



Documentação referente à API dos Microserviços Backend

Bem-vindo à documentação oficial da API dos Microserviços Backend da **EVOB**! Esta documentação foi elaborada para facilitar o entendimento e a integração dos desenvolvedores com os diversos serviços fornecidos por nossa arquitetura de microserviços.

Os Microserviços Backend são um conjunto de serviços modulares e escaláveis projetados para atender às necessidades de diferentes aplicações e plataformas. Com essa abordagem, buscamos promover a flexibilidade, a manutenibilidade e a eficiência em nossos sistemas, permitindo que desenvolvedores criem e integrem funcionalidades de maneira simples e eficaz.

Nossa documentação detalha os endpoints, parâmetros, métodos de autenticação e exemplos de solicitações e respostas para cada serviço disponível. Além disso, fornecemos informações sobre as melhores práticas de uso, possíveis casos de uso e cenários de integração para garantir uma experiência fluida e consistente para os desenvolvedores.

Estamos comprometidos em manter esta documentação atualizada e precisa, refletindo todas as mudanças e atualizações feitas em nossos microserviços. Caso tenha alguma dúvida ou precise de assistência adicional, não hesite em entrar em contato com nossa equipe de suporte técnico.

Agradecemos por escolher utilizar os Microserviços Backend em seus projetos e esperamos que esta documentação facilite sua jornada de desenvolvimento.

Índice

1.0 – SETUP, preparando o ambiente de desenvolvimento

- 1.1 – Instalando o Postman
- 1.2 – Importando a *Collection* (Coleção) **EVOB Public**
- 1.3 – Importando o Environment (Ambiente) **EVOB Public**
- 1.4 – Disponibilidade da API (Uptime)

2.0 – Realizando Requisições Abertas, sem a necessidade de *autenticação*

- 2.1 – Exibição de **Brand** (*Plataforma*)
- 2.2 – Exibição de **Brand** (*Plataforma*) por Origin
- 2.3 – Listagem de **Stores** (*Lojas*)
- 2.4 – Exibição de **Store** (*Loja*)
- 2.5 – Listagem de **Courses** (*Cursos*)
- 2.6 – Exibição de **Course** (*Curso*)
- 2.7 – Listagem de **Classrooms** (Turmas) para Curso
- 2.8 – Exibição de **Offer** (*Oferta*)
- 2.9 – Listagem de **Banners**

1.0 – SETUP

Preparando o ambiente de Desenvolvimento

1.1 – Instalando o Postman

O Postman é uma plataforma de colaboração para o desenvolvimento de APIs, permitindo aos desenvolvedores projetar, testar e documentar facilmente APIs. Com uma interface intuitiva e repleta de recursos poderosos, o Postman simplifica o processo de interação com APIs, desde a fase inicial de desenvolvimento até a implantação e manutenção.

Com o Postman, os desenvolvedores podem criar solicitações HTTP personalizadas e visualizar facilmente as respostas, testar diferentes cenários de uso, automatizar fluxos de trabalho complexos e compartilhar coleções de APIs com colegas de equipe ou a comunidade.



Além disso, o Postman oferece recursos avançados, como monitoramento de APIs em tempo real, geração automática de documentação, integração com sistemas de controle de versão e a capacidade de colaborar em projetos em equipe.

Você pode realizar o download do Postman acessando o link:

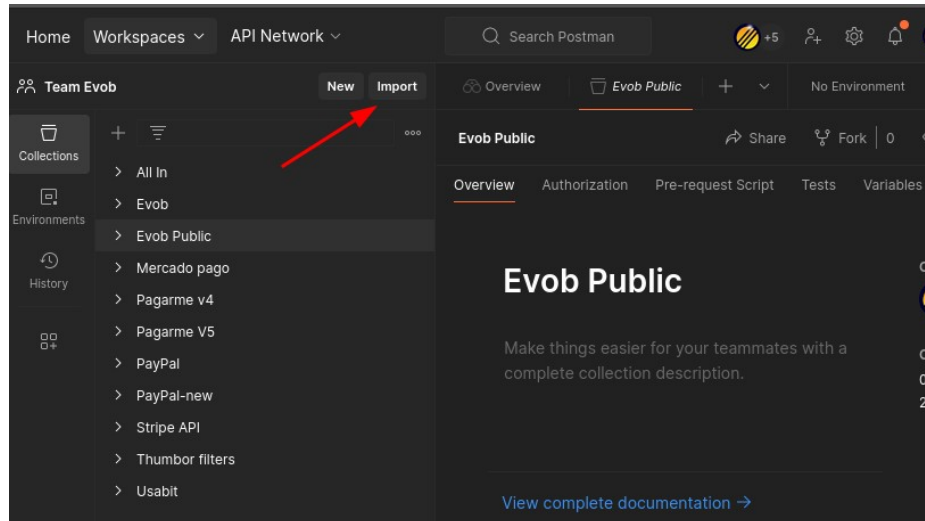
<https://www.postman.com/downloads/>

Após realizar o download do programa, siga as instruções da página indicada no link abaixo para instalá-lo em seu ambiente de desenvolvimento:

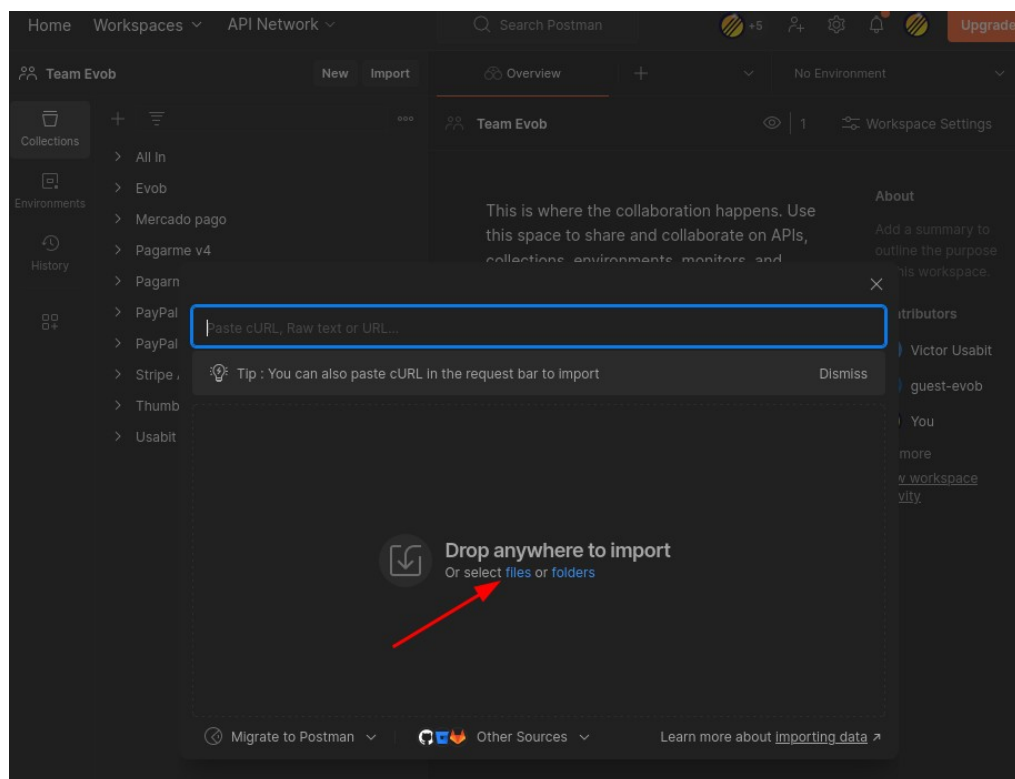
<https://learning.postman.com/docs/getting-started/installation/installation-and-updates/>

