|  |  |
| --- | --- |
| Documentation de l’API REST TANAGURU V0.1  Documentation Api Tanaguru | Résumé  Utiliser l’Api Tanaguru avec une extension Chrome  **L’équipe Tanaguru** |

Table des matières

[Introduction 2](#_Toc462219685)

[AUTHENTIFICATION 2](#_Toc462219686)

[Étape 1 : Encoder la clé et le code secret du client 3](#_Toc462219687)

[Étape 2 : Obtenir un jeton 3](#_Toc462219688)

[Étape 3 : Authentifier les requêtes API avec le jeton 4](#_Toc462219689)

[CREDITS 5](#_Toc462219690)

[Statut du retour, liste des codes 5](#_Toc462219691)

[Liste des actions disponibleS 6](#_Toc462219692)

[Lancer un audit de page 6](#_Toc462219693)

[Statistique d’utilisation de l’API 10](#_Toc462219694)

[Demander un jeton (authentification) 11](#_Toc462219695)

[Invalider un jeton (authentification) 12](#_Toc462219696)

[Cas d’erreurs 13](#_Toc462219697)

[Guide de démarrage rapide 14](#_Toc462219698)

[Installation d’une extension chrome 14](#_Toc462219699)

[Demande de jeton 14](#_Toc462219700)

[Invalider un jeton 16](#_Toc462219701)

[Lancer un audit de page 17](#_Toc462219702)

[Statistique d’utilisation de l’API 18](#_Toc462219703)

# Introduction

L’API REST fournit une API de services Web pratique, puissante, simple pour interagir avec le service d’audit des pages web de Tanaguru. Elle présente comme avantages une grande facilité d'intégration et de développement. C’est également une excellente solution pour une utilisation avec des applications mobiles et Web.

L’utilisation de l'API exige une connaissance de base des services Web.

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

Tous les appels de l'API commencent par cette URL à laquelle s'ajoute le chemin vers l'action désirée.   
Exemple : https://www.api.tanaguru.com/v1.0/service**/auditPage**

L'API est accessible uniquement via le protocole Https et accepte les requêtes HTTP **POST et GET**. Les données envoyées en POST doivent respecter le format **JSON**.

# **AUTHENTIFICATION**

L’API REST utilise le protocole OAuth pour permettre aux utilisateurs d'applications d'accéder en toute sécurité aux données sans avoir à révéler nom d'utilisateur et mot de passe.

Avant de procéder à des appels d'API REST, vous devez vous authentifier en utilisant OAuth 2.0. Pour ce faire, vous aurez besoin des éléments suivants :

* Un ID Client et le code secret fournis par Océane consulting.
* Faire une requête POST security/authenticate pour échanger ces informations d’identification et générer un jeton d’authentification via l’API. Ensuite pour toutes les requêtes, vous devez préciser dans le paramètre d’authentification sur le header de la requête, le jeton comme valeur.

**Phase 1 : Autentification**

1. Faire une requête POST sur security/authenticate en utilisant la clé et le code de client pour demander un jeton.
2. L’API Tanaguru reçoit la requête et génère un jeton et le renvoie au client sous format JSON.
3. Le client reçoit le jeton et le parse pour récupérer la valeur de jeton pour l’utiliser dans la phase

**Phase 2 : Appels vers l’API**

1. Le client lance une requête POST sur /service/auditPage avec la valeur de jeton comme un identifiant pour lancer un audit et récupérer les résultats.
2. L’API Tanaguru traite la requête et lance l’audit et renvoie les données du résultat sous format JSON au client.
3. Le client reçoit les données et les parse pour des utilisations diverses.

**Les jetons sont des mots de passe**

Gardez à l'esprit que le ID client, le code secret et le jeton sont indispensables pour accéder aux demandes liées à une application. Ces valeurs doivent être considérées comme sensibles et tout comme les mots de passe ne doivent pas être partagées ou distribuées à des personnes non fiables.

**SSL absolument nécessaire**

Cette authentification n’est fiable uniquement que si SSL est utilisé. Par conséquent, toutes les demandes pour obtenir et utiliser les jetons doivent utiliser des paramètres HTTPS, c’est également une exigence de l'API.

