
Appointment API Documentation

Base URL

/api/appointments

Create Appointment

- **URL:** /api/appointments
- **Method:** POST
- **Description:** Creates a new appointment.
- **Request Parameters:**
 - doctorUsername (String): Username of the doctor.
 - patientId (Long): ID of the patient.
 - appointmentDate (ISO 8601 DateTime): Date and time of the appointment.
 - location (String): Location of the appointment.
- **Response:** 201 Created

```
{
  "id": "Long",
  "patientId": "Long",
  "doctorId": "Long",
  "appointmentDate": "ISO 8601 DateTime",
  "location": "String",
  "status": "String",
  "createdAt": "ISO 8601 DateTime"
}
```
- **Note:** appointmentDate must be in ISO 8601 format (YYYY-MM-DDTHH:MM:SS).

Get Appointments by Doctor

- **URL:** /api/appointments/doctor/{doctorId}
- **Method:** GET
- **Description:** Retrieves all appointments for a specific doctor.
- **Path Parameter:**
 - doctorId (Long): ID of the doctor.

- **Response:** 200 OK

```
[
  {
    "id": "Long",
    "patientId": "Long",
    "doctorId": "Long",
    "appointmentDate": "ISO 8601 DateTime",
    "location": "String",
    "status": "String",
    "createdAt": "ISO 8601 DateTime"
  }
]
```

Get Appointments by Patient

- **URL:** /api/appointments/patient/{patientId}
 - **Method:** GET
 - **Description:** Retrieves all appointments for a specific patient.
 - **Path Parameter:**
 - patientId (Long): ID of the patient.
 - **Response:** 200 OK
-

Get Appointments by Location

- **URL:** /api/appointments/location
 - **Method:** GET
 - **Description:** Retrieves all appointments at a specific location.
 - **Query Parameter:**
 - location (String): Location of the appointment.
 - **Response:** 200 OK
-

Get Appointments by Status

- **URL:** /api/appointments/status
 - **Method:** GET
 - **Description:** Retrieves all appointments with a specific status.
 - **Query Parameter:**
 - status (String): Status of the appointments (e.g., “active”, “canceled”).
 - **Response:** 200 OK
-

Delete Appointment

- **URL:** /api/appointments/{appointmentId}
 - **Method:** DELETE
 - **Description:** Deletes an appointment by ID.
 - **Path Parameter:**
 - appointmentId (Long): ID of the appointment.
 - **Response:** 204 No Content
-

Update Appointment Status

- **URL:** /api/appointments/{appointmentId}/status
 - **Method:** PUT
 - **Description:** Updates the status of an appointment.
 - **Path Parameter:**
 - appointmentId (Long): ID of the appointment.
 - **Request Parameter:**
 - status (String): New status of the appointment.
 - **Response:** 200 OK

```
{
  "id": "Long",
  "patientId": "Long",
  "doctorId": "Long",
  "appointmentDate": "ISO 8601 DateTime",
  "location": "String",
  "status": "String",
  "createdAt": "ISO 8601 DateTime"
}
```
 - **Error Responses:**
 - 400 Bad Request: Invalid status provided.
-

Get Appointments by Date Range

- **URL:** /api/appointments/date-range
- **Method:** GET
- **Description:** Retrieves appointments within a specified date range.
- **Query Parameters:**
 - start (ISO 8601 DateTime): Start of the date range.
 - end (ISO 8601 DateTime): End of the date range.

- **Response:** 200 OK

Note: Dates should be provided in ISO 8601 format (YYYY-MM-DDTHH:MM:SS).

Medical Records API Documentation

Base URL: /api/medical-records

Endpoints

Create Medical Record

POST /

Creates a new medical record.

Request Body:

```
{
  "patientId": "long",
  "doctorId": "long",
  "dateOfVisit": "datetime",
  "location": "string",
  "diagnosis": "string",
  "treatmentPlan": "string",
  "notes": "string"
}
```

Response: 200 OK

```
{
  "id": "long",
  "patientId": "long",
  "doctorId": "long",
  "dateOfVisit": "datetime",
  "location": "string",
  "diagnosis": "string",
  "treatmentPlan": "string",
  "notes": "string",
  "createdAt": "datetime"
}
```

Get All Medical Records

GET /

Retrieves all medical records.

