

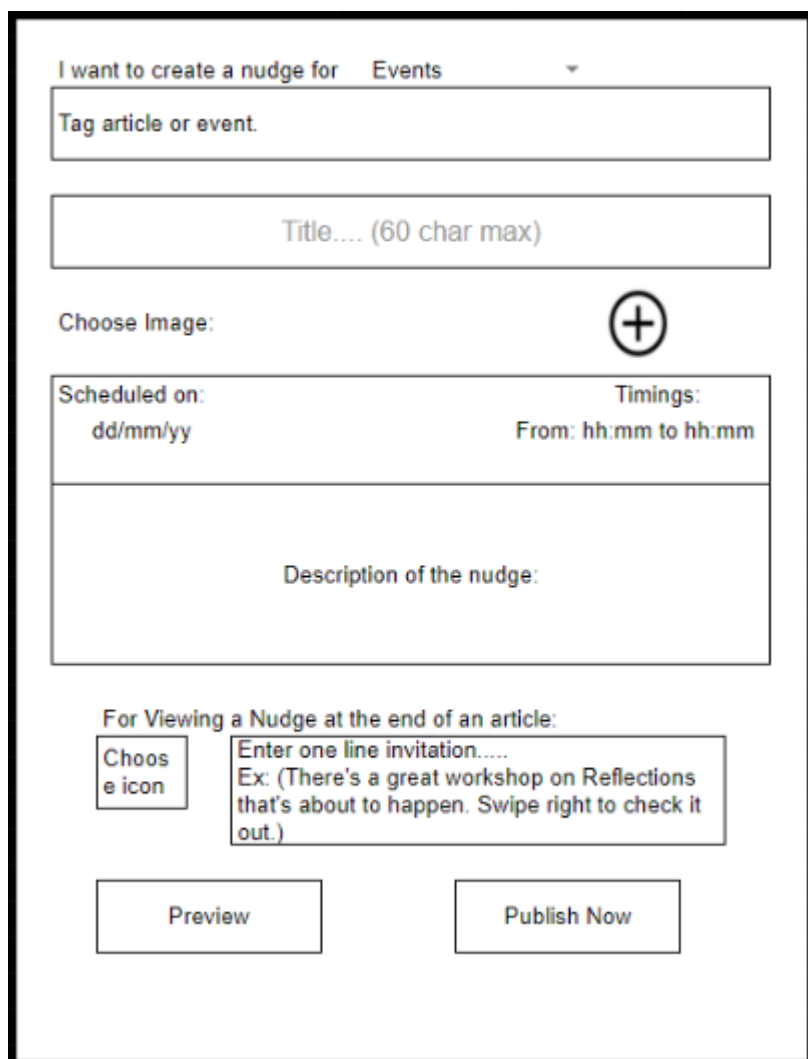
Nudge API Documentation

The Nudge API allows users to create, retrieve, update, and delete nudges for their events. Nudges are short messages with various properties that can be customized to engage the audience effectively. This documentation provides an overview of the API structure, including the base URL, API endpoints, payload structures, and descriptions.


Base URL

The base URL for the Nudge API is: <https://api/v3/>

Wireframe Analysis



The wireframe illustrates the layout for creating a nudge. It begins with a header 'I want to create a nudge for' followed by a dropdown menu currently set to 'Events'. Below this is a text input field labeled 'Tag article or event.'. A second text input field is labeled 'Title.... (60 char max)'. To the left of the next section is the label 'Choose Image:', and to its right is a circular icon containing a plus sign. This is followed by a table with two columns: 'Scheduled on:' with a sub-label 'dd/mm/yy', and 'Timings:' with a sub-label 'From: hh:mm to hh:mm'. Below the table is a large text area labeled 'Description of the nudge:'. A section titled 'For Viewing a Nudge at the end of an article:' contains a 'Choose icon' button and a text box for 'Enter one line invitation.....' with an example text. At the bottom are 'Preview' and 'Publish Now' buttons.

