

Esercizio 2: Correggere gli errori

Questo codice OpenAPI è stato scritto per rispondere al seguente esercizio. Ci sono alcuni problemi cruciali che devono essere corretti. Quindi andremo riga per riga cercando di capire cosa potrebbe essere sbagliato.

Progettare un API REST (parziale) per gestire una piattaforma di giochi e descriverla inSwagger/OpenAPI. Ogni gioco è caratterizzato da un ID (di tipo *intero* per semplicità), un nome (string) e una categoria, che può assumere valori *Shooter*, *Adventure*, *Puzzle*, *Sport*, e un numero di giocatori minimo e massimo, entrambi valori interi. Scrivere un file in formato JSON o YAML.

L'API permette di:

- ottenere l'elenco di tutti i giochi di una data categoria
- modificare il numero minimo e massimo di giocatori in un gioco
- aggiungere un insieme di giochi e le relative informazioni; è possibile quindi aggiungere anche più di un gioco con un'unica richiesta

Specificare: URL di accesso, metodi HTTP, parametri e risposte con esempi. L'API restituisce un errore, con codice 400, se i parametri in input non sono corretti.

Note:

- Non è richiesto includere le sezioni *servers*, *tags*
- Non è richiesto gestire autenticazione

```
swagger: "2.0"
info:
  version: "1.0.0"
  title: "Esercizio giochi"
  description: "API gestione piattaforma di giochi"
paths:
  /giochi:
    get:
      summary: "Elenco dei giochi di data categoria"
      operationId: "getGameByCategory"
      parameters:
        - in: "query"
          name: "categoria"
          description: "categoria del gioco"
          required: true
          type: "string"
          enum:
            - Shooter
            - Adventure
            - Puzzle
            - Sport
      responses:
        200:
          description: "Successo"
          schema:
            type: "array"
            items:
              $ref: "#/definitions/Gioco"
        400:
          description: "Input non valido"

  /giochi/{idGioco}:
    put:
      summary: "Modifica il numero minimo e il numero massimo di giocatori in un g:"
      operationId: "getGameByCategory"
      parameters:
        - in: "path"
          name: "idGioco"
          description: "id del gioco"
          required: true
          type: "integer"
          format: "int32"
          minimum: 0
        - in: "body"
          name: "Gioco aggiornato"
          description: "Oggetto gioco con informazione aggiornata"
          required: true
          schema:
            $ref: "#/definitions/Gioco"
      responses:
        200:
          description: "Successo"
        400:
          description: "Input non valido"
```

```
/giochi:
  post:
    summary: "Aggiunge un insieme di giochi"
    operationId: "addGames"
    - in: "body"
      name: "nuovo gioco"
      description: "nuovo gioco da aggiungere"
      required: true
      schema:
        type: "array"
        items:
          $ref: "#/definitions/Gioco"
    responses:
      200:
        description: "Successo"
      400:
        description: "Input non valido"

definitions:
  Gioco:
    description: "Gioco della piattaforma"
    required:
      - id
      - nome
      - categoria
      - minNum
      - maxNum
    properties:
      id:
        type: "integer"
        format: "int32"
        minimum: 0
      nome:
        type: "string"
        example: "Tetris"
      categoria:
        type: "string"
        enum:
          - Shooter
          - Adventure
          - Puzzle
          - Sport
      minNum:
        type: "integer"
        format: "int32"
        minimum: 0
      maxNum:
        type: "integer"
        format: "int32"
        minimum: 1
```