1.2 - Importando a *Collection* (Coleção) EVOB Public

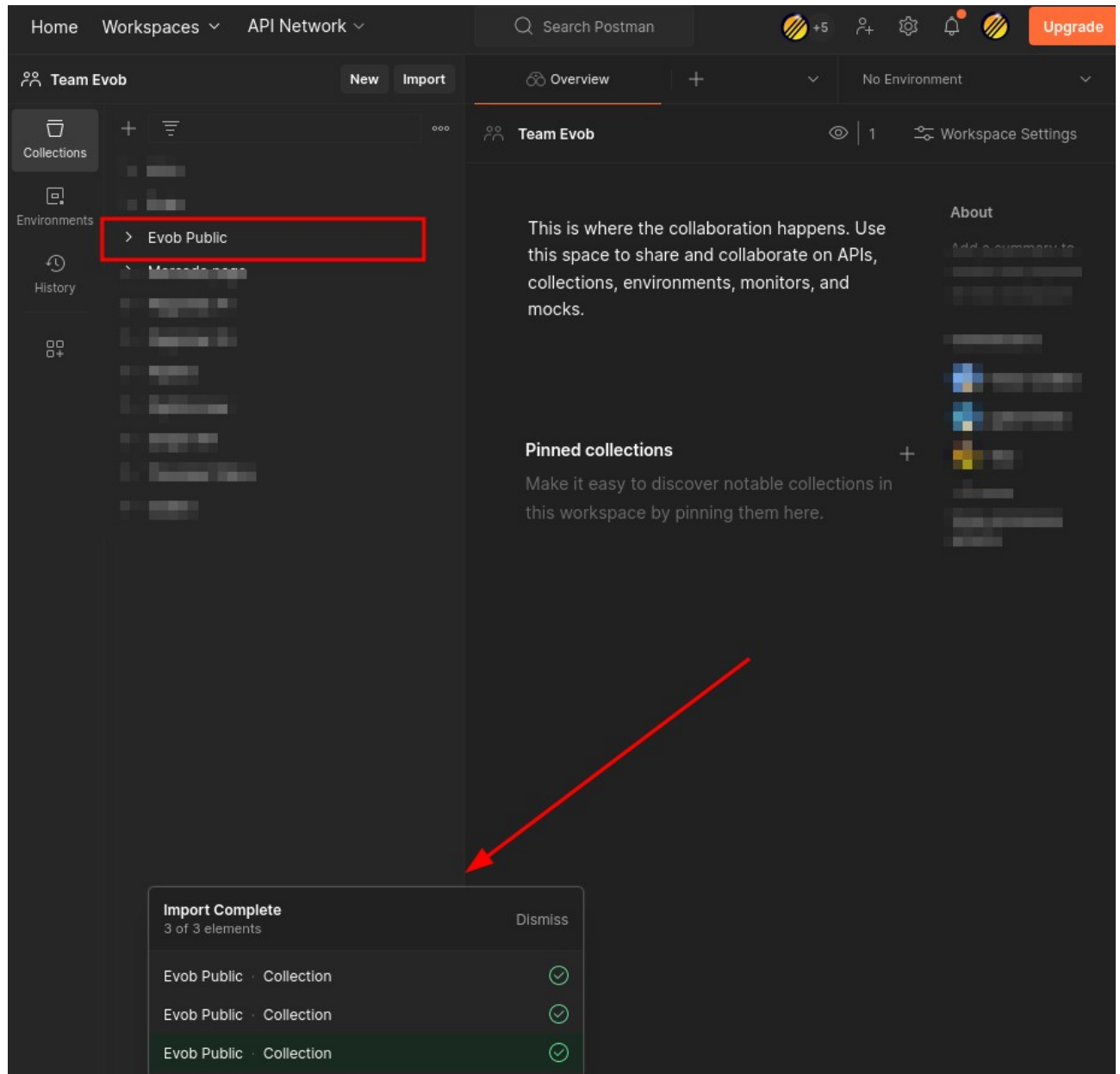
Após realizar a instalação do Postman descrita no item nº 1.1, ao iniciá-lo clique no botão **Import** (*Importar*):



Clique em **Select Files** (*Selecionar Arquivos*), e selecione o arquivo recebido junto desta documentação: `Evob Public.postman_collection`

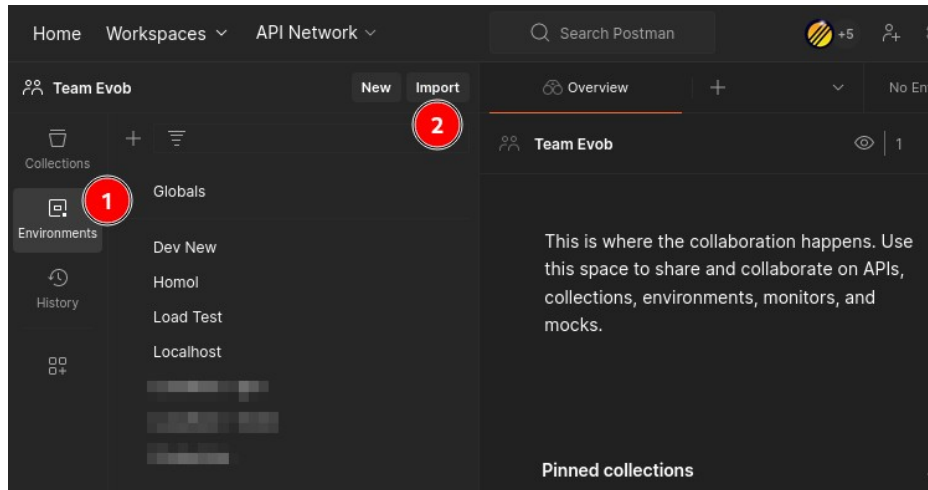


O programa apresentará um status da importação, e a coleção **EVOB Public** estará disponível para navegação:

