API Documentation

Table-1

Widget	S No.	Request Type	Base URL	Api Endpoints	Payload
Nudge	1.	GET	/api/t2/app	/nudge?id=nudge_id	--
	2.	POST	/api/t2/app	/nudge	Tagged_Event_ID, Nudge_Title, Nudge_Cover, Schedule, Nudge_Des, Nudge_Icon, OL_Invitation
	3.	PUT	/api/t2/app	/nudge?id=nudge_id	Tagged_Event_ID, Nudge_Title, Nudge_Cover, Schedule, Nudge_Des, Nudge_Icon, OL_Invitation
	4.	Delete	/api/t2/app	/nudge?id=nudge_id	--

Table-2

S No.	Description
1.	Get a Nudge by its Unique Id
2.	Create a Nudge and returns the id of the nudge
3.	Update the properties, passed in JSON format in accordance of the Unique Id
4.	Delete the Nudge based on its Unique Id

Table-3

Object Data Model of a Nudge	Type: 'nudge’
	Tagged_Event_ID : Id of the event to which the nudge is associated with
	Nudge_Title: Title of the Nudge
	Nudge_Cover: fileUpload[image], for the Cover of nudge
	Nudge_Des: Description of the Nudge
	Schedule: (Date + Time) TimeStamp
	Nudge_Icon: Icon for Identification of Nudge
	OL_Invitation: It is a One LIne Invitation which will be shown where the nudge is minimized or when it is shown along with the event.

Table-4

Directions and Limitations	Do not use mongoose library instead use mongodb library, because with mongoose, you have to create a fixed schema and for update requests you have to send the whole Object (Object Data Model) and it will be time consuming or lower the performance.
	Do not write any schemas for storing and querying data, because the writing a schema will fix the Schema, or we want to query on the basis of different variables.
	We follow this data model consistency everywhere but not as a fixed Schema
	No schema, as the data might change slightly is based on the type of the asset, for instance, the asset could be an article or an event
	You have to use '_id' I.e., the mongodb ObjectId of a document as the unique identifier, which is a 12 or 24 hexString based on the timestamp

- Table-1 provides details about the middlewares and the type of request, payload and Endpoints.
- Table-2 provides details about what should be done on request or on response corresponding to the Table-1.
- Table-3 gives an idea about what type of data will be operating on.
- Tabl-4 gives the Directions and Limitations, that must be followed by the Developer.