

# API Documentation – Video Games

---

## Table of contents

Global Information:	4
Game	5
Game – Read data of a game	5
Game – Create a new game	6
Game – Update an existing game	7
Game – Delete an existing game	8
Gender	9
Gender – Read data of an gender	9
Gender – Create a new gender	10
Gender – Update an existing gender	11
Gender – Delete an existing gender	12
Editor	13
Editor – Read data of an editor	13
Editor – Create a new editor	14
Editor – Update an existing editor	15
Editor – Delete an existing editor	16
Theme	17
Theme – Read data of a theme	17
Theme – Create a new theme	18
Theme – Update an existing theme	19
Theme – Delete an existing theme	20
Console	21
Console – Read data of a console	21
Console – Create a new console	22
Console – Update an existing console	23
Console – Delete an existing console	24
Mode	25
Mode – Read data of a mode	25
Mode – Create a new mode	26
Mode – Update an existing mode	27
Mode – Delete an existing mode	28
Support	29

Support – Read data of a support.....	29
Support – Create a new support .....	30
Support – Update an existing support.....	31
Support – Delete an existing support.....	32
Edition.....	33
Edition – Read data of an edition.....	33
Edition – Create a new edition.....	34
Edition – Update an existing edition .....	35
Edition – Delete an existing edition.....	36
Shop .....	37
Shop – Read data of a shop.....	37
Shop – Create a new shop .....	38
Shop – Update an existing shop .....	39
Shop – Delete an existing shop .....	40
Dlc.....	41
Dlc – Read data of a dlc .....	41
Dlc – Create a new dlc .....	42
Dlc – Update an existing dlc.....	43
Dlc – Delete an existing dlc.....	44
Config .....	45
Config – Read data of a config .....	45
Config – Create a new config .....	46
Config – Update an existing config .....	47
Config – Delete an existing config.....	48
Test.....	49
Test – Read data of a test .....	49
Test – Create a new test .....	50
Test – Update an existing test .....	51
Test – Delete an existing test.....	52
Comment.....	53
Comment – Read data of a comment.....	53
Comment – Create a new comment.....	54
Comment – Update an existing comment.....	55
Comment – Delete an existing comment .....	56
Analyse .....	57
Analyse – Read data of an analysis .....	57

Analyse – Create a new analysis .....	58
Analyse – Update an existing analysis .....	59
Analyse – Delete an existing analysis.....	60
Language .....	61
Language – Read data of a language.....	61
Language – Create a new language .....	62
Language – Update an existing language.....	63
Language – Delete an existing language .....	64
Article.....	65
Article – Read data of an article .....	65
Article – Create a new article .....	66
Article – Update an existing article.....	67
Article – Delete an existing article.....	68
Media .....	69
Media – Read data of an media.....	69
Media – Create a new media.....	70
Media – Update an existing media .....	71
Media – Delete an existing media .....	72
Tip.....	73
Tip – Read data of an tip .....	73
Tip – Create a new tip .....	74
Tip – Update an existing tip .....	75
Tip – Delete an existing tip.....	76
User .....	77
User – Read data of a user .....	77
User – Create a new user .....	78
User – Update an existing user .....	79
User – Delete an existing user.....	80
Role .....	81
Role – Read data of a role .....	81
Role – Create a new role.....	82
Role – Update an existing role .....	83
Role – Delete an existing role.....	84

## **Global Information:**

- 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() . 'dardnjdv382YBSdgh', 'aurezhfusdfhjdsfheztj7623BHSDG121dfdf');
  - USEREMAIL: user's email address
  - APIKEY: user's unique API key

## Game

## Game

### Game – Read data of a game

Method: GET

URL: /game/index[/id/:id]

Return: XML

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	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
<b>title</b>	String(255)	Game's title	Yes
<b>site</b>	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
<b>title</b>	Date	Game's title	Yes
<b>site</b>	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
<b>id</b>	Int	Game's ID to delete	Yes



## Gender

## 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

### Gender – Create a new gender

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

### Parameters

Field	Type	Description	Required
<b>gender</b>	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
<b>gender</b>	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

