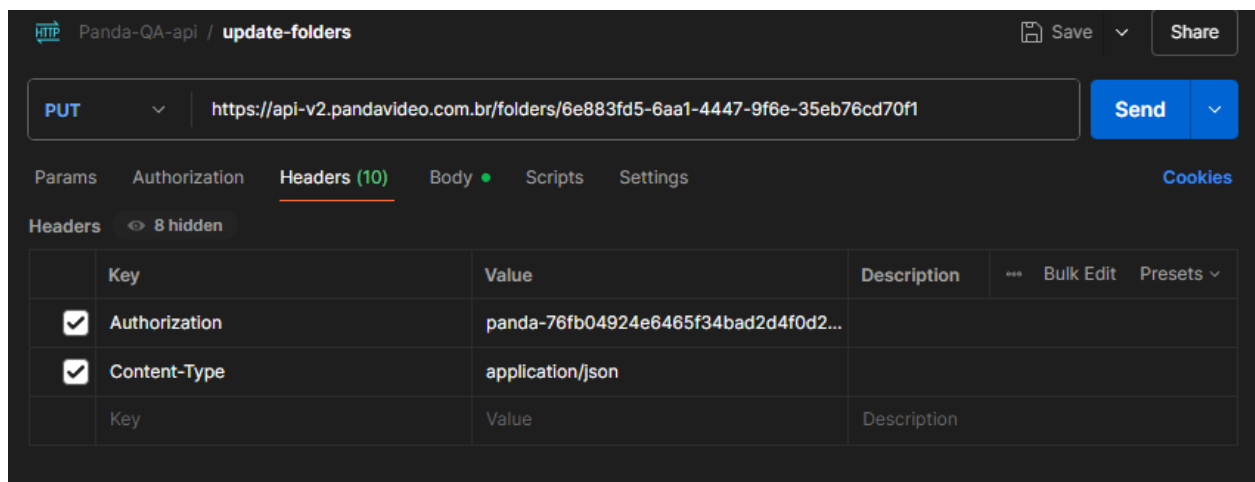
- O Postman pode demorar para processar os dados antes de enviá-los (consumindo memória e CPU local).
- A rede pode levar muito tempo para transmitir todos os bytes.
- O servidor pode demorar para processar a requisição.