I want to create a nudge for Events ▼	
Tag article or event.	
Title.... (60 char max)	
Choose Image:	
Scheduled on: dd/mm/yy	Timings: From: hh:mm to hh:mm
Description of the nudge:	
For Viewing a Nudge at the end of an article:	
Choose icon	Enter one line invitation..... Ex: (There's a great workshop on Reflections that's about to happen. Swipe right to check it out.)
Preview	Publish Now

The provided wireframe illustrates the page where a user can create a nudge for their event. By analyzing the wireframe, we can identify the various fields, images, texts, and buttons present on the page.

Fields:

1. **Event Tag:** A field where the user can enter the ID or name of the event they want to create a nudge for.
2. **Nudge Title:** A field where the user can enter the title for the nudge.
3. **Nudge Time:** A field where the user can specify the time at which the nudge should be sent.
4. **Scheduled On:** A field where the user can select the date on which the nudge should be scheduled.
5. **Timings From:** A field where the user can specify the starting time for the nudge.
6. **Nudge Description:** A field where the user can provide a description for the nudge.
7. **Nudge Icon:** An optional field where the user can upload an icon for the nudge.
8. **Nudge Invitation:** An optional field where the user can enter a one-line invitation for the nudge.

Images:

1. **Nudge Cover Image:** An image section where the user can upload an image that will be shown as the cover for the nudge.

Buttons:

1. **Publish Now:** A button that triggers the creation of the nudge and saves it for the specified event.
2. **Preview:** A button that shows the preview of the nudge.

By understanding the wireframe and its components, we can design the API endpoints and payload structure accordingly to capture and process the necessary information for creating nudges.

Object Data Model: Nudge

The Nudge object has the following properties:

- `eventId` (string): The ID of the event associated with the nudge.
- `title` (string): The title of the nudge.
- `image` (file): An image file representing the nudge.
- `time` (string): The time at which the nudge should be sent.
- `description` (string): The description of the nudge.
- `icon` (string, optional): An icon for the nudge.
- `invitation` (string, optional): A one-line invitation for the nudge.

API Endpoints

1. Create Nudge for an Event:

- URL: `/api/v3/app/nudges`
- Method: **POST**
- Description: Creates a nudge for a specific event.
- Payload:
 - `eventTag` (string): The ID or name of the event.
 - `nudgeTitle` (string): The title for the nudge.
 - `nudgeTime` (string): The time at which the nudge should be sent.
 - `nudgeDescription` (string): The description for the nudge.
 - `nudgeIcon` (file): Optional. The icon for the nudge.

- nudgeInvitation (string): Optional. The one-line invitation for the nudge.
- scheduledOn (string): The date on which the nudge should be scheduled.
- timingsFrom (string): The starting time for the nudge.
- nudgeCoverImage (file): The cover image for the nudge.
- Success Response:
 - Code: 200
 - Content: { "nudgeId": "nudge-id" }
- Error Response:
 - Code: 500
 - Content: { "error": "Internal Server Error" }

2. Get Nudge by ID:

- URL: /api/v3/app/nudges/:id
- Method: **GET**
- Description: Retrieves a nudge by its unique ID.
- URL Parameters:
 - id: The unique ID of the nudge.
- Success Response:
 - Code: 200
 - Content: JSON representation of the nudge
- Error Response:
 - Code: 404
 - Content: { "error": "Nudge not found" }

3. Update Nudge:

- URL: /api/v3/app/nudges/:id

- Method: **PUT**
- Description: Updates an existing nudge.
- URL Parameters:
 - id: The unique ID of the nudge.
- Payload: Same as the payload for creating a nudge.
- Success Response:
 - Code: 200
 - Content: { "message": "Nudge updated successfully" }
- Error Response:
 - Code: 404
 - Content: { "error": "Nudge not found" }

4. Delete Nudge:

- URL: /api/v3/app/nudges/:id
- Method: **DELETE**
- Description: Deletes a nudge based on its unique ID.
- URL Parameters:
 - id: The unique ID of the nudge.
- Success Response:
 - Code: 200
 - Content: { "message": "Nudge deleted successfully" }
- Error Response:
 - Code: 404
 - Content: { "error": "Nudge not found" }

These API endpoints provide the functionality to create, retrieve, update, and delete nudges for specific events. The payload includes various fields to provide the necessary details for creating and updating a nudge, such as the

event tag, nudge title, nudge time, nudge description, nudge icon, one-line invitation, scheduled date, timings from, and nudge cover image.

