Jibiki users REST Application Programming Interface

Mathieu Mangeot

▶ **To cite this version:**

Mathieu Mangeot. Jibiki users REST Application Programming Interface. [Technical Report] Laboratoire d'Informatique de Grenoble. 2019. hal-02165773

HAL Id: hal-02165773
https://hal.science/hal-02165773

Submitted on 26 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Le projet **Jibiki.fr**

Dictionnaire japonais-français

Consultation

→

Consultation avancée

jibiki users REST Application Programming Interface

Summary

- `GET apiusers/users/` Get a list of users
- `GET apiusers/users/[login]` Get a user
- `POST apiusers/users/[login]` Create a new user
- `PUT apiusers/users/[login]` Modify an existing user
- `DELETE apiusers/users/[login]` Delete an existing user
- `GET apiusers/users/[login]/groups` Get a list of groups that include the user
- `PUT apiusers/users/[login]/groups/[groupname]` Addmin a user in a group
- `DELETE apiusers/users/[login]/groups/[groupname]` Removing a user from a group
- `GET apiusers/groups/` Get a list of groups
- `GET apiusers/groups/[groupname]` Get the list of users in a group
- `POST apiusers/groups/[groupname]` Add a list of users in a group
- `DELETE apiusers/groups/[groupname]` Remove a list of users in a group
- `PUT apiusers/groups/[groupname]/users/[login]` Add a user in a group
- `DELETE apiusers/groups/[groupname]/users/[login]` Remove a user from a group
- `GET apiusers/dictionary/[dictname]` Get a list of roles with access to the dictionary
- `GET apiusers/dictionary/[dictname]/[role]` Get a list of users with the role that have to the dictionary
- `PUT apiusers/dictionary/[dictname]/[role]/[login]` Gives the role with access to the dictionary to a user
- `DELETE apiusers/dictionary/[dictname]/[role]/[login]` Removes the role with access to the dictionary to a user

List of available users

URL	apiusers/users
Method	GET
ContentType	application/xml
	application/json
Returns	200 OK & list of users (XML or json)
	401 Unauthorized (wrong credentials)

Note: if the user is admin, the user email and groups are displayed
Example of query:

```
curl -H 'Accept: application/json' "apiusers/users"
```

Dernière mise-à-jour : 18 mai 2015. Plate-forme: © 2001-2015, GETA-CLIPS, GETALP-LIG. Licence LGPL. Données : Licence Creative Commons CC0 (domaine public).

Description of a user

URL	apiusers/users/[login]/
Method	GET
ContentType	application/xml
	application/json
Returns	200 OK & user (XML or json)
	401 Unauthorized (wrong credentials)
	404 Not Found

Note: if the user is admin, or herself, the user email and groups are displayed
Example of query:

```
curl -H 'Accept:application/json' \
"apiusers/users/login"
```

Answer:

```
{"user": {
"xmlns": "http://www-clips.imag.fr/geta/services/dml",
"name": "Mathieu Mangeot",
"login": "mangeot"
}}
```

Creating a new user

URL	apiusers/users/[login]
Method	POST
ContentType	application/xml
	application/json
Returns	201 Created & user (XML or json)
	400 Bad Request (XML or json not valid)
	401 Unauthorized (wrong credentials)
	409 Conflict (user already existing)
	422 Unprocessable Entity (XML not semantically correct)

The data to be sent with the POST command is the description of the user in XML or json.
Note 1: the user has to identify herself.
Example of query:

```
curl -X POST \
-u mangeot:password \
-H "Content-Type: application/json;charset=UTF-8" \
-d '{"user": {
"name": "Mathieu Mangeot",
"lang": "fra",
"login": "mangeot",
```

```
    "email": "Mathieu.Mangeot_@_myemailaddress.fr"
    }}'\
    "apiusers/users/mangeot"
```

Modifying an existing user

URL	apiusers/users/[login]
Method	PUT
ContentType	application/xml
	application/json
Returns	200 OK & XML or json (user)
	400 Bad Request (XML or json not valid)
	401 Unauthorized
	404 Not Found (user not found)
	422 Unprocessable Entity (XML not semantically correct)