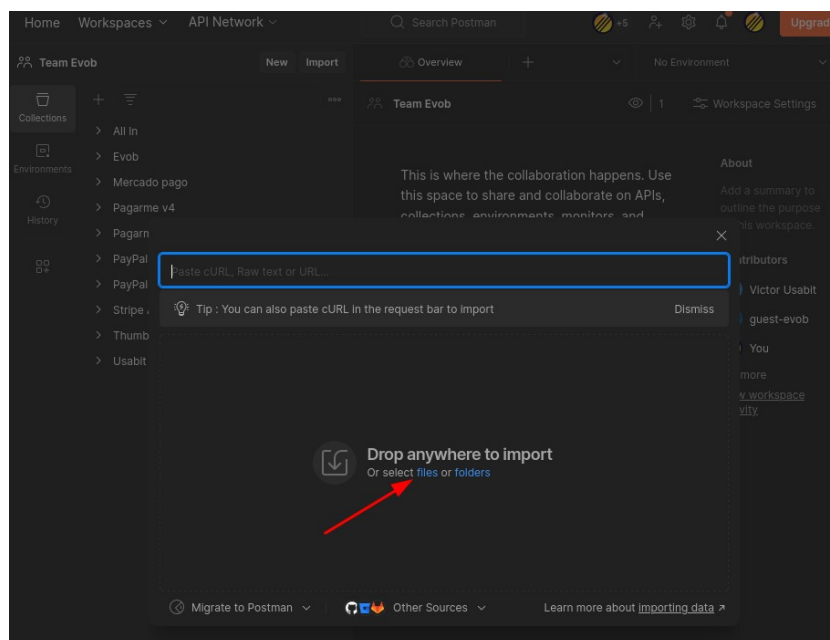


1.3 – Importando o Environment (Ambiente) EVOB Public

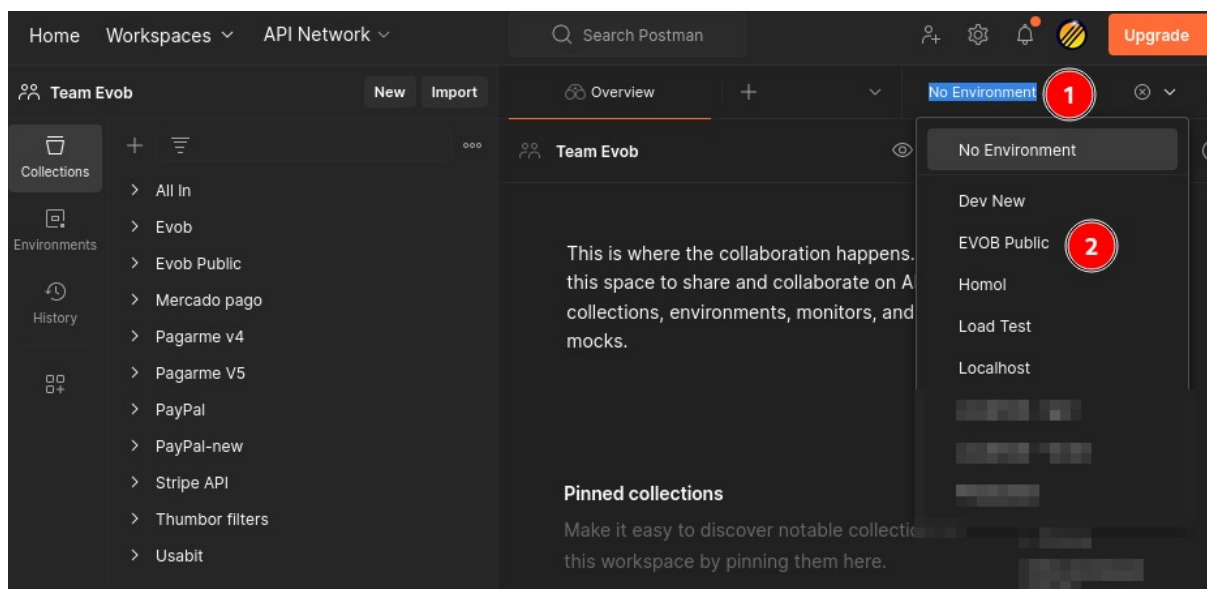
O arquivo de **Environment** (ambiente) possui todas as variáveis de ambiente necessárias para realizar requisições em nossas APIs. Para importá-lo acesse o aplicativo Postman instalando no passo nº 1.1 e clique na opção **Environments** (Ambientes) [1] e depois na opção **Import** (Importar) [2]:



Clique em **Select Files** (Selecionar Arquivos), e selecione o arquivo recebido junto desta documentação: Evob Public.postman_environment



O programa apresentará um status da importação, e o ambiente **EVOB Public** estará disponível para uso e para concluir, lembre-se de deixar selecionado o ambiente **EVOB Public** na seção Ambiente:



1.4 – Disponibilidade da API (Uptime)

A disponibilidade da nossa API de desenvolvimento, também conhecida como Uptime, é uma medida fundamental que reflete o tempo em que a API está operacional e acessível aos usuários. Estamos comprometidos em fornecer uma experiência confiável e consistente aos desenvolvedores que utilizam nossos serviços.

Atualmente, nossa API do Ambiente de Desenvolvimento está programada para ficar disponível de **segunda-feira à domingo**, das **8:00 às 23:00**, no horário de São Paulo. Durante esse período, os desenvolvedores podem acessar e utilizar os endpoints da API para integrar funcionalidades em suas aplicações com confiança.

É importante destacar que estamos abertos a ajustar o horário de disponibilidade da API conforme as necessidades dos usuários. Caso haja demanda por uma extensão do horário de funcionamento, nossa equipe de infraestrutura estará pronta para considerar e implementar essas alterações.

2.0 – Realizando Requisições Abertas

Sem a necessidade de autenticação

As requisições abertas sem autenticação de usuário representam um método direto e conveniente de interagir com a API, especialmente para recursos que são públicos e não exigem controle de acesso. Essa abordagem simplifica significativamente o processo de teste e exploração de endpoints da API, permitindo que desenvolvedores explorem facilmente funcionalidades disponíveis sem a necessidade de configurar autenticação.

Ao lidar com recursos de API que são públicos e abertos, os desenvolvedores têm a liberdade de experimentar e entender a estrutura da API sem restrições de autenticação. Isso é particularmente útil durante as fases iniciais de desenvolvimento, quando é essencial ter uma visão clara dos recursos disponíveis e das operações suportadas pela API.

É importante ressaltar que, embora as requisições abertas sejam convenientes e adequadas para recursos públicos, é fundamental entender que podem haver limitações de segurança associadas. Em alguns casos, certos dados ou funcionalidades podem não estar disponíveis sem autenticação, e é essencial compreender as políticas e práticas recomendadas pela API em questão.

Portanto, ao trabalhar com requisições abertas sem autenticação de usuário, é crucial garantir que os desenvolvedores compreendam claramente quais recursos estão acessíveis e que estejam cientes das limitações de segurança potenciais. Ao fazê-lo, os desenvolvedores podem aproveitar ao máximo os benefícios oferecidos pelas requisições abertas enquanto garantem uma experiência de desenvolvimento segura e eficaz.

