

# User Token

## Endpoints Summary

Method	Path	Swagger
GET	<a href="#">/user_token/</a>	<a href="#">Swagger ↗</a>
POST	<a href="#">/user_token/</a>	<a href="#">Swagger ↗</a>
GET	<a href="#">/user_token/{id}/</a>	<a href="#">Swagger ↗</a>
PUT	<a href="#">/user_token/{id}/</a>	<a href="#">Swagger ↗</a>
PATCH	<a href="#">/user_token/{id}/</a>	<a href="#">Swagger ↗</a>
DELETE	<a href="#">/user_token/{id}/</a>	<a href="#">Swagger ↗</a>

“ The User Token API manages users within the GATE system, including their authentication credentials, AAA (Authentication, Authorization, and Accounting) profiles, and time-based access controls. These endpoints allow you to create, retrieve, update, and delete user accounts with detailed profile information and token-based time management for access control systems.

**Base URL:** <https://gate.zequenze.com/api/v1>

**Authentication:** All endpoints require a Bearer token:

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

## Overview

The User Token API provides comprehensive user management functionality for the GATE system. This API category is designed for managing user accounts that require time-based access control, typically used in scenarios where users need allocated time credits for services or resources.

## Key Features:

- **User Account Management:** Create, read, update, and delete user accounts with detailed profile information
- **AAA Profile Integration:** Associate users with Authentication, Authorization, and Accounting profiles
- **Time-Based Access Control:** Manage user time allocations (in seconds) for token-based systems
- **Organization Filtering:** Group and filter users by organization
- **Balance Tracking:** Monitor remaining time balances for users
- **External System Integration:** Link users to external systems via external\_id references

## Common Use Cases:

- Internet café or hotspot user management
- Time-based service access control
- Guest network authentication systems
- Subscription-based time allocation services
- Multi-tenant user management with organizational boundaries

The API uses standard REST principles with JSON responses and supports pagination for list operations. User time is tracked in seconds, making it suitable for precise time-based billing and access control scenarios.

---

# Endpoints

## GET /user\_token/

**Description:** Retrieves a paginated list of all users in the system, including their AAA profile information. This endpoint supports filtering by username and organization, and can optionally include remaining time balance information.

### Use Cases:

- Display all users in an administrative dashboard
- Filter users by specific organizations for multi-tenant systems
- Search for specific users by username
- Monitor user balances across the system

### Full URL Example:

```
https://gate.zequenze.com/api/v1/user_token/?username=john_doe&organization=acme_corp&balance=true&limit=20&offset=0
```

### Parameters:

Parameter	Type	In	Required	Description
username	string	query	No	Filter users by exact username match
organization	string	query	No	Filter users by organization identifier
limit	integer	query	No	Number of results per page (default: 20)
offset	integer	query	No	Starting index for pagination (default: 0)
balance	boolean	query	No	Include remaining time balance in seconds

### cURL Example:

```
curl -X GET "https://gate.zequenze.com/api/v1/user_token/?balance=true&limit=10" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json"
```

### Example Response:

```
{  
  "count": 150,  
  "next": "https://gate.zequenze.com/api/v1/user_token/?limit=10&offset=10",  
  "previous": null,  
  "results": [  
    {  
      "id": 1,  
      "username": "john_doe",  
      "external_id": "EXT-USER-001",  
      "klass": "premium",  
      "is_active": true,  
      "organization": 1,  
      "description": "Premium user account",  
      "date_joined": "2024-01-15T10:30:00Z",
```

```
"last_login": "2024-01-20T14:25:00Z",
"expiration": "2024-12-31T23:59:59Z",
"profile": ["internet_access", "wifi_premium"],
"user_time": 3600
},
{
  "id": 2,
  "username": "jane_smith",
  "external_id": "EXT-USER-002",
  "klass": "standard",
  "is_active": true,
  "organization": 1,
  "description": "Standard user account",
  "date_joined": "2024-01-16T09:15:00Z",
  "last_login": "2024-01-19T16:45:00Z",
  "expiration": null,
  "profile": ["internet_access"],
  "user_time": 1800
}
]
}
```

### Response Codes:

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

## POST /user\_token/

**Description:** Creates a new user account with AAA profile information. This endpoint allows you to set up complete user profiles including authentication credentials, organizational assignment, and initial time allocations.

