Sobre este documento

Este documento tem como objetivo a avaliação técnica do Technical Writer e detalha os principais aspectos técnicos de uma API fictícia chamada **Petstore** (que está neste link). Este arquivo foi gerado usando a linguagem Markdown; o nome de arquivo contempla o nome do candidato, o objetivo e a data de envio: `SamuelSavickas_desafio_06092023.md` .

Conteúdo

Esta seção detalha como o conteúdo está divido:

Introdução

A API de serviços **Petstore** é uma API fictícia que simula o gerenciamento de uma loja de animais de estimação. Nesse contexto, foi projetada para permitir que donos de lojas de animais de estimação ofereçam uma variedade de serviços gerais para os animais de estimação dos clientes. Esses serviços podem incluir banho e tosa, cuidados de saúde, treinamento, produtos para animais e muito mais.

Esta API contém três objetos:

`PET` : Este grupo contém os endpoints necessários para gerenciar os registros dos animais de estimação:

Método	Endpoint	Descrição
PUT	/pet	Atualizar registro existente.
POST	/pet	Adicionar novo registro.
GET	/pet/findByStatus	Localizar registro pelo status.
GET	/pet/findByTags	Localizar registro pela tag.
GET	/pet/{petId}	Localizar registro pela ID.
POST	/pet/{petId}	Atualizar registro.
DELETE	/pet/{petId}	Deletar registro.
POST	/pet/{petId}/uploadImage	Adicionar imagem ao registro.

`STORE` : Este grupo contém os endpoints necessários para gerenciar pedidos da loja:

Método	Endpoint	Descrição
GET	/store/inventory	Retornar mapeamento do inventário por status e quantidade.
POST	/store/order	Criar um pedido.
GET	/store/order/{orderId}	Localizar uma ordem de compra pela ID.
DELETE	/store/order/{orderId}	Deletar uma ordem de compra pela ID.

`USER` : Este grupo contém os endpoints necessários para gerenciar os registros dos clientes:

Método	Endpoint	Descrição
POST	/user	Criar cliente.
POST	/user/createWithList	Criar lista de clientes.
GET	/user/login	Fazer login de cliente.
GET	/user/logout	Fazer logout de cliente.
GET	/user/{username}	Localizar cliente pelo username.
PUT	/user/{username}	Atualizar cliente.
DELETE	/user/{username}	Deletar cliente.

Observação: *Este documento irá detalhar exclusivamente o funcionamento do objeto* `STORE` .

Primeiros passos

Para acessar e configurar esta API, são necessários alguns passos iniciais.

Servidor

O endereço do servidor padrão usado pela API é:

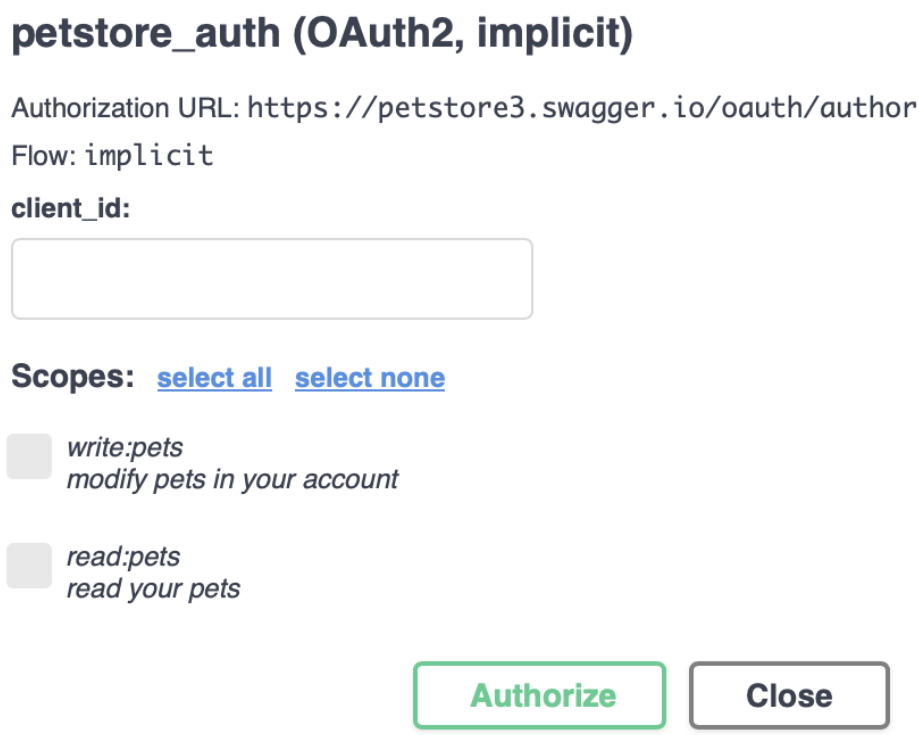
```
https://petstore3.swagger.io/api/v3/
```