2.1 – Exibição de Brand (Plataforma)

Brand (Plataforma) é a entidade que representa as marcas em nosso sistema. Deste modo, ao listar determinada **Brand** recebe-se os dados elencados abaixo:

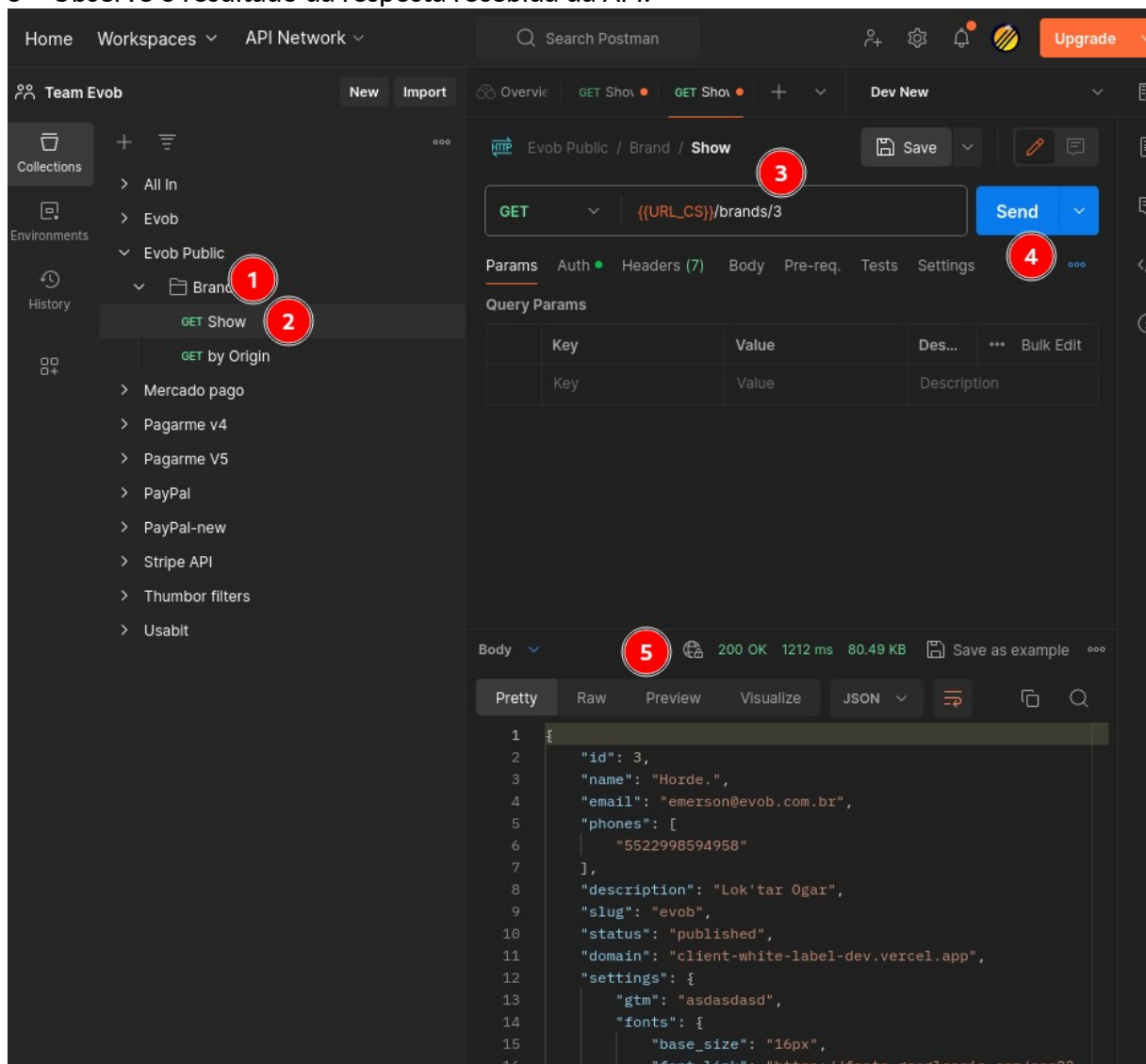
- Nome;
- E-mail;
- Telefones;
- Descrição;
- Domínio;
- Cores de Layout;
- Logotipo;

Exemplo de Requisição:**GET:** `{{URL_CS}}/brands/:id_brand`

Aonde: `id_brand` representa o código único e exclusivo da plataforma.

Recomendamos a utilização de `:id_brand` como 3 ou 2 nos testes de desenvolvimento.

- 1 - Dentro da Coleção **Evob Public**, clique na pasta **Brand**;
- 2 - Selecione a Requisição **Show**
- 3 - Insira o **ID** da Brand que deseja visualizar;
- 4 - Clique no botão **Send** (Enviar);
- 5 - Observe o resultado da resposta recebida da API.

**Resultado Esperado:**

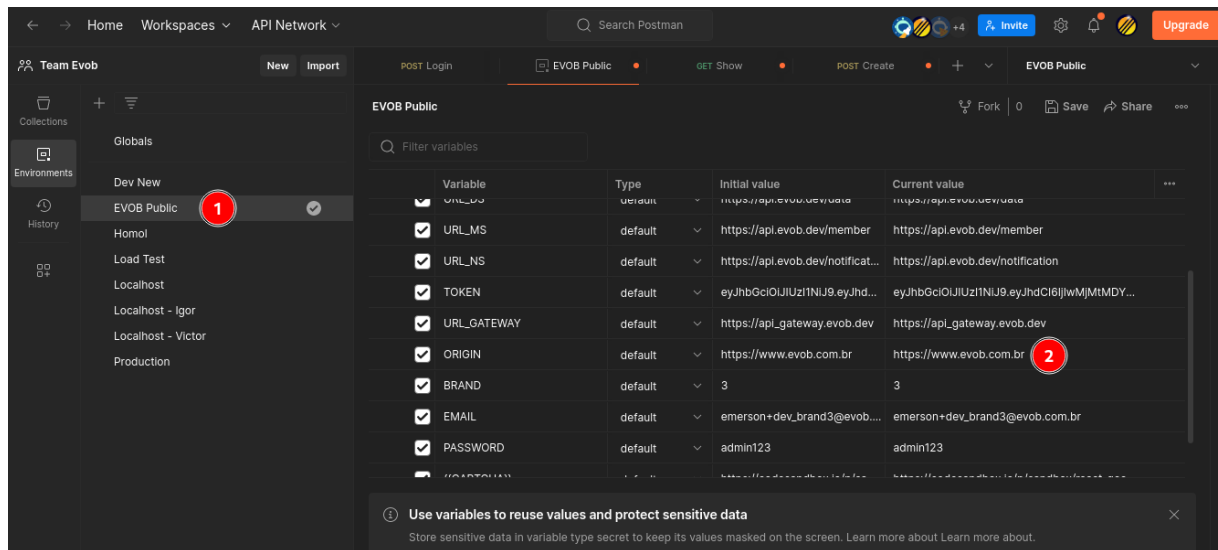
HTTP status code: 200

Body: json data

2.2 - Exibição de Brand (Plataforma) por Origin

Visando oferecer dinamismo para requisições à partir de sites e aplicativos, nossa API identifica qual **Brand** (*Plataforma*) está sendo requisitada através da Origin de requisição.

