User

Endpoints Summary

Method	Path	Swagger
`GET`	`/user/`	Swagger ↗
`POST`	`/user/`	Swagger ↗
`GET`	`/user/username/{username}/`	Swagger ↗
`PUT`	`/user/username/{username}/`	Swagger ↗
`PATCH`	`/user/username/{username}/`	Swagger ↗
`DELETE`	`/user/username/{username}/`	Swagger ↗
`GET`	`/user/{id}/`	Swagger ↗
`PUT`	`/user/{id}/`	Swagger ↗
`PATCH`	`/user/{id}/`	Swagger ↗
`DELETE`	`/user/{id}/`	Swagger ↗

> The User API endpoints provide comprehensive user management capabilities including authentication, authorization, and access control (AAA) profile management. These endpoints allow you to create, retrieve, update, and delete users in your system, manage their profiles and permissions, and handle user lifecycle operations like account expiration and status management.

Base URL: `https://gate.zequenze.com/api/v1`

Authentication: All endpoints require a Bearer token:

```
Authorization: Bearer <your-api-token>
```

Overview

The User API is the core component for managing user accounts and their associated AAA (Authentication, Authorization, and Accounting) profiles within the GATE system. This API provides full CRUD operations for user management, allowing you to integrate user provisioning, profile management, and access control into your applications.

Key Features:

- **User Lifecycle Management**: Create, read, update, and delete user accounts with comprehensive profile information
- **AAA Profile Integration**: Assign and manage Authentication, Authorization, and Accounting profiles for fine-grained access control
- **Flexible User Identification**: Support for both internal ID-based operations and username-based operations for easier integration
- **External System Integration**: Built-in support for external IDs and service class references to link users with external systems
- **Account Management**: Handle user expiration dates, activation status, and login tracking
- **Organization Support**: Associate users with specific organizations for multi-tenant environments

Common Integration Scenarios:

- **Identity Provider Integration**: Sync users from LDAP, Active Directory, or other identity systems
- **Application User Management**: Provision users for web applications, network services, or cloud resources
- **Automated Onboarding**: Create user accounts as part of employee onboarding workflows
- **Access Control Management**: Assign and modify user permissions through AAA profile assignments
- **User Analytics**: Track user login patterns and account lifecycle for reporting and compliance

The API supports both paginated listing of users with flexible filtering options and individual user operations, making it suitable for both bulk operations and real-time user management tasks.

Endpoints

GET /user/

Description: Retrieves a paginated list of all users in the system, including their AAA profile assignments and comprehensive user information. This endpoint supports flexible filtering by user ID, username, or organization, making it ideal for user search functionality, bulk operations, and administrative dashboards.

Use Cases:

- Building user management interfaces with search and filtering capabilities
- Synchronizing user data between systems
- Generating user reports and analytics
- Bulk user operations and auditing

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/?username=john.doe&organization=1&limit=50&offset=0
```

Parameters:

Parameter	Type	In	Required	Description
id	string	query	No	Filter users by specific user ID
username	string	query	No	Filter users by username (supports partial matching)
organization	string	query	No	Filter users by organization ID
limit	integer	query	No	Number of results to return per page (default: 20, max: 100)
offset	integer	query	No	The initial index from which to return the results for pagination

cURL Example:

```
curl -X GET "https://gate.zequenze.com/api/v1/user/?organization=1&limit=25&offset=0" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:

```
{
  "count": 150,
```

```
  "next": "https://gate.zequenze.com/api/v1/user/?limit=25&offset=25",
  "previous": null,
  "results": [
    {
      "id": 1,
      "username": "john.doe",
      "external_id": "EMP001234",
      "klass": "standard_employee",
      "email": "john.doe@company.com",
      "first_name": "John",
      "last_name": "Doe",
      "is_active": true,
      "organization": 1,
      "description": "Senior Network Engineer",
      "date_joined": "2024-01-15T10:30:00Z",
      "first_login": "2024-01-16T08:15:00Z",
      "last_login": "2024-03-20T14:22:00Z",
      "expiration": "2025-01-15T23:59:59Z",
      "avatar_url": "https://gate.zequenze.com/avatars/john.doe.jpg",
      "source_id": 1,
      "profile": ["network_admin", "vpn_user", "wifi_access"]
    }
  ]
}
```


Response Codes:

Status	Description
200	Success - Returns paginated list of users
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions to list users

POST /user/

Description: Creates a new user account with optional AAA profile assignments. This endpoint allows you to provision new users with comprehensive profile information, external system references, and initial access permissions. The username is the only required field, with all other user attributes being optional.

