

XRequest

XRequest

XRequest allows the developer to make an HTTP request to external (3rd party) APIs. The `<XRequest>` tag can be placed anywhere inside the `<Request>` tag.

```
<Request method="GET">
  <Sql id="allMovies" when="$q eq 1">
    SELECT * FROM movie
  </Sql>
  <XRequest id="testReq1" when="$q eq 2" url="http://localhost:7000/testx/v1.1/movies"
    method="GET" output="true">
    <Header name="Accept" value="application/json" />
  </XRequest>
</Request>
```

Attributes

1. **id** Used for uniquely identifying the XRequest
2. **url** Represents the url endpoint to which the HTTP request is to be made
3. **method** The HTTP method (GET/POST/PUT/DELETE)
4. **output** Boolean determining whether the response is to be returned (true/false/headers)
5. **classname** Name of post processable class

By default output is set to false. This is to protect the output being printed in the response.

Header and Param tags

`<Header>` tags represent the headers to be sent in the request. `<Param>` tags are used for sending query params in the request. Both of these tags are to be used inside the `<XRequest>` tag and have only two attributes

1. **name** Represents name of the Header/Param
2. **value** Represents its value

```
<XRequest id="testReq2" url="http://localhost:7000/testx/v1.1/movies"
          method="POST" output="true" >
  <Header name="Content-Type" value="application/x-www-urlencoded"/>
  <Param name="movie" value="The Godfather" />
  <Param name="rating" value="4" />
</XRequest>
```

When making post request with params, it is necessary to use Header `Content-Type: application/x-www-urlencoded`.

Body tag

`<Body>` can also be used inside `<XRequest>` representing the exact request body to be sent. This is helpful when making requests with json body.

Example POST request with JSON

```
<XRequest id="testReq2" url="http://localhost:7000/testx/v1.1/movies"
          method="POST" output="true" >
  <Header name="Content-Type" value="application/json"/>
  <Body>
    {
      "p": "The Godfather",
      "r": "4"
    }
  </Body>
</XRequest>
```

Example PUT Request with JSON body

```
<XRequest when="$q eq 5" id="testReq4" url="http://localhost:7000/testx/v1.1/movies/20"
          method="PUT" output="true" >
  <Header name="Content-Type" value="application/json"/>
  <Body>
    {
      "rating": "3"
    }
  </Body>
</XRequest>
```

Example DELETE Request

```
<XRequest id="testReq3" url="http://localhost:7000/testx/v1.1/movies/3"
          method="DELETE" output="true" >
</XRequest>
```

Note The HTTP methods used for `<Request>` and `<XRequest>` are independent of each other.

Request Parameters in XRequest

Request parameters can be accessed inside `XRequest` by using the `$` symbol. The following example shows the parameters `title` and `body` being used inside the `XRequest` body.

```
<Request method="POST">
  <XRequest id="ExampleRequest" url="https://fcm.googleapis.com/fcm/send"
    method="POST" output="true">
    <Header name="Content-Type" value="application/json"/>
    <Body>
      {
        "notification": {
          "title": "$title",
          "body": "$body"
        }
      }
    </Body>
  </XRequest>
</Request>
```

XRequest Output

Let's make XRequest to <https://jsonplaceholder.typicode.com/todos/1>

```
<Request method="GET">
  <Desc> Example XRequest persist </Desc>
  <XRequest id="testXReq" url="https://jsonplaceholder.typicode.com/todos/1"
    method="GET" output="true">
  </XRequest>
</Request>
```

The output of the above Request looks as follows

```
{
  "testXReq": {
    "id": 1,
    "completed": false,
    "title": "delectus aut autem",
    "userId": 1
  }
}
```

statusCode holds the HTTP status code received from the API and **body** consists of the actual payload.

Response Headers in Output

The response headers can be obtained by setting `output` attribute as `headers`.

```
<Request method="GET">
  <XRequest id="testXReq" url="https://jsonplaceholder.typicode.com/todos/1"
    method="GET" output="headers">
  </XRequest>
</Request>
```

This will return the response headers along with the response body as shown

```

{
  "testXReq": {
    "headers": {
      "Transfer-Encoding": "chunked",
      "Server": "cloudflare",
      "Date": "Mon, 11 Nov 2019 21:05:56 GMT",
      "Cache-Control": "private, max-age=14400",
      "Etag": "W/\\"53-hfEumeNh6YirfjyjaujcOPPT+s\\"",
      "Access-Control-Allow-Credentials": "true",
      "Expires": "-1",
      "Age": "2745",
      "Content-Type": "application/json; charset=utf-8",
      "X-Powered-By": "Express"
    },
    "body": {
      "id": 1,
      "completed": false,
      "title": "delectus aut autem",
      "userId": 1
    },
    "statusCode": 200
  }
}

```

Response processing

The response returned by `XRequest` can be given to a post processable class using the `classname` attribute.

```

<XRequest id="testReq" url="http://localhost:7000/testx/v1.1/movies"
method="GET" classname="com.metamug.plugin.ResponseExample" output="true">

```

The post processable class should implement the `ResponseProcessable` interface.

```

public class ResponseExample implements ResponseProcessable {

    @Override
    public Response process(Response response) throws Exception {
        JSONObject responseBody = (JSONObject)response.getPayload();
        // process response here

        return response;
    }
}

```