### Editor – Read data of an editor

Method: GET

URL : /editor/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	Int	Editor's ID	No

## Editor

### Editor – Create a new editor

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

### Parameters

Field	Type	Description	Required
<b>editor</b>	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
<b>editor</b>	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

### Theme – Read data of a theme

Method: GET

URL: /theme/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	Int	Theme's ID	No

## Theme

### Theme – Create a new theme

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

### Parameters

Field	Type	Description	Required
<b>theme</b>	String(255)	Theme's name	Yes

## Theme

### Theme – Update an existing theme

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

### Console – Read data of a console

Method: GET

URL: /console/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	Int	Comment's ID	No

## Console

### Console – Create a new console

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
<b>business_model</b>	String(255)	Economic mode of the game (free to play, ...)	Yes
<b>pegi</b>	String(255)	Game's PEGI according to this console	Yes
<b>release</b>	Date (Y-m-d)	Release date on this console	Yes
<b>name</b>	String(255)	Console's name	Yes
<b>description</b>	Int	Console's description	Yes
<b>cover_front</b>	String(255)	Front cover URL	Yes
<b>cover_back</b>	String(255)	Back cover URL	Yes
<b>game_idGame</b>	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
<b>business_model</b>	String(255)	Economic mode of the game (free to play, ...)	Yes
<b>pegi</b>	String(255)	Game's PEGI according to this console	Yes
<b>release</b>	Date (Y-m-d)	Release date on this console	Yes
<b>name</b>	String(255)	Console's name	Yes
<b>description</b>	Int	Console's description	Yes
<b>cover_front</b>	String(255)	Front cover URL	Yes
<b>cover_back</b>	String(255)	Back cover URL	Yes
<b>game_idGame</b>	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
<b>id</b>	Int	Comment's ID to delete	Yes



## Mode

## Mode

### Mode – Read data of a mode

Method: GET

URL: /mode/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	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
<b>mode</b>	String(255)	Mode's name	Yes

## Mode

### Mode – Update an existing mode

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
<b>mode</b>	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

### Support – Read data of a support

Method: GET

URL: /support/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	Int	The support's ID	No

## Support

### Support – Create a new support

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

### Parameters

Field	Type	Description	Required
<b>support</b>	String(255)	Support's name	Yes

## Support

### Support – Update an existing support

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
<b>support</b>	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

### Edition – Read data of an edition

Method: GET

URL: /edition/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	Int	Edition's ID	No

## 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
<b>name</b>	String(255)	Edition's name	Yes
<b>content</b>	Longtext	Edition's content	Yes
<b>console_idConsole</b>	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
<b>name</b>	String(255)	Edition's name	Yes
<b>content</b>	Longtext	Edition's content	Yes
<b>console_idConsole</b>	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
<b>id</b>	Int	Edition's ID to delete	Yes

## Shop

## 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

### Shop – Create a new shop

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

### Parameters

Field	Type	Description	Required
<b>url</b>	String(255)	Game's shop URL	Yes
<b>name</b>	String(255)	Game's shop name	Yes
<b>price</b>	Float	Game's shop price (decimal separator: dot ("."))	Yes
<b>devise</b>	String(255)	Game's shop device	Yes
<b>edition_idEdition</b>	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
<b>url</b>	String(255)	Game's shop URL	Yes
<b>name</b>	String(255)	Game's shop name	Yes
<b>price</b>	Float	Game's shop price (decimal separator: dot ( "." ))	Yes
<b>devise</b>	String(255)	Game's shop devise	Yes
<b>edition_idEdition</b>	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
<b>id</b>	Int	Shop's ID to delete	Yes



## Dlc

## Dlc

### Dlc – Read data of a dlc

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

## Dlc

### Dlc – Delete an existing dlc

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
<b>id</b>	Int	Dlc's ID to delete	Yes

## Config

## 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

### Config – Create a new config

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

### Parameters

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

## Config

### Config – Delete an existing config

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
<b>id</b>	Int	Config's ID to delete	Yes



## Test

## Test

### Test – Read data of a test

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

## Test

### Test – Update an existing test

Method: PUT

URL: /test/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 (comments and analyses) at the same time. See example file furnished by the API.

