

Bruno Affiliates API Documentation

Table of Contents

[Authorization](#)

[Getting Lead Requests](#)

[Leads Registration](#)

[Getting Autologin URL](#)

[Getting Offer Advertisers](#)

[Getting Offers](#)

[Getting Leads](#)

[Getting Conversions](#)

[Get Income](#)

[Get Postbacks](#)

[Code Samples](#)

Authorization

The API requires authorization by API key. Each API key must be passed using "Api-Key" header with each request. In addition, the API requires IP whitelisting, so you need to provide your server's IP, prior using the API.

API end point: <https://brunoaff-api.org>

API key: [883CE1D5-9E00-CF5F-7D7C-7BBBCC827688](#)

Getting Lead Requests

* Service Action to retrieve Lead Requests *

GET to <https://brunoaff-api.org/api/v2/lead-requests>

Parameters:

Parameter	Description	Required
fromDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex. 2022-04-18 00:00:00)	yes
toDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex. 2022-04-18 23:59:59)	yes
page	Page number. Used for looping through pages.	no
itemsPerPage	Items on each page. Maximum 1000.	no
errorMessages[]	Filter by Error Messages. It accepts an array. Example: errorMessages[] = 1; errorMessages[] = 2;	no

verticals[]	Filter by Verticals. It accepts an array. Example: verticals[] = 1; verticals[] = 2;	no
-------------	---	----

Sample Request:

```
curl --location -g --request GET
'https://brunoaff-api.org/api/v2/lead-requests?fromDate=2024-01-18+00%3A00%3A00&toDate=2024-01-18+14%3A17%3A40' \
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "items": {
    "ID": 1,
    "email": "example@gmail.com",
    "firstName": "John",
    "lastName": "Doe",
    "phone": 35900000001,
    "brokerAccountApiID": 100,
    "status": 0,
    "timestamp": "2023-06-27 07:09:38",
    "advertiserResponseTime": "0.39969897270203",
    "response":
    "{\\"firstName\\":\\"John\\",\\"lastName\\":\\"Doe\\",\\"email\\":\\"example@gmail.com\\"}",
    "request":
    "{\\"firstName\\":\\"John\\",\\"lastName\\":\\"Doe\\",\\"email\\":\\"example@gmail.com\\"}",
    "brokerID": 1,
    "brokerName": "Example Advertiser Name",
    "friendlyName": "Example Advertiser Friendly Name",
    "description": null,
    "countryID": 1,
    "countryName": "United States",
    "ip": "198.168.0.1",
    "whitelabelID": 1,
    "whitelabelName": "Example Whitelabel Name",
    "brokerAccountID": 1,
    "brokerAccountName": "Example Advertiser Account Name",
    "platform": "Example Platform",
    "affiliateHash": "Example Affiliate Hash",
    "affiliateName": "John Doe",
    "affiliateEmail": "example@gmail.com",
    "osSystem": "Windows 10",
    "browser": "Chrome"
  },
  "total": {
    "pages": 27,
    "items": "27"
  },
  "message": null,
  "server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
  }
}
```

```
    "executionTime": 1.183722
  }
}
```

Leads Registration

POST to <https://brunoaff-api.org/api/v2/leads>

Parameters:

Parameter	Description	Required
email	string	yes
firstName	string	yes
lastName	string	yes
password	string	yes
ip	string	yes
phone	string	yes
areaCode	numeric - /area code of the phone/	no
custom1	string	no
custom2	string	no
custom3	string	no
custom4	string	no
custom5	string	no
comment	string	no
offerName	string	no
offerWebsite	string	no
locale	string	no

Note: In case you need to record an unique identifier for each lead, please store the **“leadRequest->ID”** value and use that to match the registration to the lead/conversions responses.

