API Documentation – Video Games

Table of contents

[Global Information: 4](#_Toc421827630)

[Game 5](#_Toc421827631)

[Game – Read data of a game 5](#_Toc421827632)

[Game – Create a new game 6](#_Toc421827633)

[Game – Update an existing game 7](#_Toc421827634)

[Game – Delete an existing game 8](#_Toc421827635)

[Gender 9](#_Toc421827636)

[Gender – Read data of an gender 9](#_Toc421827637)

[Gender – Create a new gender 10](#_Toc421827638)

[Gender – Update an existing gender 11](#_Toc421827639)

[Gender – Delete an existing gender 12](#_Toc421827640)

[Editor 13](#_Toc421827641)

[Editor – Read data of an editor 13](#_Toc421827642)

[Editor – Create a new editor 14](#_Toc421827643)

[Editor – Update an existing editor 15](#_Toc421827644)

[Editor – Delete an existing editor 16](#_Toc421827645)

[Theme 17](#_Toc421827646)

[Theme – Read data of a theme 17](#_Toc421827647)

[Theme – Create a new theme 18](#_Toc421827648)

[Theme – Update an existing theme 19](#_Toc421827649)

[Theme – Delete an existing theme 20](#_Toc421827650)

[Console 21](#_Toc421827651)

[Console – Read data of a console 21](#_Toc421827652)

[Console – Create a new console 22](#_Toc421827653)

[Console – Update an existing console 23](#_Toc421827654)

[Console – Delete an existing console 24](#_Toc421827655)

[Mode 25](#_Toc421827656)

[Mode – Read data of a mode 25](#_Toc421827657)

[Mode – Create a new mode 26](#_Toc421827658)

[Mode – Update an existing mode 27](#_Toc421827659)

[Mode – Delete an existing mode 28](#_Toc421827660)

[Support 29](#_Toc421827661)

[Support – Read data of a support 29](#_Toc421827662)

[Support – Create a new support 30](#_Toc421827663)

[Support – Update an existing support 31](#_Toc421827664)

[Support – Delete an existing support 32](#_Toc421827665)

[Edition 33](#_Toc421827666)

[Edition – Read data of an edition 33](#_Toc421827667)

[Edition – Create a new edition 34](#_Toc421827668)

[Edition – Update an existing edition 35](#_Toc421827669)

[Edition – Delete an existing edition 36](#_Toc421827670)

[Shop 37](#_Toc421827671)

[Shop – Read data of a shop 37](#_Toc421827672)

[Shop – Create a new shop 38](#_Toc421827673)

[Shop – Update an existing shop 39](#_Toc421827674)

[Shop – Delete an existing shop 40](#_Toc421827675)

[Dlc 41](#_Toc421827676)

[Dlc – Read data of a dlc 41](#_Toc421827677)

[Dlc – Create a new dlc 42](#_Toc421827678)

[Dlc – Update an existing dlc 43](#_Toc421827679)

[Dlc – Delete an existing dlc 44](#_Toc421827680)

[Config 45](#_Toc421827681)

[Config – Read data of a config 45](#_Toc421827682)

[Config – Create a new config 46](#_Toc421827683)

[Config – Update an existing config 47](#_Toc421827684)

[Config – Delete an existing config 48](#_Toc421827685)

[Test 49](#_Toc421827686)

[Test – Read data of a test 49](#_Toc421827687)

[Test – Create a new test 50](#_Toc421827688)

[Test – Update an existing test 51](#_Toc421827689)

[Test – Delete an existing test 52](#_Toc421827690)

[Comment 53](#_Toc421827691)

[Comment – Read data of a comment 53](#_Toc421827692)

[Comment – Create a new comment 54](#_Toc421827693)

[Comment – Update an existing comment 55](#_Toc421827694)

[Comment – Delete an existing comment 56](#_Toc421827695)

[Analyse 57](#_Toc421827696)

[Analyse – Read data of an analysis 57](#_Toc421827697)

[Analyse – Create a new analysis 58](#_Toc421827698)

[Analyse – Update an existing analysis 59](#_Toc421827699)

[Analyse – Delete an existing analysis 60](#_Toc421827700)

[Language 61](#_Toc421827701)

[Language – Read data of a language 61](#_Toc421827702)

[Language – Create a new language 62](#_Toc421827703)

[Language – Update an existing language 63](#_Toc421827704)

[Language – Delete an existing language 64](#_Toc421827705)

[Article 65](#_Toc421827706)

[Article – Read data of an article 65](#_Toc421827707)

[Article – Create a new article 66](#_Toc421827708)

[Article – Update an existing article 67](#_Toc421827709)

[Article – Delete an existing article 68](#_Toc421827710)

[Media 69](#_Toc421827711)

[Media – Read data of an media 69](#_Toc421827712)

[Media – Create a new media 70](#_Toc421827713)

[Media – Update an existing media 71](#_Toc421827714)

[Media – Delete an existing media 72](#_Toc421827715)

[Tip 73](#_Toc421827716)

[Tip – Read data of an tip 73](#_Toc421827717)

[Tip – Create a new tip 74](#_Toc421827718)

[Tip – Update an existing tip 75](#_Toc421827719)

[Tip – Delete an existing tip 76](#_Toc421827720)

[User 77](#_Toc421827721)

[User – Read data of a user 77](#_Toc421827722)

[User – Create a new user 78](#_Toc421827723)

[User – Update an existing user 79](#_Toc421827724)

[User – Delete an existing user 80](#_Toc421827725)

[Role 81](#_Toc421827726)

[Role – Read data of a role 81](#_Toc421827727)

[Role – Create a new role 82](#_Toc421827728)

[Role – Update an existing role 83](#_Toc421827729)

[Role – Delete an existing role 84](#_Toc421827730)

# 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() . ‘dardnjdjv382YBSdgh’, ‘aurezhfusdfhjdsfheztj7623B HSDG121dfdf’);
  + USEREMAIL: user’s email address
  + APIKEY: user’s unique API key

# 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 – 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 – 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 – 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 |

# 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 – Create a new gender

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 – 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 – 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 – Create a new editor

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 – 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 – 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 – Create a new theme

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 – 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 – 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 – 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 |
| 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 – 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 – 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 |

# 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 – 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 – 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 |
| mode | String(255) | Mode’s name | Yes |

## 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 – Create a new support

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 – 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 |
| support | String(255) | Support’s name | Yes |

## 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 – Read data of an edition

Method: GET

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

Permission: user

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Description | Required |
| id | Int | Edition’s ID | No |

## 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 – 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 – 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 – Create a new shop

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 – 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 – 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 |

# 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 – 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 – 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 – 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 |
| id | Int | Dlc’s ID to delete | Yes |

# 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 – Create a new config

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 – 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 – 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 |
| id | Int | Config’s ID to delete | Yes |

# 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 – 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 – 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 |
| 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 – 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 – Create a new comment

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 – 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 – 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 |
| id | Int | Comment’s ID to delete | Yes |

# 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 – Create a new analysis

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 – 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 |
| 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 – 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 |

# 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 – Create a new language

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 – 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 |
| language | String(255) | Language’s name | Yes |

## 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 |
| 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 – Create a new article

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 – 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 – 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 – 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 – 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 – 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 |
| id | Int | Media’s ID to delete | Yes |

# 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 – Create a new tip

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 – 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 – 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 |
| id | Int | Tip’s ID to delete | Yes |

# 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 – 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 – 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 |
| 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 – 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 – Create a new role

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 – 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 |
| role | String(255) | Role’s name | Yes |

## 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).