

# Case: Event Members

## Summary

**Design** the backend of a game feature that allows players to join an event and groups the participating players according to their level.

## Functions

- Join an event via a REST API request.
  - Player ID will be obtained from the authorization (cookie, session, access token, etc...)
  - Event ID will be obtained from the request parameters (path, query string or request body)
  - The response must include the group ID and the list of Player IDs of the group members.

## Specifications

- Group categories can be defined as hard-coded and are expected as follows:

Group Category	Player Requirements
Bronze	Level < 20
Silver	Level between 20 and 49
Gold	Level >= 50

- Each group can have up to **20** members, so multiple groups must be created for a category.
- A player can only join an event once; therefore, only the player's first request should be processed for an event.
  - It would be appreciated if the solution was designed so that there are no problems if the duplicate requests come in at the same time.
- Each player should be able to join an event with just a single REST API request.

## Dependencies

- No implementation is needed on how to handle the request and provide the parameters (event ID, player ID, player level). It will suffice to use interfaces, virtual functions or simply as function parameters.
  - eg: `addEventMember(eventId, playerId, playerLevel)`

### ↳ Expected items in the solution

#### 1. Business logic of the endpoint

- The algorithm should be complete except for the dependencies mentioned above.
- Any of the following methods can be chosen to explain the logic:
  - Flowchart diagram
  - Pseudo code
  - Code sample with descriptive comment lines would be more appreciated
    - Preferably written in Go (>=1.16) or PHP (>=7.4)
    - Also acceptable: NodeJS, Python, or Java

#### 2. Database design of the feature

- Which values should be stored in what format?
- Which data storage system should be selected?
- ERD or any understandable format is acceptable.

#### 3. REST API specifications of the endpoint

- Preferably in OAS v3 format

PS: A working app is not required but you can implement and share your example if you wish.  
PARTIAL SOLUTIONS ARE WELCOME.