Sample Request:

```
curl --location -g --request POST 'https://brunoaff-api.org/api/v2/leads' \
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'email=johndor@email.com' \
--data-urlencode 'firstName=John' \
--data-urlencode 'lastName=Doe' \
--data-urlencode 'password=asd123ASD' \
--data-urlencode 'ip=1.1.1.1' \
--data-urlencode 'phone=303403403'
```

Sample Response:

```
{
  "details": {
    "leadRequest": {
      "ID": "0KMrZQ9bkLyDAYxql5n2EQBZoeYkN7eJ1Pog06mBdajzvEAp3"
    },
    "advertiser": {
      "name": "Broker Test",
      "logo": "/uploads/broker_logos/268ec14ab6f9479a99b79aba3776117e.jpg"
    },
    "offer": {
      "ID": "1",
      "name": "Offer Name Example",
      "hash": "26fd99c732"
    },
    "postbacks": [],
    "redirect": {
      "url": "https://example.com"
    }
  },
  "message": null,
  "server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
    "executionTime": 1.717107
  }
}
```

Getting Autologin URL

GET to <https://brunoaff-api.org/api/v2/brokers/login/details>

Parameters:

Parameter	Description	Required
leadRequestID	Must be the leadRequestID from the registration response.	yes

Note: By default API returns you the autologin redirect url in response after successful lead request. This method is used only if you need to get the actual autologin url after X time after the user has registered.

Sample Request:

```
curl --location -g --request GET
'https://brunoaff-api.org/api/v2/brokers/login/details?leadRequestID=fgf78df87g5867fd8gfd658g568fdg56df576g9'\
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "details": {
    "url": "https://example.com",
```

```

        "parameters": [],
        "fragment": null,
        "method": "GET"
    },
    "message": null,
    "server": {
        "date": "2024-01-18 14:17:40",
        "httpCode": 200,
        "executionTime": 0.785537
    }
}

```

Getting Offer Advertisers

* Service Action to retrieve Offer Advertisers *

GET to <https://brunoaff-api.org/api/v2/offers/advertisers/locations>

Parameters:

Parameter	Description	Required
ip	Filter by ip. Accepts string. Example: ip = 111.111.11.1; ip = 222.222.22.2;	no
userHash	Filter by Affiliate Hash. Accepts string. Example: userHash = ASDdd; userHash = ddASD;	no
activeOnly	Filter by Active advertisers. Accepts Integer. Example: activeOnly = 1; - Active activeOnly = 0; - Inactive	no
clickID	Filter by Click ID. Accepts integer.	no
countryCode	Filter by Country Code. Accepts String. Example: countryCode = US; countryCode = GB;	no
offerHash	Filter by Offer Hash. Accepts offerHash. Example: offerHash = ASD32dd; offerHash = ddA23SD;	no
offerName	Filter by Offer Name. Accepts String. Example: offerName = theOfferName;	no
offerWebsite	Filter by Offer Website. Accepts String. Example: offerWebsite = theOfferWebsite;	no

Sample Request:

```
curl --location -g --request GET 'https://brunoaff-api.org/api/v2/offers/advertisers/locations' \
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "data": {
    "items": [
      {
        "advertiserAccountID": "1",
        "advertiserAccountHash": "asd123",
        "advertiserID": "22",
        "advertiserHash": "HdsA12",
        "advertiserName": "JohnDoe",
        "advertiserLogo": "path/to/the/logo/image.png",
        "advertiserAccountOfferID": "3213",
        "advertiserAccountOfferHash": "asd3213sd",
        "link": "https://url.com",
        "custom1": null,
        "custom2": null,
        "custom3": null,
        "custom4": null,
        "custom5": null,
        "advertiserLogoBlackBackground": "path/to/the/logo/image.png",
        "advertiserAccountRegisterApi": true
      }
    ]
  },
  "message": null,
  "server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
    "executionTime": 0.827891
  }
}
```

Getting Offers

* Service Action to retrieve Offers *

GET to <https://brunoaff-api.org/api/v2/offers>

Parameters:

Parameter	Description	Required
search[campaignId]	Filter by Whitelabel. Accepts Integer.	no
search[campaignIds[]]	Filter by Whitelabel. It accepts an array. Example: search[campaignIds][] = 1; search[campaignIds][] = 2;	no
search[name]	Filter results by name	no

search[isExternal]	It's boolean param. It can be (0 for false or 1 for true)	no
page	Page number. Used for looping through pages.	no
itemPerPage	Items on each page. Maximum 1000.	no