Use Cases:

- User onboarding and account provisioning workflows
- Automated user creation from HR systems or identity providers
- Self-service account registration processes
- Bulk user import operations

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/
```

Request Body Schema:

Field	Type	Required	Description
username	string	Yes	Unique username (150 chars max, letters, digits, @/./+/-/:/
password	string	No	User password (will be hashed)
email	string	No	Valid email address
first_name	string	No	User's first name
last_name	string	No	User's last name
external_id	string	No	Reference ID for external systems
klass	string	No	Service class identifier
is_active	boolean	No	Account active status (default: true)
organization	integer	No	Organization ID to associate user with
description	string	No	Additional user description or notes
expiration	string	No	Account expiration date (ISO 8601 format)
avatar_url	string	No	URL to user's avatar image
profile	array	No	List of AAA profile short-names to assign

cURL Example:

```
curl -X POST "https://gate.zequenze.com/api/v1/user/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
```

```
  -d '{
    "username": "jane.smith",
    "email": "jane.smith@company.com",
    "first_name": "Jane",
    "last_name": "Smith",
    "password": "SecurePassword123!",
    "external_id": "EMP005678",
    "organization": 1,
    "description": "Software Developer",
    "expiration": "2025-12-31T23:59:59Z",
    "profile": ["developer_access", "vpn_user"]
  }'
```

Example Response:


```
{
  "id": 42,
  "username": "jane.smith",
  "external_id": "EMP005678",
  "klass": null,
  "email": "jane.smith@company.com",
  "first_name": "Jane",
  "last_name": "Smith",
  "is_active": true,
  "organization": 1,
  "description": "Software Developer",
  "date_joined": "2024-03-21T10:30:00Z",
  "first_login": null,
  "last_login": null,
  "expiration": "2025-12-31T23:59:59Z",
  "avatar_url": null,
  "profile": ["developer_access", "vpn_user"]
}
```


Response Codes:

Status	Description
201	Created - User successfully created
400	Bad Request - Invalid data or validation errors
401	Unauthorized - Invalid or missing API token

Status	Description
409	Conflict - Username already exists

GET /user/username/{username}/

Description: Retrieves detailed information for a specific user identified by their username. This endpoint is particularly useful when you know the username but not the internal user ID, making it ideal for user lookup operations and profile management interfaces.

Use Cases:

- User profile lookups in applications
- Username-based authentication flows
- User information display in administrative interfaces
- Integration with systems that use usernames as primary identifiers

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/username/john.doe/
```

cURL Example:

```
curl -X GET "https://gate.zequenze.com/api/v1/user/username/john.doe/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Example Response:

```
{
  "id": 1,
  "username": "john.doe",
  "external_id": "EMP001234",
  "klass": "standard_employee",
  "email": "john.doe@company.com",
  "first_name": "John",
  "last_name": "Doe",
  "is_active": true,
  "organization": 1,
  "description": "Senior Network Engineer",
  "date_joined": "2024-01-15T10:30:00Z",
```

