

Acesso automatizado por sistemas externos

Com fundamentação no Art. 8º, §3º, III da Lei 12527/2011 – Lei nº12.527/2011 – LAI e Art. 3º, XXV da Lei 14.129/2021, segue:

Os dados divulgados nos portais públicos (transparência ativa) devem ser legíveis por máquina e em formato aberto, disponibilizados por meio de uma Interface de Programa de Aplicativos (do inglês, “Application Programming Interface”), “API”.

Disponibilizamos a título de exemplo a seguinte estrutura de dados JSON referentes a licitações.

Endereço de acesso aos dados: <https://braganca.sp.gov.br/api/licitacao/listar>

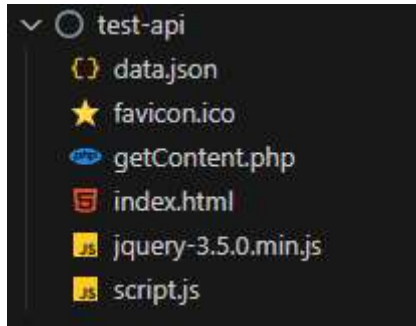
Retorno em MIME type “application/json” padrão.

Estrutura dos dados retornados: Formato JSON – array de objetos

```
[
  {
    "DataHoraPublicacao": "02/07/2024 às 08:30",
    "DataHoraSessao": "18/07/2024 às 08:00",
    "Modalidade": "Dispensa",
    "Objeto": "AQUISIÇÃO DE MATERIAL DE PINTURA...",
    "Processo": "18079",
    "Setor": "Serviços",
    "Status": "Aberto",
    "Titulo": "PROCESSO DE DISPENSA N.º 00417/2024",
    "ValorContratado": "0,00",
    "ValorEstimado": "0,00"
  },
  {
    "DataHoraPublicacao": "30/06/2023 às 00:00",
    "DataHoraSessao": "29/11/2023 às 18:00",
    "Modalidade": "Chamada Pública",
    "Objeto": "CHAMAMENTO PÚBLICO PARA HABILITAÇÃO E...",
    "Processo": "24389",
    "Setor": "Meio Ambiente",
    "Status": "Revogado",
    ....
  }
]
```

Exemplo para aplicação em HTML/JavaScript/PHP:

- Estrutura de arquivos utilizados neste exemplo:



** Utilização de jQuery apenas para exemplo.

- Arquivo “**getContent.php**” - Obtenção dos dados através do endereço da API.

```
<?php
header('Content-type: application/json');
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET");
header("Access-Control-Allow-Methods: GET, OPTIONS");
header("Access-Control-Allow-Headers: Content-Type, Content-Length, Accept-Encoding");

$url = $_POST['url'];
echo file_get_contents($url);
```

- Arquivo “**index.html**”:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Exemplo de acesso a API</title>
  </head>
  <body>
    <div id="resultSimples"></div>
    <div id="resultComplexo"></div>
    <div id="resultResponse"></div>

    <table>
      <thead></thead>
    </table>

    <script src="jquery-3.5.0.min.js"></script>
    <script src="script.js"></script>
  </body>
</html>
```

- Arquivo “script.js”:

A. Acesso aos dados.

```
// Endereço da API de origem dos dados
var endereco = "https://braganca.sp.gov.br/api/licitacao/listar";

// Definição da chamada ajax para obter os dados
var settings = {
  url: "getContent.php",
  method: "POST",
  timeout: 0,
  headers: {},
  data: {
    url: endereco,
  },
};

// Chamada ajax para obtenção dos dados.
$.ajax(settings)
  .done(function (response) {
    console.log("Dados completos", response);

