

user.login

Description

string/object `user.login(object parameters)`

Cette méthode permet de se connecter à l'API et de générer un jeton d'authentification.

Lorsque vous utilisez cette méthode, vous devez également exécuter [user.logout](#) pour empêcher la génération d'un grand nombre d'enregistrements de session ouverts.

Paramètres

Cette méthode est disponible pour les utilisateurs non authentifiés et doit être appelée sans le paramètre `auth` dans la requête JSON-RPC.

(object) Paramètres contenant le nom d'utilisateur et le mot de passe.

La méthode accepte les paramètres suivants.

Paramètre	Type	Description
password (obligatoire)	string	Mot de passe de l'utilisateur.
user (obligatoire)	string	Nom d'utilisateur.
<code>userData</code>	flag	Renvoie des informations sur l'utilisateur authentifié.

Lorsque vous utilisez l'authentification HTTP, le nom d'utilisateur dans la requête API doit correspondre à celui utilisé dans l'en-tête `Authorization`. Le mot de passe ne sera pas validé et peut être omis.

Valeurs retournées

(string/object) Si le paramètre `userData` est utilisé, retourne un objet contenant des informations sur l'utilisateur authentifié.

En plus des propriétés de l'objet [utilisateur](#) standard, les informations suivantes sont retournées :

Propriété	Type	Description
<code>debug_mode</code>	boolean	Indique si le mode debug est activé pour l'utilisateur.
<code>gui_access</code>	integer	Méthode d'authentification de l'utilisateur à l'interface. Reportez-vous à la propriété <code>gui_access</code> de l'objet groupe d'utilisateurs pour obtenir une liste des valeurs possibles.
<code>sessionid</code>	string	Jeton d'authentification, à utiliser dans les demandes API qui suivront.
<code>userip</code>	string	Adresse IP de l'utilisateur.

Si un utilisateur a été authentifié avec succès après une ou plusieurs tentatives infructueuses, la méthode renvoie les valeurs actuelles des propriétés `attempt_clock`, `attempt_failed` et `attempt_ip`, puis les réinitialise..

Si le paramètre `userData` n'est pas utilisé, la méthode renvoie un jeton d'authentification.

Le jeton d'authentification généré doit être mémorisé et utilisé dans le paramètre auth des requêtes JSON-RPC suivantes. Il est également requis lors de l'utilisation de l'authentification HTTP.

Exemples

Authentifier un utilisateur

Authentifier un utilisateur.

Requête :

```
{
  "jsonrpc": "2.0",
  "method": "user.login",
  "params": {
    "user": "Admin",
    "password": "zabbix"
  },
  "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": "0424bd59b807674191e7d77572075f33",
  "id": 1
}
```

Demander les informations d'un utilisateur authentifié

Authentifiez-vous et renvoyez des informations supplémentaires sur l'utilisateur.

Requête :

```
{
  "jsonrpc": "2.0",
  "method": "user.login",
  "params": {
    "user": "Admin",
    "password": "zabbix",
    "userData": true
  },
  "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": {
    "userid": "1",
    "alias": "Admin",
    "name": "Zabbix",
    "surname": "Administrator",
    "url": "",
    "autologin": "1",
    "autologout": "0",
    "lang": "ru_RU",
    "refresh": "0",
    "type": "3",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "127.0.0.1",
    "attempt_clock": "1355919038",
    "rows_per_page": "50",
    "debug_mode": true,
    "userip": "127.0.0.1",
    "sessionid": "5b56eee8be445e98f0bd42b435736e42",
    "gui_access": "0"
  },
  "id": 1
}
```

Voir également

- [user.logout](#)

Source

CUser::login() dans *frontends/php/include/classes/api/services/CUser.php*.

From:

<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

<https://www.zabbix.com/documentation/4.0/fr/manual/api/reference/user/login>

Last update: **2019/03/04 20:24**