```
  "first_login": "2024-01-16T08:15:00Z",
  "last_login": "2024-03-20T14:22:00Z",
  "expiration": "2025-01-15T23:59:59Z",
  "avatar_url": "https://gate.zequenze.com/avatars/john.doe.jpg",
  "source_id": 1,
  "profile": ["network_admin", "vpn_user", "wifi_access"]
}
```

Response Codes:

Status	Description
200	Success - Returns user information
401	Unauthorized - Invalid or missing API token
404	Not Found - Username does not exist

PUT /user/username/{username}/

Description: Completely replaces all user information for the specified username. This endpoint performs a full update, requiring all fields to be provided as it will overwrite the existing user record entirely. Use PATCH for partial updates instead.

Use Cases:

- Complete user profile synchronization from external systems
- User migration between systems with full data replacement
- Administrative bulk user updates where all fields are known

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/username/john.doe/
```

cURL Example:

```
curl -X PUT "https://gate.zequenze.com/api/v1/user/username/john.doe/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "username": "john.doe",
    "email": "john.doe@newcompany.com",
    "first_name": "John",
```

```
    "last_name": "Doe-Smith",
    "external_id": "EMP001234",
    "klass": "senior_employee",
    "is_active": true,
    "organization": 2,
    "description": "Principal Network Architect",
    "expiration": "2026-01-15T23:59:59Z",
    "profile": ["network_admin", "infrastructure_admin", "vpn_user"]
  }'
```

Response Codes:

Status	Description
200	Success - User successfully updated
400	Bad Request - Invalid data or validation errors
401	Unauthorized - Invalid or missing API token
404	Not Found - Username does not exist

PATCH /user/username/{username}/

Description: Performs a partial update of user information for the specified username. Only the fields provided in the request body will be updated, leaving all other user attributes unchanged. This is the preferred method for making specific changes to user accounts.

Use Cases:

- Updating specific user attributes like email or organization
- Modifying AAA profile assignments
- Activating or deactivating user accounts
- Extending or modifying user expiration dates

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/username/john.doe/
```

cURL Example:

```
curl -X PATCH "https://gate.zequenze.com/api/v1/user/username/john.doe/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
```

```
  -d '{
    "email": "john.doe@updated-email.com",
    "organization": 3,
    "profile": ["network_admin", "vpn_user", "new_access_profile"]
  }'
```

Response Codes:

Status	Description
200	Success - User successfully updated
400	Bad Request - Invalid data or validation errors
401	Unauthorized - Invalid or missing API token
404	Not Found - Username does not exist

DELETE /user/username/{username}/

Description: Permanently deletes the user account identified by the specified username. This operation cannot be undone and will remove all user data, including AAA profile assignments and historical login information.

Use Cases:

- User offboarding and account cleanup
- Removing test accounts or invalid user records
- Compliance-driven account deletion requirements
- Automated account lifecycle management

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/username/john.doe/
```

cURL Example:

```
curl -X DELETE "https://gate.zequenze.com/api/v1/user/username/john.doe/" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Response Codes:

Status	Description
204	No Content - User successfully deleted

Status	Description
401	Unauthorized - Invalid or missing API token
404	Not Found - Username does not exist

GET /user/{id}/

Description: Retrieves detailed information for a specific user identified by their internal user ID. This endpoint provides the same functionality as the username-based lookup but uses the system's internal ID as the identifier.

Use Cases:

- Retrieving user information when you have the internal ID
- Following references from other API responses that include user IDs
- Building relationships between users and other system entities

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/42/
```

cURL Example:

```
curl -X GET "https://gate.zequenze.com/api/v1/user/42/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json"
```

Response Codes:

Status	Description
200	Success - Returns user information
401	Unauthorized - Invalid or missing API token
404	Not Found - User ID does not exist

PUT /user/{id}/

Description: Completely replaces all user information for the specified user ID. This endpoint performs a full update, requiring all fields to be provided as it will overwrite the existing user record entirely.

