# Todo App API (1.0.0)

Download OpenAPI specification: Download

Todo App API Support: support@todoapp.com

A comprehensive todo management API with lists, todos, and user management

# Authentication

## Register a new user

Creates a new user account with the provided information

AUTHORIZATIONS: › *BearerAuth*

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `firstName`<br>required | string<br>User's first name |
| `lastName`<br>required | string<br>User's last name |
| `email`<br>required | string <email><br>User's email address |
| `password`<br>required | string<br>User's password (minimum 8 characters) |

### Responses

› **201** User registered successfully

› **400** Validation error or user already exists

> 500 Internal server error

POST /api/auth/register

### Request samples

Payload

Content type
application/json

Copy

```
{
    "firstName": "John",
    "lastName": "Doe",
    "email": "john.doe@example.com",
    "password": "SecurePassword123!"
}
```

### Response samples

201 400 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "user": { … },
        "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
    },
    "message": "Operation completed successfully"
}
```

## Login user

Authenticates a user with email and password

AUTHORIZATIONS: › *BearerAuth*

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `email`<br>`required` | string <email><br>User's email address |
| `password`<br>`required` | string<br>User's password |

## Responses

› **200** User logged in successfully

› **400** Validation error

› **401** Invalid credentials

› **500** Internal server error

POST `/api/auth/login`

### Request samples

**Payload**

**Content type**
application/json

Copy

```
{
    "email": "john.doe@example.com",
    "password": "SecurePassword123!"
}
```

### Response samples

**200** **400** **401** **500**

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "user": { … },
        "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
    },
    "message": "Operation completed successfully"
}
```

## Logout user

Logs out the current user and invalidates their session

AUTHORIZATIONS: › *BearerAuth*

### Responses

› **200** User logged out successfully

› **401** Unauthorized - invalid or missing token

› **500** Internal server error

POST `/api/auth/logout`

### Response samples

200 401 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
    "data": { },
    "message": "Operation completed successfully"
}
```

## Get current user

Retrieves the profile information of the currently authenticated user

AUTHORIZATIONS: > *BearerAuth*

### Responses

> **200** Current user information retrieved successfully

> **401** Unauthorized - invalid or missing token

> **500** Internal server error

GET /api/auth/me

**Response samples**

200 401 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "user": { … }
    },
    "message": "Operation completed successfully"
}
```

## Update user profile

Updates the profile information of the currently authenticated user

AUTHORIZATIONS: *BearerAuth*

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `firstName` | string<br>User's first name |
| `lastName` | string<br>User's last name |
| `password` | string<br>New password (minimum 8 characters) |

### Responses

- **200** User profile updated successfully
- **400** Validation error
- **401** Unauthorized - invalid or missing token
- **500** Internal server error

PATCH `/api/auth/me`

### Request samples

Payload

Content type
application/json

Copy

```
{
    "firstName": "John",
    "lastName": "Doe",
    "password": "NewSecurePassword123!"
}
```

### Response samples

200 400 401 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "user": { … }
    },
    "message": "Operation completed successfully"
}
```

# Todo Lists

## Create a new todo list

Creates a new todo list for the current user

AUTHORIZATIONS: > *BearerAuth*

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `name` required | string<br>Name of the todo list |
| `description` | string<br>Description of the todo list |
| `tags` | Array of strings |

Tags to associate with the list

## Responses

- › **201** Todo list created successfully
- › **400** Validation error
- › **401** Unauthorized - invalid or missing token
- › **500** Internal server error

POST `/api/todo-lists`

### Request samples

**Payload**

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "name": "Work Tasks",
    "description": "Tasks related to work projects",
  - "tags": [
        "work",
        "urgent"
    ]
}
```

### Response samples

**201** **400** **401** **500**

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todoList": { … }
    },
    "message": "Operation completed successfully"
}
```

## Get all todo lists for the current user

Retrieves all todo lists that the user has access to with optional filtering and sorting

AUTHORIZATIONS: *BearerAuth*

QUERY PARAMETERS

| | |
|---|---|
| `search` | string<br>Example: `search=work`<br>Search in todo list name and description |
| `sortBy` | string<br>Enum: `"createdAt"` `"updatedAt"` `"name"`<br>Example: `sortBy=createdAt`<br>Sort field |
| `sortOrder` | string<br>Enum: `"asc"` `"desc"`<br>Example: `sortOrder=desc`<br>Sort order |
| `page` | integer `>= 1`<br>Example: `page=1`<br>Page number for pagination |
| `limit` | integer `[ 1 .. 100 ]`<br>Example: `limit=10`<br>Number of items per page |

### Responses

› **200** Todo lists retrieved successfully

› **401** Unauthorized - invalid or missing token

> **500** Internal server error

GET `/api/todo-lists`

**Response samples**

**200** **401** **500**

**Content type**
application/json

Copy Expand all Collapse all