The data to be sent with the PUT command is the XML or json of the user.
Note: Only the user herself or an admin user in the admin group can modify a user.
Example of query:

```
    curl -X PUT \
    -u mangeot:password \
    -H "Content-Type: application/json;charset=UTF-8" \
    -d '{"user": {
    "name": "Mathieu Mangeot",
    "lang": "eng",
    "login": "mangeot",
    "email": "Mathieu.Mangeot_@_myemailaddress.fr"
    }}'\
    "apiusers/users/mangeot"
```

Deleting an existing user

URL	apiusers/users/[login]
Method	DELETE
Returns	204 No Content & XML or json (user)
	401 Unauthorized
	404 Not Found

Note: Only the user herself or an admin user in the admin group can modify an existing user.
Example of query:

```
    curl -X DELETE \
    -u login:password \
    "apiusers/users/login"
```

Answer is not significant

List of groups that include the user

URL	apiusers/users/[login]/groups
Method	GET
ContentType	application/xml
	application/json
Returns	200 OK & user (XML or json)
	401 Unauthorized
	404 Not Found

Note: the user has to be herself or in the admin group.
Example of query:

```
curl \
-u login:password \
-H 'Accept:application/json' \
"apiusers/users/login/groups"
```

Answer:

```
{"d:group-list": {
"d:group": [
{"name": "admin"},
{
"role": "admin",
"dictionary": "Cesselin",
"name": "admind_Cesselin"
},
{
"role": "admin",
"dictionary": "Kanjidic",
"name": "admind_Kanjidic"
}
],
"xmlns:d": "http://www-clips.imag.fr/geta/services/dml"
}}
```

Adding an existing user into an existing group

URL	apiusers/users/[login]/groups/[groupname]
Method	PUT
ContentType	application/xml
	application/json
Returns	200 OK & XML or json (user)
	401 Unauthorized
	404 Not Found (user not found)

Note: Only the user herself or an admin user in the admin group can modify a user.

Example of query:

```
curl -X PUT \
-u login:password \
"apiusers/users/login/groups/readerd_Cesselin"
```

Removing an existing user from an existing group

URL	apiusers/users/[login]/groups/[groupname]
Method	DELETE
ContentType	application/xml
	application/json
Returns	204 No Content & XML or json (user)
	401 Unauthorized
	404 Not Found (user or group)

Note: Only the user herself or an admin user in the admin group can delete a user.
Example of query:

```
curl -X DELETE \
-u login:password \
"apiusers/users/login/groups/readerd_Cesselin"
```

List of available groups

URL	apiusers/groups
Method	GET
ContentType	application/xml
	application/json
Returns	200 OK & list of groups (XML or json)
	401 Unauthorized

Example of query:

```
curl -H 'Accept: application/json' "apiusers/groups"
```

Answer:


```
{"d:group-list": {
  "d:group": [
    {"name": "admind_Armement"},
    {"name": "admind_Cesselin"},
    {"name": "admind_DiLAF"},
    {"name": "admind_Kanjidic"},
    {"name": "admind_Maniette"},
    {"name": "readerd_Armement"},
```

```
    {"name": "readerd_Cesselin"},
    {"name": "readerd_Maniette"},
    {"name": "specialist"},
    {"name": "validator"},
    {"name": "admin"}
  ],
  "xmlns:d": "http://www-clips.imag.fr/geta/services/dml"
}}
```

List of users members of a groups

URL	apiusers/groups/[groupname]
Method	GET
ContentType	application/xml
	application/json
Returns	200 OK & list of groups (XML or json)

Example of query:

```
curl
-u login:password \
-H 'Accept: application/json' \
"apiusers/groups/admind_Cesselin"
```

```
{"d:group": {
  "xmlns:d": "http://www-
clips.imag.fr/geta/services/dml",
  "members": {"user-ref": "mangeot"},
  "name": "admind_Cesselin",
  "admins": {"user-ref": "mangeot"}
}}
```

Adding several users in a groups

URL	apiusers/groups/[groupname]
Method	POST
ContentType	application/xml
	application/json
Returns	201 Created & user (XML or json)
	400 Bad Request (XML or json not valid)
	401 Unauthorized
	409 Conflict (user already existing)
	422 Unprocessable Entity (XML not semantically correct)