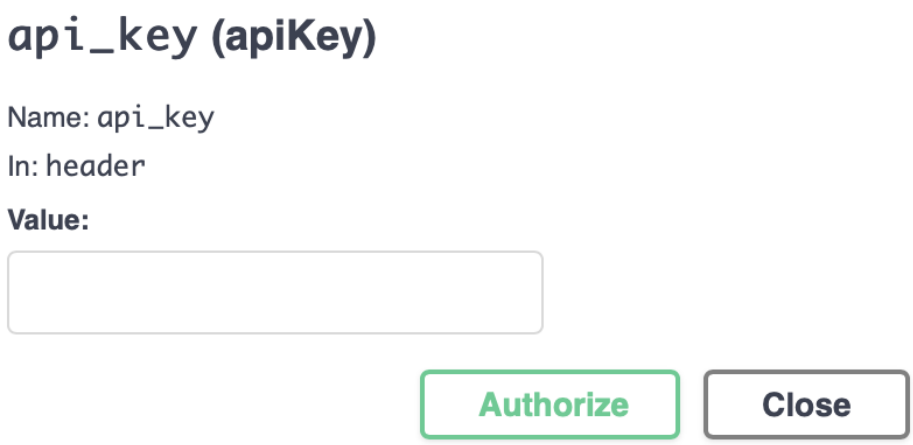
Identificação e autorização

Para receber permissões de utilização, são necessários dois tipos de verificação.

1. ID do cliente e seleção de escopos;
2. Chave de acesso da API.

Na página principal da API Petstore, clique no botão `Authorization` e preencha as informações necessárias, conforme as imagens abaixo:





Observação: *Como é um cenário de testes, pode-se escolher a ID de cliente e a chave de acesso. Essa chave de acesso deve ser utilizada no header em todas utilizações.*

Escopos

Os escopos controlam o nível de acesso que os usuários terão com a API. Estão disponíveis dois escopos:

Escopo	Descrição
read:pets	Permite apenas leitura
write:pets	Permite leitura e escrita

Objeto `STORE`

Contém informações necessárias para gerenciamento básico de pedidos da loja, como *criação*, *leitura*, *atualização* e *exclusão*.

Métodos e endpoints

Esta seção define todos os métodos e endpoints do objeto `STORE`.

`GET` /store/inventory

Retorna um inventário dos pedidos da loja, contendo status e quantidade.

Observação*: Este endpoint não requer parâmetros.*

Exemplo de request

Veja a seguir um exemplo de request do endpoint `/store/inventory`:

```
curl -X 'GET' \
  'https://petstore3.swagger.io/api/v3/store/inventory' \
  -H 'accept: application/json'
```

Exemplo de resposta

Veja a seguir um exemplo de resposta do endpoint `/store/inventory`, em caso de sucesso:

```
{
  "32": 1,
  "sold": 3,
  "string": 521,
  "unavailable": 1,
  "available1": 5,
  "pending": 4,
  "available": 267,
  "active": 47,
  "epic": 15,
  " sold": 1,
  "peric": 44,
  "nbmbm": 1,
  "not available": 1
}
```

Código de erro

Código HTTP	Descrição
500	Conteúdo não documentado.