```
{
   "status": "success",
 - "data": {
     + "todoLists": [ … ],
     + "pagination": { … }
   },
   "message": "Operation completed successfully"
}
```

# Get a specific todo list by ID

Retrieves a specific todo list by its ID

AUTHORIZATIONS: > *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| id<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

## Responses

> **200** Todo list retrieved successfully

> **401** Unauthorized - invalid or missing token

> **403** Forbidden - no access to this todo list

> **404** Todo list not found

> **500** Internal server error

GET `/api/todo-lists/{id}`

## Response samples

**200** **401** **403** **404** **500**

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todoList": { … }
    },
    "message": "Operation completed successfully"
}
```

# Update a todo list

Updates an existing todo list

AUTHORIZATIONS: > *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `id`<br>`required` | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| name | string<br>Name of the todo list |
| description | string<br>Description of the todo list |
| isArchived | boolean<br>Whether to archive the list |
| tags | Array of strings<br>Tags to associate with the list |

## Responses

> **200** Todo list updated successfully

> **400** Validation error

> **401** Unauthorized - invalid or missing token

> **403** Forbidden - no access to this todo list

> **404** Todo list not found

> **500** Internal server error

PATCH /api/todo-lists/{id}

### Request samples

Payload

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "name": "Work Tasks",
    "description": "Tasks related to work projects",
    "isArchived": false,
  - "tags": [
        "work",
        "urgent"
    ]
}
```

### Response samples

200 400 401 403 404 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todoList": { … }
    },
    "message": "Operation completed successfully"
}
```

## Delete a todo list

Deletes an existing todo list

AUTHORIZATIONS: *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| id<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

### Responses

- **200** Todo list deleted successfully
- **401** Unauthorized - invalid or missing token
- **403** Forbidden - no access to this todo list

> **404** Todo list not found

> **500** Internal server error

DELETE /api/todo-lists/{id}

### Response samples

200 | 401 | 403 | 404 | 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
    "data": { },
    "message": "Operation completed successfully"
}
```

## Add a member to a todo list

Adds a new member to the specified todo list

AUTHORIZATIONS: > *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `id`<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `userId`<br>required | string<br>User ID to add as a member |

`role`
required

string

Enum: `"admin"` `"member"` `"viewer"`

Role to assign to the new member

## Responses

- › **200** Member added successfully
- › **400** Validation error or member already exists
- › **401** Unauthorized - invalid or missing token
- › **403** Forbidden - no access to this todo list or insufficient permissions
- › **404** Todo list or user not found
- › **500** Internal server error

POST `/api/todo-lists/{id}/members`

### Request samples

**Payload**

**Content type**
application/json

Copy

```
{
    "userId": "6479a2e8b123456789abcdef",
    "role": "member"
}
```

### Response samples

**200** **400** **401** **403** **404** **500**

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todoList": { … }
    },
    "message": "Operation completed successfully"
}
```

## Update a member's role

Updates the role of an existing member in the todo list

AUTHORIZATIONS: *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| id<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| userId<br>required | string<br>User ID of the member to update |
| role<br>required | string<br>Enum: `"admin"` `"member"` `"viewer"`<br>New role to assign to the member |

### Responses

> **200** Member role updated successfully

> **400** Validation error

> **401** Unauthorized - invalid or missing token

> **403** Forbidden - no access to this todo list or insufficient permissions

> **404** Todo list or member not found

> 500 Internal server error

PATCH /api/todo-lists/{id}/members

## Request samples

Payload

Content type
application/json

Copy

```
{
    "userId": "6479a2e8b123456789abcdef",
    "role": "admin"
}
```

## Response samples

200 400 401 403 404 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todoList": { … }
    },
    "message": "Operation completed successfully"
}
```

# Remove a member from a todo list

Removes an existing member from the specified todo list

AUTHORIZATIONS: > *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `id`<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |
| `memberId`<br>required | string<br>Example: `6479a2e8b987654321fedcba`<br>Member user ID |

## Responses

› **200** Member removed successfully

› **401** Unauthorized - invalid or missing token

› **403** Forbidden - no access to this todo list or insufficient permissions

› **404** Todo list or member not found

› **500** Internal server error

DELETE `/api/todo-lists/{id}/members/{memberId}`

### Response samples

200 401 403 404 500

**Content type**

application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todoList": { … }
    },
    "message": "Operation completed successfully"
}
```

