

## Nudge API Documentation

This document outlines the structure of the API used to create, update, retrieve, and delete nudges. The nudge is an interaction trigger that can be attached to an event or article. Users can set a title, description, cover image, schedule, and additional details.

---

### API Overview

- Base URL: <http://yourdomain.com/api/nudges>
- Content-Type: [application/json](#)
- Authentication: Token-based (optional, if authentication is required)

### Object Data Model: Nudge

```
{  
  "id": "string",  
  "taggedItem": "string",  
  "title": "string",  
  "imageUrl": "string",  
  "schedule": {  
    "date": "dd/mm/yyyy",  
    "fromTime": "hh:mm",  
    "toTime": "hh:mm"  
  },  
  "description": "string",  
  "icon": "string",  
  "invitation": "string"  
}
```

---

## API Endpoints

### 1. Create Nudge

- Method: **POST**
- Endpoint: **/nudges**
- Description: Creates a new nudge for an event or article.

Request Payload:

```
{  
  "taggedItem": "string",  
  "title": "string",  
  "imageUrl": "string",  
  "schedule": {  
    "date": "dd/mm/yyyy",  
    "fromTime": "hh:mm",  
    "toTime": "hh:mm"  
  },  
  "description": "string",  
  "icon": "string",  
  "invitation": "string"  
}
```

○

Response:

```
{  
  "success": true,  
  "message": "Nudge created successfully",  
  "data": { ... }  
}
```

○

## 2. Get All Nudges

- Method: **GET**
- Endpoint: **/nudges**
- Description: Retrieve all the nudges.

Response:

```
{  
  
  "success": true,  
  
  "data": [  
  
    {  
  
      "id": "string",  
  
      "taggedItem": "string",  
  
      "title": "string",  
  
      "imageUrl": "string",  
  
      "schedule": {  
  
        "date": "dd/mm/yyyy",  
  
        "fromTime": "hh:mm",  
  
        "toTime": "hh:mm"  
  
      },  
  
      "description": "string",  
  
      "icon": "string",  
  
      "invitation": "string"  
  
    }  
  
  ]  
  
}
```

○

## 3. Get Nudge by ID

- Method: **GET**
- Endpoint: **/nudges/{id}**
- Description: Retrieve a specific nudge by its unique ID.

Response:

```
{  
  
  "success": true,  
  
  "data": {  
  
    "id": "string",  
  
    "taggedItem": "string",  
  
    "title": "string",  
  
    "imageUrl": "string",  
  
    "schedule": {  
  
      "date": "dd/mm/yyyy",  
  
      "fromTime": "hh:mm",  
  
      "toTime": "hh:mm"  
    },  
  
    "description": "string",  
  
    "icon": "string",  
  
    "invitation": "string"  
  }  
}
```

○

#### 4. Update Nudge

- Method: **PUT**
- Endpoint: **/nudges/{id}**
- Description: Update the details of an existing nudge by its ID.

Request Payload:

```
{  
  
  "taggedItem": "string",  
  
  "title": "string",  
  
  "imageUrl": "string",
```

```
"schedule": {  
  "date": "dd/mm/yyyy",  
  "fromTime": "hh:mm",  
  "toTime": "hh:mm"  
},  
"description": "string",  
"icon": "string",  
"invitation": "string"  
}
```

○

Response:

```
{  
  "success": true,  
  "message": "Nudge updated successfully",  
  "data": { ... }  
}
```

○

## 5. Delete Nudge

- Method: **DELETE**
- Endpoint: **/nudges/{id}**
- Description: Delete a specific nudge by its ID.

Response:

```
{  
  "success": true,  
  "message": "Nudge deleted successfully"  
}
```

○

---

## Error Handling

- 400 Bad Request: When the input validation fails.

```
{  
  
  "success": false,  
  
  "message": "Invalid request payload"  
}
```

- 404 Not Found: When a nudge with the given ID does not exist.

```
{  
  
  "success": false,  
  
  "message": "Nudge not found"  
}
```

- 500 Internal Server Error: For any server-side errors.

```
{  
  
  "success": false,  
  
  "message": "Something went wrong"  
}
```

This API allows users to effectively manage nudges that act as interaction triggers related to specific events or articles.