## Étape 1 : Encoder la clé et le code secret du client

Etapes pour encoder l’ID Client et le code secret du client dans un ensemble d'informations d'identification afin d’obtenir un jeton :

1. Concaténer l’ID Client plus le caractère de deux points " :", et le code secret du client, le tout en une seule chaîne.
2. Encoder en Base64 la chaîne obtenue à partir de l'étape précédente.

Vous trouverez ci-dessous des exemples de valeurs indiquant le résultat de cet algorithme. Notez que le code secret de client utilisé pour ces exemples est désactivé et ne fonctionnera pas pour les demandes réelles.

|  |  |
| --- | --- |
| Clé | b9025c56b2425053dc069585390ab7c8 |
| Code secret | 10ee8cf96f9e35f7207a9a5cb3f89ed63c5f59692e1782d82c7eb73d21067695 |
| Concaténation | b9025c56b2425053dc069585390ab7c8 : 10ee8cf96f9e35f7207a9a5cb3f89ed63c5f59692e1782d82c7eb73d21067695 |
| Porteur d’informations d’authentification (Base64) | YjkwMjVjNTZiMjQyNTA1M2RjMDY5NTg1MzkwYWI3Yzg6MTBlZThjZjk2ZjllMzVmNzIwN2E5YTVjYjNmODllZDYzYzVmNTk2OTJlMTc4MmQ4MmM3ZWI3M2QyMTA2NzY5NQ== |

## Étape 2 : Obtenir un jeton

La valeur calculée à l'étape 1 doit être échangée contre un jeton en émettant une demande de POST /service/security/authenticate :

* La demande doit être une requête HTTP POST.
* La demande doit inclure un en-tête **Authorization** avec la valeur de base : **Basic < valeur codée de l'étape 1>**.
* La demande doit inclure un en-tête Content-Type avec la valeur application/json.
* La demande doit inclure un en-tête Grant\_type : client\_credentials.

Exemple de demande :

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

Si la demande a été correctement formatée, le serveur répond avec un payload JSON codé :

Exemple de réponse :

HTTP/1.1 200 OK

Status: 200 OK

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

...

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

Les demandes doivent vérifier que la valeur associée à la clé token\_type de l'objet retourné est « **bearer »**. La valeur associée à la clé access\_token est le jeton.

Notez qu'un jeton est valide pour une application à la fois. Refaire une autre demande avec les mêmes informations d'identification va retourner le même jeton jusqu'à ce qu'il soit invalidé.

## Étape 3 : Authentifier les requêtes API avec le jeton

Le jeton est utilisé pour faire des appels vers l’API. Pour utiliser le jeton, construire une demande HTTPS normale et inclure un en-tête **Authorization** avec la valeur du jeton : **Bearer < valeur de jeton de l'étape 2>**. La signature n’est pas nécessaire.

Exemple de demande :

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

Host: api.tanaguru.com

User-Agent: My App v1.0

Authorization: Bearer AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA%2FAAAAAAAAAAAAAAA

# **CREDITS**

L'API de Tanaguru repose sur un **système de crédits**.

Lorsque vous lancez un audit depuis l'API, et que celle-ci génère un résultat finale complet, votre compte est débité d'une unité de crédit.

Les autres requêtes, (analyse échouée, demande d’information statistique, authentification, etc) ne débitent aucun crédit.

# Statut du retour, liste des codes

Chaque appel à l'API donne lieu à une réponse retournant un code spécifique en fonction du résultat obtenu. L'analyse de ce code vous permet de vous assurer que la requête a été traitée avec succès.

Tous les codes >= 400 indiquent que la requête n'a pas été traitée avec succès par nos serveurs.

