# API Technical Documentation

Contents

[API Technical Documentation 1](#_Toc185545255)

[Overview 1](#_Toc185545256)

[Base URL 1](#_Toc185545257)

[Authentication 1](#_Toc185545258)

[1. AuthController - POST /auth/token 1](#_Toc185545259)

[2. CustomerController - POST /customer/check 2](#_Toc185545260)

[Example Usage 3](#_Toc185545261)

[Error Handling 4](#_Toc185545262)

[Contact 4](#_Toc185545263)

## Overview

This document describes the structure and functionality of the API, including the endpoints, their purposes, and the expected request and response formats. The API consists of two main controllers: `AuthController` and `CustomerController`.

## Base URL

All endpoints are relative to the following base URL:  
  
```  
https://api.example.com  
```

## Authentication

The API uses a token-based authentication mechanism. Clients must obtain an access token via the `AuthController` and include it in the `Authorization` header of all subsequent requests. The format is:  
  
```  
Authorization: Bearer <access\_token>  
```

## 1. AuthController - POST /auth/token

This endpoint is used to authenticate a user and retrieve an access token.  
  
\*\*Request:\*\*  
  
- \*\*Headers:\*\* None  
- \*\*Body:\*\* JSON object containing the following parameters:  
```json  
{  
 "username": "string",  
 "password": "string"  
}  
```  
  
\*\*Response:\*\*  
  
- \*\*Headers:\*\*  
 - `Content-Type: application/json`  
- \*\*Body:\*\* JSON object containing the access token:  
```json  
{  
 "access\_token": "string"  
}  
```  
  
\*\*Response Codes:\*\*  
  
- `200 OK`: Token successfully generated.  
- `400 Bad Request`: Invalid username or password.  
- `500 Internal Server Error`: An unexpected error occurred.

## 2. CustomerController - POST /customer/check

This endpoint validates the customer information provided in the request.  
  
\*\*Request:\*\*  
  
- \*\*Headers:\*\*  
 - `Authorization: Bearer <access\_token>`  
- \*\*Body:\*\* JSON object containing the following parameters:  
```json  
{  
 "customerNo": "int",  
 "companyName": "string",  
 "cellPhone": "string",  
 "internetId": "string",  
 "iban": "string",  
 "taxNumber": "string"  
}  
```  
  
\*\*Response:\*\*  
  
- \*\*Headers:\*\*  
 - `Content-Type: application/json`  
- \*\*Body:\*\* JSON object containing the validation result:  
```json  
{  
 "flag": "boolean"  
}  
```  
  
\*\*Response Codes:\*\*  
  
- `200 OK`: Customer validation completed.  
- `400 Bad Request`: Invalid or missing parameters.  
- `401 Unauthorized`: Missing or invalid token.  
- `500 Internal Server Error`: An unexpected error occurred.

## Example Usage

### 1. Obtain Token  
  
\*\*Request:\*\*  
  
```  
POST /auth/token  
Content-Type: application/json  
  
{  
 "username": "user1",  
 "password": "password123"  
}  
```  
  
\*\*Response:\*\*  
  
```  
200 OK  
Content-Type: application/json  
  
{  
 "access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}  
```  
  
### 2. Validate Customer  
  
\*\*Request:\*\*  
  
```  
POST /customer/check  
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...  
Content-Type: application/json  
  
{  
 "customerNo": 12345,  
 "companyName": "Acme Corp",  
 "cellPhone": "555-1234",  
 "internetId": "abc123",  
 "iban": "DE89370400440532013000",  
 "taxNumber": "123456789"  
}  
```  
  
\*\*Response:\*\*  
  
```  
200 OK  
Content-Type: application/json  
  
{  
 "flag": true  
}  
```

## Error Handling

All errors will return a JSON object with the following structure:  
  
```json  
{  
 "error": "string",  
 "message": "string"  
}  
```  
  
Example:  
  
```  
400 Bad Request  
Content-Type: application/json  
  
{  
 "error": "InvalidParameter",  
 "message": "The parameter 'customerNo' is required."  
}  
```

## Contact

For support, please contact the API team at `support@example.com`.