**🧩 1. User Registration & Profile**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| RegisterView | POST | /accounts/register/ | Register a new user. Sends email verification. |
| MeView | GET | /accounts/me/ | Retrieve authenticated user info. |

**Postman Example:**

**POST** http://127.0.0.1:8000/accounts/register/

{

"email": "user@example.com",

"password": "StrongPass123!"

}

**GET** http://127.0.0.1:8000/accounts/me/  
Headers: Authorization: Bearer <access\_token>

**📧 2. Email Verification**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| VerifyEmailView | POST | /accounts/email/verify/ | Verify email after signup (via token). |
| ResendVerificationEmailView | POST | /accounts/resend-verification/ | Resend verification email. |

**POST** /accounts/email/verify/

{

"uid": "1",

"token": "abcd1234token"

}

**🔑 3. Authentication (Login, Logout, JWT)**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| LoginView | POST | /accounts/login/ | Login via email/username/phone + password. Returns JWT tokens. |
| LogoutView | POST | /accounts/logout/ | Logout user (invalidate session). |

**POST** /accounts/login/

{

"identifier": "user@example.com",

"password": "StrongPass123!",

"remember\_me": true

}

✅ Response includes:

{

"user": {"email": "user@example.com"},

"refresh": "jwt-refresh-token",

"access": "jwt-access-token"

}

**🔒 4. Password Management**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| ForgotPasswordView | POST | /accounts/password/forgot/ | Request password reset link (email). |
| ResetPasswordView | POST | /accounts/password/reset/ | Set new password using token. |
| ChangePasswordView | POST | /accounts/password/change/ | Authenticated user changes password. |

**POST** /accounts/password/forgot/

{ "email": "user@example.com" }

**POST** /accounts/password/reset/

{

"uid": "MQ",

"token": "xyztokencode",

"new\_password": "NewPass456!"

}

**POST** /accounts/password/change/  
Headers: Authorization: Bearer <token>

{

"current\_password": "OldPass123!",

"new\_password": "NewPass456!"

}

**✉️ 5. Magic Link Login (Passwordless)**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| MagicLinkRequestView | POST | /accounts/magic/request/ | Send magic link email. |
| MagicLinkConsumeView | POST | /accounts/magic/consume/ | Consume magic link and return JWT. |

**POST** /accounts/magic/request/

{ "email": "user@example.com" }

**POST** /accounts/magic/consume/

{ "token": "generated\_token\_value" }

**🔢 6. OTP (Email, SMS, Backup)**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| EmailOTPRequestView | POST | /accounts/otp/email/request/ | Request OTP via email. |
| EmailOTPVerifyView | POST | /accounts/otp/email/verify/ | Verify email OTP. |
| SMSOTPRequestView | POST | /accounts/otp/sms/request/ | Request OTP via SMS. |
| SMSOTPVerifyView | POST | /accounts/otp/sms/verify/ | Verify SMS OTP. |
| BackupCodesGenerateView | POST | /accounts/mfa/backup-codes/generate/ | Generate backup codes. |
| BackupCodeVerifyView | POST | /accounts/mfa/backup-codes/verify/ | Verify backup code. |

**POST** /accounts/otp/email/request/

{

"email": "user@example.com",

"digits": 6

}

**POST** /accounts/otp/email/verify/

{

"email": "user@example.com",

"code": "452916"

}

**🧭 7. Multi-Factor Authentication (MFA)**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| TOTPSetupBeginView | POST | /accounts/mfa/totp/setup-begin/ | Begin TOTP setup, return QR/secret. |
| TOTPConfirmView | POST | /accounts/mfa/totp/confirm/ | Confirm TOTP code. |
| MFAVerifyView | POST | /accounts/mfa/verify/ | Verify MFA code after login. |

**POST** /accounts/mfa/totp/setup-begin/  
Headers: Authorization: Bearer <token>  
*Response:*

{

"otpauth\_uri": "otpauth://totp/AuthTemplate:user@example.com?secret=ABC123..."

}

**POST** /accounts/mfa/totp/confirm/

{ "code": "123456" }

**🛡️ 8. Account Management (Admin)**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| AccountLockView | POST | /accounts/<user\_id>/lock/ | Admin locks user. |
| AccountUnlockView | POST | /accounts/<user\_id>/unlock/ | Admin unlocks user. |
| AccountDisableView | POST | /accounts/<user\_id>/disable/ | Admin disables user. |
| AccountEnableView | POST | /accounts/<user\_id>/enable/ | Admin enables user. |
| AccountSoftDeleteView | POST | /accounts/<user\_id>/soft-delete/ | Soft delete user (recoverable). |
| AccountRestoreView | POST | /accounts/<user\_id>/restore/ | Restore soft-deleted user. |
| AccountHardDeleteView | DELETE | /accounts/<user\_id>/hard-delete/ | Permanently delete user. |
| AccountActionView | POST | /accounts/<user\_id>/<action>/ | Combined admin endpoint (lock, unlock, delete, restore). |

**POST** /accounts/1/lock/  
Headers: Authorization: Bearer <admin\_token>

**🙋‍♀️ 9. User-Initiated Account Actions**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| SuspendAccountView | POST | /accounts/suspend/ | User suspends their own account. |
| RequestDeleteAccountView | POST | /accounts/request-delete/ | User requests deletion. |
| CancelDeleteAccountView | POST | /accounts/cancel-delete/ | Cancel pending delete. |
| HardDeleteAccountView | DELETE | /accounts/delete/ | Permanently delete own account. |
| ReAuthView | POST | /accounts/api/auth/reauth/ | Re-authenticate user before sensitive action. |

**POST** /accounts/suspend/  
Headers: Authorization: Bearer <token>

*Response:*

{ "detail": "Your account has been suspended." }

**🧠 10. WebAuthn / Passkeys**

| **View** | **Method** | **URL Pattern** | **Purpose** |
| --- | --- | --- | --- |
| WebAuthnRegisterBeginView | POST | /accounts/webauthn/register/begin/ | Begin WebAuthn registration. |
| WebAuthnRegisterCompleteView | POST | /accounts/webauthn/register/complete/ | Complete WebAuthn registration. |
| WebAuthnAuthBeginView | POST | /accounts/webauthn/auth/begin/ | Begin WebAuthn authentication. |
| WebAuthnAuthCompleteView | POST | /accounts/webauthn/auth/complete/ | Complete WebAuthn authentication. |

**POST** /accounts/webauthn/auth/begin/

{ "email": "user@example.com" }

*Response:*

{ "challenge": "random\_token\_123" }