

Obter conteúdo da URL / Get URL content

Bloco responsável por realizar uma requisição REST a partir do recurso (URL) informado.

Nesta página

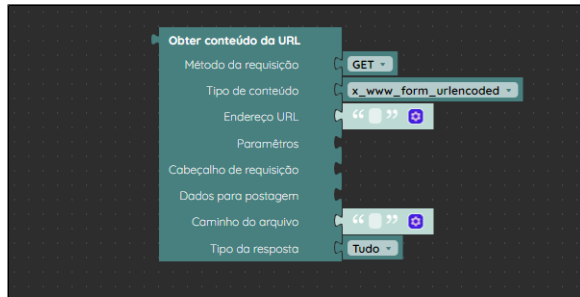


Figura 1 - Bloco Obter conteúdo da URL

- [Compatibilidade](#)
- [Retorno](#)
- [Parâmetros](#)
 - [Método da requisição](#)
 - [Tipo de conteúdo](#)
 - [Endereço URL](#)
 - [Parâmetro](#)
 - [Cabeçalho de requisição](#)
 - [Dados para a postagem](#)
 - [Caminho do arquivo](#)
 - [Tipo de resposta](#)
- [Exemplo 1](#)
- [Exemplo 2](#)
- [Exemplo 3](#)
- [Saiba mais](#)

Compatibilidade

- **Bloco servidor:** `cronapi.util.Operations.getURLFromOthers()`

Retorno

Retorna o conteúdo da requisição REST com base no que foi selecionado no parâmetro [Tipo de resposta](#).

Parâmetros

Descrição dos parâmetros do bloco.

A partir da versão do `cronapi-2.8.16`, o parâmetro **Parâmetros** passa a montar a *query string* e o parâmetro **Dados para postagem** passa a executar e montar os parâmetros no *post body*.

Método da requisição

Permite selecionar o verbo HTTP do recurso.

- **Posição:** 1
- **Inglês:** Request Method
- **Tipo:** Texto / String
- **Exemplo:** PUT
- **Opções:**
 - **GET:** solicita uma representação de dados de um recurso específico.
 - **POST:** submete dados a serem processados para um recurso específico.
 - **PUT:** usado para enviar dados ao servidor para criar ou atualizar um recurso específico. O corpo da requisição contém os dados a serem armazenados ou atualizados. Se o recurso não existir, ele pode ser criado; se já existir, ele será atualizado com os novos dados.
 - **DELETE:** apaga o recurso especificado.
 - **PATCH:** utilizado para aplicar modificações parciais em um recurso.
 - **HEAD:** solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta.
 - **OPTIONS:** usado para descrever as opções de comunicação com o recurso de destino.
 - **TRACE:** executa um teste de chamada *loop-back* junto com o caminho para o recurso de destino.

Acesse [Métodos de requisição HTTP](#) para maiores informações.

Tipo de conteúdo

Define o formato passado no corpo da requisição. Em geral, esse parâmetro é mais utilizado em requisições do tipo PUT e POST.

- **Posição:** 2
- **Inglês:** Content Type
- **Tipo:** Texto / String
- **Exemplo:** "x-www-form-urlencoded"
- **Opções:**
 - **JSON:** objeto JSON.
 - **x-www-form-urlencoded:** formato para codificação de pares chave-valor. Ex.: name=Pedro&lastName=Andrade
 - **multipart/form-data:** separa o conteúdo enviado em blocos de dados, utilizado em conjunto com o bloco [Objeto para form-Data](#).

Veja mais detalhes sobre os formatos "x-www-form-urlencoded" e "multipart/form-data" em [POST - HTTP](#).

Endereço URL

Endereço URI da requisição REST.

É comum utilizarmos o bloco [Criar texto com](#) para concatenar a URL base da API (ex: "[https://minhaapi.com/user/](#)") com o identificador do recurso obtido via parâmetro da função de bloco (ex: "345678").

- **Posição:** 3
- **Inglês:** URL address
- **Tipo:** Texto / String
- **Exemplo:** "https://minhaapi.com/user/345678/info"

Parâmetro

Parâmetros a serem utilizados como filtros da requisição. É necessário passar um mapa (ou objeto JSON) com atributo para cada parâmetro.

