API Documentation: Smart Notification System

# Authentication

## Register API

* This API is used to register the user.
* URL = /api/v1/user/register/
* METHOD = POST
* Payload =

{

"first\_name": "test",

"last\_name": "test",

"email": "user@example.com",

"mobile": "9876543210",

"password": "Test@321",

"confirm\_password": "Test@321"

}

* CURL =

curl -X 'POST' \

'http://127.0.0.1:8000/api/v1/user/register/' \

-H 'accept: application/json' \

-H 'Content-Type: application/json' \

-H 'X-CSRFTOKEN: RAL3Zw0I2pICrQXypgOsOM52uqmEb6UFUsZoQMMNYRxNmbd3umyDuJGKWWF6kXQC' \

-d '{

"first\_name": "test",

"last\_name": "test",

"email": "user1@example.com",

"mobile": "9876543219",

"password": "Test@321",

"confirm\_password": "Test@321"

}'

* Response =

{

"message": "User Registration Successful!",

"data": {

"first\_name": "test",

"last\_name": "test",

"email": "user1@example.com",

"mobile": "9876543219"

},

"errors": []

}

## Login API

* The system supports user authentication via email and password, issuing JWT tokens upon successful login. During login, the system captures the user's IP address and browser user-agent to generate a unique device fingerprint. If the device is unrecognized, it is recorded and a new\_login notification is triggered automatically. This notification is delivered to the user through their preferred channels (in-app, email, or SMS), ensuring awareness of any login from an unknown device.
* METHOD = POST
* URL = /api/v1/user/register/
* Payload =

{

"email": "user@example.com",

"password": "Test@321"

}

* CURL =

curl -X 'POST' \

'http://127.0.0.1:8000/api/v1/user/login/' \

-H 'accept: application/json' \

-H 'Content-Type: application/json' \

-H 'X-CSRFTOKEN: RAL3Zw0I2pICrQXypgOsOM52uqmEb6UFUsZoQMMNYRxNmbd3umyDuJGKWWF6kXQC' \

-d '{

"email": "user@example.com",

"password": "Test@321"

}'

* Response =

{

"message": "User Login Successful!",

"data": {

"refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tl",

"access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXV",

"user": {

"id": 1,

"email": "user@example.com",

"mobile": "9876543210",

"first\_name": "test",

"last\_name": "test"

}

},

"errors": []

}

# Notification Preferences

## Retrieve a user’s notification preferences.

* Fetches the current notification preferences for the specified user, showing selected delivery channels. Requires authentication.
* Method = GET
* URL = /api/v1/notifications/notification-preferences/{user\_id}/
* CURL=

curl -X 'GET' \

'http://127.0.0.1:8000/api/v1/notifications/notification-preferences/1/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDg1MDk1LCJpYXQiOjE3NTMwODQ3OTUsImp0aSI6IjYwZWJjYWE1YTNkYjQ5NTk4NTk0M2NkNDZiNTYxY2RlIiwidXNlcl9pZCI6MX0.H6Qkxtvk26fasl9BODTCMtxbjiY1xVU6llKT3BVDCSw'

* Response =

{

"message": "Notification Preferences Fetched Successful",

"data": {

"user": 1,

"in\_app": true,

"email": false,

"sms": false

},

"errors": []

}

## Update a user’s notification preferences.

* URL = /api/v1/notifications/preferences/<user\_id>/
* METHOD = PATCH
* PAYLOAD =

{

"in\_app": true,

"email": true,

"sms": true

}

* CURL =

curl -X 'PATCH' \

'http://127.0.0.1:8000/api/v1/notifications/notification-preferences/1/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDg1MDk1LCJpYXQiOjE3NTMwODQ3OTUsImp0aSI6IjYwZWJjYWE1YTNkYjQ5NTk4NTk0M2NkNDZiNTYxY2RlIiwidXNlcl9pZCI6MX0.H6Qkxtvk26fasl9BODTCMtxbjiY1xVU6llKT3BVDCSw' \

-H 'Content-Type: application/json' \

-H 'X-CSRFTOKEN: RAL3Zw0I2pICrQXypgOsOM52uqmEb6UFUsZoQMMNYRxNmbd3umyDuJGKWWF6kXQC' \

-d '{

"in\_app": true,

"email": true,

"sms": true

}'

* RESPONSE =

{

"message": "Notification Preference Updated Successful",

"data": {

"user": 1,

"in\_app": true,

"email": true,

"sms": true

},

"errors": []