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/42/
```

cURL Example:

```
curl -X PUT "https://gate.zequenze.com/api/v1/user/42/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "username": "jane.smith",
    "email": "jane.smith@company.com",
    "first_name": "Jane",
    "last_name": "Smith-Johnson",
    "is_active": true,
    "organization": 1,
    "profile": ["developer_access", "admin_access"]
  }'
```

Response Codes:

Status	Description
200	Success - User successfully updated
400	Bad Request - Invalid data or validation errors
401	Unauthorized - Invalid or missing API token
404	Not Found - User ID does not exist

PATCH /user/{id}/

Description: Performs a partial update of user information for the specified user ID. Only the fields provided in the request body will be updated, making this ideal for targeted modifications to user accounts.

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/42/
```

cURL Example:

```
curl -X PATCH "https://gate.zequenze.com/api/v1/user/42/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "is_active": false,
    "description": "Account suspended pending review"
  }'
```

Response Codes:

Status	Description
200	Success - User successfully updated
400	Bad Request - Invalid data or validation errors
401	Unauthorized - Invalid or missing API token
404	Not Found - User ID does not exist

DELETE /user/{id}/

Description: Permanently deletes the user account identified by the specified user ID. This operation cannot be undone and will remove all user data and associated relationships.

Full URL Example:

```
https://gate.zequenze.com/api/v1/user/42/
```

cURL Example:

```
curl -X DELETE "https://gate.zequenze.com/api/v1/user/42/" \
  -H "Authorization: Bearer YOUR_API_TOKEN"
```

Response Codes:

Status	Description
204	No Content - User successfully deleted
401	Unauthorized - Invalid or missing API token
404	Not Found - User ID does not exist

Common Use Cases

Use Case 1: Employee Onboarding Workflow

When a new employee joins your organization, you can automate their account creation and access provisioning using POST /user/ to create the account with appropriate AAA profiles, then use PATCH operations to update their information as they progress through onboarding stages.

Use Case 2: Directory Synchronization

For organizations using external identity providers, use GET /user/ with filtering to retrieve current users, compare with your external directory, then use POST to create new users and PATCH to update existing ones, ensuring your GATE system stays synchronized.

Use Case 3: Access Management and Compliance

Use GET /user/ to generate reports on user access and expiration dates, then use PATCH operations to update user profiles, modify expiration dates, or deactivate accounts that no longer require access, maintaining compliance with security policies.

Use Case 4: Self-Service Profile Management

Build user portals where users can view their information using GET /user/username/{username}/ and submit profile updates that your system processes using PATCH /user/username/{username}/ after appropriate approval workflows.

Use Case 5: Account Lifecycle Automation

Implement automated processes that monitor user activity and account status, using PATCH operations to extend expiration dates for active users, deactivate dormant accounts, or DELETE accounts that are no longer needed based on your organization's retention policies.

Best Practices

- **Use Pagination Effectively**: When listing users, always implement proper pagination using limit and offset parameters to avoid performance issues with large user datasets. Start with reasonable page sizes (25-50 users) and adjust based on your application's

needs.

- **Leverage Username-Based Operations**: For user-facing applications, prefer the username-based endpoints (/user/username/{username}/) as they're more intuitive and don't expose internal system IDs. Reserve ID-based operations for internal system integrations.
- **Implement Proper Error Handling**: Always check response codes and handle common scenarios like 404 (user not found), 409 (username conflicts), and 401 (authentication failures) gracefully in your applications.
- **Manage AAA Profiles Carefully**: When updating user profiles, retrieve the current profile list first to avoid accidentally removing necessary access permissions. Consider implementing approval workflows for profile changes that grant additional access.
- **Use Partial Updates**: Prefer PATCH over PUT operations to avoid accidentally clearing user data. PATCH operations are safer and more efficient as they only modify the specified fields.
- **Monitor Account Expiration**: Implement monitoring for user expiration dates and automate extension or notification processes. Users with expired accounts may lose access unexpectedly if not properly managed.
- **Secure Sensitive Operations**: Implement additional validation and approval processes for sensitive operations like user deletion or profile modifications that grant administrative access.
- **Cache User Information Wisely**: Consider caching user information for frequently accessed data, but ensure cache invalidation strategies are in place for when user information changes, especially for authentication and authorization data.

Revision #4
Created 2026-02-04 05:16:20 UTC by ipena@zequenze.com
Updated 2026-02-11 03:22:05 UTC by ipena@zequenze.com