POST /store/order

Cria um pedido na loja.

Request body

Parâmetro	Tipo	Exemplo	Descrição
id	integer ($int64)	10	ID do pedido
petId	integer ($int64)	198772	ID do animal
quantity	integer ($int64)	7	Quantidade
shipDate	string ($date-time)	2023-09-04T17:46:41.158Z	Data de expedição
status	string (placed, approved, delivered)	approved	Status do pedido
complete	boolean	true	Se o pedido está completo

Exemplo de request

Veja a seguir um exemplo de request do endpoint `/store/order` :

```
curl -X 'POST' \
  'https://petstore3.swagger.io/api/v3/store/order' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "id": 10,
  "petId": 198772,
  "quantity": 7,
  "shipDate": "2023-09-04T17:46:41.158Z",
  "status": "approved",
  "complete": true
}'
```

Exemplo de resposta

Veja a seguir um exemplo de resposta do endpoint `/store/order` , em caso de sucesso:

```
{
  "id": 10,
  "petId": 198772,
  "quantity": 7,
  "shipDate": "2023-09-04T17:46:41.158+00:00",
  "status": "approved",
  "complete": true
}
```

Observação: As definições da resposta estão na tabela *Request body*.

Códigos de erros

Código HTTP	Descrição
405	Entrada inválida

GET /store/order/orderId

Acessa uma ordem de compra pelo número de ID do pedido.

Observação: *Use IDs com valores inteiros entre <= 5 e > 10 para retornar respostas válidas.*

Path param

Parâmetro	Tipo	Obrigatório/Opcional	Descrição
orderId	integer ($int64)	Obrigatório	ID do pedido

Exemplo de request

Veja a seguir um exemplo de request do endpoint `/store/order/orderId` :

```
curl -X 'GET' \
  'https://petstore3.swagger.io/api/v3/store/order/10' \
  -H 'accept: application/xml'
```

Exemplo de resposta

Veja a seguir um exemplo de resposta do endpoint `/store/order/orderId` , em caso de sucesso:

```
{
  "id": 10,
  "petId": 198772,
  "quantity": 7,
  "shipDate": "2023-09-05T12:35:54.602Z",
  "status": "approved",
  "complete": true
}
```

Definições da resposta

Item	Tipo	Exemplo	Descrição
id	integer ($int64)	10	ID do pedido
petId	integer ($int64)	198772	ID do animal
quantity	integer ($int64)	7	Quantidade
shipDate	string ($date-time)	2023-09-04T17:46:41.158Z	Data de expedição
status	string (placed, approved, delivered)	approved	Status do pedido
complete	boolean	true	Se o pedido está completo

Códigos de erros

Código HTTP	Descrição
400	ID inválida
404	Pedido não localizado

DELETE /store/order/orderId

Deleta uma ordem de compra pelo número de ID do pedido.

Observação: *Use IDs com valores inteiros < 1000 para retornar respostas válidas.*

Path param

Parâmetro	Tipo	Obrigatório/Opcional	Descrição
orderId	integer ($int64)	Obrigatório	ID do pedido

Exemplo de request

Veja a seguir um exemplo de request do endpoint `/store/order/orderId` :

```
curl -X 'DELETE' \
  'https://petstore3.swagger.io/api/v3/store/order/2' \
  -H 'accept: */*'
```

Exemplo de resposta

Veja a seguir um exemplo de resposta do endpoint `/store/order/orderId` , em caso de sucesso:

```
{
  "code": 200,
  "type": "unknown",
  "message": "2"
}
```


Definições da resposta

Item	Tipo	Exemplo	Descrição
code	integer ($int64)	200 (sucesso)	Código de resposta
type	string	-	Não especificado
message	integer	2	Número do pedido deletado

Códigos de erros

Código HTTP	Descrição
400	ID inválido
404	Pedido não localizado

Taxas de limite

A Petstore API oferece limitações diferentes para as opções **gratuita** e **paga** para atrair e satisfazer uma variedade de clientes.