Response: 200 OK

```
[
  {
    "id": "long",
    "patientId": "long",
    // ... (full medical record response)
  }
]
```

```
}  
]
```

Get Records by Patient

GET /patient/{patientId}

Retrieves all medical records for a specific patient.

Path Parameters: - `patientId` (long): ID of the patient

Response: 200 OK Array of medical record responses

Get Records by Doctor

GET /doctor/{doctorId}

Retrieves all medical records for a specific doctor.

Path Parameters: - `doctorId` (long): ID of the doctor

Response: 200 OK Array of medical record responses

Get Records by Patient and Doctor

GET /patient/{patientId}/doctor/{doctorId}

Retrieves all medical records for a specific patient-doctor combination.

Path Parameters: - `patientId` (long): ID of the patient - `doctorId` (long): ID of the doctor

Response: 200 OK Array of medical record responses

Get Records by Location

GET /location

Retrieves all medical records from a specific location.

Query Parameters: - `location` (string): Location name or identifier

Response: 200 OK Array of medical record responses

Get Records by Date Range

GET /date-range

Retrieves all medical records within a specified date range.

Query Parameters: - `startDate` (datetime, ISO 8601 format): Start of date range - `endDate` (datetime, ISO 8601 format): End of date range

Example Request:

GET /api/medical-records/date-range?startDate=2024-01-01T00:00:00Z&endDate=2024-02-01T00:00:00Z

Response: 200 OK Array of medical record responses

Search Medical Records

POST /search

Searches medical records based on multiple criteria.

Request Body:

```
{
  "patientId": "long",
  "doctorId": "long",
  "startDate": "datetime",
  "endDate": "datetime",
  "location": "string",
  "diagnosisOrTreatment": "string"
}
```

Response: 200 OK Array of medical record responses

Data Models

MedicalRecordRequest

```
{
  "patientId": "long",
  "doctorId": "long",
  "dateOfVisit": "datetime",
  "location": "string",
  "diagnosis": "string",
  "treatmentPlan": "string",
  "notes": "string"
}
```

MedicalRecordResponse

```
{
  "id": "long",
  "patientId": "long",
  "doctorId": "long",
  "dateOfVisit": "datetime",
  "location": "string",
  "diagnosis": "string",
  "treatmentPlan": "string",
  "notes": "string",
}
```

```
    "createdAt": "datetime"  
  }
```

MedicalRecordSearchCriteria

```
{  
  "patientId": "long",  
  "doctorId": "long",  
  "startDate": "datetime",  
  "endDate": "datetime",  
  "location": "string",  
  "diagnosisOrTreatment": "string"  
}
```

Date Format

All datetime fields should be provided in ISO 8601 format: - Example:
2024-01-01T00:00:00Z

Error Responses

All endpoints may return these error responses: - **400 Bad Request**: Invalid request parameters or body - **404 Not Found**: Requested resource not found - **500 Internal Server Error**: Server error

Medications API Documentation

Base URL: /api/medications

Authorization Requirements

All endpoints require: - Valid username parameter - Staff role - Specific privilege level: - **editor**: Basic medication management - **supervisor**: Advanced querying and listing

Endpoints

Add Medication

POST /

Creates a new medication record.

Required Authorization: - Staff role - **editor** privilege

Query Parameters: - **username** (string): Username of the requesting user

Request Body:

```
{
  "medicationName": "string",
  "dosage": "string",
  "frequency": "string",
  "startDate": "datetime",
  "endDate": "datetime",
  "medicalRecordId": "long"
}
```

Responses: - 201 Created: Medication created successfully json {
"id": "long", "medicationName": "string", "dosage":
"string", "frequency": "string", "startDate": "datetime",
"endDate": "datetime", "medicalRecordId": "long", "createdAt":
"datetime" } - 400 Bad Request: "Invalid medication request" - 500
Internal Server Error: "An error occurred"

Get All Medications

GET /

Retrieves all medications.

Required Authorization: - Staff role - **supervisor** privilege

Query Parameters: - **username** (string): Username of the requesting user

Response: 200 OK Array of medication responses

Get Medications by Medical Record

GET /record/{medicalRecordId}

Retrieves all medications associated with a specific medical record.

