

📌 API Documentation (Base Route)

BASE_URL=https://smart-cv.worldws.pro/api

🔑 Authentication APIs

1. Login User

POST \${BASE_URL}/users/login

Request Body:

```
{  
  "email": "client@demo.com",  
  "password": "demo",  
  "loginSource": "web"  
}
```

Success Response (200):

```
{  
  "message": "login successfully...",  
  "user": {  
    "id": "user_123",  
    "username": "johndoe",  
    "email": "john.doe@example.com",  
    "roleId": 1,  
    "profile": {  
      "firstName": "John",  
      "lastName": "Doe"  
    },  
    "role": {  
      "id": 1,  
      "name": "admin"  
    },  
    "createdAt": "2024-01-05T12:00:00Z",  
    "updatedAt": "2024-01-05T12:00:00Z"  
  }  
}
```

Error Response (401):

```
{  
  "error": "Invalid credentials"  
}
```

2. Refresh Token

POST \${BASE_URL}/users/refresh-token

(Use this when the access token is expired, but refresh token is still valid)

Request Body:

(No body required, just send the refresh token in headers/cookies depending on implementation)

Success Response (200):

```
{  
  "message": "Token refreshed successfully",  
  "user": {  
    "id": "user_123",  
    "username": "johndoe",  
    "email": "john.doe@example.com",  
    "roleId": 1,  
    "profile": {  
      "firstName": "John",  
      "lastName": "Doe"  
    },  
    "createdAt": "2024-01-05T12:00:00Z",  
    "updatedAt": "2024-01-05T12:00:00Z"  
  }  
}
```

3. Logout User

POST \${BASE_URL}/users/logout *(Requires Authorization: Bearer {token})*

Request Body:

(No body required)

Success Response (200):

```
{  
  "message": "Logged out successfully"  
}
```

👤 User Management APIs

4. Create User

POST \${BASE_URL}/users/create-user (Requires **Authorization: Bearer {token}**)

Request Body:

```
{  
  "username": "johndoe",  
  "authPassword": "MySecretPassword123!",  
  "email": "john.doe@example.com",  
  "roleId": 1,  
  "profile": {  
    "firstName": "John",  
    "lastName": "Doe",  
    "phone": "+1234567890"  
  }  
}
```

Success Response (201):

```
{  
  "message": "User created successfully",  
  "user": {  
    "id": "user_123",  
    "username": "johndoe",  
    "email": "john.doe@example.com",  
    "roleId": 1,  
    "profile": {  
      "firstName": "John",  
      "lastName": "Doe"  
    },  
    "createdAt": "2024-01-05T12:00:00Z",  
    "updatedAt": "2024-01-05T12:00:00Z"  
  }  
}
```

✉️ Request Password Reset

POST \${BASE_URL}/users/request-password-reset

Request Body:

```
{  
  "email": "john.doe@example.com"  
}
```

Success Response (200):

```
{  
  "message": "Password reset email sent"  
}
```

🔒 Reset Password

POST \${BASE_URL}/users/reset-password

Request Body:

```
{  
  "token": "reset-token-from-email",  
  "newPassword": "MyNewPassword123!"  
}
```

Success Response (200):

```
{  
  "message": "Password reset successful"  
}
```

🔑 Change Password

POST \${BASE_URL}/users/change-password

Request Body (example `changePasswordSchema`):

```
{  
  "oldPassword": "MyOldPassword123!",  
  "newPassword": "MyNewPassword123!"  
}
```

Success Response (200):

```
{  
  "message": "Password changed successfully"  
}
```