API Documentation – Video Games

Table of contents

Global Information:

API URL: http://apixml.josealbea.com/

- Parts of URLs that are surrounded with brackets ("[ ]") in red are optional.
- Parts of URLs that are preceded by a two points (":") in red are parameter values to be replaced with your chosen value.
- This API furnishes you a PHP filled with a detailed test data file for every single entity (Game, Console, Gender etc.) for POST and PUT methods.
- Some parameters must be sent in the HTTP headers with the request in order to identify the user:
    - PUB: user's public key, generated by creating a hash_hmac signature, encoded in sha256, with the following parameters: idUser, email, time(), apiKey, apiSecret. Example: hash_hmac("sha256", '1' . 'test@yopmail.fr' . time() . 'dardnjdjv382YBSdgh', 'aurezhfusdfhjdsfheztj7623B HSDG121dfdf');
    - USEREMAIL: user's email address
    - APIKEY: user's unique API key
- Everything is paginated 2 per 2, which means that, by default, it will show the first page. If you want to show another page, you simply need to add this part to the URL: "/page/:page". Be careful, you can't use both id and page parameters in the URL.
- We created two users (an admin and a simple user) for you:
    - Admin:
        - Username: admin
        - Password: admin (encrypted in md5)
        - Email: admin-api@yopmail.fr
        - ApiKey: 557afa6e50b9a
        - ApiSecret: 57b59e43b10aa25
    - User:
        - Username: user
        - Password: azerty (encrypted in md5)
        - Email: user-api@yopmail.fr
        - ApiKey: 557afac355c84
        - ApiSecret: c87521e8a6ff3ac

Game

Game – Read data of a game

Method: GET
URL: /game/index[/id/:id]
Return: XML
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Game's ID	No

Game

Game – Create a new game

Method: POST
URL: /game
Return: The ID of the inserted game.
Permission: admin

Additional information: You can either create only the game, without any of its information, either create the whole game at once. If you want to do this, you can simply add the specific nodes of every other entities (that will follow this documentation) to the POST array.

Parameters

Field	Type	Description	Required
title	String(255)	Game's title	Yes
site	URI	Game's website	Yes

Game

Game – Update an existing game

Method: PUT
URL: /game/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Additional information: *You can either update only the game data, without any of its information, either update the whole game at once. If you want to do this, you can simply add the specific nodes of every other entities (that will follow this documentation) to the PUT array.*
If you decide to update the following associated entities, you have two choices:
- *Gender*
- *Editor*
- *Theme*
- *Language*

Either you specify only the id of the entry to update, which will only associate this id to the console, either you don't specify the id, in which case it will create a new entry (for example a new mode) and associate it.
You can find examples in the testData.php file.

Parameters

Field	Type	Description	Required
title	Date	Game's title	Yes
site	URI	Game's website	Yes

Game

Game – Delete an existing game

Method: DELETE
URL: /game/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Game's ID to delete	Yes

Game – Delete an existing game

Gender

Gender – Read data of an gender

Method: GET
URL: /gender/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Gender's ID	No

Gender

<u>Gender – Create a new gender</u>
Method: POST
URL: /gender
Return: The ID of the inserted gender.
Permission: admin

Parameters

Field	Type	Description	Required
gender	String(255)	Gender's name	Yes

Gender

Gender – Update an existing gender

Method: PUT
URL: /gender/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
gender	String(255)	Gender's name	Yes

Gender

Gender – Delete an existing gender

This method has been disabled as these data are not supposed to disappear. Plus, the gender can be used for several entries, which could break them if deleted.

Editor

Editor – Read data of an editor

Method: GET
URL :   /editor/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Editor's ID	No

Editor

<u>Editor – Create a new editor</u>
Method: POST
URL: /editor
Return: The ID of the inserted editor.
Permission: admin

Parameters

Field	Type	Description	Required
editor	String(255)	Editor's name	Yes

Editor

Editor – Update an existing editor

Method: PUT
URL: /editor/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
editor	String(255)	Editor's name	Yes

Editor

Editor – Delete an existing editor
This method has been disabled as these data are not supposed to disappear. Plus, the editor can be used for several entries, which could break them if deleted.