Opção gratuita

Inclui	Descrição
Limite de requisições por minuto	100 requisições por minuto.
Acesso básico	Recursos essenciais disponíveis: listar serviços, agendar serviços e visualizar detalhes do cliente.
Armazenamento de dados	Limite de 1 GB.
Suporte básico	Suporte ao cliente com tempos maiores de resposta e recursos limitados.

Opção paga

Inclui	Descrição
Limite de requisições por minuto	1.000 requisições por minuto.
Acesso completo	Inclui o acesso básico e recursos avançados: análises avançadas, como relatórios detalhados, recursos de gerenciamento de estoque avançados.
Armazenamento de dados	Limite de 50 GB.
Suporte premium	Suporte prioritário ao cliente com tempos de resposta mais rápidos e acesso a recursos de suporte avançados, como assistência técnica direta.

Melhores práticas de segurança

Proteger as credenciais de API é essencial para garantir a segurança dos sistemas e dados; é também uma parte crítica do desenvolvimento de aplicativos e serviços modernos. Veja abaixo algumas recomendações sobre como os desenvolvedores devem proteger suas credenciais de API. Seguir essas recomendações pode ajudar a proteger suas credenciais e garantir que seus sistemas permaneçam seguros e confiáveis.

Use variáveis de ambiente

Evite armazenar credenciais diretamente no código-fonte. Em vez disso, use variáveis de ambiente para armazenar informações sensíveis, como chaves de API, senhas e tokens.

Não compartilhe credenciais

Nunca compartilhe suas credenciais de API publicamente em repositórios de código, fóruns ou outros locais acessíveis ao público. Isso inclui evitar fazer push de credenciais para repositórios Git.

Utilize gerenciadores de credenciais

Considere o uso de gerenciadores de credenciais confiáveis para armazenar e gerenciar suas chaves de API. Isso pode ajudar a proteger suas credenciais com criptografia forte.

Restrinja acesso

Aplique o princípio do "princípio mínimo de privilégio". Forneça apenas as permissões necessárias para suas credenciais, limitando o acesso somente ao que é estritamente necessário.

Mantenha atualizações de segurança

Certifique-se de que as bibliotecas e frameworks que você usa para acessar a API estejam atualizadas com as correções de segurança mais recentes.

Use HTTPS

Certifique-se de que todas as comunicações com a API sejam feitas por meio de HTTPS para criptografar os dados em trânsito.

Implemente autenticação de dois fatores (2FA)

Se possível, ative a autenticação de dois fatores em suas contas, especialmente em serviços que fornecem acesso às suas credenciais de API.

Monitore e audite atividades

Implemente registros de auditoria para rastrear atividades relacionadas às suas credenciais de API, permitindo a detecção rápida de atividades suspeitas.

Rode testes de segurança

Realize testes de penetração e varreduras de segurança regulares em seu sistema para identificar vulnerabilidades e lacunas de segurança.

Rotacione chaves regularmente

Altere suas chaves de API regularmente, especialmente se você suspeitar de qualquer comprometimento de segurança.

Eduque a equipe

Treine sua equipe para entender a importância da segurança das credenciais de API e seguir boas práticas de segurança.

Implemente controle de acesso baseado em função (RBAC)

Use controles de acesso baseados em função para limitar quem pode acessar as credenciais e quais ações eles podem executar.

Perguntas frequentes

Todas perguntas importam! Por isso, selecionamos cinco perguntas recorrentes relacionadas à API Petstore.

Pergunta: Como faço para obter acesso à API Petstore?