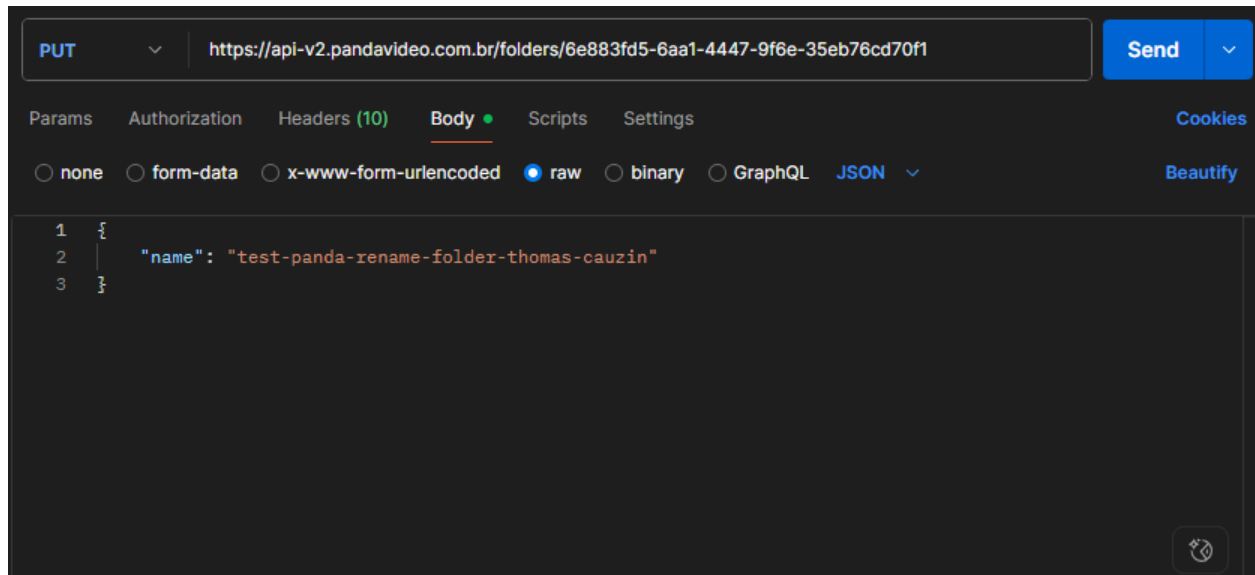


## PUT update-folders

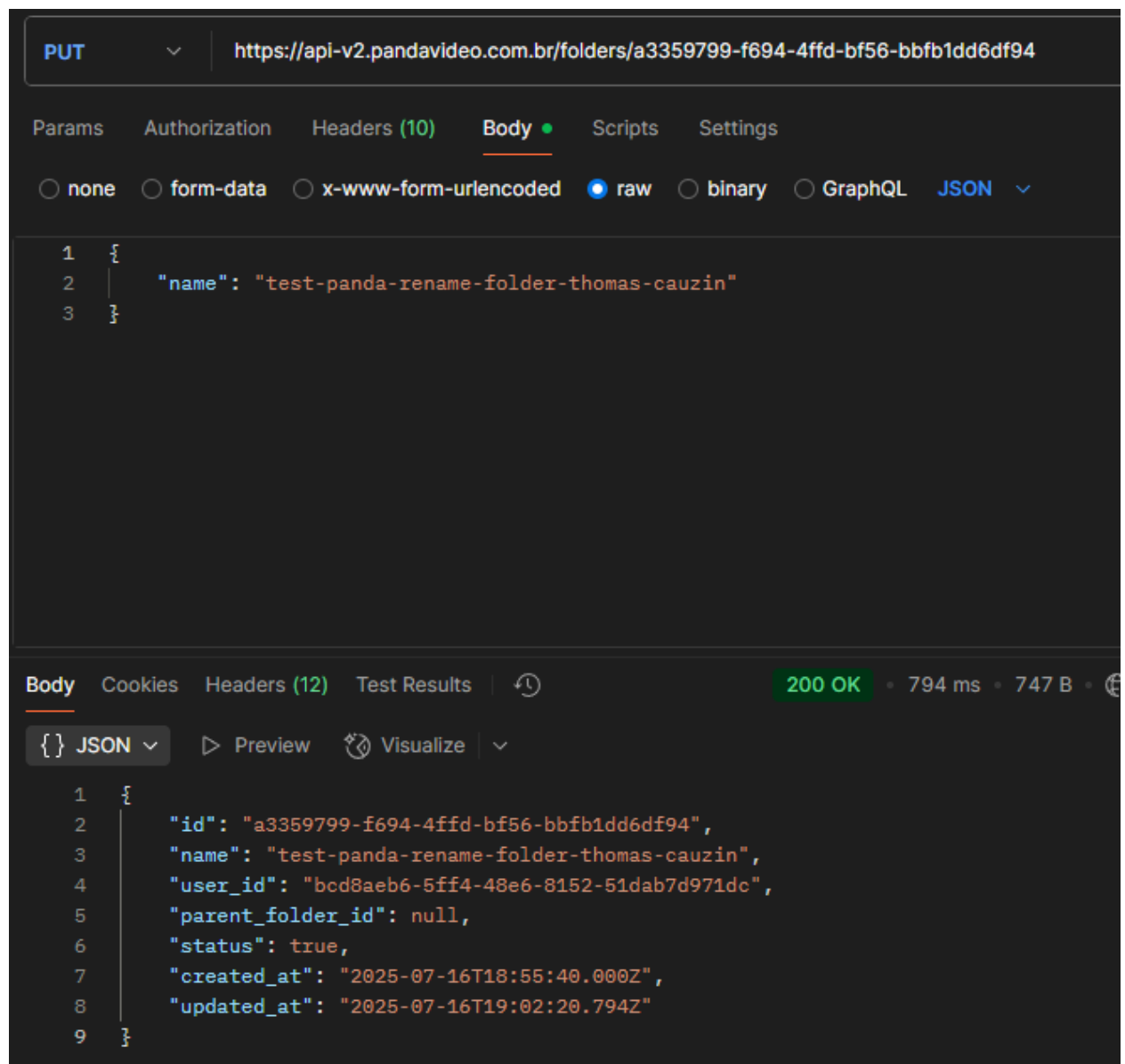
Aqui vamos tentar atualizar o campo name da página, já que pelo que entendi da regra de negócio, os outros campos são campos que são criados pela própria api, então não temos recursos para modificar os mesmos. Porém, isso vai ser testado nessa etapa também!



Mantemos o mesmo header acima



Observe nesse caso, que estamos colocando um Id que foi gerado na etapa de POST, para que possamos alterar precisamente a pasta em questão. Logo após no body, colocamos o campo “name” novamente, com o novo nome que vai ser modificado.



Peguei o Id que foi gerado quando tentei forçar um error de payload e mudei o nome dele para o campo em questão.

## 400 - Bad Request

Perceba que aqui foi alterado o id para um que não existe, com isso, não conseguimos acessar essa pasta para mudar o nome.