Sample Request:

```
curl --location -g --request GET 'https://brunoaff-api.org/api/v2/offers' \
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "data": {
    "items": [
      {
        "id": 1,
        "name": "Offer Name",
        "hash": "cxted",
        "catalogLogoPath":
"https://exampleOffer.com/ca692ec6d1e0b575d63.png",
        "website": "https://exampleOffer.com/",
        "commissions": [
          {
            "commissionEvent": "Conversion",
            "countryName": "Singapore",
            "commisionAmount": "$100"
          }
        ],
        "verticals": [
          {
            "ID": "1",
            "name": "Offers"
          }
        ]
      }
    ],
    "page": 1,
    "totalItems": 1
  },
  "message": null,
  "server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
    "executionTime": 1.1293
  }
}
```

Getting Leads

* Service Action to retrieve Leads *

GET to <https://brunoaff-api.org/api/v2/leads>

Parameters:

Parameter	Description	Required
fromDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex. 2022-04-18 00:00:00)	yes
toDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex. 2022-04-18 23:59:59)	yes
fromTime	Time filter in format HH:mm:ss (ex. 00:00:00)	no
toTime	Time filter in format HH:mm:ss (ex. 23:59:59)	no
selectedDays[]	Array with Days. Number from 1-7: (where 1 is Sunday, 2 is Monday...) Example: selectedDays[] = 1; selectedDays[] = 2;	no
page	Page number. Used for looping through pages.	no
itemsPerPage	Items on each page. Maximum 1000.	no
offerIDs[]	Array Filter by offer IDs Example: offerIDs[] = 1; offerIDs[] = 2;	no
skip	It's boolean param. It can be (0 for Real or 1 for Test Leads)	no
custom1	Filter by Custom1. Accepts string.	no
custom2	Filter by Custom2. Accepts string.	no
custom3	Filter by Custom3. Accepts string.	no
custom4	Filter by Custom4. Accepts string.	no
custom5	Filter by Custom5. Accepts string.	no
os	Filter by Operation System. Accepts string.	no
customerID	Filter by Customer ID. Accepts integer.	no
saleStatus	Filter by Sale Status Raw. Accepts string.	no
countryGroupIDs[]	Filter by Country Group. It accepts an array. Example: countryGroupIDs[] = 1; countryGroupIDs[] = 2;	no
verticals[]	Filter by Verticals. It accepts an array. Example: verticals[] = 1; verticals[] = 2;	no
hasFTD	It's boolean param. It can be (0 for false or 1 for true)	no

Note: The “**leadRequestIDEncoded**” property uniquely identifies each lead (Take into consideration that it is case sensitive). Use that to match leads records to conversions

records.

Sample Request:

```
curl --location -g --request GET
'https://brunoaff-api.org/api/v2/leads?fromDate=2024-01-18+00%3A00%3A00&toDate=2024-01-18+14%3A17%3A40'\
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "items": [
    {
      "campaignName": "Example Campaign Name",
      "countryName": "United Kingdom",
      "custom1": null,
      "custom2": null,
      "custom3": null,
      "custom4": null,
      "custom5": null,
      "platform": "Windows 10",
      "browser": "Chrome",
      "signupDate": "2022-10-27 09:00:44",
      "isSmartClick": "0",
      "customerID": "exampleCustomer@gmail.com",
      "countryCode": "GB",
      "saleStatus": null,
      "campaignHash": "b",
      "leadRequestIDEncoded":
"aaaa3aaaaAAA15A4Aa9AaAa6aaaaaA20a7aaA6AaAaAaaAaj",
      "hasFTD": 1
    }
  ],
  "total": {
    "items": 1
  },
  "message": null,
  "server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
    "executionTime": 0.898717
  }
}
```

Getting Conversions

* Service Action to retrieve Conversions *
GET to <https://brunoaff-api.org/api/v2/conversions>

Parameters:

Parameter	Description	Required
-----------	-------------	----------

fromDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex.2019-04-18 00:00:00)	yes
toDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex.2019-04-18 23:59:59)	yes
fromTime	Time filter in format HH:mm:ss (ex. 00:00:00)	no
toTime	Time filter in format HH:mm:ss (ex. 23:59:59)	no
selectedDays[]	Array with Days. Number from 1-7 (where 1 is Sunday, 2 is Monday...) Example: selectedDays[] = 1; selectedDays[] = 2;	no
page	Page number. Used for looping through pages	no
itemPerPage	Items on each page. Maximum 1000	no
campaignIDs[]	Filter by Offer. It accepts an array. Example: campaignIDs[] = 1; campaignIDs[] = 2;	no
custom1	Filter by Custom1. Accepts string.	no
custom2	Filter by Custom2. Accepts string.	no
custom3	Filter by Custom3. Accepts string.	no
custom4	Filter by Custom4. Accepts string.	no
custom5	Filter by Custom5. Accepts string.	no
os	Filter by Operation System. Accepts string.	no
customerID	Filter by Customer ID. Accepts integer.	no
lateConversionHour	Int Filter by lateConversionHour	no

Note: The **“leadRequestIDEncoded”** property uniquely identifies each lead. Use that to match leads records to conversions records.

Sample Request:

```
curl --location -g --request GET
'https://brunoaff-api.org/api/v2/conversions?fromDate=2019-04-18+00%3A00%3A00&toDate=2019-04-18+23%3A59%3A59' \
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "items": [
    {
      "depositDate": "2022-11-03 14:17:36",
      "customerID": "exampleEmail2@gmail.com",
      "custom1": null,
```

```

        "custom2": null,
        "custom3": null,
        "custom4": null,
        "custom5": null,
        "platform": "Windows 10",
        "browser": "Chrome",
        "campaignName": "Example Campaign Name",
        "countryName": "United Kingdom",
        "referrer": null,
        "isSmartClick": "0",
        "countryCode": "GB",
        "campaignHash": "b",
        "leadRequestIDEncoded": "a1aa1a1aaAa1AAaAAaaaaaaaaAAaA"
    }
},
"total": {
    "items": "1"
},
"message": null,
"server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
    "executionTime": 1.783154
}
}

```

Get Income

* Service Action to retrieve Income *

GET to <https://brunoaff-api.org/api/v2/income>

Parameters:

Parameter	Description	Required
page	Page number. Used for looping through pages. (int)	no
itemsPerPage	Items on each page. Maximum 1000. (int)	no
fromDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex. 2019-04-18 00:00:00)	yes
toDate	Date filter in YYYY-MM-DD HH:mm:ss format (ex. 2019-04-18 23:59:59)	yes
affiliateID	Affiliate ID (int)	no
affiliateIDs[]	Affiliate IDs (array). Example: affiliateIDs[]=1 affiliateIDs[]=2	no
campaignID	Campaign ID (int)	no
campaignIDs[]	Campaign IDs (array). Example: campaignIDs[]=1 campaignIDs[]=2	no

brokerID	Broker ID (int)	no
qualified	Qualified (int)	no
skip	Skip (int)	no
brokerWorkingHours	Broker Working Hours (int)	no
brokerGroups[]	Broker Groups (array). Example: brokerGroups[]=1 brokerGroups[]=2	no
countryID	Country ID (int)	no
countryIDs[]	Country IDs (array). Example: countryIDs[]=1 countryIDs[]=2	no

Sample Request:

```
curl --location -g --request GET
'https://brunoaff-api.org/api/v2/income?fromDate=2022-01-09+00%3A00%3A00&toDate=2022-01-09+00%3A00%3A00' \
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "cost": 123.123,
  "income": 123.123,
  "message": null,
  "server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
    "executionTime": 1.427113
  }
}
```

Get Postbacks

* Service Action to retrieve Postbacks *

GET to <https://brunoaff-api.org/api/v2/postbacks>

Parameters:

Parameter	Description	Required
page	between 1 and ∞ . Default 1 (int)	no
itemsPerPage	Allowed values [5, 10, 15, 20, 25, 100, 250, 500, 750, 1000]. Default 25 (int)	no
whitelabelID	Whitelabel ID (int)	no

offerID	Offer ID (int)	no
type	Postback type (int): Signup (1),	no
title	Postback title (string)	no
status	Postback status (int)	no
affiliateHash	Affiliate Hash (string)	no
url	Postback URL (string)	no
postbackID	Postback ID (int)	no

Sample Request:

```
curl --location -g --request GET 'https://brunoaff-api.org/api/v2/postbacks' \
--header 'Api-Key: 883CE1D5-9E00-CF5F-7D7C-7BBBCC827688'
```

Sample Response:

```
{
  "items": [
    {
      "ID": 1,
      "projectID": 1,
      "whitelabelID": 1,
      "title": "Test",
      "type": 1,
      "url": "http://test.com",
      "check_condition": "test",
      "time_registered": "2020-01-09 13:32:34",
      "status": 1,
      "comment": "test",
      "amount": 1,
      "event": 1,
      "projectTitle": "Test",
      "hash": "test",
      "whitelabelName": "Test",
      "affiliateEmail": "test@test.com",
      "affiliateName": "Test",
      "affiliateHash": "test"
    }
  ],
  "total": {
    "items": 1
  },
  "message": null,
  "server": {
    "date": "2024-01-18 14:17:40",
    "httpCode": 200,
    "executionTime": 1.168598
  }
}
```

Code Samples

Registration

```
<?php
$endpoint = "https://brunoaff-api.org/api/v2/leads";
$api_key = "883CE1D5-9E00-CF5F-7D7C-7BBCC827688";
try {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $endpoint);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, [
        "Api-Key: $api_key",
    ]);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query([
        "firstName" => "Test",
        "lastName" => "Test",
        "email" => "test-random-email" . time() . "@mailinator.com",
        "password" => "123456Asd",
        "phone" => "2071566154",
        "areaCode" => "44",
        "ip" => "5.62.29.4", //you can use getUserIP() to get user`s IP
        "custom" => null, //custom fields for your own use
        "custom1" => null, //custom fields for your own use
        "custom2" => null, //custom fields for your own use
        "custom3" => null, //custom fields for your own use
        "custom4" => null, //custom fields for your own use
        "custom5" => null, //custom fields for your own use
        "comment" => null, //put here your comment
        "offerName" => null, //put your offer name
        "offerWebsite" => null, //put your domain
        "currency" => "USD",
        "locale" => "af_NA",
    ]));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $output = curl_exec($ch);
    if (empty($output)) {
        throw new \ErrorException("Can`t make request");
    }
    $response = json_decode($output);
    if (empty($response)) {
        throw new \ErrorException("Can`t parse request");
    }
    if (!in_array(curl_getinfo($ch, CURLINFO_HTTP_CODE), [200, 201])) {
        throw new \ErrorException($response->messages[0]);
    }
    //SUCCESS - Your code here
    echo "ok";
    curl_close($ch);
} catch (\ErrorException $exc) {
    http_response_code(400);
    echo $exc->getMessage();
}
```

Get User's IP

```
function getUserIP(){
    foreach (array("HTTP_LSWCDN_FORWARDED_FOR", "HTTP_CLIENT_IP",
"HTTP_X_FORWARDED_FOR",
        "HTTP_X_FORWARDED", "HTTP_X_CLUSTER_CLIENT_IP", "HTTP_FORWARDED_FOR",
"HTTP_FORWARDED",
        "REMOTE_ADDR") as $key) {
        if (array_key_exists($key, $_SERVER) === true) {
            foreach (explode(",", $_SERVER[$key]) as $ip) {
                if (filter_var($ip, FILTER_VALIDATE_IP,
FILTER_FLAG_NO_PRIV_RANGE |
                FILTER_FLAG_NO_RES_RANGE) !== false) {
                    return $ip;
                }
            }
        }
        foreach (explode(",", $_SERVER[$key]) as $ip) {
            if (filter_var($ip, FILTER_VALIDATE_IP) !== false) {
                return $ip;
            }
        }
    }
}
```