Resposta: Para obter acesso à API Petstore, você precisa registrar sua loja em nossa plataforma. Isso fornecerá as credenciais de autenticação necessárias para começar a usar a API, como chaves de acesso e nome de usuário. Confira o tópico Introdução > Primeiros passos.

Pergunta: Quais são os endpoints da API?

Resposta: Esta API contém três objetos com os seguintes endpoints:

PET

Método	Endpoint
PUT	/pet
POST	/pet
GET	/pet/findByStatus
GET	/pet/findByTags
GET	/pet/{petId}
POST	/pet/{petId}
DELETE	/pet/{petId}
POST	/pet/{petId}/uploadImage

STORE

Método	Endpoint
GET	/store/inventory
POST	/store/order
GET	/store/order/{orderId}
DELETE	/store/order/{orderId}

USER

Método	Endpoint
POST	/user
POST	/user/createWithList
GET	/user/login
GET	/user/logout
GET	/user/{username}
PUT	/user/{username}
DELETE	/user/{username}

Confira o tópico Introdução para descrições detalhadas de cada objeto e endpoint.

Pergunta: Quais são os passos para iniciar um pedido para um cliente por meio da API?

Resposta: Para iniciar um pedido, você deve enviar uma solicitação `POST` para o endpoint `/store/order`, fornecendo os detalhes relevantes, como:

- ID do pedido
- ID do animal
- Quantidade
- Data de expedição
- Status do pedido
- Se o pedido está completo

Confira o tópico Objeto STORE > POST/store/order.

Pergunta: Como é possível visualizar um pedido feito por meio da API?

Resposta: É possível visualizar os pedidos realizados fazendo uma solicitação `GET` para o endpoint `/store/order/orderId`, indicando o número do pedido. Isso retornará detalhes relevantes, como:

- ID do pedido
- ID do animal
- Quantidade
- Data de expedição
- Status do pedido
- Se o pedido está completo

Confira o tópico Objeto STORE > GET/store/order/orderId.

Pergunta: O uso da API é gratuito?

Resposta: Existem duas opções de uso: gratuita e paga, com diferentes limitações e recursos. Confira o tópico Taxas de limite.

Apêndice - Propostas para controle de versão de documentação de API

O controle de versão da documentação da API é fundamental para garantir que desenvolvedores e clientes tenham acesso às informações corretas, precisas e atualizadas sobre como usar a API. Isso facilita o uso correto da API e ajuda a evitar confusões causadas por versões desatualizadas da documentação.

A seguir, estão alguns passos que podem ser adotados para controlar a versão de uma documentação de API.

Sistema de controle de versão (VCS)

Utilize uma ferramenta de controle de versão, como Git, para gerenciar a documentação. Isso permite acompanhar todas as alterações ao longo do tempo. Nesse sistema, a versão do documento deve seguir a versão da API e pode estar no formato de tag na primeira página ou em um arquivo de README no repositório da documentação.

Crie um repositório para a documentação

Crie um repositório dedicado para armazenar a documentação da API. Isso pode ser feito no mesmo serviço onde você mantém o código-fonte da API ou em um repositório separado.

Utilize ramificações (branches)

Ao fazer alterações na documentação, crie ramificações separadas no repositório para cada nova versão ou conjunto de alterações. Isso ajuda a manter as versões anteriores da documentação intactas, pois os clientes podem estar usando uma versão anterior da API e devem ter acesso à documentação correspondente.

Ciclo de desenvolvimento

A documentação da API deve estar prevista no ciclo de desenvolvimento da API. Sempre que ocorrerem alterações significativas na API, deve-se atualizar a documentação correspondente. Isso inclui adicionar novos recursos, remover recursos obsoletos ou alterar a estrutura da API.

Forneça um histórico de alterações

Inclua um registro de alterações na documentação para que os desenvolvedores possam acompanhar as atualizações ao longo do tempo.