|  |  |
| --- | --- |
| QUICK START API REST TANAGURU V0.1  Quickstart Api Tanaguru | Résumé  Use Api Tanaguru with a chrome extension  **Team Tanaguru** |

**Summary**

[Introduction 3](#_Toc471387957)

[AUTHENTICATION 3](#_Toc471387958)

[Step 1 : ENCODING THE Client'S KEY AND SECRET CODE 4](#_Toc471387959)

[Step 2 : Obtain a token 5](#_Toc471387960)

[Step 3 : AUTHENTIFY API REQUESTS WITH THE TOKEN 5](#_Toc471387961)

[CREDITS 6](#_Toc471387962)

[RETURN STATUS, LIST OF CODES 6](#_Toc471387963)

[LIST OF AVAILABLE ACTIONS 6](#_Toc471387964)

[LAUNCH AN AUDIT OF PAGE 6](#_Toc471387965)

[API USER STATISTICS 10](#_Toc471387966)

[REQUEST A TOKEN (AUTHENTICATION) 11](#_Toc471387967)

[INVALIDATE A TOKEN (AUTHENTICATION) 12](#_Toc471387968)

[CASES OF ERRORS 13](#_Toc471387969)

[Quick start Guide 14](#_Toc471387970)

[Installing a chrome extension 14](#_Toc471387971)

[Request token 14](#_Toc471387972)

[Invalidate a token 15](#_Toc471387973)

[Launch a page audit 16](#_Toc471387974)

[API USE STATISTICS 17](#_Toc471387975)

# Introduction

The API REST provides a convenient, powerful and simple Web services API to interact with the Audit Service of Tanaguru web pages. It has advantages such as great ease of integration and development. It is also an excellent solution for use with mobile and web applications.

The use of the API requires a basic knowledge of Web services.

API base URL: https://www.api.tanaguru.com/v1.0/service

All API calls begin with this URL with the path to the desired action.   
Sample : https://www.api.tanaguru.com/v1.0/service**/auditPage**

The API is accessible only with the Https protocol and accepts HTTP **POST and GET** requests. Data sent in POST must conform to the **JSON** format.

# **AUTHENTICATION**

The REST API uses the OAuth protocol to allow application users to access data securely without having to reveal user name and password.

Before making REST API calls, you must authenticate using OAuth 2.0. To do this, you will need the following:

* A client ID and secret code provided by Océane consulting.
* Description belowMake a POST security / authenticate request to exchange these authentication informations and generate an authentication token through the API. Then for all queries, you must specify in the authentication parameter on the query header the token as the value.

**Step 1 : Autentication**

1. Make a POST request on security / authenticate using the key and client code to request a token.
2. The Tanaguru API receives the request and generates a token and returns it to the client in JSON format.
3. The client recieves the token and parse it to retrieve the token value for use in step

**Step 2 : Calls to the API**

1. The client initiates a POST request on / service / auditPage with the token value as an identifier to start an audit and retrieve the results.
2. The Tanaguru API processes the query, starts the audit and returns the result data in JSON format to the client.
3. The customer receives the data and parses it for various uses.

**The tokens are passwords**

Keep in mind that the client ID, PIN, and token are required to access application-related requests. These values should be considered sensitive and just as passwords should not be shared or distributed to untrusted people.

**SSL absolutely necessary**

This authentication is only reliable when SSL is used. Therefore, all requests to obtain and use tokens must use HTTPS parameters, it is also a requirement of the API.

## Step 1 : ENCODING THE Client'S KEY AND SECRET CODE

Steps for encoding client ID and client secret code into a set of identification informations to obtain a token:

1. Concatenate the Client ID plus the character of two points " :", and the secret code of the client, all in a single chain.
2. Encoding in Base64 the chain obtained from the previous step.