Theme

Theme – Read data of a theme

Method: GET
URL: /theme/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Theme's ID	No

Theme

<u>Theme – Create a new theme</u>
Method: POST
URL: /theme
Return: The ID of the inserted theme.
Permission: admin

Parameters

Field	Type	Description	Required
theme	String(255)	Theme's name	Yes

Theme

<u>Theme – Update an existing theme</u>
Method: PUT
URL: /theme/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
theme	String(255)	Theme's name	Yes

Theme

Theme – Delete an existing theme

This method has been disabled as these data are not supposed to disappear. Plus, the theme can be used for several entries, which could break them if deleted.

Console

Console – Read data of a console
Method: GET
URL: /console/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Comment's ID	No

Console

<u>Console – Create a new console</u>
Method: POST
URL: /comment
Return: The ID of the inserted console.
Permission: admin

Additional information: *You can also insert every associated entities (modes, supports, editions, shops, dlcs, configs, tests, comments and analyses) at the same time. See example file furnished by the API.*

Parameters

Field	Type	Description	Required
business_model	String(255)	Economic mode of the game (free to play, …)	Yes
pegi	String(255)	Game's PEGI according to this console	Yes
release	Date (Y-m-d)	Release date on this console	Yes
name	String(255)	Console's name	Yes
description	Int	Console's description	Yes
cover_front	String(255)	Front cover URL	Yes
cover_back	String(255)	Back cover URL	Yes
game_idGame	Int	Game's ID to which the console is attached to	Yes

Console

Console – Update an existing console

Method: PUT
URL: /console/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Additional information: You can also update every associated entities (modes, supports, editions, shops, dlcs, configs, tests, comments and analyses) at the same time. See example file furnished by the API.
If you decide to update the following associated entities, you have two choices:
- *Mode*
- *Support*
Either you specify only the id of the entry to update, which will only associate this id to the console, either you don't specify the id, in which case it will create a new entry (for example a new mode) and associate it.
You can find examples in the testData.php file.

Parameters

Field	Type	Description	Required
business_model	String(255)	Economic mode of the game (free to play, …)	Yes
pegi	String(255)	Game's PEGI according to this console	Yes
release	Date (Y-m-d)	Release date on this console	Yes
name	String(255)	Console's name	Yes
description	Int	Console's description	Yes
cover_front	String(255)	Front cover URL	Yes
cover_back	String(255)	Back cover URL	Yes
game_idGame	Int	Game's ID to which the console is attached to	Yes

Console

Console – Delete an existing console

Method: DELETE
URL :   /console/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Comment's ID to delete	Yes

Console – Delete an existing console

Mode

Mode – Read data of a mode

Method: GET
URL: /mode/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Mode's ID	No

Mode

Mode – Create a new mode

Method: POST
URL: /mode
Return: The ID of the inserted mode.
Permission: admin

Parameters

Field	Type	Description	Required
mode	String(255)	Mode's name	Yes

Mode

<u>Mode – Update an existing mode</u>
Method: PUT
URL: /mode/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
mode	String(255)	Mode's name	Yes

Mode

Mode – Delete an existing mode
This method has been disabled as these data are not supposed to disappear. Plus, the mode can be used for several entries, which could break them if deleted.

Support

Support – Read data of a support
Method: GET
URL: /support/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	The support's ID	No

Support

<u>Support – Create a new support</u>
Method: POST
URL: /support
Return: The ID of the inserted support.
Permission: admin

Parameters

Field	Type	Description	Required
support	String(255)	Support's name	Yes

Support

<u>Support – Update an existing support</u>
Method: PUT
URL: /support/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
support	String(255)	Support's name	Yes

Support

Support – Delete an existing support

This method has been disabled as these data are not supposed to disappear. Plus, the support can be used for several entries, which could break them if deleted.

Edition

Edition – Create a new edition

Method: POST
URL: /edition
Return: The ID of the inserted edition.
Permission: admin

Additional information: You can also insert every associated shops at the same time. See example file furnished by the API.

Parameters

Field	Type	Description	Required
name	String(255)	Edition's name	Yes
content	Longtext	Edition's content	Yes
console_idConsole	Int	Console's ID to which the edition is attached to	Yes

Edition

Edition – Update an existing edition

Method: PUT
URL: /edition/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Additional information: You can also update every associated shops at the same time. See example file furnished by the API.

