TimeTranslator Data Models

Scenario

An instance of Scenario is created when the user taps the Start Button, after a successful network request fetches the Scenario and Event data.

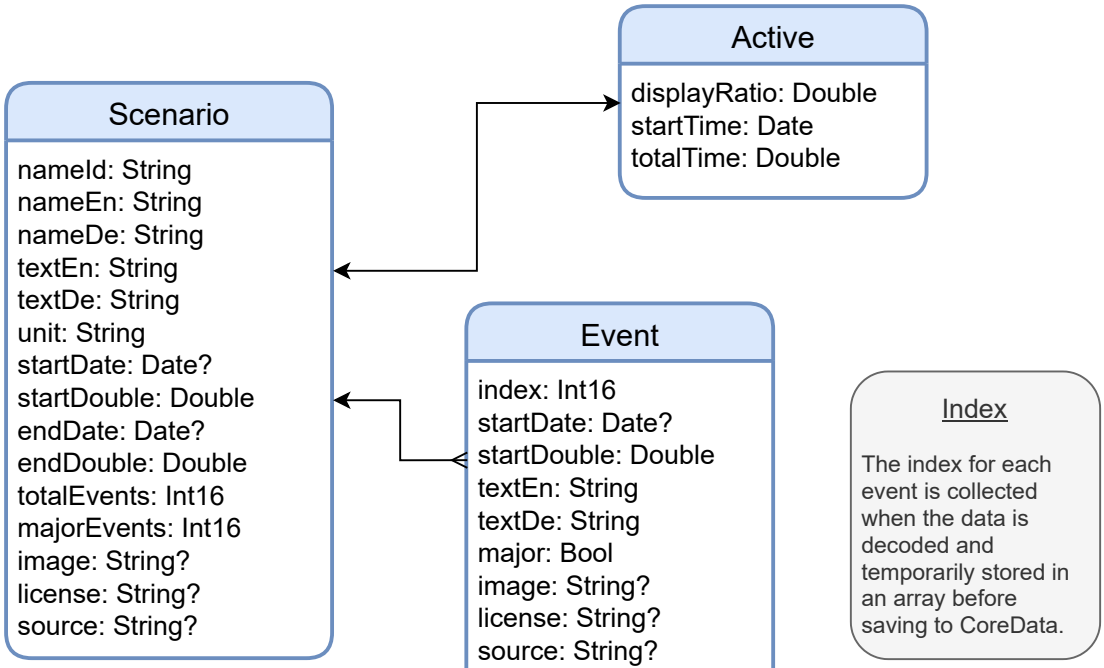
The Scenario and its associated Event array and instance of Active are removed from CoreData when the run time has completed or the user taps the cancel button, once the user confirms that they are finished.

Active

An instance of Active is created along with the Scenario, and stores the current time (plus 4 seconds to allow for UI transitions) in startTime. The totalTime is the user-defined RunTime in seconds. The displayRatio is calculated by dividing totalTime by the Scenario's length (endDouble - startDouble).

NameId

The nameId is used to fetch all scenario and event data from the api.



Name & Text

TimeTranslator is a bilingual app, so each String intended for display is provided in both English (En) and German (De).

Index

The index for each event is collected when the data is decoded and temporarily stored in an array before saving to CoreData.

Unit & Start/End

There are currently four defined unit types:
- time = "datetime"
- date = "date"
- astro = "ae, miokm, miomiles"
- distance = "km"

If a scenario has datetime or date in the Unit property, the values for the "start" and "end" keys should be a String in the format "yyyy-MM-dd'T'HH:mm:ss". The decoder will convert this to a date to store in the StartDate/EndDate properties, and convert the date into a Double to store in the StartDouble/EndDouble properties.

If a scenario has any other unit type, the values for the "start" and "end" keys should be a Double. The StartDate and EndDate will be nil, and the value will be stored in the StartDouble/EndDouble properties.

Event Timing

The displayTiming is calculated by applying the displayRatio from Active to the difference between the Event's startDouble and the Scenario's startDouble. The displayTiming is then used to schedule local notifications for major events.

As the Scenario is running, the displayTiming is checked against elapsed time to determine if the Event should be displayed on the table view. When elapsed time exceeds the displayTiming, the Event's displayed property is set to true. This triggers the table view to update, adding the Event to the top of the list.