

Notifications contain references to User(s), Trophies, Tournaments, Quests,... these are weak references (-> serializedContent)

Setting
type: Enum value: JSON-String

ImageResource
UUID: String locationType: Enum uri: String

Notification
UUID: String createdAt: Datetime type: Enum version: Integer serializedContent: JSON-String

Trophy
UUID: String type: Enum serializedContent: JSON-String

Quest is a one-time occurrence of a "quiz" - everytime a user goes back to a place and answers more questions, a new Quest object is created.

Questions are only stored locally if they have been answered correctly - but they may be used as transient objects for encapsulation in all situations.
- Depending on "type", "serializedData" is displayed accordingly (image, multi-choice,...)
- Question structures (serializedData) may change overtime, so migrations must be supported -> "version" is necessary

Companionship
UUID: String confirmed confirmedAt: Datetime

TrophyForUser
UUID: String receivedAt: Datetime

Question
UUID: String type: Enum question: String serializedData: JSON-String correctAnswer: JSON-String answeredCorrectly: boolean

Questions might be contained in multiple places (e.g. questions about city/nation)?

User may also be a Companion

User
email: String (=UUID) name: String signedUpAt: Datetime serializedProfile: JSON-String

TournamentTaskDone
UUID: String completed: boolean completedAt: Datetime

Quest
UUID: String answeredCorrectly: Integer completed: Enum completedAt: Datetime

TournamentInvitation
UUID: String invitedAt: Datetime

Tournament
UUID: String createdAt: Datetime started: boolean startedAt: Datetime type: Enum serializedContent: JSON-String

TournamentTask
UUID: String

Place
UUID: String name: String lastVisitedAt: Datetime

Places might get deleted over time, TournamentTasks not.

A Tournament is made up of quests (we call it TournamentTask here)

Initiator
Confirmer

hasInvited

isInvitedTo

isInvitedTo

takesPartIn

isMadeUpOf