Pode-se configurar o domínio no **Environment** (*Ambiente*) **EVOB Public** [1] variável **ORIGIN** [2].



Caso esteja utilizando um domínio customizado, solicite sua adição ao nosso time de infraestrutura. Para simular ambientes locais de desenvolvimento, o padrão é:

http://localhost:300:id_brand

aonde :id_brand é o ID da Plataforma que deseja utilizar.

Deste modo, pode-se executar em ambiente localhost qualquer projeto simulando determinada Plataforma, apenas subindo o aplicativo no padrão de porta 3000 + (mais) o ID desejável. Exemplo: <http://localhost:3024> para Plataforma ID 24.

Exemplo de Requisição:

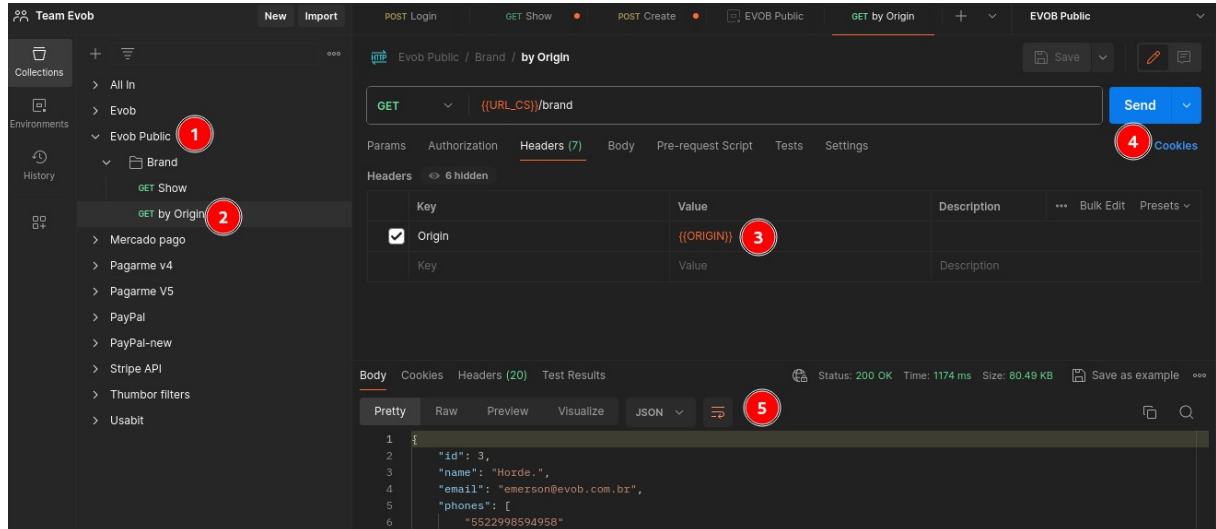
GET: {{URL CS}}/brand

Com o Header (cabeçalho) de requisição Obrigatório: Origin

- 1 - Dentro da Coleção **Evob Public**, clique na pasta **Brand**;
- 2 - Selecione a Requisição **By Origin**
- 3 - Insira o **Header** da Origin que deseja visualizar;

4 - Clique no botão **Send** (Enviar);

5 - Observe o resultado da resposta recebida da API.



Resultado Esperado:

HTTP status code: 200

Body: json data

Informação Importante:

Com este padrão de Origin no cabeçalho, qualquer consulta aos endpoints será filtrada pela **Brand** (Plataforma) desejada. Impossibilitando a exibição de dados cruzados dentre as marcas.

2.3 - Listagem de Stores (Lojas)

Store (Loja) é a entidade que representa franquias ou filias pertencentes a uma determinada **Brand** (Plataforma).

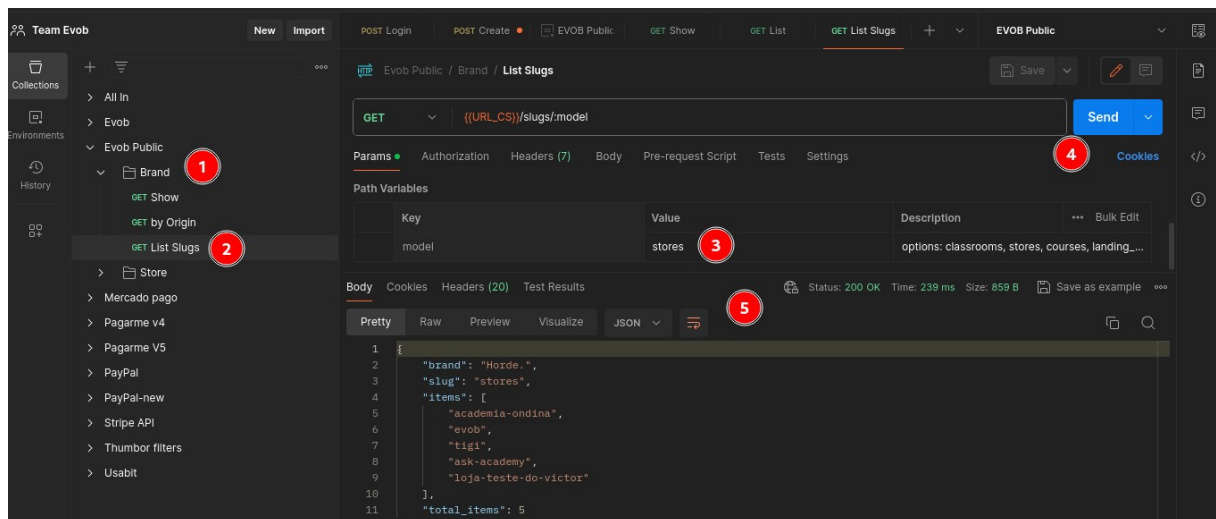
Deste modo uma Plataforma pode ter Lojas distintas, com Cursos exclusivos, Split de pagamento, e financeiro separados. Possibilitando regras e configurações únicas para cada Loja.

Para consultar quais Lojas estão disponíveis para a Plataforma, realizamos a listagem de Stores (Lojas).

Exemplo de Requisição:**GET:** `{{URL_CS}}/slugs/:model`

Com o Header (cabeçalho) de requisição Obrigatório: Origin
Aonde: *model* representa a entidade que deseja-se listar.

- 1 – Dentro da Coleção **Evob Public**, clique na pasta **Brand**;
- 2 – Selecione a Requisição **List Slugs**
- 3 – Adicione o **Parâmetro** :model como '*courses*'
- 4 – Clique no botão **Send** (Enviar);
- 5 – Observe o resultado da resposta recebida da API.