Below are examples of values indicating the result of this algorithm. Note that the client secret code used for these samples is disabled and will not work for actual requests.

|  |  |
| --- | --- |
| Key | b9025c56b2425053dc069585390ab7c8 |
| Secret code | 10ee8cf96f9e35f7207a9a5cb3f89ed63c5f59692e1782d82c7eb73d21067695 |
| Concatenate | b9025c56b2425053dc069585390ab7c8 : 10ee8cf96f9e35f7207a9a5cb3f89ed63c5f59692e1782d82c7eb73d21067695 |
| Authentication identification informations (Base64) | YjkwMjVjNTZiMjQyNTA1M2RjMDY5NTg1MzkwYWI3Yzg6MTBlZThjZjk2ZjllMzVmNzIwN2E5YTVjYjNmODllZDYzYzVmNTk2OTJlMTc4MmQ4MmM3ZWI3M2QyMTA2NzY5NQ== |

## Step 2 : Obtain a token

The value calculated in step 1 must be exchanged against a token by issuing a POST / service / security / authenticate request:

* The request must be an HTTP POST request.
* The request must include an **Authorization** header with the base value : **Basic < coded value of step 1>**.
* The request must include a Content-Type header with the value application / json.
* The request must include a Grant\_type header: client\_credentials.

Sample request:

POST /v1.0/service/security/authenticate HTTP/1.1

Host: api.tanaguru.com

User-Agent: My App v1

Authorization: Basic YjkwMjVjNTZiMjQyNTA1M2RjMDY5NTg1MzkwYWI3Yzg6MTBlZThjZjk2ZjllMzVmNzIwN2E5YTVjYjNmODllZDYzYzVmNTk2OTJlMTc4MmQ4MmM3ZWI3M2QyMTA2NzY5NQ==

Content-Type: application/json

Grant\_type: client\_credentials

If the request has been formatted correctly, the server responds with a coded JSON payload:

Sample of answer:

HTTP/1.1 200 OK

Status: 200 OK

Content-Type: application/json; charset=utf-8

...

{‘’status’’: 200 ,’’token\_type’’: ‘’bearer’’ , ‘’access\_token’’: ‘’AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA%2FAAAAAAAAAAAA’’}

Requests must verify that the value associated with the token\_type key of the returned object is "**bearer**". The value associated with the access\_token key is the token.

Note that a token is valid for one application at a time. Redoing another request with the same identification informations will return the same token until it is invalidated.

## Step 3 : AUTHENTIFY API REQUESTS WITH THE TOKEN

The token is used to make calls to the API. To use the token, build a normal HTTPS request and include an **Authorization** header with the value of the token: **Bearer < Token value of step 2>**. Signature is not required.

Sample request:

GET /v1.0/service/limit\_stat HTTP/1.1

Host: api.tanaguru.com

User-Agent: My App v1.0

Authorization: Bearer AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA%2FAAAAAAAAAAAAAAA

# **CREDITS**

The Tanaguru API is based on a **credit system**.

When you run an audit from the API, and it generates a complete end result, your account is debited from a credit unit.

The other queries, (failed analysis, statistical information request, authentication, etc.) do not debit any credit.

# RETURN STATUS, LIST OF CODES

Each call to the API gives rise to a response returning a specific code according to the result obtained. Scanning this code ensures that the query has been successfully processed.

All codes >= 400 indicate that the request has not been processed successfully by our servers.

* **200**: OK
* **400**: Missing parameter, or incorrect value
* **401**: Authentication required (token not specified or invalid)
* **403**: Unauthorized action (out-of-print credits, unauthorized URL, etc.)
* **404**: Non-accessible page (URL unknown / not access the address)
* **406**: The JSON indicated in POST data is invalid
* **408**: Exceeding the maximum time allowed for audit
* **500**: Unknown Error, contact us
* **503**: The API is temporarily unavailable, try again in a few minutes