### Use Cases:

- Register new users in the system
- Bulk user creation for organizations

- Set up guest accounts with specific time limits
- Create users with predefined AAA profiles

### Full URL Example:

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

### Request Body Parameters:

Parameter	Type	Required	Description
username	string	Yes	Unique username (150 chars max, letters/digits/@/./+/-:/)
password	string	No	User password for authentication
email	string	No	User 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 reference (e.g., "premium", "standard")
is_active	boolean	No	Whether user account is active (default: true)
organization	integer	No	Organization ID to associate user with
description	string	No	Descriptive text about the user
expiration	string	No	ISO 8601 date-time when user expires
avatar_url	string	No	URL to user's avatar image
profile	array	No	Array of AAA profile short-names

### cURL Example:

```
curl -X POST "https://gate.zequenze.com/api/v1/user_token/" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "username": "new_user",
```

```
"password": "secure_password123",
"email": "newuser@example.com",
"first_name": "New",
"last_name": "User",
"external_id": "EXT-USER-003",
"klass": "premium",
"is_active": true,
"organization": 1,
"description": "Newly created premium user",
"expiration": "2024-12-31T23:59:59Z",
"profile": ["internet_access", "wifi_premium", "email_access"]
}'
```

### Example Response:

```
{
  "id": 3,
  "username": "new_user",
  "external_id": "EXT-USER-003",
  "klass": "premium",
  "email": "newuser@example.com",
  "first_name": "New",
  "last_name": "User",
  "is_active": true,
  "organization": 1,
  "description": "Newly created premium user",
  "date_joined": "2024-01-21T11:30:00Z",
  "first_login": null,
  "last_login": null,
  "expiration": "2024-12-31T23:59:59Z",
  "avatar_url": null,
  "profile": ["internet_access", "wifi_premium", "email_access"]
}
```

### Response Codes:

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

Status	Description
403	Forbidden - Insufficient permissions

## GET /user\_token/{id}/

**Description:** Retrieves detailed information for a specific user by their ID, including AAA profile information and optionally their remaining time balance.

### Use Cases:

- Display user profile in administrative interface
- Check user details before granting access
- Retrieve user information for authentication systems
- Monitor individual user time balances

### Full URL Example:

```
https://gate.zequenze.com/api/v1/user_token/123/?balance=true
```

### Parameters:

Parameter	Type	In	Required	Description
id	integer	path	Yes	Unique identifier of the user
balance	boolean	query	No	Include remaining time balance in seconds

### cURL Example:

```
curl -X GET "https://gate.zequenze.com/api/v1/user_token/1/?balance=true" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json"
```

### Example Response:

```
{  
  "id": 1,  
  "username": "john_doe",  
  "external_id": "EXT-USER-001",  
  "klass": "premium",
```

```
"is_active": true,
"organization": 1,
"description": "Premium user account",
"date_joined": "2024-01-15T10:30:00Z",
"last_login": "2024-01-20T14:25:00Z",
"expiration": "2024-12-31T23:59:59Z",
"profile": ["internet_access", "wifi_premium", "email_access"],
"user_time": 3600
}
```

### Response Codes:

Status	Description
200	Success - Returns user details
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions
404	Not Found - User with specified ID does not exist

## PUT /user\_token/{id}/

**Description:** Completely updates an existing user account, replacing all modifiable fields. This is a full update operation that requires all fields you want to preserve to be included in the request.

### Use Cases:

- Complete user profile updates
- Bulk user information changes
- Resetting user configurations
- Migrating user data between systems

### Full URL Example:

```
https://gate.zequenze.com/api/v1/user_token/123/
```

### cURL Example:

```
curl -X PUT "https://gate.zequenze.com/api/v1/user_token/1/" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json" \
-d '{
```

```
"username": "john_doe_updated",
"external_id": "EXT-USER-001-UPDATED",
"klass": "enterprise",
"is_active": true,
"organization": 2,
"description": "Updated to enterprise user",
"expiration": "2025-12-31T23:59:59Z",
"profile": ["internet_access", "wifi_premium", "email_access", "vpn_access"],
"user_time": 7200
}'
```

### Example Response:

```
{
  "id": 1,
  "username": "john_doe_updated",
  "external_id": "EXT-USER-001-UPDATED",
  "klass": "enterprise",
  "is_active": true,
  "organization": 2,
  "description": "Updated to enterprise user",
  "date_joined": "2024-01-15T10:30:00Z",
  "last_login": "2024-01-20T14:25:00Z",
  "expiration": "2025-12-31T23:59:59Z",
  "profile": ["internet_access", "wifi_premium", "email_access", "vpn_access"],
  "user_time": 7200
}
```

### Response Codes:

Status	Description
200	Success - User updated successfully
400	Bad Request - Invalid data provided
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions
404	Not Found - User with specified ID does not exist

# PATCH /user\_token/{id}/

**Description:** Partially updates a user account, allowing you to modify only specific fields without affecting other user data. This is ideal for making incremental changes to user accounts.

