****

**PLAYSTUDIOS TECHNICAL CHALLENGE**

**Explanations for Quest Configuration JSON properties**

## **Background:**

* This file is a single source of truth to run the Player Quest Progress and it contains all the required properties of all the Quests accessed in the services.
* This configuration JSON file can be located at “***PSQuest.Core -> Common -> questConfig.json***” in the solution.
* This is a JSON formatted file containing an array of Quests that are active or inactive and each quest contains further an array of milestones a quest can contain in it.
* Since this file is created and handled by the backend team, make sure it contains ONLY one active quest at a time.

## **Explanation and usage of each property in the JSON configuration file:**

1. questId [String]

* This property is a unique identifier used to store the Player Quest State (example: Q0001)

1. questName [String]

* This property is used to identify quest with a meaningful name. It is not used in the main logic but mainly added for internal use only (example: Christmas Fiesta Quest)

1. totalMilestones [Number]

* This property determines how many milestones a quest contains. It is used to validate if a player has finished all the milestones in a quest or not. (example: 5 or 10)

1. requiredPoints [Number]

* This property is used to check how many points are required to finish the Quest competition. It is also summation of all pointsToCompleteMilestone property in the milestones array in all the milestones in a quest. (example: 1000 points)

1. rateFromBet [Number]

* This property is used to compute the quest points a player will receive upon a single bet progression request (example: 2)

1. levelBonusRate [Number]

* This property is used to compute the quest points a player will received upon participating in a quest progress (example: 200)

1. isActive [Boolean]

* This property is used to determine what quest is active when a player quest progress is requested. This flag must hold the value *true* for ONLY 1 quest from all the quests stored in this JSON configuration file; all other quests must have *false* value. (example: true or false)

1. milestones [Array of Milestones]

* As explained earlier, a quest can have multiple milestones and that too are configurable. The number of milestones in a quest can be added as an array object using this property. The array of milestones have below properties in it:
  + milestoneIndex [Number]
    - This property provides the index of a milestone completed by a player by Quest Progress request. It is an incremental number of successive milestones in a quest. This milestoneIndex property is also used to determine the player quest state like how many milestones a player have finished in a quest. (example: 1 or 3 or 5)
  + pointsToCompleteMilestone [Number]
    - This property is used to determine how many points are required to complete a milestone. The sum of pointsToCompleteMilestone for all the milestones should match with requiredPoints quest property declared above. (example: 1000)
  + chipsAwarded [Number]
    - This property is passed to the response of quest progress request upon completing the number of milestones on each bet. This property is configurable and you can update as per your choice. The client then updates the chips awarded to a Player in the Player database to match the available chip balance.

(example: 200)