# LIST OF AVAILABLE ACTIONS

## LAUNCH AN AUDIT OF PAGE

**POST** https://www.api.tanaguru.com/v1.0/service/auditPage

**Request format**

**Headers**

**Authorization:** bearer eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI.oGxndCq3Ocz6CKi2aUb4uA

**Accept:** application/json; charset=utf-8

**Content-Type:** application/json

**Payload**

**{**

**"page\_url": "",**

**"referentiel ": "",**

**"level ": "",**

**"language": "",**

**"dt\_tbl\_marker": "" ,**

**"cplx\_tbl\_marker": "",**

**"pr\_tbl\_marker": "",**

**"dcr\_img\_marker": "",**

**"inf\_img\_marker": "",**

**"screen\_width ": 1920,**

**"screen\_height": 1080,**

**"description\_ref": true,**

**"html\_tags": false**

**}**

**Parameters**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Required** | **Type** | **Default value** | **Description** | **Available value** |
| Authorization | Yes | String | Any | Authentication token used | bearer <token value> |
| page\_url | Yes | String | Any | URL of the page to be audited | <http://www>. ….. |
| referentiel | No | String | RGAA30 | The accessibility guidelines to be used | RGAA32016, RGAA30, RGAA22, AW22 |
| level | No | String | AA | The level of accessibility to be used | A, AA, AAA, Bz, Or, Ar, LEVEL\_1, LEVEL\_2, LEVEL\_3 |
| language | No | String | All | Message language and audit result remarks | All, Fr, En, Es\_En, Es\_Fr |
| dt\_tbl\_marker | No | String | Any | HTML marker for data tables | Corresponds to the attribute "id", "class" or "role" of complex arrays. Several markers can be entered, separated by one ";" |
| cplx\_tbl\_marker | No | String | Any | HTML marker for complex tables |
| pr\_tbl\_marker | No | String | Any | HTML marker for presentation tables |
| dcr\_img\_marker | No | String | Any | Decorative image marker |
| inf\_img\_marker | No | String | Any | Marker of images with information |
| screen\_width | No | Number | 1920 | Width of screen in pixel | Maximum value is 2048 |
| screen\_height | No | Number | 1080 | Screen height in pixels | Maximum value is 2048 |
| description\_ref | No | Boolean | False | To provide or not the titles of the tests, criteria, theme of the repository. | True/False |
| html\_tags | No | Boolean | False | To provide remarks and titles encoded in HTML or not | True/False |

**Answer format**

{

"http\_status\_code": 200,

"url": "",

"status": "COMPLETED",

"score": 100,

"ref": "",

"level": "",

"language": "All",

"nb\_w3c\_invalidated": 0,

"nb\_passed": 0,

"nb\_failed": 0,

"nb\_not\_tested": 0,

"nb\_na": 0,

"nb\_failed\_occurences": 0,

"nb\_detected": 0,

"nb\_suspected": 0,

"nb\_nmi": 0,

"themes\_description\_en": {

"Rgaa30-13": "…",

…

},

"themes\_description\_fr": {

"Rgaa30-13": "…",

…

},

"themes\_description\_es": {

"Rgaa30-13": "…",

…

},

"criterions\_description\_en": {

"Rgaa30-12-11": "…",

…

},

"criterions\_description\_fr": {

"Rgaa30-12-11": "…",

…

},

"tests\_description\_en": {

"Rgaa30-11-1-2":"…",

…

},

"tests\_description\_fr": {

"Rgaa30-11-1-2":"…",

…

},

"test\_na": [

{

"criterion": "…",

"test": "…",

"theme": "…"

},

{…}

],

"test\_passed": [

{

"criterion": "…",

"test": "…",

"theme": "…"

},

{…}

],

"remarks": [

{

"criterion": "",

"theme": "",

"test": "",

"issue": "",

"message\_en": "",

"message\_fr": "",

"message\_es": "",

"line\_number": 0,

"snippet": ""

},

{…….}

]

}

**Object**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| http\_status\_code | Number | Return status query code |
| url | String | Audited URL |
| status | String | Status of the audit |
| score | Number | Audit score in % |
| ref | String | The repository used for the audit |
| level | String | The level of accessibility used for auditing |
| language | String | Message language and audit result remarks |
| nb\_w3c\_invalidated | Number | Number of w3c errors |
| nb\_passed | Number | Number of valided tests |
| nb\_failed | Number | Number of invalided tests |
| nb\_not\_tested | Number | Number of non-tested tests |
| nb\_na | Number | Number of tests not applicable |
| nb\_failed\_occurences | Number | Number of test cases invalidated |
| nb\_detected | Number | See (Deleting this value) always zero |
| nb\_suspected | Number | See (Deleting this value) always zero |
| nb\_nmi | Number | Number of pre-qualified tests |
| themes\_description\_en | Objet | Description of topics in English |
| themes\_description\_fr | Objet | Description of topics in French |
| themes\_description\_es | Objet | Description of topics in Spanish |
| criterions\_description\_en | Objet | Description of criteria in English |
| criterions\_description\_fr | Objet | Description of criteria in French |
| tests\_description\_en | Objet | Description of tests in English |
| tests\_description\_fr | Objet | Description of tests in French |
| test\_na | Array | Vector contains all tests not applicable |
| test\_passed | Array | Vector contains all validated tests |
| remarks | Array | Vector contains all the notes of the audit |
| criterion | String | Code of the criterion (the description to be found in criterions\_description\_en for the English and criterions\_description\_fr for the French) |
| theme | String | Code of Theme (the description to look for in themes\_description\_en for English and themes\_description\_en for the French, themes\_description\_es for the Spanish) |
| test | String | Test code (description to be found in tests\_description\_en for english and tests\_description\_en for french) |
| issue | String | Remark code |
| message\_en | String | Description of the remark in English |
| message\_fr | String | Description of the remark in French |
| message\_es | String | Description of the remark in Spanish |
| line\_number | Number | Number of lines of the audited code |
| snippet | String | The HTML code concerned by the remark |