    // Exemplo 1 - Filtro simples;
    exemploFiltroSimples(response);

    exemploFiltroComplexo(response);

    result = "<H3>Dados completos retornados pela chamada ajax</H3> ";
    result += montarTabela(response);
    $("#resultResponse").html(result);
  })
  .fail(function (jqXHR, exception) {
    var msg = "";
    if (jqXHR.status === "(failed)net::ERR_INTERNET_DISCONNECTED") {
      msg = "Uncaught Error.\n" + jqXHR.responseText;
    }
    if (jqXHR.status === 0) {
      msg = "Not connect.\n Verify Network.";
    } else if (jqXHR.status == 413) {
      msg = "Image size is too large.";
    } else if (jqXHR.status == 404) {
      msg = "Requested page not found. [404]";
    } else if (jqXHR.status == 405) {
      msg = "Image size is too large.";
    } else if (jqXHR.status == 500) {
      msg = "Internal Server Error [500].";
    } else if (exception === "parsererror") {
      msg = "Requested JSON parse failed.";
    } else if (exception === "timeout") {
      msg = "Time out error.";
    } else if (exception === "abort") {
      msg = "Ajax request aborted.";
    } else {
      msg = "Uncaught Error.\n" + jqXHR.responseText;
    }
    console.log(msg);
  });
```

B. Exemplo de uso com filtro simples:

```
function exemploFiltroSimples(dados) {
  // Filtros de dados Exemplo 1
  var filters = {
    Setor: "Esportes e Lazer",
    Modalidade: "Dispensa",
  };

  result = "<H3>Filtrado exemplo Simples: </h3> <h4>";
  $.each(filters, function (key, val) {
    result += key + ": " + val + ", <br>";
  });
  result += "</h4>";

  // Usando filtro de exemplo 1
  dadosFiltrados = $(dados).filter(function (i, item) {
    return (
      item.Modalidade == filters.Modalidade && item.Setor == filters.Setor
    );
  });
  result += montarTabela(dadosFiltrados);
  $("#resultSimples").html(result);

  console.log("Filtro exemplo simples", filters);
  console.log("Dados filtrados", dadosFiltrados);
}
```

C. Exemplo de uso com filtro complexo:

```
function exemploFiltroComplexo(dados) {
  // Filtros de dados Exemplo 2
  // os valores podem ser expressos como string ou array de strings
  var filters = {
    // array de valores a serem retornados para o campo "Setor"
    Setor: ["Esportes e Lazer", "Civil", "Saúde"],
    // String para retornar uma modalidade específica
    Modalidade: "Dispensa",
    // array para data a ser retornada.
    DataHoraPublicacao: ["20/06/2024", "25/06/2024", "01/07/2024"],
  };

  result = "<H3>Filtrado exemplo Complexo: </h3> <h4>";
  $.each(filters, function (key, val) {
    result += key + ": " + val.toString() + ", <br>";
  });
  result += "</h4>";

  dadosFiltrados = find_in_object(dados, filters);
  result += montarTabela(dadosFiltrados);
  $("#resultComplexo").html(result);

  console.log("Filtro exemplo complexo", filters);
  console.log("Dados filtrados", dadosFiltrados);
}
```

D. Funções de apoio para os exemplos

```
/**
 * Realiza o filtro nos dados com base em objeto chave -> valor onde os
 * valores podem ser cadeias simples ou array de valores
 * @param {json} my_array Objeto json a ser filtrado
 * @param {array|string} my_criteria valor a ser procurado
 * @returns {json} Objeto resultante;
 */
function find_in_object(my_array, my_criteria) {
    return my_array.filter(function (obj) {
        return Object.keys(my_criteria).every(function (key) {
            return (
                (Array.isArray(my_criteria[key]) &&
                    my_criteria[key].some(function (criteria) {
                        return obj[key].indexOf(criteria) > -1;
                    }))) ||
                obj[key] == my_criteria[key]
            );
        });
    });
}
```

E. Montagem de tabela html para exibir os dados

```
/**
 * Monta uma tabela html percorrendo os dados informados por parâmetro
 * @param {json} dados Informações a serem exibidas.
 * @returns String
 */
function montarTabela(dados) {
    var flag = false;
    var header = "";
    var result = "";
    $.each(dados, function (k, item) {
        result += "<tr>";
        $.each(item, function (key, val) {
            if (flag == false) {
                header += "<th>" + key + "</th>";
            }
            result += "<td>" + val + "</td>";
        });
        result += "</tr>";
        flag = true;
    });
    result =
        "<table border='1'><thead><tr>" +
        header +
        "</tr></thead><tbody>" +
        result +
        "</tbody></table>";
    return result;
}
```