* **200**: OK
* **400**: Paramètre manquant, ou valeur incorrecte
* **401**: Authentification nécessaire (jeton non précisé ou invalide)
* **403**: Action non autorisée (crédits épuisés, URL non autorisée, etc)
* **404**: Page inaccessible (URL inconnue / impossible d'accéder à l'adresse)
* **406**: Le JSON indiqué en données POST n'est pas valide
* **408**: Dépassement du temps maximal autorisé pour l’audit
* **500**: Erreur inconnue, contactez-nous
* **503**: L'API est momentanément indisponible, réessayez dans quelques minutes

# Liste des actions disponibleS

## Lancer un audit de page

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

**Format de la requête**

**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**

**}**

**Paramètres**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom** | **Requis** | **Type** | **Valeur par défaut** | **Description** | **Valeur possible** |
| Authorization | Oui | String | Aucune | Jeton d'authentification utilisé | bearer <valeur de jeton> |
| page\_url | Oui | String | Aucune | URL de la page à auditer | <http://www>. ….. |
| referentiel | Non | String | RGAA30 | Le référentiel d’accessibilité à utiliser | RGAA32016, RGAA30, RGAA22, AW22 |
| level | Non | String | AA | Le niveau d’accessibilité à utiliser | A, AA, AAA, Bz, Or, Ar, LEVEL\_1, LEVEL\_2, LEVEL\_3 |
| language | Non | String | All | La langue des messages et les remarques de résultat d’audit | All, Fr, En, Es\_En, Es\_Fr |
| dt\_tbl\_marker | Non | String | Aucune | Marqueur HTML des tableaux de données | Correspond à l'attribut "id", "class" ou "role" des tableaux complexes. Plusieurs marqueurs peuvent être saisis, séparées par un ; |
| cplx\_tbl\_marker | Non | String | Aucune | Marqueur HTML des tableaux complexes |
| pr\_tbl\_marker | Non | String | Aucune | Marqueur HTML des tableaux de présentation |
| dcr\_img\_marker | Non | String | Aucune | Marqueur des images décoratives |
| inf\_img\_marker | Non | String | Aucune | Marqueur des images porteuses d'informations |
| screen\_width | Non | Number | 1920 | Largeur de l’écran en pixel | Valeur maximum est 2048 |
| screen\_height | Non | Number | 1080 | Hauteur de l’écran en pixel | Valeur maximum est 2048 |
| description\_ref | Non | Boolean | False | Pour fournir ou non les intitulés des tests, critères, thème du référentiel. | True/False |
| html\_tags | Non | Boolean | False | Pour fournir les remarques et les intitulés encodé en HTML ou non | True/False |

**Format de la réponse**

{

"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": ""

},

{…….}

]

}

**Objet**

|  |  |  |
| --- | --- | --- |
| **Nom** | **Type** | **Description** |
| http\_status\_code | Number | Code retour du statut de la requête |
| url | String | URL auditée |
| status | String | Status de l’audit |
| score | Number | Score de l’audit en % |
| ref | String | Le référentiel utilisé pour l’audit |
| level | String | Le niveau d’accessibilité utilisé pour l’audit |
| language | String | La langue des messages et les remarques de résultat d’audit |
| nb\_w3c\_invalidated | Number | Nombre des erreurs w3c |
| nb\_passed | Number | Nombre de teste validé |
| nb\_failed | Number | Nombre de teste invalidé |
| nb\_not\_tested | Number | Nombre de teste non testé |
| nb\_na | Number | Nombre de teste non applicable |
| nb\_failed\_occurences | Number | Nombre d’occurrence de testes invalidé |
| nb\_detected | Number | A voir (Suppression de cette valeur) toujours zéro |
| nb\_suspected | Number | A voir (Suppression de cette valeur) toujours zéro |
| nb\_nmi | Number | Nombre de teste de pré-qualifié |
| themes\_description\_en | Objet | Description des thèmes en anglais |
| themes\_description\_fr | Objet | Description des thèmes en français |
| themes\_description\_es | Objet | Description des thèmes en espagnole |
| criterions\_description\_en | Objet | Description des critères en anglais |
| criterions\_description\_fr | Objet | Description des critères en français |
| tests\_description\_en | Objet | Description des tests en anglais |
| tests\_description\_fr | Objet | Description des tests en français |
| test\_na | Array | Vecteur contient tous les tests non applicable |
| test\_passed | Array | Vecteur contient tous les tests validés |
| remarks | Array | Vecteur contient tous les remarques de l’audit |
| criterion | String | Code du critère (la description à chercher dans criterions\_description\_en pour l’anglais et criterions\_description\_fr pour le français) |
| theme | String | Code de la Thématique (la description à chercher dans themes\_description\_en pour l’anglais et themes\_description\_fr pour le français, themes\_description\_es pour l’espagnole) |
| test | String | Code du test (la description à chercher dans tests\_description\_en pour l’anglais et tests\_description\_fr pour le français) |
| issue | String | Code de la remarque |
| message\_en | String | Description de la remarque en anglais |
| message\_fr | String | Description de la remarque en français |
| message\_es | String | Description de la remarque en espagnole |
| line\_number | Number | Nombre de ligne du code audité |
| snippet | String | Le bout de code HTML concerné par la remarque |