**Sample Curl**

curl --ssl-reqd -H "Authorization**:** bearer eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI.oGxndCq3Ocz6CKi2aUb4uA" -H "Accept**:** application/json; charset=utf-8" -H "Content-Type: application/json" -X POST -d ' {"page\_url": "http://www.oceaneconsulting.com", "language":"all", "dt\_tbl\_marker":"data", "cplx\_tbl\_marker":"complexe", "pr\_tbl\_marker":"presentation", "dcr\_img\_marker":"decorative", "inf\_img\_marker":"informative"}' https://www.api.tanaguru.com/v1.0/service/auditPage

## API USER STATISTICS

**GET** https://www.api.tanaguru.com/v1.0/service/limit\_stat

**Request format**

**Headers**

**Authorization:** bearer eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI.oGxndCq3Ocz6CKi2aUb4uA

**Accept:** application/json; charset=utf-8

**Content-type:** application/json

**Parameters**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Required** | **Type** | **Default value** | **Description** | **Available value** |
| Authorization | Yes | String | Any | Authentication token used | bearer <token value> |

**Request format**

{

"status" : 200,

"quotas\_limit" : 0,

"quotas\_used": 0,

"number\_call": 0,

"number\_call\_error": 0

}

**Object**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| status | Number | Return status query code |
| quotas\_limit | Number | Quota limitation (number of calls allowed) |
| quotas\_used | Number | Number of calls made successfully |
| number\_call | Number | Number of calls made with or without error |
| number\_call\_error | Number | Number of calls returned with an error |

**Sample Curl**

curl --ssl-reqd -H "Authorization**:** bearer eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI.oGxndCq3Ocz6CKi2aUb4uA" -H "Accept**:** application/json; charset=utf-8" -H "Content-Type: application/json" -X GET

https://www.api.tanaguru.com/v1.0/service/limit\_stat

## REQUEST A TOKEN (AUTHENTICATION)

**POST** https://www.api.tanaguru.com/v1.0/security/authenticate

**Request format**

**Headers**

**Authorization:** basic YjkwMjVjNTZiMjQyNTA1M2RjMDYMmM3ZWI3M2QyMTA2NzY5NQ==

**Content-Type:** application/json

**Grant\_type:** client\_credentials

**Parameters**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Required** | **Type** | **Default value** | **Description** | **Available value** |
| Authorization | Yes | String | Any | Authentication information | basic < Key concatenation and base64 encoded client secret code> |

**Request format**

{

"status ": 200,

"token\_type": "bearer”,

"access\_token": ""

}

**Object**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| status | Number | Return status query code |
| token\_type | String | Token type |
| access\_token | String | Authentication token |

**Sample Curl**

curl --ssl-reqd -H "Authorization**:** basic YjkwMjVjNTZiMjQyNTA1M2RjMDYMmM3ZWI3M2QyMTA2NzY5NQ==" -H "Content-Type: application/json" -H "Grant\_type: client\_credentials" -X POST

https://www.api.tanaguru.com/v1.0/service/security/authenticate

## INVALIDATE A TOKEN (AUTHENTICATION)

**POST** https://www.api.tanaguru.com/v1.0/security/invalidate\_token

**Request format**

**Headers**

**Authorization:** basic YjkwMjVjNTZiMjQyNTA1M2RjMDYMmM3ZWI3M2QyMTA2NzY5NQ==

**Content-Type:** application/json

**access\_token: …**

**Parameters**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Required** | **Type** | **Default** | **Description** | **Available value** |
| Authorization | Yes | String | Any | Authentication information | basic < Key concatenation and base64 encoded client secret code> |
| access\_token | Yes | String | Any | Authentication Token to Disable | Token value |