## Leave a todo list

Allows the current user to leave a todo list they are a member of

AUTHORIZATIONS: › *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| id<br>`required` | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

### Responses

› **200** Successfully left the todo list

› **400** Cannot leave - user is the owner or last member

› **401** Unauthorized - invalid or missing token

› **403** Forbidden - not a member of this todo list

› **404** Todo list not found

› **500** Internal server error

POST `/api/todo-lists/{id}/leave`

### Response samples

200 | 400 | 401 | 403 | 404 | 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
    "data": { },
    "message": "Operation completed successfully"
```

```
  }
```

## Get all todos for a specific list

Retrieves all todos belonging to a specific list with optional filtering and sorting

AUTHORIZATIONS: *BearerAuth*

PATH PARAMETERS

`listId` required

string

Example: `6479a2e8b123456789abcdef`

Todo list ID

QUERY PARAMETERS

`status`

string

Enum: `"not_started"` `"in_progress"` `"completed"`

Example: `status=in_progress`

Filter by todo status

`priority`

string

Enum: `"low"` `"medium"` `"high"` `"urgent"`

Example: `priority=high`

Filter by todo priority

`tags`

string

Example: `tags=work,urgent`

Filter by tags (comma-separated)

`search`

string

Example: `search=project`

Search in todo name and description

`sortBy`

string

Enum: `"createdAt"` `"updatedAt"` `"dueDate"` `"priority"` `"name"`

Example: `sortBy=dueDate`

Sort field

`sortOrder`

string

Enum: `"asc"` `"desc"`

Example: `sortOrder=asc`

Sort order

`page`

integer >= 1

Example: `page=1`

Page number for pagination

`limit`

integer [ 1 .. 100 ]
Example: `limit=10`
Number of items per page

## Responses

> 200 Todos retrieved successfully

> 401 Unauthorized - invalid or missing token

> 403 Forbidden - no access to the specified list

> 404 Todo list not found

> 500 Internal server error

GET `/api/todo-lists/{listId}/todos`

### Response samples

200 401 403 404 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todos": [ … ],
      + "pagination": { … }
    },
    "message": "Operation completed successfully"
}
```

## Get todo statistics for a specific list

Retrieves aggregated statistics about todos in a specific list

AUTHORIZATIONS: › *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `listId`<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

## Responses

› **200** Todo statistics retrieved successfully

› **401** Unauthorized - invalid or missing token

› **403** Forbidden - no access to the specified list

› **404** Todo list not found

› **500** Internal server error

GET `/api/todo-lists/{listId}/todos/stats`

### Response samples

200 401 403 404 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "stats": { … }
    },
    "message": "Operation completed successfully"
}
```

# Todos

## Create a new todo

Creates a new todo item in a specified list

AUTHORIZATIONS: › *BearerAuth*

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `name`<br>required | string<br>Name/title of the todo |
| `description` | string<br>Detailed description of the todo |
| `listId`<br>required | string<br>ID of the todo list this belongs to |
| `status` | string<br>Enum: `"not_started"` `"in_progress"` `"completed"`<br>Initial status of the todo |
| `priority` | string<br>Enum: `"low"` `"medium"` `"high"` `"urgent"`<br>Priority level of the todo |
| `dueDate` | string <date-time><br>Due date for the todo |
| `tags` | Array of strings<br>Tags to associate with the todo |

### Responses

› **201** Todo created successfully

› **400** Validation error

> 401 Unauthorized - invalid or missing token

> 403 Forbidden - no access to the specified list

> 500 Internal server error

POST `/api/todos`

## Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

```
{
    "name": "Complete project documentation",
    "description": "Write comprehensive documentation for the API",
    "listId": "6479a2e8b123456789abcdef",
    "status": "not_started",
    "priority": "medium",
    "dueDate": "2023-06-15T10:00:00.000Z",
  - "tags": [
        "documentation",
        "urgent"
    ]
}
```

## Response samples

201 400 401 403 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todo": { … }
    },
    "message": "Operation completed successfully"
  }
```

## Get all todos for the current user

Retrieves all todos belonging to the current user with optional filtering and sorting