}

# Thread

## Retrieve all discussion threads.

* Fetches and returns a list of all existing threads. This endpoint is publicly accessible and does not require authentication.
* URL = /api/v1/notifications/thread/fetch/
* CURL =

curl -X 'GET' \

'http://127.0.0.1:8000/api/v1/notifications/thread/fetch/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDc5MTcwLCJpYXQiOjE3NTMwNzg4NzAsImp0aSI6IjM0ZjEwZjZmYzQyMTRlN2RhZTVmOGVjMWU1ZTVhZWE2IiwidXNlcl9pZCI6MX0.CbhjCr21J1RPKIlt5vNjk4m-lr2HyhDhdbB2IG719H4'

* RESPONSE =

{

"message": "Threads Fetched Successful",

"data": [

{

"id": 1,

"title": "Test Thread"

},

{

"id": 2,

"title": "Test Thread 1"

}],

"errors": []}

## Create a new discussion thread.

* Allows an authenticated user to create a new thread. The user is automatically subscribed to the thread upon creation.
* URL = /api/v1/notifications/thread/create/
* CURL =

{

"title": "Test Thread 1"

}

curl -X 'POST' \

'http://127.0.0.1:8000/api/v1/notifications/thread/create/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDc5MTcwLCJpYXQiOjE3NTMwNzg4NzAsImp0aSI6IjM0ZjEwZjZmYzQyMTRlN2RhZTVmOGVjMWU1ZTVhZWE2IiwidXNlcl9pZCI6MX0.CbhjCr21J1RPKIlt5vNjk4m-lr2HyhDhdbB2IG719H4' \

-H 'Content-Type: application/json' \

-H 'X-CSRFTOKEN: RAL3Zw0I2pICrQXypgOsOM52uqmEb6UFUsZoQMMNYRxNmbd3umyDuJGKWWF6kXQC' \

-d '{

"title": "Test Thread 1"

}'

* PAYLOAD =

{

"title": "Test Thread 1"

}

* RESPONSE =

{

"message": "Threads Created Successful",

"data": {

"title": "Test Thread 1"

},

"errors": []

}

## Subscribe to a discussion thread.

* Allows an authenticated user to subscribe to a thread to receive notifications about new comments. The system prevents duplicate subscriptions and ensures that users do not subscribe to the same thread more than once. On success, it returns the subscription details.
* METHOD = POST
* URL = /api/v1/notifications/thread-subscription/create/
* CURL =

curl -X 'POST' \

'http://127.0.0.1:8000/api/v1/notifications/thread-subscription/create/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDc5MTcwLCJpYXQiOjE3NTMwNzg4NzAsImp0aSI6IjM0ZjEwZjZmYzQyMTRlN2RhZTVmOGVjMWU1ZTVhZWE2IiwidXNlcl9pZCI6MX0.CbhjCr21J1RPKIlt5vNjk4m-lr2HyhDhdbB2IG719H4' \

-H 'Content-Type: application/json' \

-H 'X-CSRFTOKEN: RAL3Zw0I2pICrQXypgOsOM52uqmEb6UFUsZoQMMNYRxNmbd3umyDuJGKWWF6kXQC' \

-d '{

"thread": 1

}

* PAYLOAD =

{

"thread": 2

}

* RESPONSE =

{

"message": "Threads Subscription Created Successful",

"data": {

"thread": 1

},

"errors": []

}

# Comment Triggers Notifications

## Create a comment on a discussion thread.

* This endpoint allows authenticated users to post a comment on an existing thread. Once a comment is created, all users subscribed to the thread (excluding the commenter) are notified based on their individual preferences via in-app, email, or SMS notifications. The API enforces rate limiting and uses a signal-based notification dispatch system backed by Celery tasks for asynchronous delivery.
* URL = /api/v1/notifications/trigger/
* CURL =

curl -X 'POST' \

'http://127.0.0.1:8000/api/v1/notifications/trigger/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDc5MTcwLCJpYXQiOjE3NTMwNzg4NzAsImp0aSI6IjM0ZjEwZjZmYzQyMTRlN2RhZTVmOGVjMWU1ZTVhZWE2IiwidXNlcl9pZCI6MX0.CbhjCr21J1RPKIlt5vNjk4m-lr2HyhDhdbB2IG719H4' \

-H 'Content-Type: application/json' \

-H 'X-CSRFTOKEN: RAL3Zw0I2pICrQXypgOsOM52uqmEb6UFUsZoQMMNYRxNmbd3umyDuJGKWWF6kXQC' \

-d '{

"content": "test test",

"thread": 1

}'

