French Jeopardy Game – API Specifications

## Purpose

To describe how to develop the APIs for the French Jeopardy game, specifically for the Categories and the Questions/Answers used to play the game.

## API Description

#### Categories

The French Jeopardy game Category API returns a list of categories stored in a cloud hosted SQL database. Attributes include Category Name (EN) and Category Name (FR) and can reference an array of relevant questions. Additionally, it provides the ability to add, update and remove items (indirectly) from the database.

Questions (includes answers)

The French Jeopardy game Questions API returns a list of questions stored in a cloud hosted SQL database. Attributes include Question (EN), Question (FR), Answer (EN), Answer (FR), their relevant Category and so on. Additionally, it provides the ability to add, update and remove items (indirectly) from the database.

## URI Endpoints

#### Dev

* (Local) [https://localhost:3306/api/{categories](https://localhost:3306/api/%7bcategories)}
* (Local) [https://localhost:3306/api/{questions](https://localhost:3306/api/%7bquestions)}
* (Cloud) <https://jeopardyappservice.azurewebsites.net>

#### QA

* TBD…

#### Staging

* TBD…

#### Production

* TBD…

## RESTful Routing

### General Behaviours

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Route** | **GET** | **POST** | **PUT** | **DELETE** |
| /categories | Get list of Categories | Creates new Category | *Error* | *Error* |
| /categories/12 | Get the Category that corresponds to the ID provided | *Error* | Update Category corresponding to the ID provided | Deletes the Category corresponding to the ID provided |
| /questions | Get list of Questions | Creates new Question | *Error* | *Error* |
| /questions/board | Get random list of 25 Questions | *Error* | *Error* | *Error* |
| /questions/100 | Get the Question that corresponds to the points value provided | *Error* | Update Question corresponding to the ID provided | Deletes the Question corresponding to the ID provided |

### Standard Request Header

|  |  |
| --- | --- |
| **Key** | **Value** |
| **Authorization** |  |
| **Date** | today’s date |
| **Accept** | application/json, ~~text/xml~~ |
| **X-Version** | 1.0 |
| **Accept-Encoding** | Gzip, Deflate |
| **Accept-Language** | “en” or “fr” (default to “en” if not specified) |
| **Accept-Charset** | UTF-8 |
| **Content-Type** | Mime type of request body (PUT/POST/PATCH) |
| **Prefer** | “minimal“ or “representation“ |
| **If-Match, If-None-Match, If-Range** | ETag number from previous request |

### Standard Response Header

|  |  |
| --- | --- |
| **Key** | **Value** |
| **Date** | today’s date |
| **Content-Type** | application/json |
| **Content-Encoding** | Gzip, Deflate (if result is returned compressed) |
| **Content-Length** | Response size in Kb |
| **Content-Language** | “en” or “fr” (as specified in the request, default to “en” if not specified in the request) |
| **Preferred-Applied** | “minimal“ or “representation“ (if specified in request) |
| **api-version-supported** | 1.0 |
| **api-version-deprecated** | (leave empty if not applicable) |
| **ETag** | Generated ETag number |
| **Add following if result is a list of objects** | |
| **X-TotalCount** | Number of records returned |
| **X-NextPage** | /categories?page=n+1 |
| **X-PrevPage** | /categories?page=n-1 |

## HTTP GET

#### Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| Get() | None -> return all records | GET /categories | 200 OK | List of all Jeopardy Categories in JSON format. |
| 404 Not Found | Error with info  Message: “Categories not found” |
| 500 Internal Server Error | Returns the exception |
| Get(Id) | ?id=n (n = #)  Alternative:  ?queryString | GET /categories/{id}  Alternative:  /categories/{queryString} | 200 OK | Return the single category corresponding to ID |
| 400 Bad Request | Error with info |
| 404 Not Found | Error with info  Message: “Category does not exist” |
| 500 Internal Server Error | Returns the exception |

#### Questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| Get() | None -> return all records | GET /questions | 200 OK | List of all Jeopardy Questions in JSON format. |
| 404 Not Found | Error with info  Message: “No questions found” |
| 500 Internal Server Error | Returns the exception |
| Get(Points) | ?points=n (n = #)  Alternative:  ?queryString | GET /questions/{points}  Alternative:  /questions/{queryString} | 200 OK | Return the single question corresponding to ID |
| 400 Bad Request | Error with info |
| 404 Not Found | Error with info  Message: “No questions found” |
| 500 Internal Server Error | Returns the exception |
| Get(board) |  | GET /questions/{board} | 200 OK | Return the single category corresponding to ID |
| 404 Not Found | Error with info  Message: “No questions found” |
| 500 Internal Server Error | Returns the exception |

## HTTP POST

#### Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| /categories  POST | n/a | {Category object} | 201 Created | Returns location of new resource |
| 400 Bad Request | Error with info Message: “Category name cannot be empty” |
| 404 Not Found | Error with info  Message: “Item not found, please check request” |
| 409 Conflict | Error with info  Message: “The category already exists.” |
| 500 Internal Server Error | Returns the exception |
|  |  |
|  |  |
|  |  |

#### Questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| /questions  POST | n/a | {Questions object} | 201 Created | Returns location of new resource |
| 400 Bad Request | Error with info |
| 404 Not Found | Error with info  Message: “Item not found, please check request” |
| 409 Conflict | Error with info  Message: “Question already exists.” |
| 500 Internal Server Error | Returns the exception |
|  |  |
|  |  |
|  |  |

## HTTP PUT

#### Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| /categories(id)  PUT | n/a | {Category object} | 200 OK | Message: “Request successful, item updated.” |
| 400 Bad Request | Error with info  Message: “Id must be provided” |
| 404 Not Found | Error with info |
| 500 Internal Server Error | Returns the exception |
|  |  |
|  |  |
|  |  |

#### Questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| /questions(id)  PUT | n/a | {Category object} | 200 OK | Message: “Request successful, item updated.” |
| 400 Bad Request | Error with info  Message: “Not allowed or request not accepted” |
| 404 Not Found | Error with info  Message: “Item not found, please check request” |
| 500 Internal Server Error | Returns the exception |
|  |  |
|  |  |
|  |  |

## HTTP DELETE

#### Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| Delete(categoryId) | ?id=n (n = #) | n/a | 204 No Content | No response body |
| 400 Bad Request | Error with info  Message: “Not allowed or request not accepted” |
| 404 Not Found | Error with info |
| 500 Internal Server Error | Returns the exception |
|  |  |
|  |  |
|  |  |

#### Questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Query Params** | **Request Body** | **Response Status Code** | **Response Body (JSON format)** |
| Delete(questionId) | ?id=n (n = #) | n/a | 204 No Content | No response body |
| 400 Bad Request | Error with info  Message: “Not allowed or request not accepted” |
| 404 Not Found | Error with info  Message: “Database not found, please check request” |
| 500 Internal Server Error | Returns the exception |
|  |  |
|  |  |
|  |  |

# Models

### Categories: Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Display Name** | **Description** |
| id | Integer | Identifier | Unique identifier for a category. |
| categoryName\_En | String (varchar) | Category Name (EN) | The English name of a category. |
| categoryName\_Fr | String (varchar) | Category Name (FR) | The French name of a category. |

### Questions: Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Display Name** | **Description** |
| Id | Integer | Identifier | Unique identifier for a question. |
| question\_En | String (varchar) | Question (EN) | The English translation of a question. |
| question\_Fr | String (varchar) | Question (FR) | The French translation of a question. |
| answer\_en | String (varchar) | Answer (EN) | The English translation of an answer. |
| answer\_fr | String (varchar) | Answer (FR) | The French translation of an answer. |
| points | Integer | Points | Value awarded to players for correct responses. |
| timeLimit | Integer | Time Limit | A fixed time interval to answer the question before expiring. |
| hint | String (varchar) | Hint | An optional prompt to assist players in answering the question correctly. |
| category\_id | Integer | Category | Foreign key of the Category for which the question belongs. |