AUTHORIZATIONS: › *BearerAuth*

QUERY PARAMETERS

`status`

string
Enum: `"not_started"` `"in_progress"` `"completed"`
Example: `status=in_progress`
Filter by todo status

`priority`

string
Enum: `"low"` `"medium"` `"high"` `"urgent"`
Example: `priority=high`
Filter by todo priority

`tags`

string
Example: `tags=work,urgent`
Filter by tags (comma-separated)

`search`

string
Example: `search=project`
Search in todo name and description

`sortBy`

string
Enum: `"createdAt"` `"updatedAt"` `"dueDate"` `"priority"` `"name"`
Example: `sortBy=dueDate`
Sort field

`sortOrder`

string
Enum: `"asc"` `"desc"`
Example: `sortOrder=asc`
Sort order

`page`

integer >= 1
Example: `page=1`
Page number for pagination

`limit`

integer [ 1 .. 100 ]
Example: `limit=10`
Number of items per page

### Responses

> 200 Todos retrieved successfully

> 401 Unauthorized - invalid or missing token

> 500 Internal server error

GET /api/todos

### Response samples

200 401 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todos": [ … ],
      + "pagination": { … }
    },
    "message": "Operation completed successfully"
}
```

## Get todo statistics for the current user

Retrieves aggregated statistics about the user's todos

AUTHORIZATIONS: > *BearerAuth*

### Responses

> **200** Todo statistics retrieved successfully

> **401** Unauthorized - invalid or missing token

> **500** Internal server error

GET /api/todos/stats

### Response samples

200 401 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "stats": { … }
    },
    "message": "Operation completed successfully"
}
```

## Get all todos for a specific list

Retrieves all todos belonging to a specific list with optional filtering and sorting

AUTHORIZATIONS: > *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `listId`<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

QUERY PARAMETERS

| | |
|---|---|
| `status` | string<br>Enum: `"not_started"` `"in_progress"` `"completed"` |

Example: `status=in_progress`
Filter by todo status

`priority`

string
Enum: `"low"` `"medium"` `"high"` `"urgent"`
Example: `priority=high`
Filter by todo priority

`tags`

string
Example: `tags=work,urgent`
Filter by tags (comma-separated)

`search`

string
Example: `search=project`
Search in todo name and description

`sortBy`

string
Enum: `"createdAt"` `"updatedAt"` `"dueDate"` `"priority"` `"name"`
Example: `sortBy=dueDate`
Sort field

`sortOrder`

string
Enum: `"asc"` `"desc"`
Example: `sortOrder=asc`
Sort order

`page`

integer >= 1
Example: `page=1`
Page number for pagination

`limit`

integer [ 1 .. 100 ]
Example: `limit=10`
Number of items per page

## Responses

> **200** Todos retrieved successfully

> **401** Unauthorized - invalid or missing token

> **403** Forbidden - no access to the specified list

> **404** Todo list not found

> **500** Internal server error

GET /api/todos/list/{listId}

### Response samples

200 401 403 404 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todos": [ … ],
      + "pagination": { … }
    },
    "message": "Operation completed successfully"
}
```

## Get todo statistics for a specific list

Retrieves aggregated statistics about todos in a specific list

AUTHORIZATIONS: *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `listId`<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo list ID |

### Responses

- 200 Todo statistics retrieved successfully
- 401 Unauthorized - invalid or missing token
- 403 Forbidden - no access to the specified list

> 404 Todo list not found

> 500 Internal server error

GET /api/todos/list/{listId}/stats

### Response samples

200 | 401 | 403 | 404 | 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "stats": { … }
    },
    "message": "Operation completed successfully"
}
```

## Get a specific todo by ID

Retrieves a specific todo item by its ID

AUTHORIZATIONS: > *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| id<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo ID |

### Responses

> 200 Todo retrieved successfully

> **401** Unauthorized - invalid or missing token

> **403** Forbidden - no access to this todo

> **404** Todo not found

> **500** Internal server error

GET /api/todos/{id}

**Response samples**

**200** **401** **403** **404** **500**

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todo": { … }
    },
    "message": "Operation completed successfully"
}
```

# Update a todo

Updates an existing todo item

AUTHORIZATIONS: > *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| id<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo ID |

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `name` | string<br>Name/title of the todo |
| `description` | string<br>Detailed description of the todo |
| `status` | string<br>Enum: `"not_started"` `"in_progress"` `"completed"`<br>Status of the todo |
| `priority` | string<br>Enum: `"low"` `"medium"` `"high"` `"urgent"`<br>Priority level of the todo |
| `dueDate` | string <date-time><br>Due date for the todo |
| `tags` | Array of strings<br>Tags to associate with the todo |

## Responses

- **200** Todo updated successfully
- **400** Validation error
- **401** Unauthorized - invalid or missing token
- **403** Forbidden - no access to this todo
- **404** Todo not found
- **500** Internal server error

PATCH `/api/todos/{id}`

## Request samples

**Payload**

**Content type**

application/json

Copy Expand all Collapse all