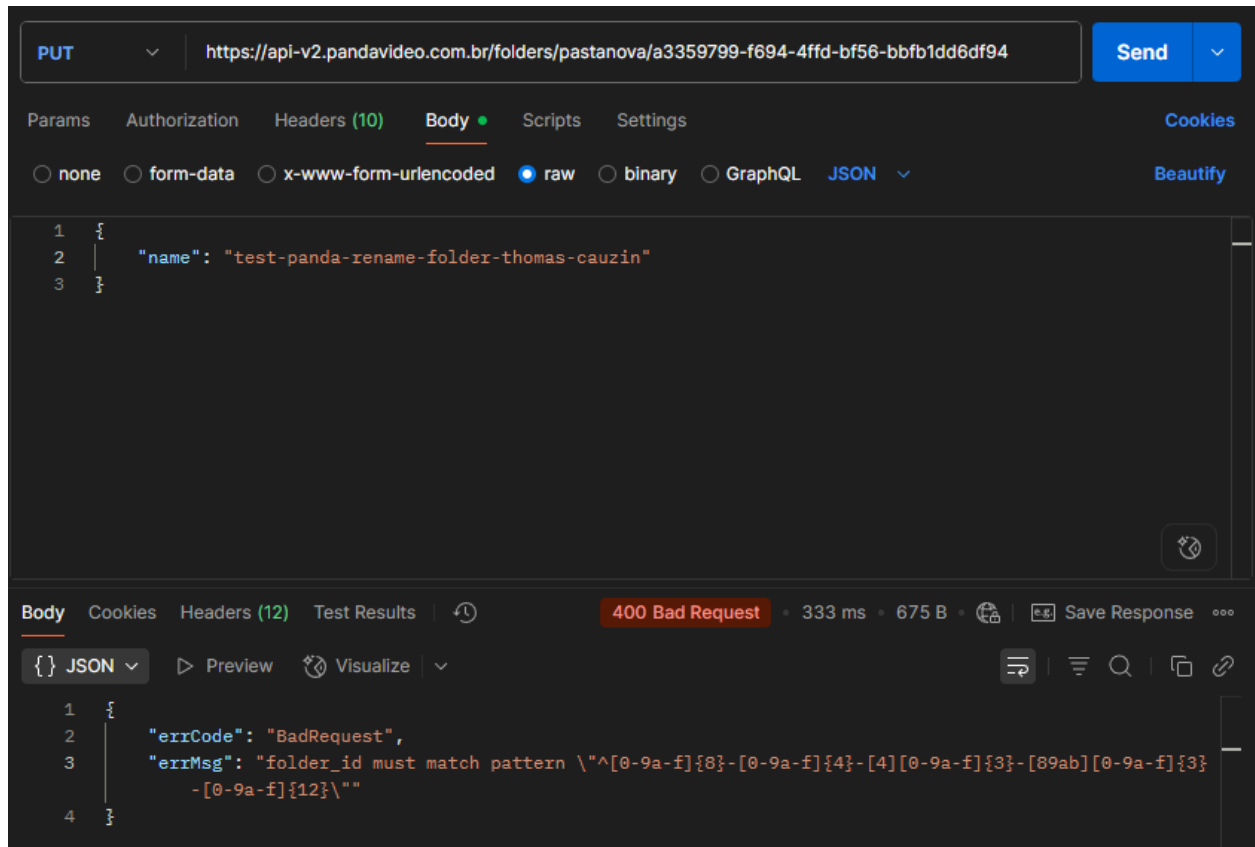
The screenshot shows a REST client interface with a PUT request to `https://api-v2.pandavideo.com.br/folders/a3359799-f694-4ffd-bf56-b`. The request body is a JSON object: `{ "name": "test-panda-rename-folder-thomas-cauzin" }`. The response is a 400 Bad Request with the following JSON body: `{ "errCode": "BadRequest", "errMsg": "folder_id must match pattern \"^[0-9a-f]{8}-[0-9a-f]{4}-[4][0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}\"" }`.

```
PUT https://api-v2.pandavideo.com.br/folders/a3359799-f694-4ffd-bf56-b

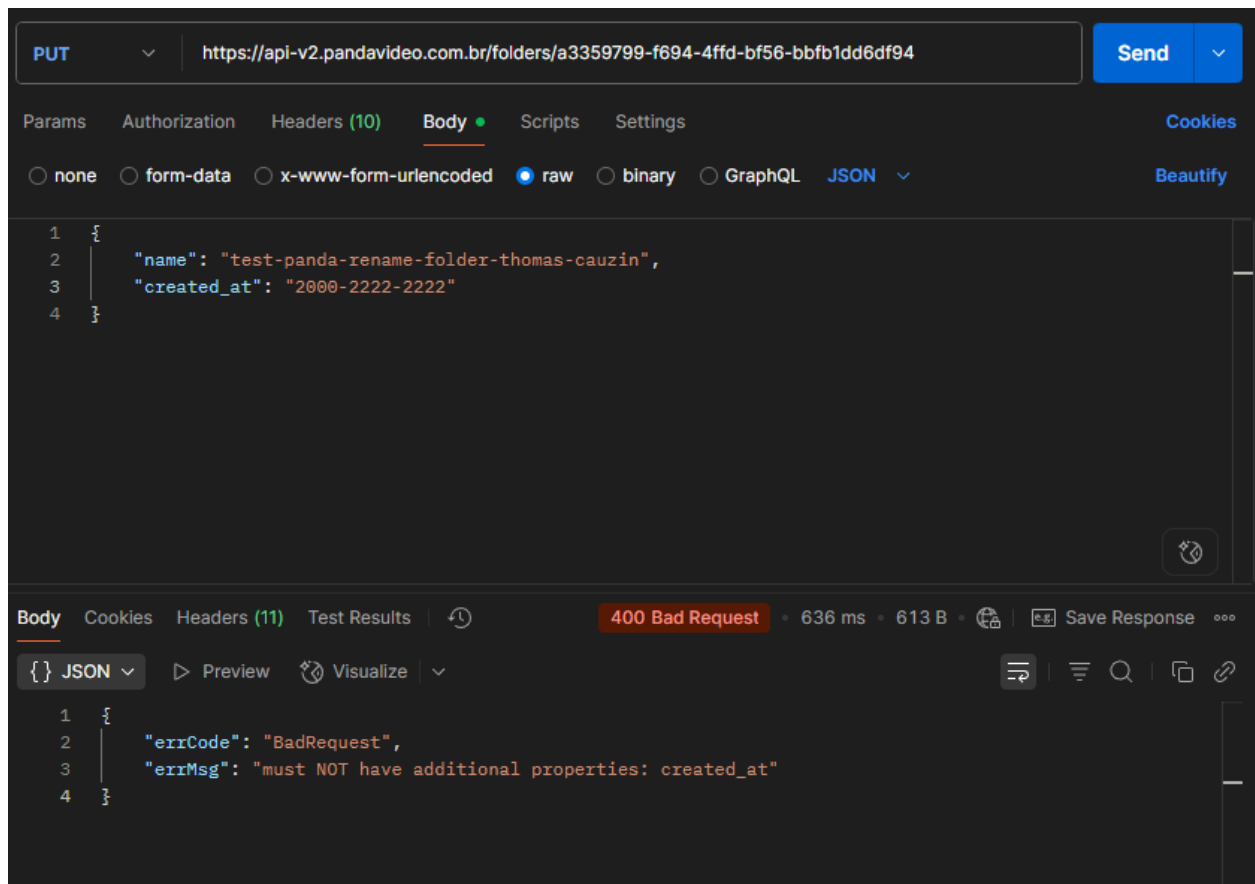
{
  "name": "test-panda-rename-folder-thomas-cauzin"
}
```