* PAYLOAD=

{

"content": "test test",

"thread": 1

}

* RESPONSE =

{

"message": "Comments Created Successful",

"data": {

"content": "test test",

"thread": 1,

"user": 1,

"created\_at": "2025-07-21T06:22:34.227595Z"

},

"errors": []

}

# In-App Notifications (via API)

## Fetch full notification history for the user.

* Returns all notifications (read and unread) sorted by creation time. Supports pagination and provides thread and comment context where applicable.
* Query Parameters
  + page (optional)
  + page\_size (optional)
* METHOD = GET
* URL = /api/v1/notifications/notifications/history/
* CURL =

curl -X 'GET' \

'http://127.0.0.1:8000/api/v1/notifications/notifications/history/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDg2NzcwLCJpYXQiOjE3NTMwODY0NzAsImp0aSI6IjlmNDkwMzNjOWI5ZjQxMzhiYTRhMmM5OWM5ZWY5MzhmIiwidXNlcl9pZCI6Mn0.30hoSZFNjrD2RESbUceToPhcYKEf9xSeCmtrgSYgAx8'

* RESPONSE =

{

"message": "Notifications fetched successfully.",

"data": [

{

"id": 2,

"comment": {

"content": "test test",

"thread": 1,

"user": 1,

"created\_at": "2025-07-21T08:27:21.566825Z"

},

"thread": {

"id": 1,

"title": "Test Thread"

},

"message": "New comment on 'Test Thread' by user@example.com",

"created\_at": "2025-07-21T08:27:21.571656Z",

"is\_read": false,

"in\_app\_status": true,

"email\_status": false,

"sms\_status": false

}

],

"errors": []

}

## Fetch unread notifications for the authenticated user.

* This endpoint retrieves all unread notifications belonging to the logged-in user. Supports pagination and returns a structured response including the related comment and thread info
* METHOD = GET
* URL = /api/v1/notifications/notifications/unread/
* CURL =

curl -X 'GET' \

'http://127.0.0.1:8000/api/v1/notifications/notifications/unread/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDg2NzcwLCJpYXQiOjE3NTMwODY0NzAsImp0aSI6IjlmNDkwMzNjOWI5ZjQxMzhiYTRhMmM5OWM5ZWY5MzhmIiwidXNlcl9pZCI6Mn0.30hoSZFNjrD2RESbUceToPhcYKEf9xSeCmtrgSYgAx8'

* Query Parameters
  + page (optional)
  + page\_size (optional)
* RESPONSE =

{

"message": "Unread Notification fetched successfully.",

"data": [

{

"id": 2,

"comment": {

"content": "test test",

"thread": 1,

"user": 1,

"created\_at": "2025-07-21T08:26:48.858583Z"

},

"thread": {

"id": 1,

"title": "Test Thread"

},

"message": "New comment on 'Test Thread' by user@example.com",

"created\_at": "2025-07-21T08:26:48.865552Z",

"is\_read": false,

"in\_app\_status": true,

"email\_status": false,

"sms\_status": false

},

],

"errors": []

}

## Mark selected notifications as read.

* This endpoint allows the user to mark specific notification(s) as read by passing a list of valid notification IDs that belong to them.
* METHOD = POST
* URL = /api/v1/notifications/notifications/read/
* CURL =

curl -X 'POST' \

'http://127.0.0.1:8000/api/v1/notifications/notifications/read/' \

-H 'accept: application/json' \

-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzUzMDg2NzcwLCJpYXQiOjE3NTMwODY0NzAsImp0aSI6IjlmNDkwMzNjOWI5ZjQxMzhiYTRhMmM5OWM5ZWY5MzhmIiwidXNlcl9pZCI6Mn0.30hoSZFNjrD2RESbUceToPhcYKEf9xSeCmtrgSYgAx8' \

-H 'Content-Type: application/json' \

-H 'X-CSRFTOKEN: RAL3Zw0I2pICrQXypgOsOM52uqmEb6UFUsZoQMMNYRxNmbd3umyDuJGKWWF6kXQC' \

-d '{

"notification\_ids": [

1

]

}'

* PAYLOAD =

{

"notification\_ids": [

1

]

}

* RESPONSE =

{

"message": "Notifications Marked as read.",

"data": {

"marked\_read": {

"notification\_ids": [

1

]

}

},

"errors": []

}

# Weekly Summary Notification (Scheduled Task)

* This project includes a Celery Beat task that automatically sends users a weekly summary of their unread notifications.
* What it Does:
  + - Iterates over all users in the system.
    - Fetches unread notifications created within the last 7 days.
    - If any exist:
      * Creates a new in-app notification summarizing the count of those unread messages.
      * This summary is not sent via email or SMS (by default).