



# Como acessar dados em APIs públicas?

## Um tutorial para usuários não técnicos

América Aberta 2024

Este workshop visa apresentar os conceitos fundamentais sobre APIs e explorar, com exemplos práticos, seu funcionamento.



Este trabalho está sob a licença [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/). Mediante atribuição de crédito à organização autora, pode ser copiado e redistribuído em qualquer suporte ou formato; remixado e adaptado para qualquer fim, inclusive comercial (nestes casos, as alterações feitas devem ser indicadas).

## Introdução

É comum que órgãos públicos disponibilizem dados através de APIs (Application Programming Interface), o que nem sempre é de fácil acesso e uso pelos usuários, em especial usuários não programadores. Pensando nisso, este workshop visa apresentar os conceitos fundamentais sobre APIs e explorar, com exemplos práticos, seu funcionamento. Ao final, os participantes estarão aptos a realizar consultas e estarão a par dos erros mais comuns que podem surgir nesse processo.

Os tópicos abordados neste documento estão listados abaixo.

[Introdução](#)

[Conceitos Fundamentais sobre APIs](#)

[Explorando uma API](#)

[Consultando APIs: Mãos na Massa](#)

[Erros Comuns e Como Resolvê-los](#)

[Recursos para explorar mais](#)

Neste workshop iremos usar como exemplo prático a API de integração do Portal Nacional de Contratações Públicas (PNCP), disponível [neste endereço](#). O manual de uso da API está disponível [nesta página](#) e serve como material de apoio.

## Conceitos Fundamentais sobre APIs

### O que é uma API?

O termo **API** é um acrônimo para **Application Programming Interface** (Interface de Programação de Aplicações). Uma API é um tipo de software que provê um conjunto de operações (chamadas **métodos**) que deverão ser executadas em algum sistema. Um uso comum para APIs é prover operações de

**consulta, adição, atualização e remoção** de dados em um banco de dados. Desta forma, um usuário poderia acessar uma base de dados de forma indireta, controlada e segura.

## Por que é comum fazer uso de APIs para disponibilizar dados?

O uso de APIs é comum e se tornou um padrão no desenvolvimento de software pois estas interfaces oferecem diversos benefícios tanto para quem disponibiliza os dados quanto para quem os consome. Exemplos destes benefícios são:

- **Atualização em tempo real** - As APIs operam diretamente sobre a base de dados, retornando os dados mais atuais.
- **Facilidade na documentação e suporte** - As APIs geralmente vem acompanhadas de documentação técnica, que facilita o seu uso.
- **Controle e segurança** - Os provedores de dados podem usar autenticação, autorização e registro de logs para controlar quem acessa os dados. Além disso, as operações possíveis são predefinidas, limitando o poder de acesso dos usuários.

## Principais componentes de uma API

**Recursos** - É qualquer objeto que possa ser acessado, manipulado e com o qual se possa interagir através dos métodos da API. Exemplos de recursos da API do PNCP são usuários, planos de contratação e contratações.

**Endpoints** - É uma URL que serve como ponto de contato entre o cliente (usuário da API) e o servidor (provedor de dados). Através de um endpoint o cliente envia uma requisição para o servidor com o intuito de acessar algum dado ou funcionalidade da API. Cada método (operação) da API é executado através de um endpoint.

**HTTP verbs (Métodos de requisição HTTP)** - O protocolo REST é um padrão de arquitetura de sistemas web amplamente utilizado. Este protocolo define 5 tipos de interações entre um cliente e os recursos de um servidor, conhecidos como verbos HTTP. São eles:

- **GET** (obter) - Utilizado quando se deseja **consultar** recursos.
- **POST** (publicar) - Utilizado quando se deseja **criar** novos recursos na base de dados.
- **PUT** (pôr) - Utilizado quando se deseja **substituir** um recurso existente por outro.
- **PATCH** (remendar) - Utilizado quando se deseja **alterar** um recurso existente.
- **DELETE** (apagar) - Utilizado quando se deseja **apagar** recursos da base de dados.

**Parâmetros** - São variáveis passadas aos métodos para especificar a operação a ser realizada. Por exemplo, o método [consultar contratação](#) da API do PNCP possui 3 parâmetros:

- **cnpj** - o CNPJ do órgão cuja contratação se deseja acessar.
- **ano** - o ano em que a contratação foi realizada.
- **sequencial** - o número que identifica unicamente a contratação daquele órgão naquele ano.