```
{
    "name": "Complete project documentation",
    "description": "Write comprehensive documentation for the API",
    "status": "in_progress",
    "priority": "high",
    "dueDate": "2023-06-15T10:00:00.000Z",
  - "tags": [
        "documentation",
        "urgent"
    ]
}
```

### Response samples

200 400 401 403 404 500

**Content type**

application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "todo": { … }
    },
    "message": "Operation completed successfully"
}
```

## Delete a todo

Deletes an existing todo item

AUTHORIZATIONS: *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| id<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>Todo ID |

### Responses

> **200** Todo deleted successfully

> **401** Unauthorized - invalid or missing token

> **403** Forbidden - no access to this todo

> **404** Todo not found

> **500** Internal server error

DELETE /api/todos/{id}

### Response samples

**200** **401** **403** **404** **500**

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
    "data": { },
    "message": "Operation completed successfully"
}
```

# Users

## Get all users

Retrieves a list of all users in the system (admin only)

AUTHORIZATIONS: > *BearerAuth*

### Responses

› **200** Users retrieved successfully

› **401** Unauthorized - invalid or missing token

› **403** Forbidden - admin access required

› **500** Internal server error

GET `/api/users`

### Response samples

**200** **401** **403** **500**

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "users": [ … ]
    },
    "message": "Operation completed successfully"
}
```

## Get user by ID

Retrieves a specific user by their ID (admin only)

AUTHORIZATIONS: › *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `id`<br>`required` | string<br>Example: `6479a2e8b123456789abcdef` |

User ID

## Responses

- › **200** User retrieved successfully
- › **401** Unauthorized - invalid or missing token
- › **403** Forbidden - admin access required
- › **404** User not found
- › **500** Internal server error

GET /api/users/{id}

### Response samples

200 | 401 | 403 | 404 | 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "user": { … }
    },
    "message": "Operation completed successfully"
}
```

## Update user by ID

Updates a specific user's information (admin only)

AUTHORIZATIONS: › *BearerAuth*

PATH PARAMETERS

| | |
|---|---|
| `id`<br>required | string<br>Example: `6479a2e8b123456789abcdef`<br>User ID |

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `firstName` | string<br>User's first name |
| `lastName` | string<br>User's last name |
| `email` | string <email><br>User's email address |
| `role` | string<br>Enum: `"user"` `"admin"`<br>User's role in the system |
| `isActive` | boolean<br>Whether the user account is active |

## Responses

> **200** User updated successfully

> **400** Validation error

> **401** Unauthorized - invalid or missing token

> **403** Forbidden - admin access required

> **404** User not found

> **500** Internal server error

PATCH `/api/users/{id}`

### Request samples

Payload

Content type
application/json

Copy

```
{
    "firstName": "John",
    "lastName": "Doe",
    "email": "john.doe@example.com",
    "role": "user",
    "isActive": true
}
```

### Response samples

200 400 401 403 404 500

Content type
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "user": { … }
    },
    "message": "Operation completed successfully"
}
```

## Update current user

Updates the profile information of the currently authenticated user

AUTHORIZATIONS: > *BearerAuth*

REQUEST BODY SCHEMA: application/json
required

| | |
|---|---|
| `firstName` | string<br>User's first name |

| | |
|---|---|
| `lastName` | string<br>User's last name |
| `password` | string<br>New password (minimum 8 characters) |

## Responses

- **200** User profile updated successfully
- **400** Validation error
- **401** Unauthorized - invalid or missing token
- **500** Internal server error

PATCH `/api/users/me`

### Request samples

Payload

**Content type**
application/json

Copy

```
{
    "firstName": "John",
    "lastName": "Doe",
    "password": "NewSecurePassword123!"
}
```

### Response samples

200 400 401 500

**Content type**
application/json

Copy Expand all Collapse all

```
{
    "status": "success",
  - "data": {
      + "user": { … }
    },
    "message": "Operation completed successfully"
}
```