Parameters

Field	Type	Description	Required
name	String(255)	Edition's name	Yes
content	Longtext	Edition's content	Yes
console_idConsole	Int	Console's ID to which the edition is attached to	Yes

Edition

Edition – Delete an existing edition

Method: DELETE
URL :   /edition/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Edition's ID to delete	Yes

Shop

Shop – Read data of a shop
Method: GET
URL: /shop/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Shop's ID	No

Shop

<u>Shop – Create a new shop</u>
Method: POST
URL: /shop
Return: The ID of the inserted shop.
Permission: admin

Parameters

Field	Type	Description	Required
url	String(255)	Game's shop URL	Yes
name	String(255)	Game's shop name	Yes
price	Float	Game's shop price (decimal separator: dot ("."))	Yes
devise	String(255)	Game's shop device	Yes
edition_idEdition	Int	Edition's ID to which is attached the shop	Yes

Shop

Shop – Update an existing shop

Method: PUT
URL: /shop/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
url	String(255)	Game's shop URL	Yes
name	String(255)	Game's shop name	Yes
price	Float	Game's shop price (decimal separator: dot ("."))	Yes
devise	String(255)	Game's shop device	Yes
edition_idEdition	Int	Edition's ID to which is attached the shop	Yes

Shop

Shop – Delete an existing shop
Method: DELETE
URL :   /shop/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Shop's ID to delete	Yes

Shop – Delete an existing shop

Dlc

Dlc

<u>Dlc – Read data of a dlc</u>
Method: GET
URL: /dlc/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Dlc's ID	No

Dlc

Dlc – Create a new dlc

Method: POST
URL: /dlc
Return: The ID of the inserted dlc.
Permission: admin

Parameters

Field	Type	Description	Required
title	String(255)	Dlc's title / name	Yes
description	Longtext	Dlc's description	Yes
price	Float	Dlc price (decimal separator: dot ("."))	Yes
devise	String(255)	Dlc's devise	Yes
console_idConsole	Int	Console's ID to which is attached the shop	Yes

Dlc

Dlc – Update an existing dlc

Method: PUT
URL: /dlc/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
title	String(255)	Dlc's title / name	Yes
description	Longtext	Dlc's description	Yes
price	Float	Dlc price (decimal separator: dot ("."))	Yes
devise	String(255)	Dlc's devise	Yes
console_idConsole	Int	Console's ID to which is attached the dlc	Yes

Dlc

<u>Dlc – Delete an existing dlc</u>
Method: DELETE
URL :   /dlc/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Dlc's ID to delete	Yes

<u>Dlc – Delete an existing dlc</u>

Config

Config – Read data of a config
Method: GET
URL: /config/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Config's ID	No

Config

<u>Config – Create a new config</u>
Method: POST
URL: /config
Return: The ID of the inserted config.
Permission: admin

Parameters

Field	Type	Description	Required
config	Longtext	Config's name	Yes
type	String(255)	Config's type (minimal, optimal etc.)	Yes
console_idConsole	Int	Console's ID to which is attached the config	Yes

Config

Config – Update an existing config

Method: PUT
URL: /config/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
config	Longtext	Config's name	Yes
type	String(255)	Config's type (minimal, optimal etc.)	Yes
console_idConsole	Int	Console's ID to which is attached the config	Yes

Config

<u>Config – Delete an existing config</u>
Method: DELETE
URL :   /config/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Config's ID to delete	Yes

Test

Test

<u>Test – Read data of a test</u>
Method: GET
URL: /test/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Test's ID	No

Test

Test – Create a new test

Method: POST
URL: /test
Return: The ID of the inserted test.
Permission: admin

Additional information: You can also insert every associated entities (comments and analyses) at the same time. See example file furnished by the API.

Parameters

Field	Type	Description	Required
report	Longtext	Test's name	Yes
date	Datetime	Test's creation date (format Y-m-d H:i:s)	Yes
user_name	String(255)	Test's author	Yes
note	Int(2)	Test's note	Yes
console_idConsole	Int	Console's ID to which is attached the config	Yes

Test

<u>Test – Update an existing test</u>

Method: PUT
URL: /test/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

<u>Additional information:</u> You can also update every associated entities (comments and analyses) at the same time. See example file furnished by the API.