**Exemple 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

## Statistique d’utilisation de l’API

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

**Format de la requête**

**Headers**

**Authorization:** bearer eyJhbGciOiJkaXIiLCJlbmMiOiJBMTI.oGxndCq3Ocz6CKi2aUb4uA

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

**Content-type:** application/json

**Paramètres**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom** | **Requis** | **Type** | **Valeur par défaut** | **Description** | **Valeur possible** |
| Authorization | Oui | String | Aucune | Jeton d'authentification utilisé | bearer <valeur de jeton> |

**Format de la réponse**

{

"status" : 200,

"quotas\_limit" : 0,

"quotas\_used": 0,

"number\_call": 0,

"number\_call\_error": 0

}

**Objet**

|  |  |  |
| --- | --- | --- |
| **Nom** | **Type** | **Description** |
| status | Number | Code retour du statut de la requête |
| quotas\_limit | Number | Limitation de quotas (nombre d’appels autorisé) |
| quotas\_used | Number | Nombre d’appels effectué avec sucée |
| number\_call | Number | Nombre d’appels effectué avec ou sans erreur |
| number\_call\_error | Number | Nombre d’appels retourné avec une erreur |

**Exemple 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

## Demander un jeton (authentification)

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

**Format de la requête**

**Headers**

**Authorization:** basic YjkwMjVjNTZiMjQyNTA1M2RjMDYMmM3ZWI3M2QyMTA2NzY5NQ==

**Content-Type:** application/json

**Grant\_type:** client\_credentials

**Paramètres**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom** | **Requis** | **Type** | **Valeur par défaut** | **Description** | **Valeur possible** |
| Authorization | Oui | String | Aucune | Informations d'authentification | basic <concaténation des clés et code secret de client encodé en Base64> |

**Format de la réponse**

{

"status ": 200,

"token\_type": "bearer”,

"access\_token": ""

}

**Objet**

|  |  |  |
| --- | --- | --- |
| **Nom** | **Type** | **Description** |
| status | Number | Code retour du statut de la requête |
| token\_type | String | Type de jeton |
| access\_token | String | Jeton d’authentification |

**Exemple 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

## Invalider un jeton (authentification)

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

**Format de la requête**

**Headers**

**Authorization:** basic YjkwMjVjNTZiMjQyNTA1M2RjMDYMmM3ZWI3M2QyMTA2NzY5NQ==

**Content-Type:** application/json

**access\_token: …**

**Paramètres**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nom** | **Requis** | **Type** | **Valeur par défaut** | **Description** | **Valeur possible** |
| Authorization | Oui | String | Aucune | Informations d'authentification | basic <concaténation des clés et code secret de client encodé en Base64> |
| access\_token | Oui | String | Aucune | Jeton d’authentification à invalider | Valeur de jeton |

**Format de la réponse**

{

"status ": 200,

"access\_token": ""

}

**Objet**

|  |  |  |
| --- | --- | --- |
| **Nom** | **Type** | **Description** |
| status | Number | Code retour du statut de la requête |
| access\_token | String | Jeton d’authentification qui a été invalidé |

**Exemple 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

# Cas d’erreurs

Cette section décrit quelques erreurs communes impliquées dans l'utilisation de jetons. Sachez que toutes les réponses d'erreur possibles ne sont pas couvertes ici !

**Obtenir un jeton avec une demande non valide (en laissant de côté Grant\_type : client\_credentials)**

{

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

"status": 412

}

**Obtenir un jeton avec une demande non valide (en laissant de côté Authorization)**

{

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

"status": 412

}

**Obtenir ou invalider un jeton avec des informations d'identification d'applications incorrectes ou périmées**

{

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

"status": 403

}

**Invalider un jeton incorrect ou périmé**

{

"message": "Invalid or expired token",

"status": 403

}

# Guide de démarrage rapide

## Installation d’une extension chrome

Depuis un navigateur chrome, rendez-vous à : <https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo>

Lancer l’extension une fois installée.