Required Authorization: - Staff role - `editor` privilege

Path Parameters: - `medicalRecordId` (long): ID of the medical record

Query Parameters: - `username` (string): Username of the requesting user

Responses: - 200 OK: Array of medication responses - 404 Not Found: "Medical record not found" - 500 Internal Server Error: "An error occurred"

Delete Medication

DELETE /{medicationId}

Deletes a specific medication.

Required Authorization: - Staff role - `editor` privilege

Path Parameters: - `medicationId` (long): ID of the medication to delete

Query Parameters: - `username` (string): Username of the requesting user

Responses: - 204 No Content: "Medication deleted successfully" - 404 Not Found: "Medication not found" - 500 Internal Server Error: "An error occurred"

Get Active Medications

GET /active

Retrieves all currently active medications.

Required Authorization: - Staff role - `supervisor` privilege

Query Parameters: - `username` (string): Username of the requesting user

Response: 200 OK Array of medication responses

Search Medications by Name

GET /search

Searches medications by name pattern.

Required Authorization: - Staff role - `supervisor` privilege

Query Parameters: - `name` (string): Medication name search pattern - `username` (string): Username of the requesting user

Response: 200 OK Array of medication responses

Get Expiring Medications

GET /expiring

Retrieves medications that are expiring soon.

Required Authorization: - Staff role - editor privilege

Query Parameters: - username (string): Username of the requesting user

Response: 200 OK Array of medication responses

Data Models

MedicationRequest

```
{
  "medicationName": "string",
  "dosage": "string",
  "frequency": "string",
  "startDate": "datetime",
  "endDate": "datetime",
  "medicalRecordId": "long"
}
```

MedicationResponse

```
{
  "id": "long",
  "medicationName": "string",
  "dosage": "string",
  "frequency": "string",
  "startDate": "datetime",
  "endDate": "datetime",
  "medicalRecordId": "long",
  "createdAt": "datetime"
}
```

Common Error Responses

All endpoints may return these error responses: - 400 **Bad Request**: Invalid request parameters or body - 401 **Unauthorized**: Missing or invalid username - 403 **Forbidden**: Insufficient privileges or invalid role - 404 **Not Found**: Requested resource not found - 500 **Internal Server Error**: Server error

Date Format

All datetime fields should be provided in ISO 8601 format: - Example: 2024-01-01T00:00:00Z

Patient Management API Documentation

Base URL: /api/patients

Authorization Levels

This API uses role-based access control with the following privileges: - **editor**: Can create and view patients - **supervisor**: Can perform searches, deletions, and advanced operations

Endpoints

Create Patient

POST /

Creates a new patient record.

Required Role: editor

Query Parameters: - username (string): Username of the requesting user

Request Body:

```
{
  "firstName": "string",
  "lastName": "string",
  "dob": "date",
  "gender": "string",
  "address": "string",
  "phoneNumber": "string",
  "emergencyContact": "string",
  "emergencyContactPhone": "string"
}
```

Responses: - 201 Created: Patient created successfully json { "id": "long", "firstName": "string", "lastName": "string", "dob": "date", "gender": "string", "address": "string", "phoneNumber": "string", "emergencyContact": "string", "emergencyContactPhone": "string", "createdAt": "datetime" } - 400 Bad Request: Invalid request data - 403 Forbidden: Insufficient privileges

Get All Patients

GET /

Retrieves all patients.

Required Role: editor

Query Parameters: - username (string): Username of the requesting user

Response: 200 OK

```
[
  {
    "id": "long",
    "firstName": "string",
    "lastName": "string",
    // ... (full patient response)
  }
]
```

Get Patient by Login

GET /

Retrieves a patient's own information using their login credentials.

Request Body:

```
{
  "username": "string",
  "password": "string"
}
```

Responses: - 200 OK: Full patient information - 400 Bad Request: Invalid credentials - 403 Forbidden: Access denied

Delete Patient

DELETE /{patientId}

Deletes a patient record.