Examples

Create a Nudge

Request:

```
POST /nudges
Content-Type: application/json

{
  "eventId": "event-123",
  "nudgeTitle": "Join our exciting event!",
  "nudgeTime": "2023-07-01T15:00:00",
  "nudgeDescription": "Don't miss out on this amazing event.",
  "nudgeIcon": "https://example.com/nudge-icon.png",
  "nudgeInvitation": "Join us for an unforgettable experience!",
  "scheduledOn": "2023-06-30",
  "timingsFrom": "10:00 AM",
  "nudgeCoverImage": "https://example.com/nudge-cover.png"
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": "nudge-456",
  "eventId": "event-123",
  "nudgeTitle": "Join our exciting event!",
  "nudgeTime": "2023-07-01T15:00:00",
  "nudgeDescription": "Don't miss out on this amazing event.",
  "nudgeIcon": "https://example.com/nudge-icon.png",
  "nudgeInvitation": "Join us for an unforgettable experience!",
  "scheduledOn": "2023-06-30",
  "timingsFrom": "10:00 AM",
  "nudgeCoverImage": "https://example.com/nudge-cover.png"
}
```

Get Nudge by ID

Request:

```
GET /nudges/nudge-456
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "nudge-456",
  "eventId": "event-123",
  "nudgeTitle": "Join our exciting event!",
  "nudgeTime": "2023-07-01T15:00:00",
  "nudgeDescription": "Don't miss out on this amazing event.",
  "nudgeIcon": "https://example.com/nudge-icon.png",
  "nudgeInvitation": "Join us for an unforgettable experience!",
  "scheduledOn": "2023-06-30",
  "timingsFrom": "10:00 AM",
  "nudgeCoverImage": "https://example.com/nudge-cover.png"
}
```

Get Nudge by ID

Request:

```
GET /nudges/nudge-456
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "nudge-456",
  "eventId": "event-123",
  "nudgeTitle": "Join our exciting event!",
  "nudgeTime": "2023-07-01T15:00:00",
  "nudgeDescription": "Don't miss out on this amazing event.",
}
```

```
"nudgeIcon": "https://example.com/nudge-icon.png",
"nudgeInvitation": "Join us for an unforgettable experience!",
"scheduledOn": "2023-06-30",
"timingsFrom": "10:00 AM",
"nudgeCoverImage": "https://example.com/nudge-cover.png"
}
```

Update Nudge

Request:

```
PUT /nudges/nudge-456
Content-Type: application/json

{
  "eventId": "event-123",
  "nudgeTitle": "Join our exciting event!",
  "nudgeTime": "2023-07-01T16:00:00",
  "nudgeDescription": "Don't miss out on this amazing event. Limited
spots available!",
  "nudgeIcon": "https://example.com/nudge-icon.png",
  "nudgeInvitation": "Join us for an unforgettable experience! RSVP
now!",
  "scheduledOn": "2023-06-30",
  "timingsFrom": "10:00 AM",
  "nudgeCoverImage": "https://example.com/nudge-cover.png"
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "nudge-456",
  "eventId": "event-123",
  "nudgeTitle": "Join our exciting event!",
  "nudgeTime": "2023-07-01T16:00:00",
  "nudgeDescription": "Don't miss out on this amazing event. Limited
spots available!",
  "nudgeIcon": "https://example.com/nudge-icon.png",
```



```
"nudgeInvitation": "Join us for an unforgettable experience! RSVP now!",  
  "scheduledOn": "2023-06-30",  
  "timingsFrom": "10:00 AM",  
  "nudgeCoverImage": "https://example.com/nudge-cover.png"  
}
```

Delete Nudge

Request:

```
DELETE /nudges/nudge-456
```

Response:

```
HTTP/1.1 200 OK  
Content-Type: application/json  
  
{  
  "message": "Nudge deleted successfully"  
}
```

That concludes the CRUD functionality of the Nudge API. Feel free to use these endpoints to manage nudges for your events.