Parameters

Field	Type	Description	Required
report	Longtext	Test's name	Yes
date	Datetime	Test's creation date (format Y-m-d H:i:s)	Yes
user_name	String(255)	Test's author	Yes
note	Int(2)	Test's note	Yes
console_idConsole	Int	Console's ID to which is attached the config	Yes

Test

Test – Delete an existing test

Method: DELETE
URL :   /test/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Test's ID to delete	Yes

Comment

Comment – Read data of a comment

Method: GET
URL :   /comment/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	The Comment's-ID	Yes

Comment

<u>Comment – Create a new comment</u>
Method: POST
URL: /comment
Return: The ID of the inserted comment.
Permission: admin

Parameters

Field	Type	Description	Required
date	Date	Date with Y-m-d format	Yes
user_name	String	User's name	Yes
note	Int	Note of the comment	Yes
like	Int	Number of likes	Yes
dislike	Int	Number of dislikes	Yes
text	String	Comment's content	Yes
test_idTest	Int	ID of the test associated with	Yes

Comment

Comment – Update an existing comment
Method: PUT
URL: /comment/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
date	Date	Date with Y-m-d format	Yes
user_name	String	User's name	Yes
note	Int	Note of the comment	Yes
like	Int	Number of likes	Yes
dislike	Int	Number of dislikes	Yes
text	String	Comment's content	Yes
test_idTest	Int	ID of the test associated with	Yes

Comment

<u>Comment – Delete an existing comment</u>
Method: DELETE
URL: /comment/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an
error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Comment's ID to delete	Yes

<u>Comment – Delete an existing comment</u>

Analyse

Analyse – Read data of an analysis
Method:　　　GET
URL:　　　　/analyse/index[/id/:id]
Permission:　user

Parameters

Field	Type	Description	Required
id	Int	Analysis' ID	No

Analyse

<u>Analyse – Create a new analysis</u>
Method: POST
URL: /analyse
Return: The ID of the inserted analysis.
Permission: admin

Parameters

Field	Type	Description	Required
analyse	Longtext	Analysis' name	Yes
type	Datetime	Analysis' type (positive, negative etc.)	Yes
test_idTest	Int	Test's ID to which is attached the analysis	Yes

Analyse

<u>Analyse – Update an existing analysis</u>
Method: PUT
URL: /analyse/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
analyse	Longtext	Analysis' name	Yes
type	Datetime	Analysis' type (positive, negative etc.)	Yes
test_idTest	Int	Test's ID to which is attached the analysis	Yes

Analyse

Analyse – Delete an existing analysis

Method: DELETE
URL :   /analyse/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Analysis' ID to delete	Yes

Analyse – Delete an existing analysis

Language

Language

<u>Language – Read data of a language</u>
Method: GET
URL: /language/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Language's ID	No

BOUAZIZ Axel – Albea José – Teisseire Yohann – Traing Philippe   WebService & XML

Language

<u>Language – Create a new language</u>
Method: POST
URL: /language
Return: The ID of the inserted language.
Permission: admin

Parameters

Field	Type	Description	Required
language	String(255)	Language's name	Yes

Language

<u>Language – Update an existing language</u>
Method: PUT
URL: /language/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an
error.
Permission: admin

Parameters

Field	Type	Description	Required
language	String(255)	Language's name	Yes

<u>Language – Update an existing language</u>

Language

<u>Language – Delete an existing language</u>
Method: DELETE
URL: /language/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Language's ID to delete	Yes

Article

Article – Read data of an article
Method: GET
URL: /article/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Article's ID	No

Article

<u>Article – Create a new article</u>
Method: POST
URL: /article
Return: The ID of the inserted article.
Permission: admin

Parameters

Field	Type	Description	Required
type	String(255)	Article's type (e.g.: news etc.)	Yes
title	String(255)	Article's title	Yes
user_name	String(255)	Article's author	Yes
date	Datetime	Article's creation date (format Y-m-d H:i:s)	Yes
console_ names	String(255)	Article's console names, comma separated (",")	Yes
game_idGame	Int	Game's ID to which is attached the article	Yes

Article

Article – Update an existing article

Method: PUT
URL: /article/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
type	String(255)	Article's type (e.g.: news etc.)	Yes
title	String(255)	Article's title	Yes
user_name	String(255)	Article's author	Yes
date	Datetime	Article's creation date (format Y-m-d H:i:s)	Yes
console_ names	String(255)	Article's console names, comma separated (",")	Yes
game_idGame	Int	Game's ID to which is attached the article	Yes

Article

Article – Delete an existing article

Method: DELETE
URL :   /article/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Language's ID to delete	Yes