Neste endpoint, todos estes parâmetros são obrigatórios, isto é, o usuário precisa fornecer estas informações para que a operação de consulta seja realizada. Porém, parâmetros também podem ser opcionais.

**Paginação** - Às vezes um endpoint retorna muitos dados em uma única consulta e para evitar sobrecarregar o servidor este resultado é dividido em várias partes (páginas). Para acessar estas diferentes partes, o usuário deve passar como parâmetro o **número da página** que deseja acessar e geralmente é possível especificar o **tamanho da página**, ou seja, a quantidade de elementos retornados em cada página.

**Respostas em JSON** - JSON é um formato de arquivo assim como .xlsx (planilha excel), pdf, entre outros. Arquivos JSON são amplamente utilizados para transmitir informações em aplicações web e são o formato padrão de resultados dos endpoints de uma API. Um arquivo JSON é composto por **objetos**, **atributos** de objetos e **coleções** de objetos. Tomando como exemplo o resultado [deste endpoint da API do PNCP](#), identificamos os objetos como qualquer coisa

circundada por chaves “{ }”. Os atributos vêm antes dos dois pontos e os seus valores vêm após os dois pontos. As coleções são definidas por colchetes “[ ]”.

## Explorando uma API

### Como inserir parâmetros diretamente na URL?

Iremos explorar 2 tipos de parâmetros que aparecem em um API, são eles:

- **Path/route parameters** (parâmetros de caminho ou rota) - São os parâmetros que aparecem dentro da URL entre as barras simples. Por exemplo, no endpoint [consultar contratação](#), há 3 parâmetros de rota, rodeados por chaves “{ }”: `cnpj`, `ano` e `sequencial`. Estes parâmetros são sempre obrigatórios.

```
GET /v1/orgaos/{cnpj}/compras/{ano}/{sequencial} Consultar Contratação
```

- **Query parameters** (parâmetros de consulta) - São parâmetros opcionais que aparecem sempre após um ponto de interrogação (?) na URL do endpoint. O símbolo ? delimita o fim dos parâmetros obrigatórios e o início dos parâmetros opcionais. Por exemplo, [nesta consulta da API do PNCP](#), os parâmetros de consulta “categoria”, “pagina” e “tamanhoPagina” aparecem após o ?. Vale notar que o nome do parâmetro aparece na URL sucedido por um “=” e o valor do parâmetro. Os parâmetros de consulta são separados por um símbolo “&”.

### Como encontrar uma API por trás de um site?

Podemos inspecionar a rede de requisições de uma determinada aplicação web para descobrir a API que fornece os dados para a aplicação. Siga os passos abaixo para descobrir a API do PNCP a partir de uma consulta na interface do portal.

## Realizando uma consulta

1. Acesse a [página de consulta a Planos de Contratações Anuais](#) do PNCP.
2. No filtro `ano`, selecione 2024.
3. No filtro `órgãos`, pesquise e selecione “INSTITUTO FEDERAL DE EDUCACAO, CIENCIA E TECNOLOGIA DA PARAIBA”
4. Clique em `pesquisar`.
5. Clique no PCA que apareceu como resultado.

## Inspecionando a rede de requisições

6. Clique com o botão direito do mouse em algum lugar da página e selecione `inspecionar`.
7. No painel que abriu, selecione a aba `network` (rede).
8. Recarregue a página para capturar todas as requisições feitas pelo site.
9. Use o campo de filtro para procurar por tipos específicos de requisições:
  - a. `XHR` ou `Fetch`: Requisições de APIs geralmente aparecem aqui.
  - b. `JSON`: APIs retornam dados no formato JSON, então você pode buscar por URLs que contenham essa extensão ou `application/json` no cabeçalho de resposta.
10. Ao navegar nos resultados verá que alguns possuem uma URL que se inicia como `https://pncp.gov.br/api/pncp/v1/...`
11. Copie e cole esta URL no navegador e obterá os dados que alimentam a página inspecionada.

## Consultando APIs: Mãos na Massa

Para praticar o que você aprendeu até agora, realize o exercício abaixo.

### Exercício 1 - Alterando consultas diretamente na URL

**PASSO 1:** Acesse a URL

<https://pncp.gov.br/api/pncp/v1/orgaos/10783898000175/pca/2025/1/itens?categoria=1&pagina=1&tamanhoPagina=2>

Este endpoint retorna itens do Plano Anual de Contratações (ano 2025) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (campus Itaporanga).