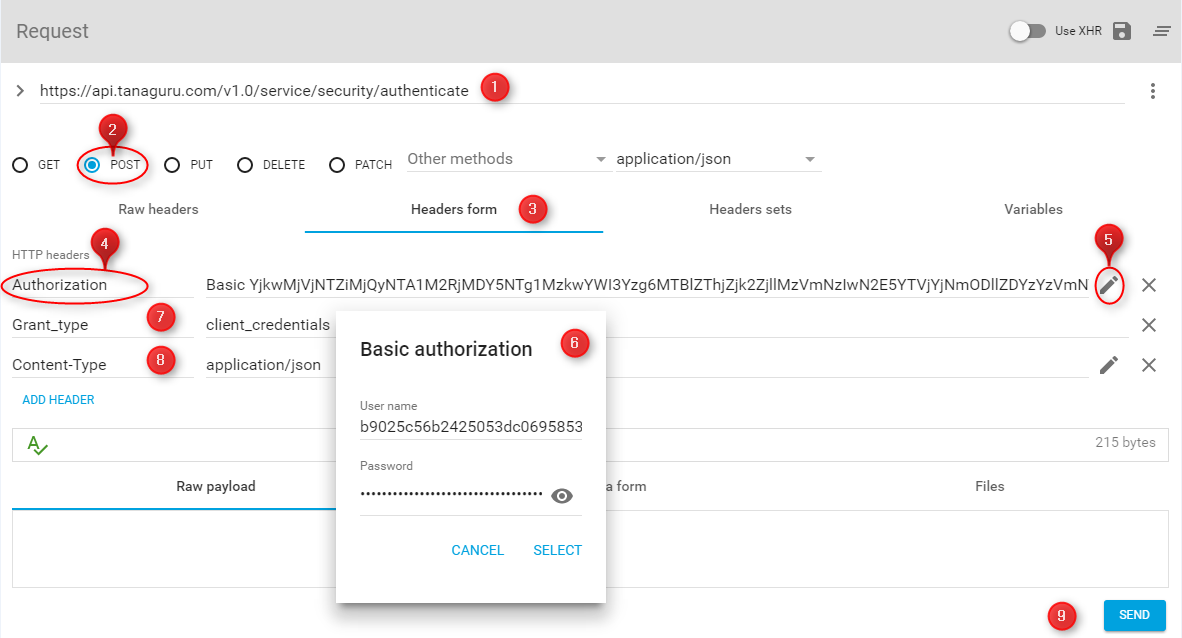
## Demande de jeton

En supposant que votre ID BETA est : **b9025c56b2425053dc069585390ab7c8** et votre

code secret BETA est : **10ee8cf96f9e35f7207a9a5cb3f89ed63c5f59692e1782d82c7eb73d21067695**.

**Remarque :** Pour la première version BETA de l'API Tanaguru, nous autorisons 5000 requêtes gratuite.

**Remarque :** Les informations d’authentification mentionnée sont seulement pour la version BETA, c’est juste un exemple. Rapprochez-vous de l'équipe Tanaguru pour avoir un ID client et un code secret valide (contact@tanaguru.com).

1. Saisissez l’url pour l’authentification : **https://api.tanaguru.com/v1.0/service/security/authenticate**
2. Choisissez la méthode de la requête **POST**
3. Allez à l’onglet **Headers form**
4. Ajoutez le paramètre **Authorization**
5. Cliquez sur l’icône éditer la valeur du paramètre **Authorization**
6. Saisissez votre **ID Client** comme User name et **le code secret** comme Password, et validez.
7. Ajoutez le paramètre **Grant\_type** et lui donner comme valeur **client\_credentials**
8. Ajoutez le paramètre **Content-type** et lui donner comme valeur **application/json**
9. Après avoir saisi toutes ces informations, cliquez sur **envoyer** pour faire la demande. 

