Stellar Auth Service

API Documentation Reference

Last Updated: 11/02/2026

Stellar Auth Service API DocumentationÐ

Welcome to the Stellar Auth Service API. This service handles user authentication, session management, and account recovery for the Stellar platform.Ð

Base URLÐ

`http://localhost:3000` (Development)Ð

`https://auth.susindran.in` (Production)Ð

Authentication EndpointsÐ

1. Sign UpÐ

Create a new user account.Ð

- **URL**: `/api/auth/signup`
- **Method**: `POST`
- **Access**: Public
- **Request Body**:

```jsonÐ
{Ð
  "email": "user@example.com",Ð
  "password": "StrongPassword123!",Ð
  "name": "John Doe"Ð
}Ð
```Ð

- **Success Response (201 Created)**:

```jsonÐ
{Ð
  "success": true,Ð
  "message": "Signup successful. Please check your email to verify your account.",Ð
  "data": {Ð
    "user": { "id": "uuid", "email": "user@example.com", "name": "John Doe", "verified": false },Ð
```

"token": "jwt_access_token",Đ

"refresh_token": "jwt_refresh_token"Đ

}Đ

}Đ

```Đ

## 2. Sign InĐ

Authenticate an existing user.Đ

- **URL**: `/api/auth/signin`
- **Method**: `POST`
- **Access**: Public
- **Request Body**:

```jsonĐ

{Đ

"email": "user@example.com",Đ

"password": "StrongPassword123!"Đ

}Đ

```Đ

- **Success Response (200 OK)**:

```jsonĐ

{Đ

"success": true,Đ

"message": "Sign in successful",Đ

"data": {Đ

"user": { "id": "uuid", "email": "user@example.com", "name": "John Doe", "verified": true },Đ

"token": "jwt_access_token",Đ

"jwt_refresh_token": "jwt_refresh_token"Đ

}Đ

}Đ

```Đ

## 3. Sign OutĐ

Invalidate the current user session (Handled client-side by purging tokens).Đ

- **URL**: `/api/auth/signout`
- **Method**: `POST`
- **Access**: Private (Requires Bearer Token)
- **Headers**: `Authorization: Bearer <token>`
- **Success Response (200 OK)**:

```jsonĐ

{ "success": true, "message": "Sign out successful" }Đ

```Đ

---

## Session & Token ManagementĐ

### 4. Get Current SessionĐ

Retrieve user details using an access token.Đ

- **URL**: `/api/auth/session`
- **Method**: `POST`
- **Access**: Private (Requires Bearer Token)
- **Headers**: `Authorization: Bearer <token>`
- **Request Body**:

```jsonĐ

{ "access_token": "jwt_access_token" }Đ

```Đ

- **Success Response (200 OK)**:

```jsonĐ

{Đ

  "success": true,Đ

  "data": {Đ

    "user": { "id": "uuid", "email": "user@example.com", "name": "John Doe", "verified": true }Đ

  }Đ

}Đ

```Đ

### 5. Refresh JWT TokenĐ

Obtain a new access token using a refresh token.Đ

- **URL**: `/api/auth/refresh-jwt`
- **Method**: `POST`
- **Access**: Public
- **Request Body**:

```jsonĐ

{ "jwt_refresh_token": "your_refresh_token" }Đ

```Đ

- **Success Response (200 OK)**:

```jsonĐ

{Đ

  "success": true,Đ

"message": "JWT token refreshed successfully",Đ

"data": {Đ

"token": "new_access_token",Đ

"refresh_token": "new_refresh_token"Đ

}Đ

}Đ

```Đ

---

## Account Recovery & VerificationĐ

### 6. Verify EmailĐ

Verify a user account using the token sent to their email.Đ

- **URL**: `/api/auth/verify-email`
- **Method**: `POST`
- **Access**: Public
- **Request Body**:

```jsonĐ

{Đ

"email": "user@example.com",Đ

"verificationToken": "email_token_string"Đ

}Đ

```Đ

### 7. Reset Password RequestĐ

Send a password reset link to the user's email.Đ

- **URL**: `/api/auth/reset-password`
- **Method**: `POST`
- **Access**: Public
- **Request Body**:

```jsonĐ

{ "email": "user@example.com" }Đ

```Đ

### 8. Confirm Password ResetĐ

Set a new password using a reset token.Đ

- **URL**: `/api/auth/reset-password/confirm`
- **Method**: `POST`
- **Access**: Public
```

• **Request Body**:
```jsonĐ
{Đ
  "email": "user@example.com",Đ
  "resetToken": "reset_token_string",Đ
  "newPassword": "NewStrongPassword123!"Đ
}Đ
```Đ

---

## OAuth EndpointsĐ

### 9. Google Sign-In (Initiate)Đ

Generate the Google OAuth URL.Đ

• **URL**: `/api/auth/google`
• **Method**: `POST`
• **Access**: Public
• **Success Response (200 OK)**:
```jsonĐ
{Đ
  "success": true,Đ
  "data": { "url": "https://accounts.google.com/...", "state": "unique_state" }Đ
}Đ
```Đ

---

## Error CodesĐ

| Status Code | Description |Đ
| :--- | :--- |Đ
| `400` | Bad Request (Missing fields, invalid format) |Đ
| `401` | Unauthorized (Invalid token, wrong credentials) |Đ
| `403` | Forbidden (Insufficient permissions) |Đ
| `429` | Too Many Requests (Rate limit hit - currently disabled in production) |Đ
| `500` | Internal Server Error (Database or SMTP failure) |Đ