

# Hi-Lo Game (part 2)

WASA: Web and Software Architecture

---

Prof. Emanuele Panizzi

## Let's specify the methods for the /games/ path (continued from hilo-game-part1)

```
get:
  summary: list all games
  description: |
    Obtain the list of all games, with the final result (win/lose)
    and the number of guesses
  responses:
    "200":
      description: |
        An array of objects, each one containing the game id,
        the final outcome, and the number of guesses
      content:
        application/json:
          schema:
            type: array
            items:
              type: object
              properties:
                id:
                  type: integer
                outcome:
                  type: string
                enum:
                  - win
                guesses:
```

# Swagger description - get

**GET** /games list all games ⬆

Obtain the list of all games, with the final result (win/lose) and the number of guesses

**Parameters** Try it out

No parameters

**Responses**

Code	Description	Links
200	An array of objects, each one containing the game id, the final outcome, and the number of guesses	No links

Media type

⌵

Controls Accept header

**Example Value** | Schema

```
[
  {
    "id": 0,
    "outcome": "win",
    "guesses": 0
  }
]
```

## Now let's specify the methods for the `/games/{id}` path

```
/games/{id}:  
  parameters:  
    - name: id  
      in: path  
      required: true  
      description: this is the game id  
      schema:  
        type: integer  
        description: e.g., /games/1234  
  put:
```

A parameter in the path is always required

## /games/{id} path: PUT

```
put:
  summary: start or reset a game
  description: |
    Start a game with given id, generating the secret number;
    or reset an existing game, re-generating the secret number
    and zeroing the guesses counter
  responses:
    "200":
      description: An existing game is reset
    "201":
      description: A new game is created
```

## /games/{id} path: POST

```
post:
  summary: make a guess
  description: |
    Try to guess the secret number; return hi, lo, or correct,
    plus the guess count.
  parameters:
    - name: guess
      in: query
      required: true
      description: the guess
      schema:
        type: integer
        minimum: 1
        maximum: 100
  responses:
```

## /games/{id} path: POST (continued)

```
responses:
  "200":
    description: Guess accepted, the result is in the the content
    content:
      application/json:
        schema:
          type: object
          properties:
            guess-count:
              type: integer
              minimum: 1
              maximum: 10
            guess-outcome:
              type: string
              enum:
                - hi
                - lo
                - correct
  "403":
```

## /games/{id} path: POST (continued)

```
"403":
  description: game over
  content:
    application/json:
      schema:
        type: object
        properties:
          id:
            type: integer
          outcome:
            type: string
            enum:
              - win
              - lose
          guesses:
            type: integer
"404":
  description: game not found
get:
```



## /games/{id} path: GET

```
get:
  summary: list guesses
  description: |
    Return a list of all guesses in reverse chronological order,
    including the responses received (hi/lo/correct)
  responses:
    "200":
      description: Request accepted, all guesses listed in content
      content:
```

## /games/{id} path: GET (continued)

```
content:
  application/json:
    schema:
      type: array
      items:
        type: object
        properties:
          guess-count:
            type: integer
            minimum: 1
            maximum: 10
          guess-outcome:
            type: string
            enum:
              - hi
              - lo
              - correct
```

## References

- <https://www.openapis.org>
- <https://oai.github.io/Documentation/>
- <http://gamificationlab.uniroma1.it/notes/wasa/hilo-game.yaml>

->