**Resultado Esperado:**

HTTP status code: 200

Body: *json data*

2.4 – Exibição de Store (Loja)

Estando em posse das Lojas publicadas, (vide item nº 2.3), pode-se requisitar seus dados como:

- Nome;
- E-mail;
- Telefones;
- Descrição;
- Domínio;

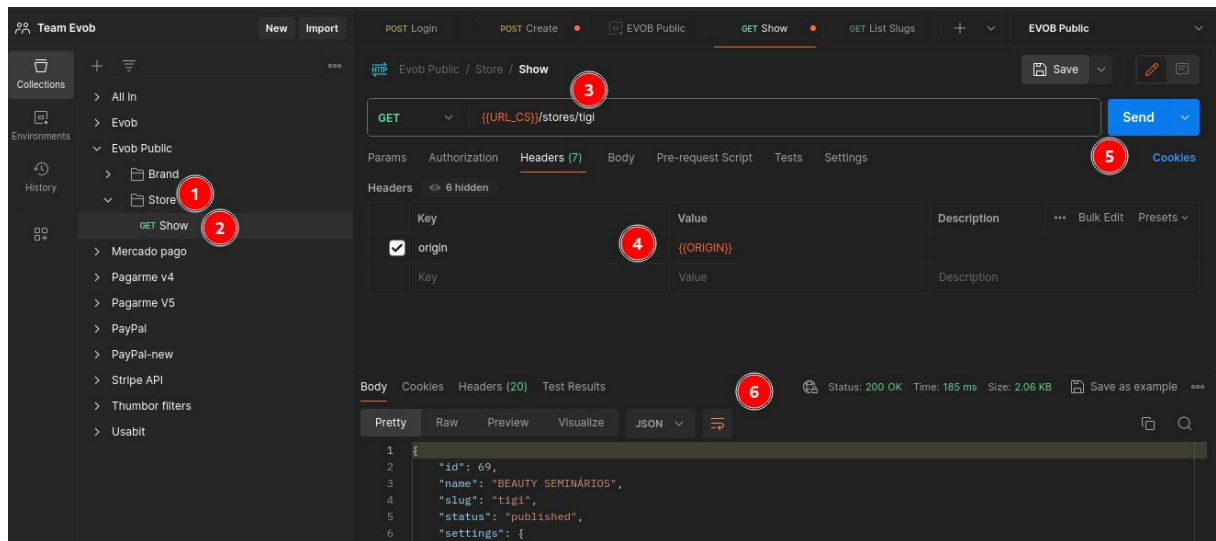
- Endereço;
- Logotipo;

Exemplo de Requisição:**GET:** `{{URL_CS}}/stores/:slug`

Com o Header (cabeçalho) de requisição Obrigatório: Origin

Aonde: *slug* representa a Loja que deseja-se exibir. Podendo passar a Slug ou o ID da Loja.

- 1 – Dentro da Coleção **EvoB Public**, clique na pasta **Store**;
- 2 – Selecione a Requisição **Show**;
- 3 – Insira a slug da Loja que deseja visualizar;
- 4 – Certifique-se de passar o **Header** da Origin da sua Plataforma;
- 5 – Clique no botão **Send** (Enviar);
- 6 – Observe o resultado da resposta recebida da API.

**Resultado Esperado:**

HTTP status code: 200

Body: *json data*

2.5 – Listagem de Courses (Cursos)

Course (*Curso*) é a entidade que representa o produto distribuído/comercializado pelas Plataformas. Sendo responsável pelo agrupamento de todo o conteúdo didático. Como Turmas, Módulos, Aulas e Atividades.

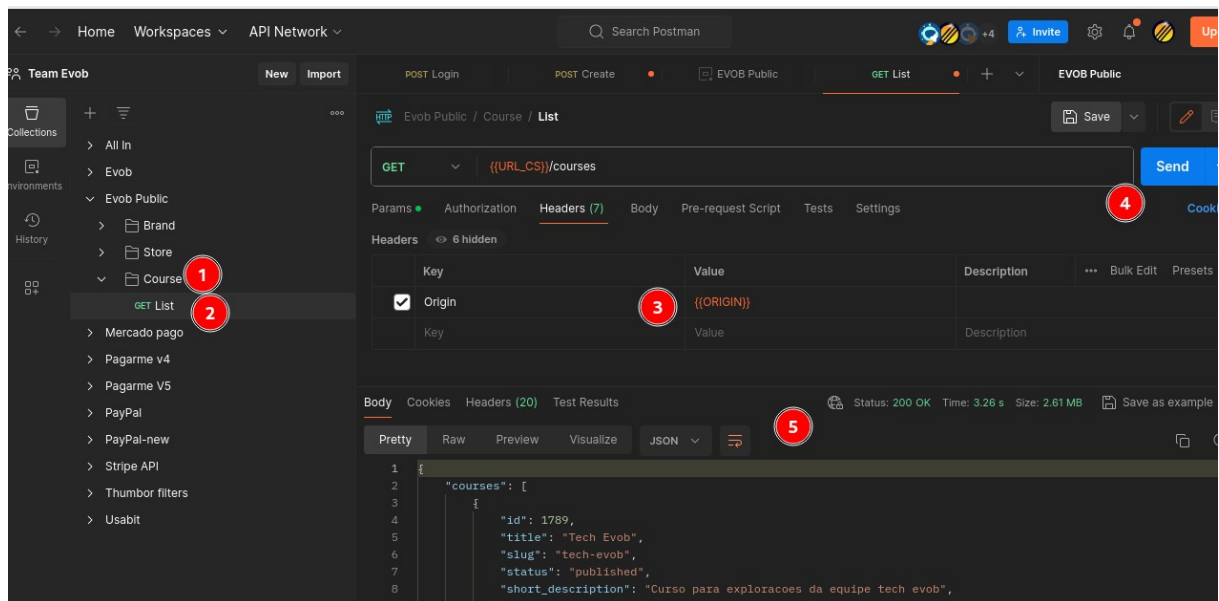
Para consultar quais cursos estão disponíveis para determinada Brand (Plataforma) ou Store (Loja), realizamos a consulta exemplificada abaixo:

Exemplo de Requisição:

GET: {{URL_CS}}/courses

Com o Header (cabeçalho) de requisição Obrigatório: Origin

- 1 – Dentro da Coleção **EvoB Public**, clique na pasta **Course**;
- 2 – Selecione a Requisição **List**;
- 3 – Certifique-se de passar o **Header** da Origin da sua Plataforma;
- 4 – Clique no botão **Send** (Enviar);
- 5 – Observe o resultado da resposta recebida da API.



