As of A28.2, Aerie provides a "plan service" that manages a repository of plans. Access to this repository is mediated by a REST-based interface exposed by the service. The repository itself is currently implemented with MongoDB.

The repository of plans may be queried for a list of all plans, and new plans may be added to the repository. Existing plans may be retrieved in full, replaced in full or in part, or deleted in full. The list of activities in a plan may be appended to (by creating a new activity) and retrieved in full. Individual activities in a plan may be retrieved in full, replaced in full or in part, and deleted in full.

Operations on plans are validated to ensure consistency with the adaptation-specific activity model with which they are associated. Stored plans shall contain activities whose parameter names and types are defined by the associated activity type.

Data composition

The plan repository contains a set of JSON-style documents with the following top-level attributes: - id: A database-specific unique key identifying the plan. - adaptationId: A database-specific foreign key identifying the adaptation that the plan was built against. - name: A user-meaningful string specifying the name of the plan. - startTimestamp: A timestamp describing the instant at which the plan is considered to be applicable. - endTimestamp: A timestamp describing the instant at which the plan is considered no longer to be applicable. - activityInstances: A list of activity instances which comprise the plan.

A plan is conceptually composed of activity instances. Each activity instance has the following attributes:

- activityId: A database-specific unique key identifying the activity instance in the context of its containing plan.
- activityType: A database-specific foreign key identifying the type of activity this instance represents.
- name: A user-meaningful string specifying the name of this activity instance.
- intent: A user-meaningful string specifying the purpose of this activity instance in the context of its containing plan.
- startTimestamp: A timestamp describing the instant at which this activity instance is considered to begin.
- endTimestamp: A timestamp describing the instant at which this activity instance is considered to cease.
- start: The same semantic timestamp as described by startTimestamp, but in the UNIX epoch numeric format.
- end: The same semantic timestamp as described by endTimestamp, but in the UNIX epoch numeric format.
- duration: The numeric difference between startTimestamp and endTimestamp. (Units unclear – seconds?)

- backgroundColor: A string representing the RGB color associated with this activity instance.
- listeners: A list of foreign keys identifying listeners interested in this activity instance.
- textColor: A color for text associated with this activity instance.
- y: A vertical offset for figures associated with this activity instance.
- constraints: A list of constraints upon this activity instance.
- parameters: A list of parameters specifying the fine-grained behavior of this activity instance.

An activity may conceptually have multiple constraints imposed upon it. Each instance of a constraint has the following attributes:

- type: A database-specific foreign key identifying the constraint to be applied.
- name: A user-meaningful string specifying the name of this constraint instance.

The behavior of an activity may be parametrized on some user input. These parameters are specified as part of an activity instance. Each instance of a parameter has the following attributes:

- type: A database-specific foreign key identifying the parameter type to be instantiated.
- name: A database-specific foreign key identifying the parameter in the activity type for which a value is to be specified.
- value: A string representing the value of this parameter.

For more information on how activity Instances are declared in a plan, refer to the Activity Instance SIS at charlie web, Collection 298416 and Document 2579270.