### Use Cases:

- Update user time balance without changing other details
- Modify user status (active/inactive)
- Add or remove specific AAA profiles
- Update expiration dates or organizational assignments

### Full URL Example:

```
https://gate.zequenze.com/api/v1/user_token/123/
```

### cURL Example:

```
curl -X PATCH "https://gate.zequenze.com/api/v1/user_token/1/" \  
-H "Authorization: Bearer YOUR_API_TOKEN" \  
-H "Content-Type: application/json" \  
-d '{  
  "user_time": 5400,  
  "description": "Time balance updated",  
  "profile": ["internet_access", "wifi_premium"]  
}'
```

### Example Response:

```
{  
  "id": 1,  
  "username": "john_doe",  
  "external_id": "EXT-USER-001",  
  "klass": "premium",  
  "is_active": true,  
  "organization": 1,  
  "description": "Time balance updated",  
  "date_joined": "2024-01-15T10:30:00Z",  
  "last_login": "2024-01-20T14:25:00Z",  
  "expiration": "2024-12-31T23:59:59Z",  
  "profile": ["internet_access", "wifi_premium"],  
  "user_time": 5400  
}
```

## Response Codes:

Status	Description
200	Success - User updated successfully
400	Bad Request - Invalid data provided
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions
404	Not Found - User with specified ID does not exist

# DELETE /user\_token/{id}/

**Description:** Permanently removes a user account from the system. This action cannot be undone and will delete all associated user data and tokens.

## Use Cases:

- Remove expired or unused accounts
- Clean up test or temporary users
- Comply with data retention policies
- Decommission user accounts

## Full URL Example:

```
https://gate.zequenze.com/api/v1/user_token/123/
```

## cURL Example:

```
curl -X DELETE "https://gate.zequenze.com/api/v1/user_token/1/" \  
-H "Authorization: Bearer YOUR_API_TOKEN"
```

## Response Codes:

Status	Description
204	No Content - User successfully deleted
401	Unauthorized - Invalid or missing API token
403	Forbidden - Insufficient permissions
404	Not Found - User with specified ID does not exist

# Common Use Cases

## Use Case 1: Internet Café User Management

Create time-based user accounts for internet café customers with specific time allocations. Use POST to create users with `user_time` values, then use PATCH to update remaining time as customers use the service.

## Use Case 2: Guest Network Access

Set up temporary guest accounts with expiration dates and limited AAA profiles. Use the `expiration` field to automatically invalidate accounts and the `profile` array to restrict access to specific services.

## Use Case 3: Multi-Tenant Organization Management

Filter and manage users across different organizations using the `organization` parameter. Use GET with organization filtering to display users per tenant, and ensure proper isolation between organizations.

## Use Case 4: Subscription Service Time Tracking

Monitor user time balances across your subscriber base using the `balance=true` parameter with GET requests. Use PATCH operations to add time when users purchase additional credits.

## Use Case 5: External System Integration

Synchronize users with external databases or CRM systems using the `external_id` field to maintain references between systems while managing authentication through GATE.

---

# Best Practices

- **Time Management:** Always specify time values in seconds for consistency. Consider implementing automatic time deduction mechanisms that integrate with these API endpoints.
- **Pagination:** When retrieving large user lists, use appropriate `limit` and `offset` values to prevent timeouts and improve performance. Start with `limit=50` and adjust based on your

needs.

- **Balance Monitoring:** Use the `balance=true` parameter judiciously as it adds computational overhead. Only include it when you specifically need current balance information.
  - **Security Considerations:** Never store passwords in plain text. The API handles password hashing automatically, but ensure your client applications don't log or expose password data.
  - **Error Handling:** Implement retry logic for network failures and handle 404 errors gracefully when users might have been deleted by other processes.
  - **Bulk Operations:** For creating or updating multiple users, implement batching with appropriate delays to avoid overwhelming the API server.
  - **Profile Management:** Maintain a reference list of available AAA profiles in your application to ensure you're assigning valid profiles to users.
- 

Revision #4

Created 2026-02-04 05:17:09 UTC by ipena@zequenze.com

Updated 2026-02-11 03:24:15 UTC by ipena@zequenze.com