Vous obtiendrez la réponse ci-dessous :

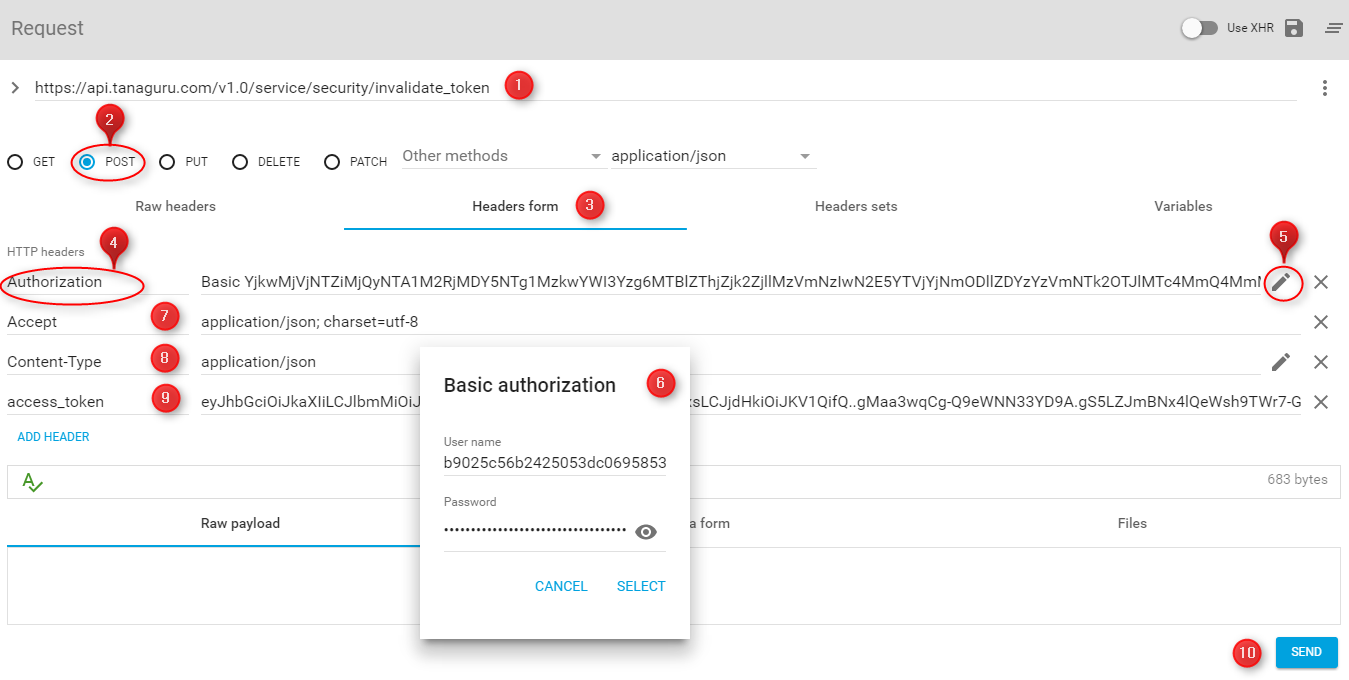


1. La réponse contient des informations à propos du **statut** de la réponse, ainsi que le type de **jeton**
2. La valeur du **jeton** que vous allez utiliser ensuite pour faire des demandes d’audit.

## Invalider un jeton

Après avoir utilisé le jeton, si vous souhaitez l’invalider pour une raison quelconque,