**Answer format**

{

"status ": 200,

"access\_token": ""

}

**Object**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| status | Number | Return status query code |
| access\_token | String | Authentication token that has been invalidated |

**Sample Curl**

curl --ssl-reqd -H "Authorization**:** bearer eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI.oGxndCq3Ocz6CKi2aUb4uA" -H "Content-Type: application/json" -H "access\_token:" -X POST

https://www.api.tanaguru.com/v1.0/service/security/invalidate\_token

# CASES OF ERRORS

This section describes some common mistakes involved in using tokens. Be aware that not all possible error responses are covered here!

**Obtain a token with an invalid request (leaving Grant\_type: client\_credentials aside)**

{

"message": "grant\_type value is missing",

"status": 412

}

**Obtain a token with an invalid request (leaving aside Authorization)**

{

"message": "Consumer key & Consumer secret value are missing",

"status": 412

}

**Obtain or invalidate a token with invalid or outdated application credentials**

{

"message": "Unable to verify your credentials",

"status": 403

}

**Disable invalid or outdated token**

{

"message": "Invalid or expired token",

"status": 403

}

# Quick start Guide

## Installing a chrome extension

From a chrome browser, download [Advanced Rest client](https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo)

Once installed, start the extension.

## Request token

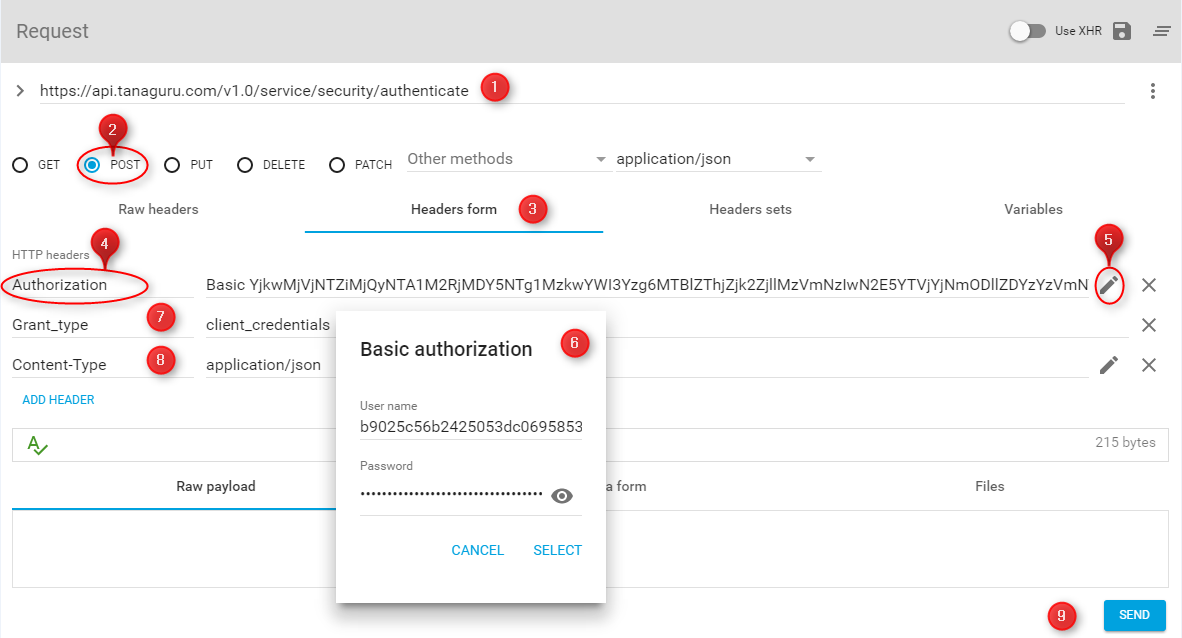
Assuming that your Beta ID is: **b9025c56b2425053dc069585390ab7c8** and your

Beta Password is: **10ee8cf96f9e35f7207a9a5cb3f89ed63c5f59692e1782d82c7eb73d21067695**.