Client Autologin

```
<?php
//CHANGE THESE VARIABLES WITH THE DETAILS YOU RECEIVED FOR YOUR API
//DO NOT FORGET TO IMPLEMENT ERROR HANDLING BASED ON YOUR FLOW AND
REQUIREMENTS

$endpoint = "https://brunoaff-api.org";
$leadRequestID = "USER SIGNUP REQUEST ID HERE";
$apiKey = "883CE1D5-9E00-CF5F-7D7C-7BBBCC827688";
$curl = curl_init();

//THIS CALL MUST BE DONE ON YOUR SERVER SIDE IN ORDER TO HIDE YOUR API KEY
curl_setopt_array($curl, array(
    CURLOPT_URL =>
"$endpoint/api/v2/brokers/login/details?leadRequestID=$leadRequestID",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "Api-Key: $apiKey",
    ),
));
$response = curl_exec($curl);
    curl_close($curl);
?>
```

```

<!DOCTYPE html>
<html>
  <head>
  </head>
<body>
  <script>
    //DO NOT DIRECTLY CALL THE API FROM JAVASCRIPT SINCE THIS WILL EXPOSE
    YOUR API KEY
    var rawResponse = "<?php echo $response ?? \"\"; ?>";//WE INITIALIZE THE
    RESPONSE FROM THE
    SERVER SIDE SCRIPT
    if (rawResponse.length <= 0) {
      throw new Error("Something went wrong...");//YOUR ERROR HANDLING HERE
    }
    var response = JSON.parse(rawResponse);
    var url = response.data.url;
    if (response.data.method == "POST") { //HANDLING THE POST METHOD
    if (response.data.fragment) {
      url = url + "#" + response.data.fragment;
    }
    var formJS = createElement("form", "", {
      method: "POST",
      action: url,
    });
    for (var key in response.data.parameters) {
      var item = response.data.parameters[key];
      var input = createElement("input", "", {
        type: "hidden",
        name: key,
        value: item
      });
      appendChild(formJS, input);
    }
    document.body.appendChild(formJS);
    formJS.submit();
  } else {
    //HANDLING THE GET METHOD
    var parameters = [];
    for (var key in response.data.parameters) {
      parameters.push(key + "=" + response.data.parameters[key]);
    }
    if (url.indexOf("?") === -1 && parameters.length > 0) {

    }
    if (parameters.length > 0) {
      url += parameters.join("&");
    }
    if (response.data.fragment) {
      url = url + "#" + response.data.fragment;
    }
    window.location.href = url;
  }
  //UTILITY FUNCTIONS TO CREATE THE HTML ELEMENTS
  function createElement(tag, content, attributes) {
    if (typeof content === "undefined") {
      content = "";
    }
    if (typeof attributes === "undefined") {

```



```

        attributes = [];
    }
    var newElement = document.createElement(tag);
    newElement.innerHTML = content;
    Object.keys(attributes).forEach(function (key) {
        setAttribute(newElement, key, attributes[key]);
    });
    return newElement;
}
function setAttribute(selector, name, value) {
    if (!value) {
        value = "";
    }
    if (name) {
        findElement(selector).forEach(function (element) {
            element.setAttribute(name, value);
        });
    }
}
}
function appendChild(selector, html) {
    if (!html) {
        html = "";
    }
    findElement(selector).forEach(function (element) {
        if (typeof html === "string") {
            element.innerHTML = element.innerHTML + html;
        } else if (typeof html === "object") {
            element.appendChild(html);
        }
    });
}
function findElement(selector, element) {
    if (typeof element === "undefined") {
        element = document;
    }
    if (typeof selector === "string") {
        return
        Array.prototype.slice.call(element.querySelectorAll(selector));
    } else if (typeof selector instanceof Array) {
        return selector;
    } else if (typeof selector === "object") {
        return [selector];
    }
    return [];
}
}
<script>
<body>
<html>

```