Resultado Esperado:

HTTP status code: 200

Body: json data

Opções de Filtros Disponíveis através de URL Parâmetros Requests:

a-) Filtrar por Loja:

```
{{URL_CS}}/courses?q[store_id_eq]=:id_da_loja
```

b-) Tipo de Curso: (live | presencial | online)

```
{{URL_CS}}/courses?course_type=online
```

c-) Em destaque:

```
{{URL_CS}}/courses?q[featured_true]=true
```

d-) Com Oferta Gratuita:

```
{{URL_CS}}/courses?q[has_free_offer_false]=true
```

d-) Busca por palavras chave:

```
{{URL_CS}}/courses?search_string=cabelos,corte
```

e-) Por data de Criação:

```
{{URL_CS}}/courses?q[created_at_gteq]=2021-10-01T00:00:00.000Z&q[created_at_lteq]=2021-11-01T00:00:00.000Z
```

Aonde:

```
q[created_at_gteq] = Data Inicial
```

```
q[created_at_lteq] = Data Final
```

Informação Importante:

Todos as opções de filtros podem ser utilizadas em conjunto, possibilitando várias combinações de listagem. Por Exemplo:

```
{{URL_CS}}/courses?q[store_id_eq]=69&q[has_free_offer_false]=true&search_string=cabelos,corte&q[created_at_gteq]=2021-10-01T00:00:00.000Z&q[created_at_lteq]=2021-11-01T00:00:00.000Z
```

2.6 – Exibição de Course (Curso)

Em posse da Listagem de Cursos, pode-se consultar especificamente uma unidade retornando informações como:

- Título;
- Slug;
- Status;
- Descrição Curta e Longa;
- Tags;
- Nível;
- Banners;

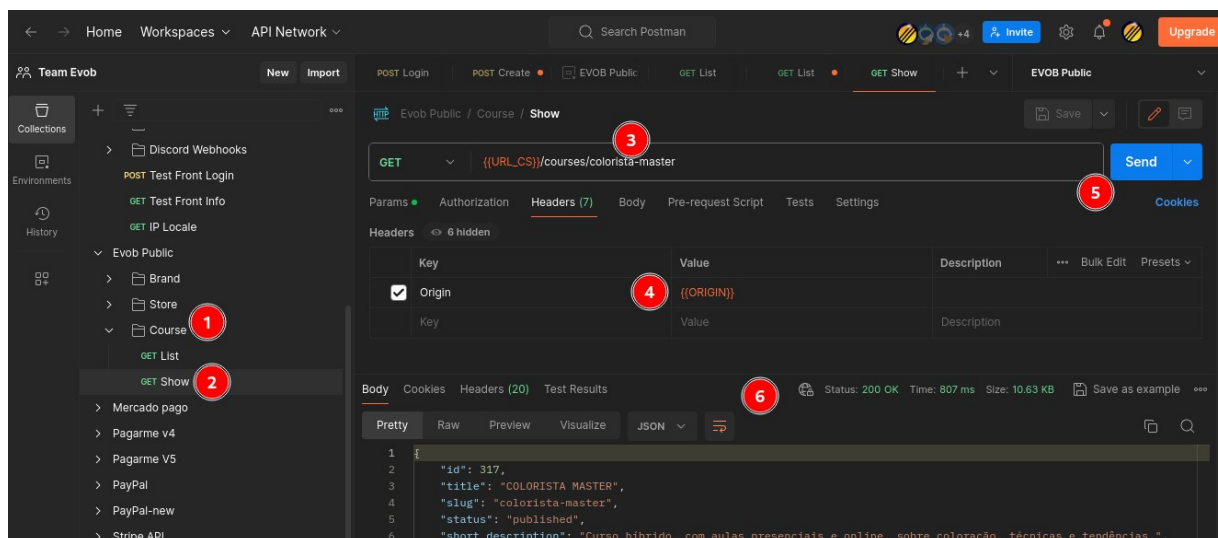
Exemplo de Requisição:

GET: `{{URL_CS}}/courses/:slug`

Com o Header (cabeçalho) de requisição Obrigatório: Origin

Aonde: *slug* representa o Curso que deseja-se exibir. Podendo passar a Slug ou o ID do Curso.

- 1 – Dentro da Coleção **EvoB Public**, clique na pasta **Course**;
- 2 – Selecione a Requisição **Show**;
- 3 – Insira a slug ou ID do Curso que deseja visualizar;
- 4 – Certifique-se de passar o **Header** da Origin da sua Plataforma;
- 5 – Clique no botão **Send** (Enviar);
- 6 – Observe o resultado da resposta recebida da API.



Resultado Esperado:

HTTP status code: 200

Body: json data

2.7 - Listagem de Classrooms (Turmas) para Curso

Classroom (Turma) é a entidade que representa as Salas de Aula dos **Cursos**. Sendo responsável pelo agrupamento de alunos, assim como a especificação de quais Módulos e Aulas pertencerão a grade de ensino. Podendo um mesmo Curso ter várias turmas, cada uma com mais ou menos aulas.

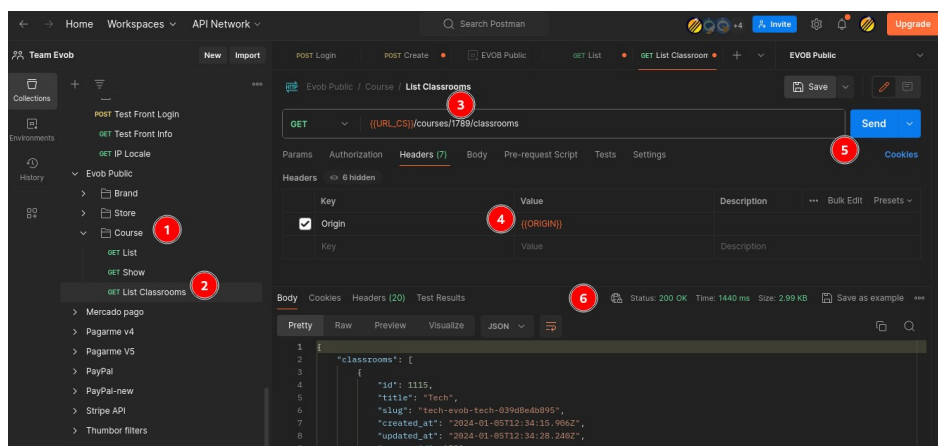
Para consultar quais Turmas estão disponíveis para determinado Curso, realizamos a consulta exemplificada abaixo:

Exemplo de Requisição:**GET:** `{{URL_CS}}/courses/:slug/classrooms`

Com o Header (cabeçalho) de requisição Obrigatório: Origin

Aonde: *slug* representa o Curso que deseja-se filtrar. Podendo passar a Slug ou o ID do Curso.

- 1 – Dentro da Coleção **EvoB Public**, clique na pasta **Course**;
- 2 – Selecione a Requisição **List Classrooms**;
- 3 – Insira a slug ou ID do Curso que deseja visualizar;
- 4 – Certifique-se de passar o **Header** da Origin da sua Plataforma;
- 5 – Clique no botão **Send** (Enviar);
- 6 – Observe o resultado da resposta recebida da API.



Resultado Esperado:

HTTP status code: 200

Body: json data

No response da requisição, consta uma chave chamada ['offers'] contendo as ofertas de cada turma.