400 Bad Request • 847 ms • 675 B

```
{
  "errCode": "BadRequest",
  "errMsg": "folder_id must match pattern \"^[0-9a-f]{8}-[0-9a-f]{4}-[4][0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}\""
}
```



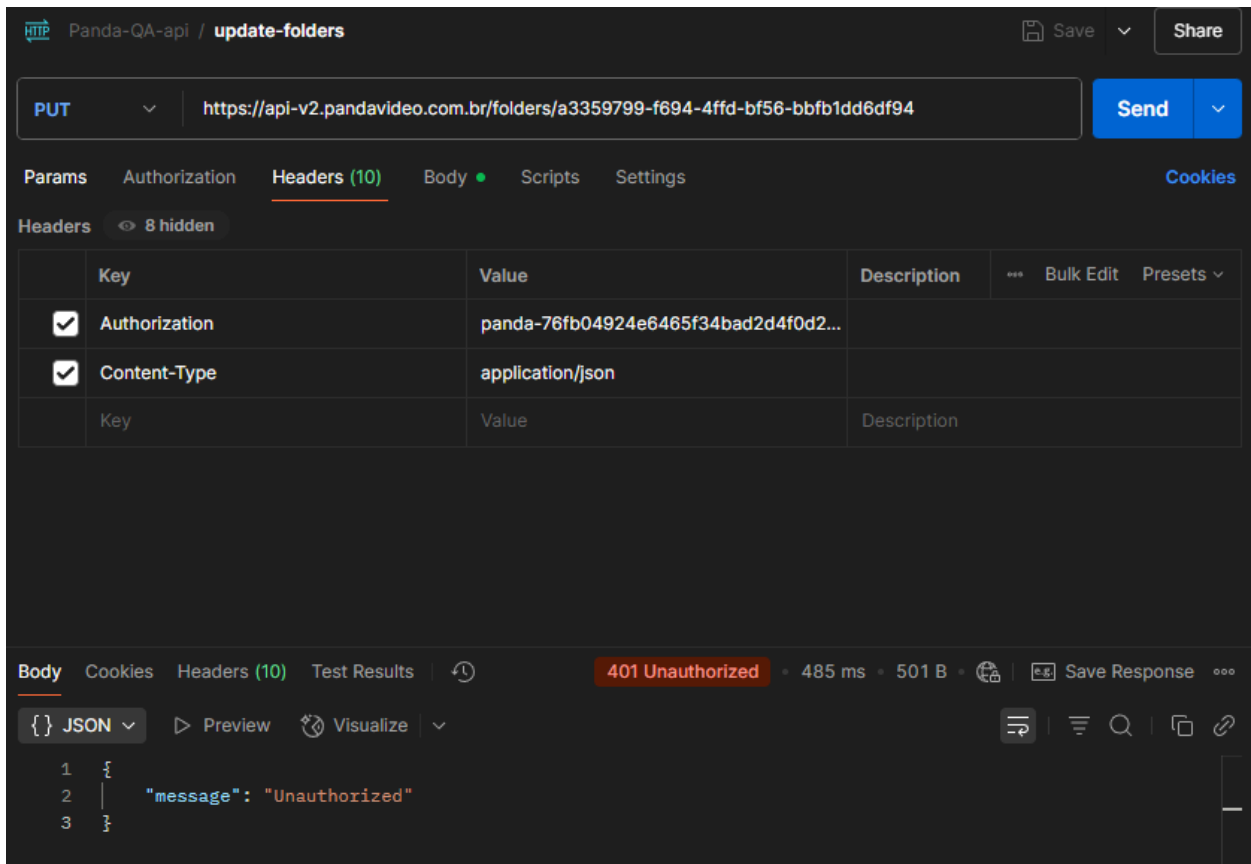
Na imagem acima eu tentei também alterar o endpoint para um endereço inexistente.



Como um extra, aqui eu coloquei tentando mudar campos que são gerados automaticamente pela API, como “created\_at”, validando nosso raciocínio de que não é possível modificar eles.

## 401 - Unauthorized

Esse error ele ocorre porque eu tentei modificar a key e sendo assim, não é autorizado a utilizar a api já que não temos uma chave valida, tornando inválido todo o processo após.



The screenshot shows a REST client interface for a PUT request to `https://api-v2.pandavideo.com.br/folders/a3359799-f694-4ffd-bf56-bbfb1dd6df94`. The request headers are set to `Authorization: panda-76fb04924e6465f34bad2d4f0d2...` and `Content-Type: application/json`. The response status is **401 Unauthorized** with a response time of 485 ms and a body size of 501 B. The response body is a JSON object: `{ "message": "Unauthorized" }`.

Key	Value	Description
Authorization	panda-76fb04924e6465f34bad2d4f0d2...	
Content-Type	application/json	

```
1 {
2   "message": "Unauthorized"
3 }
```

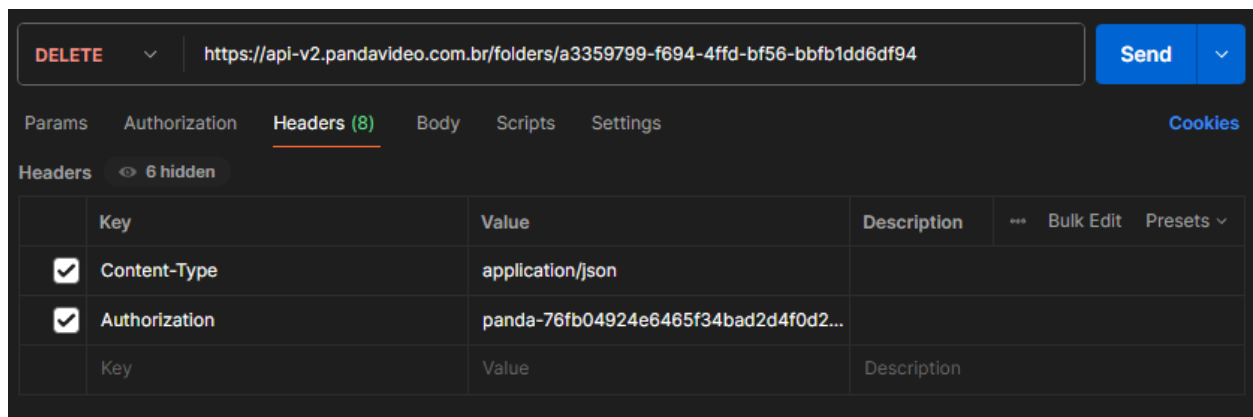
### Exemplo extra:

Abaixo eu fui no meu DELETE e deletei a pasta para ver o que retornava se eu tentasse alterar uma pasta que não existe mais, ou seja, mantinha tudo, porém ela teoricamente já não existia. A requisição apresentou success, porém alterou o status para false, sendo assim, podemos interpretar como o status realmente sendo se aquela pasta existe ou não. Ele até faz a mudança, porém mudar em

uma pasta que não existe? Me parece um retrabalho que a longo prazo pode ser um pouco custoso dependendo da aplicação, deixo aqui minha sugestão de melhoria nesse campo.