Required Role: supervisor

Path Parameters: - patientId (long): ID of the patient to delete

Query Parameters: - username (string): Username of the requesting user

Responses: - 204 No Content: Patient successfully deleted - 403 Forbidden: Insufficient privileges - 404 Not Found: Patient not found

Search Endpoints

Search by Last Name

GET /search/lastname

Required Role: supervisor

Query Parameters: - lastName (string): Last name to search for - username (string): Username of the requesting user

Response: 200 OK

```
[
  {
    "id": "long",
    "firstName": "string",
    "lastName": "string",
    "phoneNumber": "string"
  }
]
```

Keyword Search

GET /search/keyword

Required Role: supervisor

Query Parameters: - keyword (string): Search term - username (string): Username of the requesting user

Response: 200 OK

```
[
  {
    "id": "long",
    "firstName": "string",
    "lastName": "string",
    "dob": "date",
    "gender": "string",
    "phoneNumber": "string"
  }
]
```

Search by Age Range

GET /search/age-range

Required Role: supervisor

Query Parameters: - minAge (integer): Minimum age - maxAge (integer): Maximum age - username (string): Username of the requesting user

Response: 200 OK

```
[
  {
    "id": "long",
    "firstName": "string",
    "lastName": "string",
    "dob": "date",
    "gender": "string",
  }
]
```

```
    "phoneNumber": "string"
  }
]
```

Data Models

PatientRequest

```
{
  "firstName": "string",
  "lastName": "string",
  "dob": "date",
  "gender": "string",
  "address": "string",
  "phoneNumber": "string",
  "emergencyContact": "string",
  "emergencyContactPhone": "string"
}
```

PatientResponse

```
{
  "id": "long",
  "firstName": "string",
  "lastName": "string",
  "dob": "date",
  "gender": "string",
  "address": "string",
  "phoneNumber": "string",
  "emergencyContact": "string",
  "emergencyContactPhone": "string",
  "createdAt": "datetime"
}
```

PatientSearch

```
{
  "id": "long",
  "firstName": "string",
  "lastName": "string",
  "dob": "date",
  "gender": "string",
  "phoneNumber": "string"
}
```

PatientSummary

```
{
  "id": "long",
  "firstName": "string",
  "lastName": "string",
  "phoneNumber": "string"
}
```

Error Responses

All endpoints may return these error responses: - **400 Bad Request**: Invalid request parameters or body - **403 Forbidden**: Insufficient privileges or unauthorized access - **404 Not Found**: Requested resource not found - **500 Internal Server Error**: Server error

Prescription API Documentation

Base URL

/api/prescriptions

Create Prescription

- **URL:** /api/prescriptions
- **Method:** POST
- **Description:** Creates a new prescription.
- **Request Body:**

```
{
  "medicalRecordId": "Long",
  "medicationId": "Long",
  "doctorId": "Long",
  "instructions": "String",
  "issuedDate": "ISO 8601 DateTime"
}
```

- **Response:** 201 Created

```
{
  "id": "Long",
  "medicalRecordId": "Long",
  "medicationId": "Long",
  "doctorId": "Long",
  "instructions": "String",
  "issuedDate": "ISO 8601 DateTime",
  "createdAt": "ISO 8601 DateTime"
}
```

- **Error Responses:**
 - 500 Internal Server Error: An error occurred while creating the prescription.
-

Delete Prescription

- **URL:** /api/prescriptions/{id}
- **Method:** DELETE

- **Description:** Deletes a prescription by ID.
 - **Path Parameter:**
 - id: ID of the prescription to delete.
 - **Response:** 200 OK
 - Success message: “Prescription deleted successfully”
 - **Error Responses:**
 - 404 Not Found: Prescription not found.
 - 500 Internal Server Error: An error occurred while deleting the prescription.
-

Get All Prescriptions

- **URL:** /api/prescriptions
- **Method:** GET
- **Description:** Retrieves all prescriptions.
- **Response:** 200 OK

```
[
  {
    "id": "Long",
    "medicalRecordId": "Long",
    "medicationId": "Long",
    "doctorId": "Long",
    "instructions": "String",
    "issuedDate": "ISO 8601 DateTime",
    "createdAt": "ISO 8601 DateTime"
  }
]
```

- **Error Responses:**
 - 500 Internal Server Error: An error occurred while fetching prescriptions.
-

Get Prescriptions by Medical Record

- **URL:** /api/prescriptions/medicalRecord/{medicalRecordId}
- **Method:** GET
- **Description:** Retrieves prescriptions associated with a specific medical record.
- **Path Parameter:**
 - medicalRecordId: ID of the medical record.
- **Response:** 200 OK

- **Error Responses:**
 - 404 Not Found: Medical record not found.
 - 500 Internal Server Error: An error occurred while fetching prescriptions by medical record.
-