- **Posição:** 4
- **Inglês:** Parameters
- **Tipo:** Map / Mapa
- **Exemplo:** "{ \"id\" : \"345678\" }"

Cabeçalho de requisição

Objeto com atributos a serem tratados pelo serviço. É necessário passar um mapa (ou objeto JSON) com os atributos.

- **Posição:** 5
- **Inglês:** Headers
- **Tipo:** Map / Mapa
- **Exemplo:** "{ \"X-AUTH-TOKEN\" : \"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50a3kiLCJ1eWciOiJ0bm90cnkiLCJ1eWciOiJ0bm90cnkiLCJ1eWciOiJ0bm90cnki\" }"

Dados para a postagem

Conteúdo a ser adicionado no corpo na requisição. O formato é definido pelo recurso a ser requisitado, normalmente JSON.

Para o envio de arquivos, também é possível passar nesse parâmetro um mapa contendo os dados de um arquivo ou utilizar o bloco [Objeto para Form-Data / Form-Data Object](#).

- **Posição:** 6
- **Inglês:** Data to Post
- **Tipo:** Map / Mapa (JSON)
- **Exemplo:** "{ \"id\" : \"123\" , \"nome\" : \"Paulo\" }"

Caminho do arquivo

Endereço (caminho) onde o arquivo será salvo ao realizar uma requisição no formato **stream**. Deve-se informar o caminho com o nome e extensão do arquivo. É recomendado utilizar o caminho de uma pasta temporária da aplicação (exemplo [Pasta reciclável da aplicação](#)), evitando manter arquivos obtidos pela aplicação no servidor.

Esse campo deve ser utilizado quando o retorno for um *array* de bytes referente a um arquivo.

- **Posição:** 7
- **Inglês:** Path of the file
- **Tipo:** Texto / String
- **Exemplo:** `"/home/cronapp/project/src/"`

Tipo de resposta

Define o retorno do bloco.

- **Posição:** 8
- **Inglês:** Response
- **Tipo:** Texto / String
- **Exemplo:** `"body"`
- **Opções:**
 - **Tudo** (inglês: **All**): obtém todos os tipos de respostas em um único objeto, utilize o bloco **Obter item da resposta HTTP** para obter o body, stream e status do retorno desse objeto.
 - **Body:** corpo da requisição.
 - **Stream:** corpo da requisição em formato [Stream](#) (arquivos).
Ao selecionar essa opção, o bloco **Obter conteúdo da URL** não retornará valor. O retorno ocorrerá apenas no arquivo informado no parâmetro "Caminho do arquivo", seja o arquivo desejado ou alguma mensagem de erro gerado ao não encontrar o arquivo ou problema de autenticação. Veja [Exemplo 2](#).
 - **Status:** [código HTTP](#) da resposta.

Se o status da requisição for maior ou igual a 400, a opção "Body" retornará a mensagem de erro apenas se ela existir. Para retornar todo o erro, utilize a opção "Tudo" ou "Stream" no parâmetro **Tipo de resposta**.

Exemplo 1

A função abaixo inicia criando um mapa contendo o atributo "X-AUTH-TOKEN", usado para autenticação, na variável "autenticacao". Em seguida, faz uma requisição GET em um serviço web e utiliza a entrada **Parâmetros** para realizar um filtro do tipo *query string*, no parâmetro **Cabeçalho da requisição** inclui a variável de "autenticacao". O conteúdo obtido pela requisição é impresso no **Console do Depurador** (Figura 2).

Endereço montado na requisição da figura 1: `https://minhaapi.com.br/users?id=82`

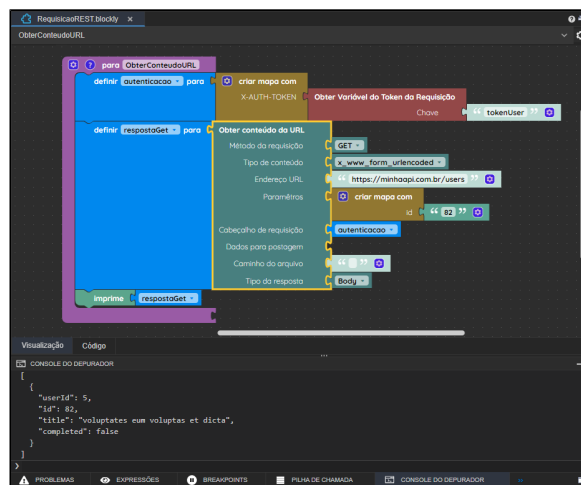


Figura 2 - Exemplo de requisição utilizando o bloco Obter conteúdo da URL

Exemplo 2

Também é possível realizar uma requisição que retorna um *array* de bytes de arquivo.

