

CLIENTES HTTP

Testando o ClientURL - CURL:

```
curl https://www.ifpr.edu.br/
curl -L https://www.ifpr.edu.br/
curl -i https://www.ifpr.edu.br/ - requisição com os cabeçalhos
curl -I https://www.ifpr.edu.br/ - somente cabeçalhos
```

Requisição:

▼ Cabeçalhos de solicitação

```
:authority: www.ifpr.edu.br
:method: GET
:path: /
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,.
n/signed-exchange;v=b3;q=0.9
accept-encoding: gzip, deflate, br
accept-language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7
cache-control: max-age=0
cookie: _gid=GA1.3.1728568396.1651512343; _ga_Z8DEPESPKF=GS1.1.1651516034.2.1.1651516443.60; _ga=GA1.3.19
51512343
referer: https://www.google.com/
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
sec-fetch-dest: document
sec-fetch-mode: navigate
sec-fetch-site: cross-site
```

Resposta:

▼ Cabeçalhos de resposta

```
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Mon, 02 May 2022 18:43:04 GMT
link: <https://reitoria.ifpr.edu.br/wp-json/>; rel="https://api.w.org/"
server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16
vary: Accept-Encoding
x-powered-by: PHP/5.4.16
```

Fazendo Downloads

```
curl -O https://releases.ubuntu.com/20.04.1/ubuntu-20.04.1-desktop-amd64.iso
```

Caso um download seja interrompido, é possível continuá-lo usando a opção -C (mesmo que não tenha sido iniciado com o curl):

```
curl -C - -O https://releases.ubuntu.com/20.04.1/ubuntu-20.04.1-desktop-amd64.iso
```

CONSUMINDO API's

O curl também é útil para realizar testes de API's REST. Tomemos como exemplo a aplicação <https://jsonplaceholder.typicode.com/> que foi feita justamente para fins de testes.

Ela possui as seguintes rotas:

```
GET /posts
GET /posts/:id
GET /posts/:id/comments
GET /comments?postId={:id}
POST /posts
PUT /posts/:id
PATCH /posts/:id
DELETE /posts/:id
```

Por padrão, o curl usa o método HTTP GET para realizar as requisições. Para fazer uma requisição GET, basta executar

```
curl https://jsonplaceholder.typicode.com/posts
```

Também é possível especificar qual método usar, passando a opção -X e o método como argumento:

```
curl -X GET https://jsonplaceholder.typicode.com/posts
```

Para realizar a criação de um post, você pode executar:

```
curl -i -X POST https://jsonplaceholder.typicode.com/comments \ -H
'Content-Type: application/json' \ -d ${'
  "postId": 101,
  "name"   : "Jefferson Chaves",
  "email": "jefferson.chaves@ifpr.edu.br",
  "body": "De acordo ao manual do curl ( disponível online [...])"
}'
```

A **opção -H** permite adicionar um cabeçalho e a **-d** permite passar os dados da requisição. Nesse caso, definimos que o conteúdo é no formato JSON e passamos o conteúdo da requisição que será usado para criar uma conta. Podemos ver no resultado que o status foi 201, indicando que um novo recurso foi criado.

Onde encontrar uma lista de comandos:

<https://gist.github.com/subfuzion/08c5d85437d5d4f00e58>

OUTROS EXEMPLOS

```
curl -X OPTIONS http://example.org/ -i
```

```
curl -X OPTIONS http://example.org -i
```

```
curl -X
```

```
curl -d "param1=value1&param2=value2" -H "Content-Type:
application/x-www-form-urlencoded" -X POST http://example.org -i
```

```
curl -i https://viacep.com.br/ws/85867295/json/
```

FERRAMENTAS

Postman

<https://redbot.org/?uri=http%3A%2F%2Fexample.org>

Teste de ferramentas de rede do Firefox

https://developer.mozilla.org/pt-BR/docs/Tools/Network_Monitor

Sites para testar

<http://example.org>

<https://jsonplaceholder.typicode.com/>

<https://pokeapi.co/>

<https://foz.ifpr.edu.br/wp-json>

Source Code for RequestHeader Example

```
public class RequestHeaderExample extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Enumeration e = request.getHeaderNames();

        while (e.hasMoreElements()) {
            String name = (String)e.nextElement();
            String value = request.getHeader(name);
            out.println(name + " = " + value);
        }
    }
}
```

Source Code for Request Info Example

```
public class RequestInfo extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body>");
        out.println("<head>");
        out.println("<title>Request Information Example</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h3>Request Information Example</h3>");
        out.println("Method: " + request.getMethod());
        out.println("Request URI: " + request.getRequestURI());
        out.println("Protocol: " + request.getProtocol());
        out.println("PathInfo: " + request.getPathInfo());
        out.println("Remote Address: " + request.getRemoteAddr());
        out.println("</body>");
        out.println("</html>");
    }
}
```

Source Code for Request Parameter Example

```
public class RequestParamExample extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Request Parameters Example</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h3>Request Parameters Example</h3>");
        out.println("Parameters in this request:<br>");

        String firstName = request.getParameter("firstname");
        String lastName = request.getParameter("lastname");

        if (firstName != null || lastName != null) {
            out.println("First Name:");
            out.println(" = " + HTMLFilter.filter(firstName) + "<br>");
            out.println("Last Name:");
            out.println(" = " + HTMLFilter.filter(lastName));
        } else {
            out.println("No Parameters, Please enter some");
        }
        out.println("<P>");
        out.print("<form action=\"");
        out.print("RequestParamExample\" ");
        out.println("method=POST>");
        out.println("First Name:");
        out.println("<input type=text size=20 name=firstname>");
        out.println("<br>");
        out.println("Last Name:");
        out.println("<input type=text size=20 name=lastname>");
        out.println("<br>");
        out.println("<input type=submit>");
        out.println("</form>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

```
response.setContentType("text/plain"); //Set content type of the  
response so that jQuery knows what it can expect.  
  
response.setCharacterEncoding("UTF-8"); //You want world domination?
```