Get Prescriptions by Doctor

- **URL:** /api/prescriptions/doctor/{doctorId}
 - **Method:** GET
 - **Description:** Retrieves prescriptions created by a specific doctor.
 - **Path Parameter:**
 - doctorId: ID of the doctor.
 - **Response:** 200 OK
 - **Error Responses:**
 - 404 Not Found: Doctor not found.
 - 500 Internal Server Error: An error occurred while fetching prescriptions by doctor.
-

Get Prescriptions by Medication

- **URL:** /api/prescriptions/medication/{medicationId}
 - **Method:** GET
 - **Description:** Retrieves prescriptions for a specific medication.
 - **Path Parameter:**
 - medicationId: ID of the medication.
 - **Response:** 200 OK
 - **Error Responses:**
 - 404 Not Found: Medication not found.
 - 500 Internal Server Error: An error occurred while fetching prescriptions by medication.
-

Get Active Prescriptions

- **URL:** /api/prescriptions/active
 - **Method:** GET
 - **Description:** Retrieves all active prescriptions.
 - **Response:** 200 OK
 - **Error Responses:**
 - 500 Internal Server Error: An error occurred while fetching active prescriptions.
-

Get Prescriptions by Date Range

- **URL:** /api/prescriptions/dateRange
 - **Method:** GET
 - **Description:** Retrieves prescriptions issued within a specified date range.
 - **Query Parameters:**
 - **startDate:** Start date in ISO 8601 DateTime format.
 - **endDate:** End date in ISO 8601 DateTime format.
 - **Response:** 200 OK
 - **Error Responses:**
 - **500 Internal Server Error:** An error occurred while fetching prescriptions by date range.
-

Get Prescriptions by Patient ID

- **URL:** /api/prescriptions/patient/{patientId}
 - **Method:** GET
 - **Description:** Retrieves prescriptions associated with a specific patient.
 - **Path Parameter:**
 - **patientId:** ID of the patient.
 - **Response:** 200 OK
 - **Error Responses:**
 - **404 Not Found:** Patient not found.
 - **500 Internal Server Error:** An error occurred while fetching prescriptions by patient.
-

Progress Report API Documentation

Base URL

/api/progress-reports

Create Progress Report

- **URL:** /api/progress-reports
- **Method:** POST
- **Description:** Creates a new progress report.
- **Request Body:**

```
{
  "patientId": "Long",
  "caregiverId": "Long",
  "date": "ISO 8601 DateTime",
  "summary": "String",
  "recommendations": "String"
}
```

- **Response:** 200 OK

```
{
  "id": "Long",
  "patientId": "Long",
  "caregiverId": "Long",
  "date": "ISO 8601 DateTime",
  "summary": "String",
  "recommendations": "String",
  "createdAt": "ISO 8601 DateTime"
}
```

Delete Progress Report

- **URL:** /api/progress-reports/{id}
- **Method:** DELETE
- **Description:** Deletes a progress report by ID.
- **Path Parameter:**
 - id (Long): ID of the progress report.
- **Response:** 204 No Content

Get All Progress Reports

- **URL:** /api/progress-reports
- **Method:** GET
- **Description:** Retrieves all progress reports.
- **Response:** 200 OK

```
[
  {
    "id": "Long",
    "patientId": "Long",
    "caregiverId": "Long",
    "date": "ISO 8601 DateTime",
    "summary": "String",
    "recommendations": "String",
    "createdAt": "ISO 8601 DateTime"
  }
]
```

Get Reports by Patient

- **URL:** /api/progress-reports/patient/{patientId}
 - **Method:** GET
 - **Description:** Retrieves all progress reports for a specific patient.
 - **Path Parameter:**
 - patientId (Long): ID of the patient.
 - **Response:** 200 OK
-

Get Reports by Caregiver

- **URL:** /api/progress-reports/caregiver/{caregiverId}
 - **Method:** GET
 - **Description:** Retrieves all progress reports for a specific caregiver.
 - **Path Parameter:**
 - caregiverId (Long): ID of the caregiver.
 - **Response:** 200 OK
-

Get Reports by Date Range

- **URL:** /api/progress-reports/range
 - **Method:** GET
 - **Description:** Retrieves progress reports within a specified date range.
 - **Query Parameters:**
 - **start** (ISO 8601 DateTime): Start of the date range.
 - **end** (ISO 8601 DateTime): End of the date range.
 - **Response:** 200 OK
-

Search Reports by Keyword

- **URL:** /api/progress-reports/search
 - **Method:** GET
 - **Description:** Searches progress reports by a keyword in the summary or recommendations.
 - **Query Parameter:**
 - **keyword** (String): Keyword to search in the report content.
 - **Response:** 200 OK
-

Note: Dates should be provided in ISO 8601 format (YYYY-MM-DDTHH:MM:SS).