## DEL delete-folders

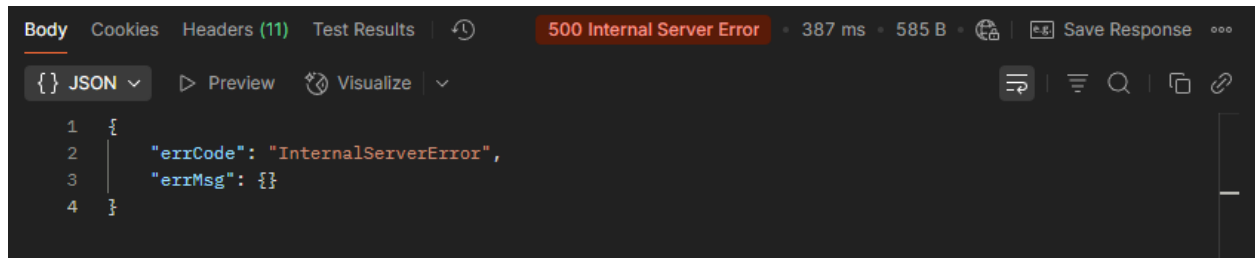
Aqui como o nome já diz trata-se da parte em que iremos deletar pastas que não fazem mais sentido manter.



Mantenho o mesmo padrão para o header do projeto, seguindo o mesmo raciocínio das anteriores. E assim como no put, precisamos colocar o id na url para deletar corretamente a pasta. Porém assim como o value de Authorization, podemos também setar uma variável no escopo da url e somente alterar o valor dessa variável futuramente, protegendo assim o id da pasta na url.

## 500 - Internal Server error





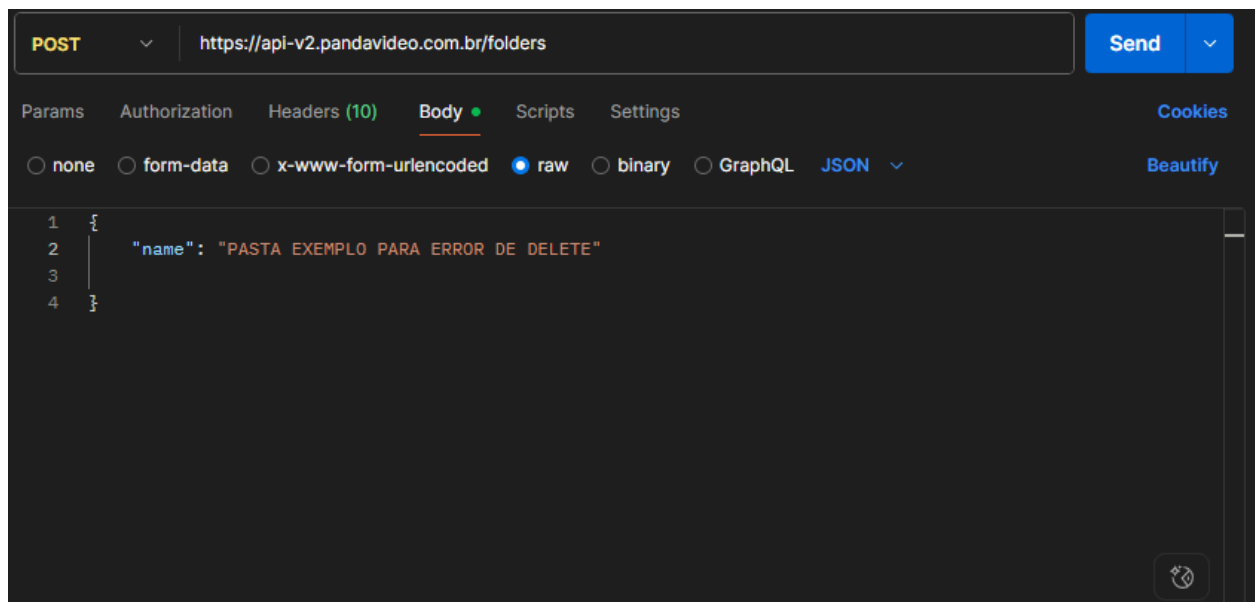
A screenshot of a REST client interface. The top bar shows the response status as "500 Internal Server Error" with a duration of 387 ms and a size of 585 B. The "Body" tab is selected, displaying a JSON object with the following structure:

```
1 {
2   "errCode": "InternalServerError",
3   "errMsg": {}
4 }
```

Aqui eu percebi que a pasta ela é deletada, porém apresenta um error de servidor, então é interessante contatar ao administrador para entender um pouco mais o que pode estar causando isso.

