

Onboarding

Overview

The `/v1/auth/onboard` endpoint handles user and organization onboarding for both individual and business users. This endpoint creates new users, organizations, and wallets in the Vudy ecosystem.

Endpoint Details

- **Method:** `POST`
- **Path:** `/v1/auth/onboard`
- **Content-Type:** `application/json`

Request Schema

Required Fields

Field	Type	Validation	Description
<code>email</code>	string	Valid email, no sub-addresses(+) lowercase	User's email address
<code>firstName</code>	string	Min 2 characters, trimmed	User's first name
<code>lastName</code>	string	Min 2 Characters, trimmed	User's last name
<code>username</code>	string	Min 4 characters, trimmed, no special characters except <code>-</code>	Unique Username
<code>country</code>	string	Exactly 2 Characters	ISO country code
<code>isBusiness</code>	boolean	-	Define whether this is a business account
<code>termsOfService</code>	boolean	MUST BE TRUE	Accepting Terms of service

Optional Fields

Field	Type	Validation	Description
<code>language</code>	string	Must be an accepted language enum <code>en es</code>	Defaults to <code>en</code> - User's preferred language
<code>businessName</code>	string	Min 2 characters, trimmed	Required if <code>isBusiness</code> is true

Conditional Validation

- If `isBusiness` is `true`, then `businessName` is required and must be at least 2 characters long

Response Format

Success Response

```
{  
  "success": true,  
  "data": {  
    "nextStep": "OTP" | "WALLET_SETUP",  
    "message": "User created successfully"  
  },  
  "error": null  
}
```

Error Response

```
{  
  "success": false,  
  "data": null,  
  "error": {  
    "code": "VUDY#OB##",  
    "message": "Error description",  
    "details": "Additional error details (optional)"  
  }  
}
```

Error Codes and Debugging

VUDY#OB01 - Invalid Request Body

- **Message:** "Invalid or missing request body"
- **Status:** 400
- **Debug Steps:**
 - a. Check if request body is properly formatted JSON
 - b. Verify Content-Type header is set to `application/json`
 - c. Ensure request body is not empty

VUDY#OB02 - Validation Failed

- **Message:** "Validation failed"
- **Status:** 400
- **Details:** Array of validation error details
- **Debug Steps:**
 - a. Check the `details` field for specific validation errors
 - b. Verify all required fields are present
 - c. Check field format requirements (email format, string lengths, etc.)
 - d. Ensure `termsOfService` is set to `true`

- e. If `isBusiness` true, verify `bu `businessName` provided

VUDY#OB03 - Username Already in Use

- **Message:** "Username is already in use"
- **Status:** 400
- **Debug Steps:**
 - a. Check database for existing username
 - b. Verify username uniqueness constraint
 - c. Check for case sensitivity issues (usernames are stored lowercase)

VUDY#OB04 - Turnkey Organization Creation Failed

- **Message:** "Failed to create turnkey organization"
- **Status:** 400
- **Debug Steps:**
 - a. Check Turnkey API connectivity
 - b. Verify Turnkey credentials and configuration
 - c. Check Turnkey API logs for specific error details
 - d. Verify sub-organization name format
 - e. Check if email is already associated with another Turnkey organization

VUDY#OB05 - Organization Creation Failed

- **Message:** "Failed to create organization"
- **Status:** 400
- **Debug Steps:**
 - a. Check database connectivity
 - b. Verify organization table schema
 - c. Check for database constraint violations
 - d. Review organization name uniqueness

VUDY#OB10 - Business name is required

- **Message:** "Business name is required"
- **Status:** 400
- **Debug Steps:**
 - a. Check that business name is being sent when `isBusiness` is true.

Business Logic Flow

New User Onboarding

1. **Validation:** Validate request body against schema
2. **Email Check:** Verify email is not already in use in Turnkey
3. **Username Check:** Verify username is available in database
4. **Turnkey Org:** Create sub-organization in Turnkey
5. **Database Org:** Create organization in local database
6. **User Creation:** Create user record with encrypted email
7. **User-Org Link:** Associate user with organization as ADMIN
8. **Country Check:** Determine next step based on country
 - **El Salvador (SV):** Return `WALLET_SETUP` next step
 - **Other Countries:** Create wallet and return `OTP` next step
9. **Wallet Creation:** Create EVM and SOLANA wallets in Turnkey
10. **Wallet Association:** Link wallets to user account

Existing User Onboarding → **USE OTHER API**

- **Status:** Not yet implemented (returns 418)

Country-Specific Behavior

El Salvador (SV)

- Users from El Salvador require additional compliance step
- Wallet creation is deferred to a separate process
- Next step: `WALLET_SETUP`

Other Countries

- Wallets are created immediately during onboarding
- Both EVM and SOLANA wallets are created
- Next step: `OTP` verification