**PASSO 2:** Altere os seguintes parâmetros de consulta diretamente na URL e observe o resultado mudar:

1. Altere o tamanho da página para 10. O resultado agora deverá retornar uma coleção de 10 itens.
2. Altere a categoria para 2, que é o código para a categoria "Serviço". Os itens retornados agora devem estar filtrados pela categoria 2 (serviço) ao invés de categoria 1 (material).
3. Remova o filtro de categoria e altere a página para 2. Os itens retornados agora são de ambas as categorias.

**PASSO 3:** Altere parâmetros de rota para consultar outro PCA.

1. Remova os parâmetros opcionais e altere o ano para 2024. Os itens retornados agora são do PCA 2024 do mesmo órgão.
2. Altere o CNPJ para 10783898000175. O resultado agora deve ser referente ao PCA 2024 do IFECT-PB campus Guarabira.
3. Altere o ano para 2030 e veja o que acontece.

## Exercício 2 - Download em massa de dados usando código Python.

Para este exercício, iremos utilizar a linguagem de programação Python para automatizar a consulta a vários endpoints e salvar o resultado em um arquivo `.csv`. O código está disponível [neste notebook](#). Siga as instruções no notebook para executar o código.

## Erros Comuns e Como Resolvê-los

Às vezes uma requisição a um endpoint pode retornar um erro ao invés do resultado esperado. Todos os tipos de resposta a uma requisição, seja erros ou sucessos, são categorizados e padronizados. Uma ótima fonte de consulta a este catálogo de respostas é [www.httpstatus.com.br](http://www.httpstatus.com.br).

Abaixo listamos alguns erros comuns que podem ocorrer ao acessar uma API.

### Erros de autorização - 401 (unauthorized), 403 (forbidden)

Algumas APIs exigem uma identificação do usuário através de credenciais. Caso você esteja tentando acessar um recurso para o qual não tenha autorização de acesso, receberá um erro com código 401 (unauthorized) ou 403 (forbidden). Para resolver este tipo de erro, você precisa acessar o recurso utilizando credenciais que permitam o seu acesso. O manual de uso da API deve informar onde obter tais credenciais.

Para reproduzir este erro:

1. Acesse o [catálogo de APIs governamentais](#).
2. Acesse a API Benefícios Previdenciários.
3. Tente fazer uma consulta no endpoint de benefício.
4. Veja que um erro 403 foi retornado.

### Erro de formatação da consulta - 400 (bad request)

Este erro ocorre quando o servidor não conseguiu processar a solicitação porque está malformada ou contém parâmetros inválidos. Para resolver este erro, certifique-se que a URL está bem formatada e em acordo com a documentação da API.

Para reproduzir este erro:

1. Acesse a API do comprasGOV [neste endpoint](#).
2. Troque o valor do parâmetro `pagina` para "abc"

## Erro para “recurso não encontrado” - 404 (not found)

O recurso solicitado não foi encontrado no servidor. Neste caso, certifique-se que o identificador do recurso pesquisado está correto. Caso esteja, então o recurso realmente não existe.

Para reproduzir este erro:

1. Acesse a API do PNCP neste endpoint.
2. Troque o ano para 2030.
3. Veja que retornou um erro 404

Em seguida:

1. Acesse a API do comprasGOV neste endpoint.
2. Troque o valor do parâmetro `idCompra` para “abc”
3. Veja que retornou um resultado vazio.

A mensagem de erro a ser retornada depende de como a API foi implementada.

## Erros do lado do servidor (5xx)

Os erros do tipo 5xx são sempre erros do lado do servidor. Não há nada que o usuário possa fazer para resolver o problema. Abaixo listamos erros comuns deste tipo.

- **500 (Internal Server Error)** - Um erro genérico indicando que algo deu errado no servidor, mas sem detalhes específicos.
- **503 (Service Unavailable)** - O servidor está temporariamente indisponível devido a sobrecarga ou manutenção.

## Recursos para explorar mais

O [catálogo de APIs governamentais](#) é um repositório que lista várias APIs públicas. Vale a pena checar, mas saiba que muitas das APIs exigem um cadastro para se ter acesso aos recursos.

Existem vários recursos online disponíveis e gratuitos sobre como consultar APIs utilizando Python ou qualquer outra linguagem de programação. Deixamos como exemplo as páginas abaixo:

- [Consumindo API's com Python](#)
- [Primeiros passos com solicitações HTTP Python para APIs REST](#)
- [Python e APIs: conhecendo a biblioteca Requests](#)