The data to be sent with the POST command is the description of the user in XML or json.
Note 1: the user has to identify herself.
Example of query:

```
curl -X POST \
-u mangeot:password \
-H "Content-Type: application/json;charset=UTF-8" \
-d '{"user-list": {
"user": [
{
"login": "titi",
},
"login": "toto",
}
]
}}'\
"apiusers/groups/readerd_Cesselin/"
```

Removing a list of users from an existing group

URL	apiusers/groups/[groupname]
Method	DELETE
ContentType	application/xml
	application/json
Returns	204 No Content & XML or json (user)
	401 Unauthorized
	404 Not Found (user or group)

Note: Only the user herself or an admin user in the admin group can delete a user.
Example of query:

```
curl -X DELETE \
-u mangeot:password \
-H "Content-Type: application/json;charset=UTF-8" \
-d '{"user-list": {
"user": [
{
"login": "titi",
},
"login": "toto",
}
]
}}'\
"apiusers/groups/readerd_Cesselin/"
```

Adding an existing user into an existing group

URL	apiusers/groups/[groupname]/users/[login]
Method	PUT
ContentType	application/xml
	application/json
Returns	200 OK & XML or json (user)

	401 Unauthorized
	404 Not Found (user not found)

Note: Only the user herself or an admin user in the admin group can modify a user.
Example of query:

```
curl -X PUT \
-u login:password \
"apiusers/groups/readerd_Cesselin/users/toto"
```

Removing an existing user from an existing group

URL	apiusers/groups/[groupname]/users/[login]
Method	DELETE
ContentType	application/xml
	application/json
Returns	204 No Content & XML or json (user)
	401 Unauthorized
	404 Not Found (user or group)

Note: Only the user herself or an admin user in the admin group can delete a user.
Example of query:

```
curl -X DELETE \
-u login:password \
"apiusers/users/login/groups/readerd_Cesselin"
```

List of roles with access to a dictionary

URL	apiusers/dictionary/[dictname]
Method	GET
ContentType	application/xml
	application/json
Returns	200 OK & list of groups (XML or json)
	401 Unauthorized
	404 Not Found (dictionary)

Example of query:

```
curl -H 'Accept: application/json'
"apiusers/dictionary/Cesselin"
```

List of users with a role with access to a dictionary

URL	apiusers/dictionary/[dictname]/[role]
Method	GET

ContentType	application/xml
	application/json
Returns	200 OK & list of groups (XML or json)
	401 Unauthorized
	404 Not Found (dictionary or role)

Example of query:

```
    curl -H 'Accept: application/json'
 "apiusers/dictionary/Cesselin/reader"
```

Gives a role related to a dictionary to a user

URL	apiusers/dictionary/[dictname]/[role]/[login]
Method	PUT
ContentType	application/xml
	application/json
Returns	200 OK & XML or json (user)
	401 Unauthorized
	404 Not Found (dictionary or user not found)

Note: Only an admin or dictionary admin can give a role to a user.
Example of query:

```
    curl -X PUT \
    -u login:password \
    "apiusers/dictionary/Cesselin/reader/toto"
```

Removing a role from a user in a dictionary

URL	apiusers/dictionary/[dictname]/[role]/[login]
Method	DELETE
ContentType	application/xml
	application/json
Returns	204 No Content & XML or json (user)
	401 Unauthorized
	404 Not Found (dictionary, role or user)

Note: Only an admin or dictionary admin can remove a role to a user.
Example of query:

```
    curl -X DELETE \
    -u login:password \
    "apiusers/users/login/groups/readerd_Cesselin"
```

Authentication methods

Authentication method is one of the following:

BasicAuth	login and password are encoded in base 64 and sent to the server in the request header. This method is not secure because the encryption method is easy to decrypt.
Authentication cookie	cookies are sent to the server in the request header. In order to ask for an authentication cookie, please visit the login page.
Querystring	login=login
	password=password
	This method is not secure at all : login and password are sent in cleartext. They will even be registered on the web server logs!

Misc

If you need more help, send an email to Mathieu Mangeot.