O exemplo abaixo faz uma requisição que retorna um arquivo (`logo.svg`), usamos o parâmetro **Caminho do arquivo** (destaque 1 da figura 3) para informar o local onde será salvo este conteúdo "Pasta da aplicação/Cronapp.svg" e definimos o **Tipo da resposta** como "Stream" (destaque 2 da figura 3). O resultado da execução do bloco pode ser visto no destaque 3 da figura 3.

No formato "Stream", o bloco **Obter conteúdo da URL** não retorna valores, apenas executa a ação e salva o arquivo obtido no endereço informado no parâmetro **Caminho do arquivo**.

Ao gerar algum erro na requisição, essa informação será salva no arquivo informado no parâmetro **Caminho do arquivo**. Nesses casos, recomendamos renomear a extensão do arquivo para `.txt` e abrir em um editor de texto para identificar o problema.

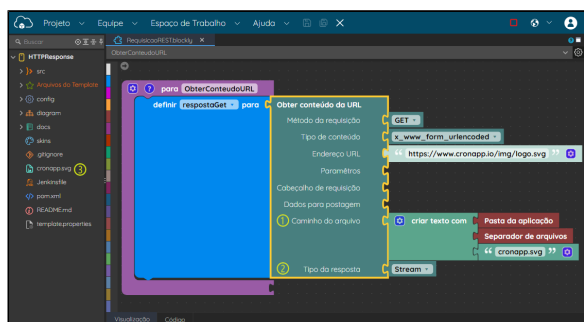


Figura 3 - Exemplo de requisição com *array* de bytes

Exemplo 3

Mostraremos agora como realizar uma requisição de envio (POST) com arquivos.

No exemplo da figura 4, obtemos um arquivo enviado pelo usuário da aplicação com o uso do bloco **Upload de arquivos** e encaminhamos esse arquivo para um recurso em uma API qualquer. Independente da configuração feita no parâmetro **Múltiplo** do bloco **Upload de arquivos** (destaque 1 da figura 4), ele sempre retornará uma lista com os endereços dos arquivos, dessa forma, utilizamos o bloco **Na lista** para obter o endereço desse *array* e incluí-lo em um mapa, cujo nome da chave foi criado como "file" (2). Foi necessário **criar um mapa** pois o parâmetro **Dados para postagem** sempre espera receber um mapa ou JSON.

Utilizamos o bloco **Obter conteúdo da URL** e configuramos o parâmetro **Método da requisição** como "POST", no **Endereço URL** informamos o recurso que espera receber o arquivo e no parâmetro **Dados para a postagem** incluímos o mapa com o arquivo obtido do bloco **Upload de arquivo**.

Se não ocorrer nenhum problema durante o processo de requisição, o retorno do bloco **Obter conteúdo da URL** será "200", pois o parâmetro **Tipo da resposta** está configurado com a opção "Status".

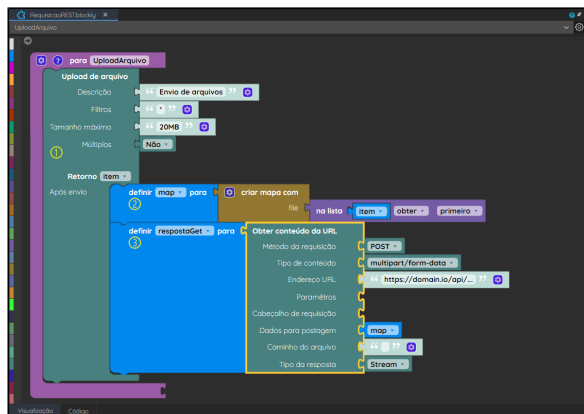



Figura 4 - Exemplo de requisição POST com o uso de arquivo

Também é possível enviar arquivos informando outros parâmetros de uma requisição "multipart/form-data".  acesse a documentação do bloco [Objeto para Form-Data](#) para mais detalhes.

Saiba mais

No Cronapp, requisições REST também podem ser tratadas pela Fonte de dados, acesse a documentação [Fonte de dados tipo Web Service \(REST / SOAP\)](#) para mais detalhes.