Media

Media – Read data of an media

Method: GET
URL: /media/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Media's ID	No

Media

Media – Create a new media

Method: POST
URL: /media
Return: The ID of the inserted media.
Permission: admin

Parameters

Field	Type	Description	Required
type	String(255)	Media's type (e.g.: picture, video etc.)	Yes
url	String(255)	Media's URL path	Yes
unit	String(255)	Media's unit (e.g.: cm, px etc.)	Yes
width	Float	Media's width (in the unit specified in unit column)	Yes
height	Float	Media's height (in the unit specified in unit column)	Yes
console_names	String(255)	Media's console names, comma separated (",")	Yes
game_idGame	Int	Game's ID to which is attached the article	Yes

Media

Media – Update an existing media

Method: PUT
URL: /media/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
type	String(255)	Media's type (e.g.: picture, video etc.)	Yes
url	String(255)	Media's URL path	Yes
unit	String(255)	Media's unit (e.g.: cm, px etc.)	Yes
width	Float	Media's width (in the unit specified in unit column)	Yes
height	Float	Media's height (in the unit specified in unit column)	Yes
console_names	String(255)	Media's console names, comma separated (",")	Yes
game_idGame	Int	Game's ID to which is attached the article	Yes

Media

<u>Media – Delete an existing media</u>
Method: DELETE
URL :   /media/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Media's ID to delete	Yes

Tip

Tip

<u>Tip – Read data of an tip</u>
Method: GET
URL: /tip/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Tip's ID	No

Tip

<u>Tip – Create a new tip</u>
Method: POST
URL: /tip
Return: The ID of the inserted tip.
Permission: admin

Parameters

Field	Type	Description	Required
content	Longtext	Tip itself	Yes
console_names	String(255)	Tip's console names, comma separated (",")	Yes
game_idGame	Int	Game's ID to which is attached the article	Yes

Tip

Tip – Update an existing tip
Method: PUT
URL: /tip/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
content	Longtext	Tip itself	Yes
console_names	String(255)	Tip's console names, comma separated (",")	Yes
game_idGame	Int	Game's ID to which is attached the article	Yes

Tip

<u>Tip – Delete an existing tip</u>
Method: DELETE
URL :   /tip/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	Tip's ID to delete	Yes

<u>Tip – Delete an existing tip</u>

User

User – Read data of a user
Method: GET
URL: /user/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	User's ID	No

User

User – Create a new user
Method: POST
URL: /user
Return: The ID of the inserted user.
Permission: admin

Additional information: When a user is created, both apiKey and apiSecret are generated automatically and can be found with the entries "apiKey" and "apiSecret" in the XML generated by a GET method. These values are used to identify the request and manage roles.

Parameters

Field	Type	Description	Required
email	String(255)	User's email	Yes
username	String(255)	User's username	Yes
password	String(255)	User's password (md5 encrypted)	Yes
role	Int	User's role ID	Yes

User

<u>User – Update an existing user</u>
Method: PUT
URL: /user/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
email	String(255)	User's email	Yes
username	String(255)	User's username	Yes
password	String(255)	User's password (md5 encrypted)	Yes
role	Int	User's role ID	Yes

User

User – Delete an existing user

Method: DELETE
URL :   /user/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
id	Int	User's ID to delete	Yes

Role

Additional information: There are two roles defined by the API: user (id 1) and admin (id 2). These roles **CAN NOT, IN ANY WAY** be deleted, otherwise the API won't work as they are used for the authentication. Only users with "ADMIN" role can access a page regarding the roles.

Role – Read data of a role

Method: GET
URL: /role/index[/id/:id]
Permission: user

Parameters

Field	Type	Description	Required
id	Int	Role's ID	No

Role

<u>Role – Create a new role</u>
Method: POST
URL: /role
Return: The ID of the inserted role.
Permission: admin

Parameters

Field	Type	Description	Required
role	String(255)	Role's name	Yes

Role

<u>Role – Update an existing role</u>
Method: PUT
URL: /role/index[/id/:id]
Return: Boolean. If everything went well, returns true, otherwise it returns an error.
Permission: admin

Parameters

Field	Type	Description	Required
role	String(255)	Role's name	Yes

Role

Role – Delete an existing role

This method has been disabled as these data are not supposed to be deleted without breaking the whole API (used to authenticate the users when they make a request).