

Account Privacy APIs

Author: Kibuti, Backend Developer

Last Updated: 2025-12-11

Version: v1.0

Base URL: `https://api.nexgate.com/api/v1`

Short Description: The Account Privacy API allows users to manage their account privacy settings, controlling whether their account is private or public. Private accounts require follow request approval before users can see their content and follow them.

Hints:

- All endpoints require authentication via Bearer token
- Privacy settings are automatically created with default values (public account) on first access
- Changing privacy settings does not affect existing followers
- Private accounts require follow request approval for new followers

Standard Response Format

All API responses follow a consistent structure using our Globe Response Builder pattern:

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2025-09-23T10:30:45",
  "data": {
    // Actual response data goes here
  }
}
```

Error Response Structure

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Error description",
  "action_time": "2025-09-23T10:30:45",
  "data": "Error description"
}
```

Standard Response Fields

| Field | Type | Description |
|-------------|---------------|---|
| success | boolean | Always <code>true</code> for successful operations, <code>false</code> for errors |
| httpStatus | string | HTTP status name (OK, BAD_REQUEST, NOT_FOUND, etc.) |
| message | string | Human-readable message describing the operation result |
| action_time | string | ISO 8601 timestamp of when the response was generated |
| data | object/string | Response payload for success, error details for failures |

HTTP Method Badge Standards

For better visual clarity, all endpoints use colored badges for HTTP methods with the following standard colors:

- **GET** - `GET` - Green (Safe, read-only operations)
- **POST** - `POST` - Blue (Create new resources)
- **PUT** - `PUT` - Yellow (Update/replace entire resource)
- **PATCH** - `PATCH` - Orange (Partial updates)
- **DELETE** - `DELETE` - Red (Remove resources)

Endpoints

1. Get Privacy Settings

Purpose: Retrieve the current privacy settings for the authenticated user's account

Endpoint: `GET` `{base_url}/e-social/privacy/account`

Access Level: `Protected` (Requires Bearer Token Authentication)

Authentication: Bearer Token

Request Headers:

| Header | Type | Required | Description |
|---------------|--------|----------|--|
| Authorization | string | Yes | Bearer token for authentication (format: <code>Bearer <token></code>) |
| Content-Type | string | Yes | Must be <code>application/json</code> |

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Privacy settings retrieved successfully",
  "action_time": "2025-12-11T10:30:45",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "userId": "123e4567-e89b-12d3-a456-426614174000",
    "isPrivate": false,
    "createdAt": "2025-01-15T08:20:30",
    "updatedAt": "2025-12-11T10:30:45"
  }
}
```

Success Response Fields:

| Field | Description |
|--------|---|
| id | Unique identifier for the privacy settings record (UUID format) |
| userId | Unique identifier of the user who owns these settings (UUID format) |

| Field | Description |
|-----------|--|
| isPrivate | Boolean flag indicating if the account is private (true) or public (false) |
| createdAt | Timestamp when the privacy settings were first created (ISO 8601 format) |
| updatedAt | Timestamp when the privacy settings were last modified (ISO 8601 format) |

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2025-12-11T10:30:45",
  "data": "Token has expired"
}
```

Standard Error Types:

Application-Level Exceptions (400-499)

- `401 UNAUTHORIZED`: Authentication issues (empty, invalid, expired, or malformed tokens)
- `500 INTERNAL_SERVER_ERROR`: Unexpected server errors

Error Response Examples:

Unauthorized - Token Issues (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2025-12-11T10:30:45",
  "data": "Token has expired"
}
```

Unauthorized - Missing Token (401):

```
{
  "success": false,
```

```
"httpStatus": "UNAUTHORIZED",
"message": "Authentication token is required",
"action_time": "2025-12-11T10:30:45",
"data": "Authentication token is required"
}
```

2. Update Privacy Settings

Purpose: Update the privacy settings for the authenticated user's account (switch between private and public)

Endpoint: **PUT** `{base_url}/e-social/privacy/account`

Access Level: Protected (Requires Bearer Token Authentication)

Authentication: Bearer Token

Request Headers:

| Header | Type | Required | Description |
|---------------|--------|----------|--|
| Authorization | string | Yes | Bearer token for authentication (format: <code>Bearer <token></code>) |
| Content-Type | string | Yes | Must be <code>application/json</code> |

Request JSON Sample:

```
{
  "isPrivate": true
}
```

Request Body Parameters:

| Parameter | Type | Required | Description | Validation |
|-----------|---------|----------|---|------------------|
| isPrivate | boolean | Yes | Flag to set account as private (true) or public (false) | Must not be null |

Success Response JSON Sample:

```

{
  "success": true,
  "httpStatus": "OK",
  "message": "Privacy settings updated successfully",
  "action_time": "2025-12-11T10:30:45",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "userId": "123e4567-e89b-12d3-a456-426614174000",
    "isPrivate": true,
    "createdAt": "2025-01-15T08:20:30",
    "updatedAt": "2025-12-11T10:30:45"
  }
}

```

Success Response Fields:

| Field | Description |
|-----------|--|
| id | Unique identifier for the privacy settings record (UUID format) |
| userId | Unique identifier of the user who owns these settings (UUID format) |
| isPrivate | Boolean flag indicating if the account is private (true) or public (false) |
| createdAt | Timestamp when the privacy settings were first created (ISO 8601 format) |
| updatedAt | Timestamp when the privacy settings were last modified (ISO 8601 format) |

Error Response JSON Sample:

```

{
  "success": false,
  "httpStatus": "UNPROCESSABLE_ENTITY",
  "message": "Validation failed",
  "action_time": "2025-12-11T10:30:45",
  "data": {
    "isPrivate": "must not be null"
  }
}

```

Standard Error Types:

Application-Level Exceptions (400-499)

- `401 UNAUTHORIZED`: Authentication issues (empty, invalid, expired, or malformed tokens)
- `422 UNPROCESSABLE_ENTITY`: Validation errors with detailed field information
- `500 INTERNAL_SERVER_ERROR`: Unexpected server errors

Error Response Examples:

Validation Error (422):

```
{
  "success": false,
  "httpStatus": "UNPROCESSABLE_ENTITY",
  "message": "Validation failed",
  "action_time": "2025-12-11T10:30:45",
  "data": {
    "isPrivate": "must not be null"
  }
}
```

Unauthorized - Token Issues (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2025-12-11T10:30:45",
  "data": "Token has expired"
}
```

Quick Reference Guide

Common HTTP Status Codes

- `200 OK`: Successful GET/PUT request
- `401 Unauthorized`: Authentication required/failed
- `422 Unprocessable Entity`: Validation errors
- `500 Internal Server Error`: Server error

Authentication

- **Bearer Token:** Include `Authorization: Bearer your_token` in headers

Data Format Standards

- **Dates:** Use ISO 8601 format (2025-12-11T14:30:00Z)
- **IDs:** UUID format (e.g., 550e8400-e29b-41d4-a716-446655440000)

Revision #3

Created 11 December 2025 09:05:15 by Admin Qbit

Updated 11 December 2025 09:06:49 by Admin Qbit