**Note:** For First Beta version of Tanaguru Api, we authorize 5000 requests for free.

**Note:** The authentication informations mentioned is only for beta version, this is a sample. Contact team Tanaguru to have a valid client ID and PIN (contact@tanaguru.com)

1. Enter the url for authentication: **https://api.tanaguru.com/v1.0/service/security/authenticate**
2. Choose the request method POST
3. Go to the tab **Headers form**
4. Add the parameter **Authorization**
5. Click to the « Edit » icone to set the parameter value **Authorization**
6. Enter your **client ID** at User name, your **secret code** at Password, and click on select button.
7. Add the **Grant\_type** parameter and give it the value **client\_credentials**
8. Add the **Content-type** parameter and give it the value **application/json**
9. After entering all these informations, click send to apply.



You will get the following answer:



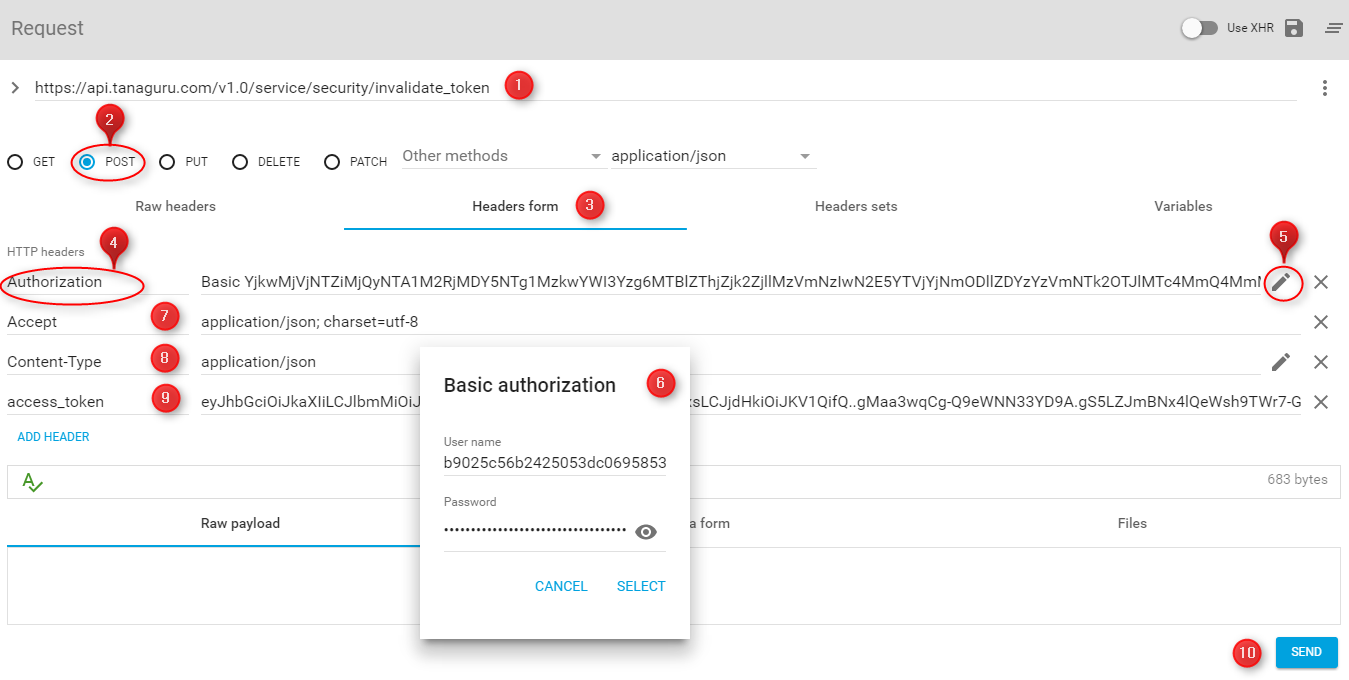
1. The answer contain informations about the answer **statut** and the **token** type
2. The token value that you will use to make audit requests.