### Parameters

Field	Type	Description	Required
<b>report</b>	Longtext	Test's name	Yes
<b>date</b>	Datetime	Test's creation date (format Y-m-d H:i:s)	Yes
<b>user_name</b>	String(255)	Test's author	Yes
<b>note</b>	Int(2)	Test's note	Yes
<b>console_idConsole</b>	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
<b>id</b>	Int	Test's ID to delete	Yes

## Comment

## Comment

### Comment – Read data of a comment

Method: GET

URL : /comment/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	Int	The Comment's-ID	Yes

## Comment

### Comment – Create a new comment

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

### Parameters

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

## Comment

### Comment – Delete an existing comment

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
<b>id</b>	Int	Comment's ID to delete	Yes



## Analyse

## Analyse

### Analyse – Read data of an analysis

Method: GET

URL: /analyse/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
<b>id</b>	Int	Analysis' ID	No

## Analyse

### Analyse – Create a new analysis

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

### Parameters

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

## Analyse

### Analyse – Update an existing analysis

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
<b>analyse</b>	Longtext	Analysis' name	Yes
<b>type</b>	Datetime	Analysis' type (positive, negative etc.)	Yes
<b>test_idTest</b>	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
<b>id</b>	Int	Analysis' ID to delete	Yes

## Language

## Language

### Language – Read data of a language

Method: GET

URL: /language/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
id	Int	Language's ID	No

## Language

### Language – Create a new language

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

### Parameters

Field	Type	Description	Required
<b>language</b>	String(255)	Language's name	Yes

## Language

### Language – Update an existing language

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
<b>language</b>	String(255)	Language's name	Yes

## Language

### Language – Delete an existing language

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
<b>id</b>	Int	Language's ID to delete	Yes



## Article

## 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

### Article – Create a new article

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

### Parameters

Field	Type	Description	Required
<b>type</b>	String(255)	Article's type (e.g.: news etc.)	Yes
<b>title</b>	String(255)	Article's title	Yes
<b>user_name</b>	String(255)	Article's author	Yes
<b>date</b>	Datetime	Article's creation date (format Y-m-d H:i:s)	Yes
<b>console_names</b>	String(255)	Article's console names, comma separated (",")	Yes
<b>game_idGame</b>	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
<b>type</b>	String(255)	Article's type (e.g.: news etc.)	Yes
<b>title</b>	String(255)	Article's title	Yes
<b>user_name</b>	String(255)	Article's author	Yes
<b>date</b>	Datetime	Article's creation date (format Y-m-d H:i:s)	Yes
<b>console_names</b>	String(255)	Article's console names, comma separated (",")	Yes
<b>game_idGame</b>	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
<b>id</b>	Int	Language's ID to delete	Yes

## Media

## Media

### Media – Read data of an media

Method: GET

URL: /media/index[/id/:id]

Permission: user

### Parameters

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

## Media

### Media – Delete an existing media

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
<b>id</b>	Int	Media's ID to delete	Yes



## Tip

## Tip

### Tip – Read data of an tip

Method: GET

URL: /tip/index[/id/:id]

Permission: user

### Parameters

Field	Type	Description	Required
id	Int	Tip's ID	No

## Tip

### Tip – Create a new tip

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

### Parameters

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

## Tip

### Tip – Delete an existing tip

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
<b>id</b>	Int	Tip's ID to delete	Yes

## User

## User

### User – Read data of a user

Method: GET

URL: /user/index[/id/:id]

Permission: user

### Parameters

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

## User

### User – Update an existing user

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
<b>email</b>	String(255)	User's email	Yes
<b>username</b>	String(255)	User's username	Yes
<b>password</b>	String(255)	User's password (md5 encrypted)	Yes
<b>role</b>	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
<b>id</b>	Int	User's ID to delete	Yes



## Role

## 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

### Role – Create a new role

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

### Parameters

Field	Type	Description	Required
<b>role</b>	String(255)	Role's name	Yes

## Role

### Role – Update an existing role

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
<b>role</b>	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).