Mas como podemos checar se realmente está sendo deletado? Simples, vamos fazer um passo a passo (com prints) desde a criação até o delete final e depois vamos listar todas as pastas que temos.

## CREATE



A screenshot of a REST client interface. The top bar shows the request method as "POST" and the URL as "https://api-v2.pandavideo.com.br/folders". The "Body" tab is selected, displaying a JSON object with the following structure:

```
1 {
2   "name": "PASTA EXEMPLO PARA ERROR DE DELETE"
3 }
4 }
```

```
Body Cookies Headers (12) Test Results 201 Created 911 ms 757 B Save Response
{} JSON Preview Visualize
1 {
2   "id": "cf718666-72c1-449b-bc2a-78430d95f11a",
3   "status": true,
4   "name": "PASTA EXEMPLO PARA ERROR DE DELETE",
5   "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
6   "updated_at": "2025-07-16T19:23:52.000Z",
7   "created_at": "2025-07-16T19:23:52.000Z",
8   "parent_folder_id": null
9 }
```

## UPDATE

```
PUT https://api-v2.pandavideo.com.br/folders/cf718666-72c1-449b-bc2a-78430d95f11a Send
Params Authorization Headers (10) Body Scripts Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
1 {
2   "name": "MUDANDO O NOME DA PASTA DE EXEMPLO DELETE"
3 }
```

```
Body Cookies Headers (12) Test Results 200 OK • 333 ms • 763 B Save Response
{} JSON Preview Visualize
1 {
2   "id": "cf718666-72c1-449b-bc2a-78430d95f11a",
3   "name": "MUDANDO O NOME DA PASTA DE EXEMPLO DELETE",
4   "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
5   "parent_folder_id": null,
6   "status": true,
7   "created_at": "2025-07-16T19:23:52.000Z",
8   "updated_at": "2025-07-16T19:24:43.331Z"
9 }
```

LIST

HTTP

Panda-QA-api / list-folders

Save

Share

GET

https://api-v2.pandavideo.com.br/folders/

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Headers

6 hidden

	Key	Value	Description		Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	panda-76fb04924e6465f34bad2d4f0d2...				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Body

Cookies

Headers (12)

Test Results

200 OK • 326 ms • 1.04 KB • Save Response

JSON

Preview

Visualize

```
1 {
2   "folders": [
3     {
4       "id": "072ec3bf-b9ac-46bd-a6a9-887d93b67478",
5       "name": "$(dd if=/dev/zero bs=20M count=1 | base64)",
6       "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
7       "parent_folder_id": null,
8       "status": true,
9       "created_at": "2025-07-16T18:55:20.000Z",
10      "updated_at": "2025-07-16T18:55:20.000Z",
11      "videos_count": "0"
12    },
13    {
14      "id": "cf718666-72c1-449b-bc2a-78430d95f11a",
15      "name": "MUDANDO O NOME DA PASTA DE EXEMPLO DELETE",
16      "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
17      "parent_folder_id": null,
18      "status": true,
19      "created_at": "2025-07-16T19:23:52.000Z",
20      "updated_at": "2025-07-16T19:24:43.000Z",
21      "videos_count": "0"
22    },
23  ]
24 }
```

DELETE

HTTP Panda-QA-api / delete-folders Save Share

DELETE https://api-v2.pandavideo.com.br/folders/cf718666-72c1-449b-bc2a-78430d95f11a Send

Params Authorization Headers (8) Body Scripts Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (11) Test Results 500 Internal Server Error 1.41 s 585 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "errCode": "InternalServerError",
3   "errMsg": {}
4 }
```

LIST

HTTP Panda-QA-api / list-folders Save Share

GET https://api-v2.pandavideo.com.br/folders/ Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Headers 6 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	panda-76fb04924e6465f34bad2d4f0d2...				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Body Cookies Headers (12) Test Results 200 OK 329 ms 974 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "folders": [
3     {
4       "id": "072ec3bf-b9ac-46bd-a6a9-887d93b67478",
5       "name": "${dd if=/dev/zero bs=20M count=1 | base64}",
6       "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
7       "parent_folder_id": null,
8       "status": true,
9       "created_at": "2025-07-16T18:55:20.000Z",
10      "updated_at": "2025-07-16T18:55:20.000Z",
11      "videos_count": "0"
12    },
13    {
14      "id": "367ff9e3-1481-4ddb-b1c5-e1932a7b1d94",
15      "name": "test-panda-qa-create-test",
16      "user_id": "bcd8aeb6-5ff4-48e6-8152-51dab7d971dc",
17      "parent_folder_id": null,
18      "status": true,
19      "created_at": "2025-07-16T18:33:13.000Z",
20      "updated_at": "2025-07-16T18:33:13.000Z",
21      "videos_count": "0"
22    },
23  ]
24 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Como podemos ver, ele deleta sim a pasta, porém apresenta um erro de servidor que não posso indicar diretamente o que está acontecendo, porque não conheço o código. Sendo assim, é de suma importância contactar o desenvolvedor do mesmo para entender a situação. Após isso, verificar se ao deletar ele retorna uma mensagem padrão de debug como “sua pasta foi deletada com

sucesso!”, isso seria de excelente ajuda para orientar o usuário da api dos passos que foram executados, garantindo assim uma excelência no produto final de nossa aplicação.

## Explicando console do código Robot