User Management API Documentation

Base URL: /api/users

Authentication Endpoints

Register User

POST /register

Creates a new user account.

Request Body:

```
{
  "username": "string",
  "firstName": "string",
  "secondName": "string",
  "password": "string",
  "email": "string",
  "primaryLocation": "string",
  "role": "string",
  "privileges": "string"
}
```

Response: 201 Created

```
{
  "id": "long",
  "username": "string",
  "firstName": "string",
  "secondName": "string",
  "email": "string",
  "primaryLocation": "string",
  "secondaryLocation": "string",
  "phoneNumber": "string",
  "role": "string",
  "privileges": "string",
  "createdAt": "datetime"
}
```

Login

POST /login

Authenticates a user.

Request Body:

```
{
  "username": "string",
}
```



```
    "password": "string"
  }
```

Responses: - 200 OK: “Login successful” - 401 Unauthorized: Error message

User Management Endpoints

Update User

PUT /{id}

Updates an existing user’s information.

Request Body:

```
{
  "email": "string",
  "primaryLocation": "string",
  "secondaryLocation": "string",
  "phoneNumber": "string",
  "role": "string",
  "privileges": "string"
}
```

Response: 200 OK

```
{
  "id": "long",
  "username": "string",
  // ... (same as registration response)
}
```

Delete User

DELETE /delete

Deletes a user account.

Request Body:

```
{
  "username": "string",
  "password": "string"
}
```

Responses: - 204 No Content: User deleted successfully - 404 Not Found: Error message

Validate User Access

GET /{id}/validate

Validates user access privileges.

Parameters: - `id` (path): User ID - `required_access` (query): Required access level

Responses: - 200 OK: "Access granted" - 403 Forbidden: "Access denied" - 404 Not Found: "User not found" - 500 Internal Server Error: "An error occurred"

Search and Filter Endpoints

Find by Username

GET `/username/{username}`

Response: 200 OK

```
{
  "id": "long",
  "username": "string",
  // ... (same as registration response)
}
```

Find by Role

GET `/role/{role}`

Response: 200 OK Array of user objects

Find by Privileges

GET `/privileges/{privileges}`

Response: 200 OK Array of user objects

Find by Email

GET `/email/{email}`

Response: 200 OK Array of user objects

Find by Primary Location

GET `/primary-location/{primaryLocation}`

Response: 200 OK Array of user objects

Find by Secondary Location

GET `/secondary-location/{secondaryLocation}`

Response: 200 OK Array of user objects

Find by Role and Privileges

GET /role/{role}/privileges/{privileges}

Response: 200 OK Array of user objects

Search Users

GET /search/{searchTerm}

Performs a general search across user data.

Response: 200 OK Array of user objects

Error Responses

All endpoints may return the following error responses:

- 400 Bad Request: Invalid request parameters or body
- 401 Unauthorized: Authentication required or failed
- 403 Forbidden: Insufficient permissions
- 404 Not Found: Resource not found
- 500 Internal Server Error: Server-side error

Data Models

UserRegistrationRequest

```
{
  "username": "string",
  "firstName": "string",
  "secondName": "string",
  "password": "string",
  "email": "string",
  "primaryLocation": "string",
  "role": "string",
  "privileges": "string"
}
```

UserUpdateRequest

```
{
  "email": "string",
  "primaryLocation": "string",
  "secondaryLocation": "string",
  "phoneNumber": "string",
  "role": "string",
  "privileges": "string"
}
```

UserResponse

```
{
  "id": "long",
  "username": "string",
  "firstName": "string",
  "secondName": "string",
  "email": "string",
  "primaryLocation": "string",
  "secondaryLocation": "string",
  "phoneNumber": "string",
  "role": "string",
  "privileges": "string",
  "createdAt": "datetime"
}
```

LoginRequest

```
{
  "username": "string",
  "password": "string"
}
```