

# API Documentation

This document describes the endpoints exposed by the Challenge backend. The API is divided into two logical services **User** and **Todo** and uses JSON for requests and responses.

## Authentication

Authenticated endpoints require a Bearer token in the Authorization header. Use the **Login user** endpoint to obtain a JSON Web Token (JWT). Include Authorization: Bearer token in requests to protected resources.

## User Service

### Health check (User)

**GET** http://localhost:4000/

Returns a simple status message indicating that the User service is online.

#### Response

Status	Body
200	{ "status": "success", "data": { "message": "User Service is running" } }

### Register user

**POST** /user/register

Creates a new user account. Add the new user's email and password in JSON format. If the email already exists, the service responds with a 409 Conflict.

#### Request body

```
{  
  "email": "test4@test.com",  
  "password": "123456"  
}
```

#### Responses

Status	Body
201	{ "status": "success", "data": { "id": "3b3e9ba3-d11d-4c13-b632-429bb2c6f040", "email": "test@yahoo.com" } }

409	{ "status": "error", "message": "Email already in use" }
-----	---

## Login user

**POST** /user/login

Authenticates a user with their email and password. A successful login returns a JWT and the user profile. Incorrect credentials produce a 401 error.

**Request body**

```
{  
  "email": "test@yahoo.com",  
  "password": "123456"  
}
```

**Responses**

Status	Body
200	{ "status": "success", "data": { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...." } }
401	{ "status": "error", "message": "Invalid credentials" }

## Get user by ID

**GET** /user/{id}

Retrieves a user profile. If the user does not exist, the service returns a 404 error.

**Path parameter**

Name	Description
id	Unique identifier of user

**Responses**

Status	Body
200	{ "status": "success", "data": { "id": "3b3e9ba3-d11d-4c13-b632-429bb2c6f040", "email": "test@yahoo.com", "createdAt": "2025-08-11T19:41:28.888Z" } }

404	{ "status": "error", "message": "User not found" }
-----	---

# Todo Service

## Health check (Todo)

**GET** http://localhost:4001

Returns a status message indicating that the Todo service is operational.

**Response**

Status	Body
200	{ "status": "success", "data": { "message": "Todo Service is running" } }

## Create todo

**POST** /api/todo/  
Requires authentication.

Creates a new todo item for the authenticated user. Provide a title and an optional description in JSON format.

**Request body**

```
{  
  "content": "Build App"  
}
```

**Responses**

Status	Body
201	{ "status": "success", "data": { "content": "Build App", "userId": "3b3e9ba3-d11d-4c13-b632-429bb2c6f040", "id": "072ace85-9f8d-469a-9d9a-4047c7d65ce2", "createdAt": "2025-08-11T19:56:59.075Z" } }
401	{ "status": "error", "message": "Missing or invalid Authorization header" }

## List todos

**GET** /api/todo/

Requires authentication.

Returns a list of todo items for the authenticated user.

### Responses

Status	Body
200	<pre>{   "status": "success",   "data": [     {       "id": "072ace85-9f8d-469a-9d9a-4047c7d65ce2",       "content": "Build App",       "userId": "3b3e9ba3-d11d-4c13-b632-429bb2c6f040",       "createdAt": "2025-08-11T19:56:59.075Z"     }   ] }</pre>
401	<pre>{   "status": "error",   "message": "Missing or invalid Authorization header" }</pre>

## Update todo

**PUT** /api/todo/{todoId}

Requires authentication.

Updates the fields of an existing todo. Only the provided fields are modified. If the todo does not exist, a 404 error is returned.

### Path parameter

Name	Description
todoId	Unique identifier of todo

### Request body

```
{
  "content": "Build Mobile App"
}
```

## Responses

Status	Body
200	<pre>{   "status": "success",   "data": {     "id": "072ace85-9f8d-469a-9d9a-4047c7d65ce2",     "content": "Build Mobile App",     "userId": "3b3e9ba3-d11d-4c13-b632-429bb2c6f040",     "createdAt": "2025-08-11T19:56:59.075Z"   } }</pre>
404	<pre>{   "status": "error",   "message": "Not found" }</pre>
401	<pre>{   "status": "error",   "message": "Missing or invalid Authorization header" }</pre>

## Delete todo

**DELETE** /api/todo/{todoId}

Requires authentication.

Deletes a todo item by its identifier[1]. If the todo does not exist, the service returns a 404 error.

### Path parameter

Name	Description
todoId	Unique identifier of todo

## Responses

Status	Body
204	<i>(no content)</i>
404	<pre>{   "status": "error",   "message": "Not found" }</pre>
401	<pre>{   "status": "error",   "message": "Missing or invalid Authorization header" }</pre>