```
Tests.Add To Cart
```

```
=====
```

```
Adicionar Item Ao Carrinho
```

```
| PASS |
```

Está informando que o arquivo `add_to_cart.robot` foi executado e o caso de teste chamado "Adicionar Item Ao Carrinho" foi executado com sucesso (PASS).

```
Tests.Checkout
```

```
=====
```

```
Finalizar Compra Com Sucesso
```

```
| PASS |
```

O caso de teste dentro do arquivo `checkout.robot` também foi executado e passou com sucesso. Este é o teste que automatiza o processo de compra.

```
Tests.Login
```

```
=====
```

```
Login Com Sucesso
```

```
| PASS |
```

Indica que o login foi realizado com sucesso no site do Sauce Demo.

```
[ERROR:sandbox\policy\win\sandbox_win.cc:762] Sandbox cannot access executable.
```

```
[ERROR:content\browser\network_service_instance_impl.cc:597] Network service crash
```

Essas mensagens são avisos do navegador Chrome ou do ChromeDriver no Windows.

Elas ocorrem por limitações de sandbox no Windows, mas não afetam o resultado dos testes.

```
Tests
```

```
| PASS |
```

```
3 tests, 3 passed, 0 failed
```

O Robot Framework finaliza a execução mostrando um **resumo completo**:

- Foram executados **3 testes**
- Todos **3 passaram**
- Nenhum teste falhou

**Arquivos Gerados:**



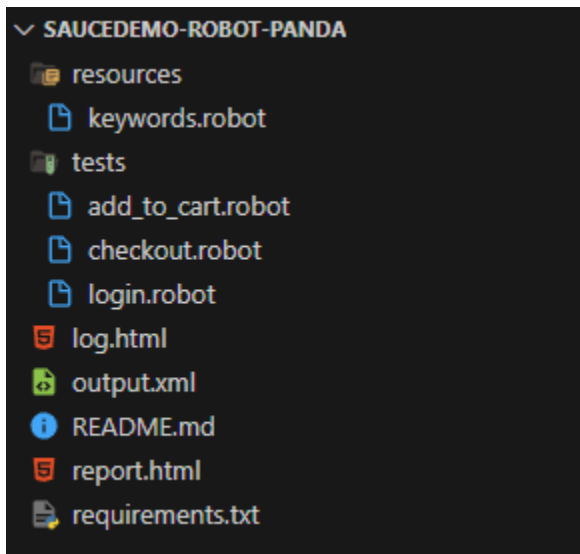
```
Output: C:\Users\cauzin\saucedemo-robot-panda\output.xml
Log:    C:\Users\cauzin\saucedemo-robot-panda\log.html
Report: C:\Users\cauzin\saucedemo-robot-panda\report.html
```

**output.xml**: arquivo bruto, usado internamente

**log.html**: relatório detalhado com **passo a passo da execução**

**report.html**: relatório resumido por teste (útil para o gerente ou analista de qualidade)

## Estrutura do Projeto:



# Conclusão e Recomendações Finais

Tentei neste documento explorar o máximo que pude de testes no postman, tentando encontrar os cenários mais adversos de erros e claro, de operações de sucesso na api da panda, com o objetivo final de assegurar uma API de qualidade para os usuários e desenvolvedores da plataforma. Também fiz questão de explicar algumas saídas do console do projeto implementado com o framework “Robot” para que fosse possível abranger todo o conteúdo proposto, reforço que grande parte do código em si foi explicado no arquivo [README.md](#) presente no github, como por exemplo passos futuros e melhorias no próprio código. Aqui, foi mais um extra para tentar ser o mais completo possível para a compreensão final.

No mais, meu muito obrigado pelo desafio e pela oportunidade de aprender com vocês. Espero estar junto em breve. Obrigado!

Em caso de alguma dúvida:

[thomascauzin@gmail.com](mailto:thomascauzin@gmail.com)

[Linkedin Thomás Causin](#)