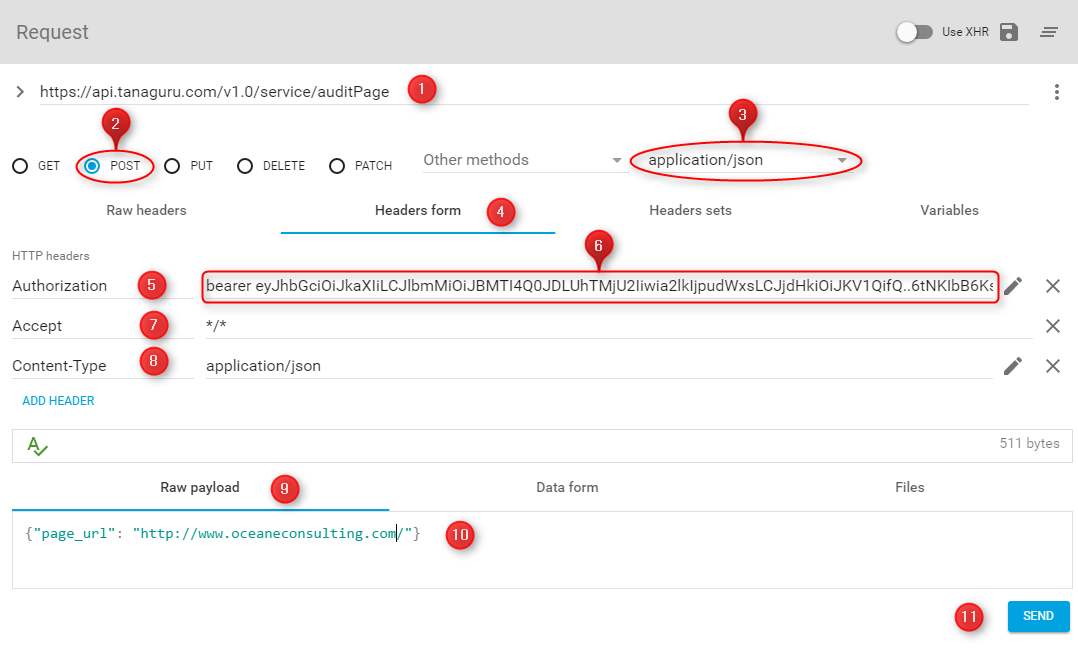
1. Saisir l’url pour l’authentification : **https://api.tanaguru.com/v1.0/service/security/invalidate\_token**
2. Vous devrez faire une demande **POST**,
3. Allez à l’onglet **Headers form**
4. Ajoutez le paramètre **Authorization** ,
5. Cliquez sur l’icône éditer la valeur du paramètre **Authorization**
6. Saisissez votre **ID Client** comme User name et le **code secret** comme Password, et validez.
7. Ajoutez le paramètre **Accept** et lui donner comme valeur **application/json ; charset=utf-8**
8. Ajoutez le paramètre **Content-type** et lui donner comme valeur **application/json**
9. Ajoutez le paramètre **access\_token** et saisissez le jeton que vous voulez invalider.
10. Cliquez sur **Envoyer** pour faire la demande.



## Lancer un audit de page

Pour lancer un audit sur le site <http://www.oceaneconsulting.com> avec toutes les langues,

1. Saisir l’url pour l’authentification : **https://api.tanaguru.com/v1.0/service/auditPage**
2. Faites une demande **POST**,
3. Choisissez **application/json** comme type de donnée à envoyer.
4. Allez à l’onglet **Headers form**
5. Ajoutez le paramètre **Authorization**
6. Utilisez le type de jeton **bearer** concaténé avec votre **jeton** comme : **bearer eyJH...**
7. Ajoutez le paramètre **Accept** et lui donner comme valeur **\*/\***
8. Ajoutez le paramètre **Content-type** et lui donner comme valeur **application/json**
9. Allez à l’onglet **Raw payload**
10. Vous saisissez en format Json **{"page\_url" : "http://www.oceaneconsulting.com"}**, vous n’êtes pas obligé de rajouter les attributs **language**, **referentiel** et **level** parce qu’ils ont des valeurs par default (All, Rgaa30, AA).
11. Cliquez sur **envoyer** pour faire la demande.



## Statistique d’utilisation de l’API

Pour connaître les statistiques d’utilisation de l’API avec votre compte

1. Saisir l’url pour l’authentification : **https://api.tanaguru.com/v1.0/service/limit\_stat**
2. Faites une demande **GET**
3. Allez à l’onglet **Headers form**
4. Ajoutez le paramètre **Authorization**
5. Utilisez le type de jeton **bearer** concaténé avec votre **jeton** comme : **bearer eyJH...**
6. Ajoutez le paramètre **Accept** et lui donner comme valeur **application/json ; charset=utf-8**
7. Ajoutez le paramètre **Content-type** et lui donner comme valeur **application/json**
8. Cliquez sur **envoyez** pour faire la demande.
9. La réponse contient des informations d’utilisation de l’api.