**Note :** Keep the token generated to use it in the authentication for audit requests.

## Invalidate a token

After using the token, if you want disable it for any reason,

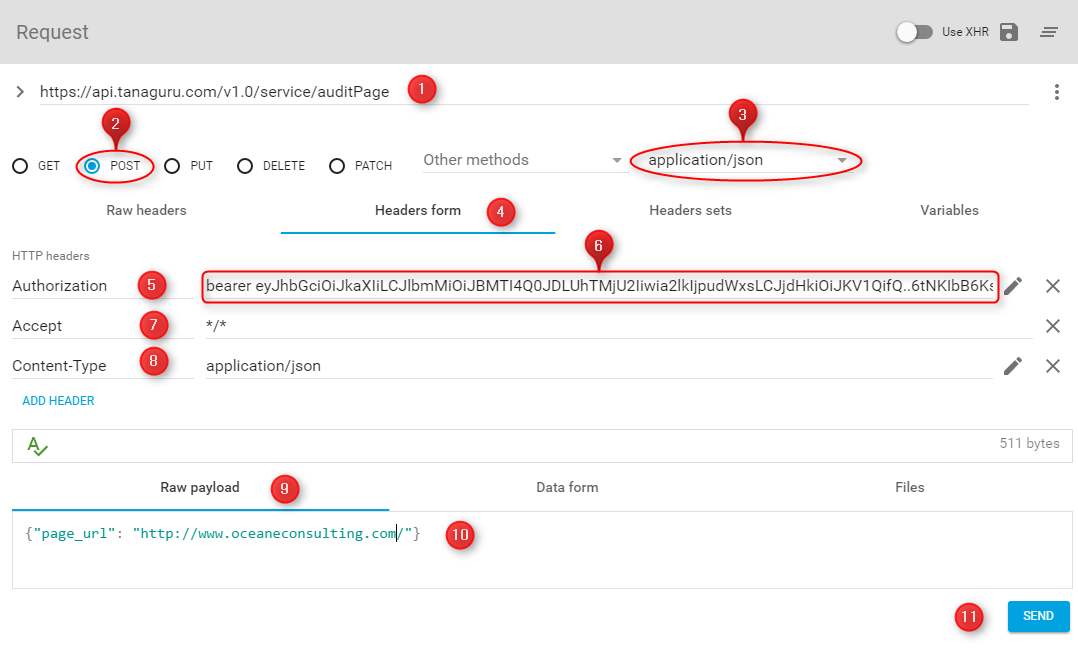
1. Enter the url authentication: **https://api.tanaguru.com/v1.0/service/security/invalidate\_token**
2. Make a **POST** request,
3. Go to the tab **Headers form**
4. Add the parameter **Authorization**,
5. Click to the « Edit » icone to set the parameter value **Authorization**
6. Enter your **Client** **ID** at User name, your **secret code** at Password, and validate.
7. Add the **Accept** parameter and give it the value **application/json ; charset=utf-8**
8. Add the **Content-type** parameter and give it the value **application/json**
9. Add the **access\_token** parameter and enter the token that you want invalidate.
10. Click send to apply.



## Launch a page audit

To launch an audit on the site <http://www.oceaneconsulting.com> with all languages,

1. Enter the url authentication: **https://api.tanaguru.com/v1.0/service/auditPage**
2. Make a **POST** request,
3. Choose **application/json** as data type to send.
4. Go to tab **Headers form**
5. Add the parameter **Authorization**
6. Use the Token type **bearer** with your token as : **bearer eyJH...**
7. Add the parameter **Accept** and give it the value **\*/\***
8. Add the parameter **Content-type** and give it the value **application/json**
9. Got to tab **Raw payload**
10. Enter at the Json format **{"page\_url" : "http://www.oceaneconsulting.com"}**, add the **language**, **referentiel** et **level** attributes is not required because they have a default value (All, Rgaa30, AA).
11. Click send to apply.



## API USE STATISTICS

To see API usage statistics with your account

1. Enter the url authentication: **https://api.tanaguru.com/v1.0/service/limit\_stat**
2. Make a **GET** request
3. Go to tab **Headers form**
4. Add the parameter **Authorization**
5. Use the Token type **bearer** with your token as : **bearer eyJH...**
6. Add the parameter **Accept** and give it the value **\*/\***
7. Add the parameter **Content-type** and give it the value **application/json**
8. Click send to apply.
9. The answer contain the API usage informations.