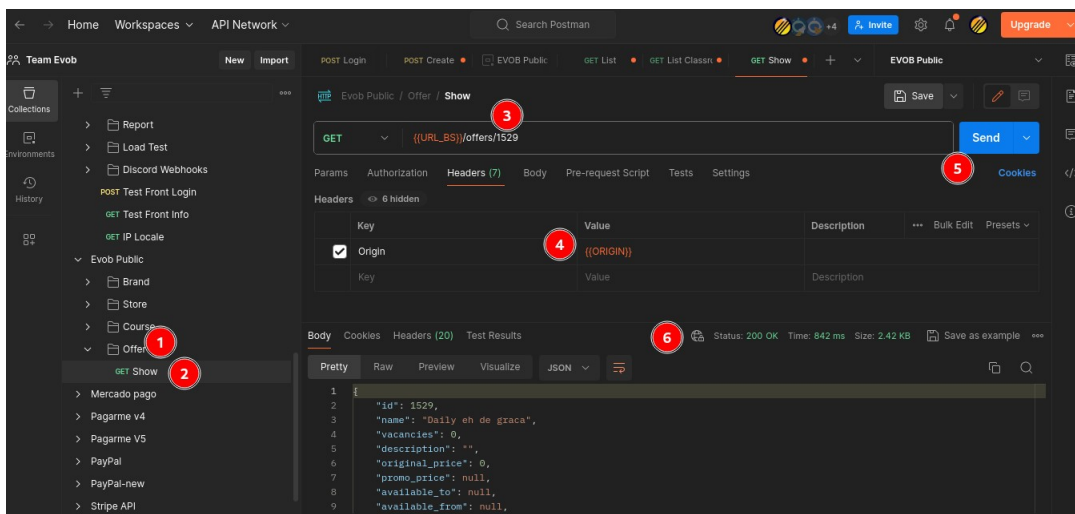
2.8 – Exibição de Offer (Oferta)

Offer (Oferta) é responsável por gerenciar a forma de se comercializar determinada Turma. Possibilitando gerenciar a quantidade de vagas disponíveis, período liberado para receber matrículas, assim como as formas de pagamento. Existem Ofertas do tipo Paga e Gratuita.

Exemplo de Requisição:**GET:** {{URL_BS}}/offers/:id_offer

Com o Header (cabeçalho) de requisição Obrigatório: Origin
Aonde: *id_offer* representa a Oferta que desejamos consultar.

- 1 – Dentro da Coleção **Evob Public**, clique na pasta **Offer**;
- 2 – Selecione a Requisição **Show**;
- 3 – Insira o ID da Oferta que deseja visualizar;
- 4 – Certifique-se de passar o **Header** da Origin da sua Plataforma;
- 5 – Clique no botão **Send** (Enviar);
- 6 – Observe o resultado da resposta recebida da API.



Resultado Esperado:

HTTP status code: 200

Body: *json data*

Detalhamento de Campos:

"vacancies": vagas (0 para infinito)

"original_price": valor da oferta (em centavos)

"promo_price": valor promocional (em centavos)

"tickets_used": nº de matrículas já realizadas

"starts_at": data de início período de matrículas

"ends_at": data de término período de matrículas

"type_offer": free (gratuita) ou subscription/standard (Paga),

"available_payment_methods": formas de pgto disponíveis

"currency": Moeda (padrão BRL)

"installments": parcelamento (caso disponível)

"classroom": dados como nomes da loja. Plataforma e curso

"validate_status": valid ou invalid

"validate_reason": motivo de retornar invalid

URL Parâmetros Requests disponíveis:

- Quantidade:

{{URL_BS}}/offers/1529?qty=2

Retorna os valores e parcelamentos multiplicando, assim como valida vagas disponíveis pela quantidade solicitada.

- Cupom de Desconto:

{{URL_BS}}/offers/1529?coupon_code=123

Valida Cupom de Desconto retornando as chaves:

"coupon_status": "valid" ou "invalid"

"coupon_reason": motivo de retonar invalid. Ex: "coupon not found for this course",

2.9 – Listagem de Banners

Banners são as imagens de publicidade pertencentes a uma determinada **Brand** (Plataforma).

Deste modo uma Plataforma pode ter diversos Banners, com publicidade para determinado curso, evento ou ação direcionada.

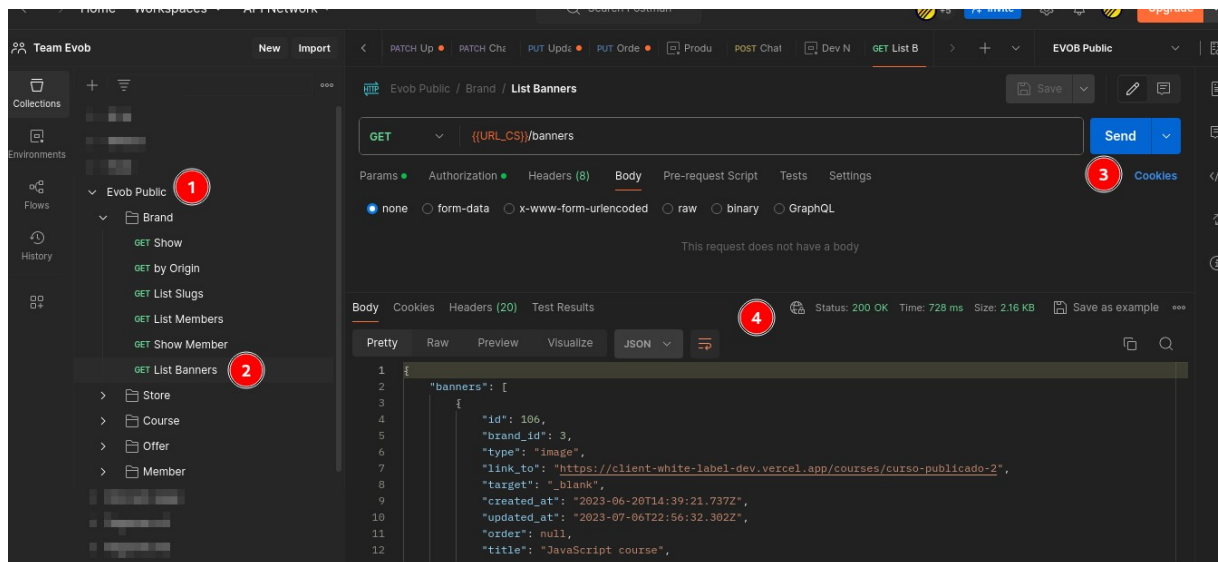
Para consultar quais Banners estão disponíveis para a Plataforma, realizamos a requisição de listagem de Banners.

Exemplo de Requisição:

GET: `{{URL_CS}}/banners`

Com o Header (cabeçalho) de requisição Obrigatório: Origin

- 1 – Dentro da Coleção **Evob Public**, clique na pasta **Brand**;
- 2 – Selecione a Requisição **List Banners**
- 3 – Clique no botão **Send** (Enviar);
- 4 – Observe o resultado da resposta recebida da API.



Resultado Esperado:

HTTP status code: 200

Body: json data

Opções de Filtros Disponíveis através de URL Parâmetros Requests:

a-) Por data de Criação:

{{URL_CS}}/banners?q[created_at_gteq]=2021-10-01T00:00:00.000Z&q[created_at_lteq]=2021-11-01T00:00:00.000Z

Aonde:

q[created_at_gteq] = Data Inicial

q[created_at_lteq] = Data Final

b-) Registros por Página:

{{URL_CS}}/banners?per_page=12

** não informado este parâmetro o sistema sempre retorna de 10 em 10 registros.*

c-) Paginando e Ordenando:

{{URL_CS}}/banners?page=3&sort_column=created_at&sort_direction=desc