PONA Auth Secondary Onboarding

Author: Josh S. Sakweli, Backend Lead Team
Last Updated: 2026-04-27
Version: v1.0

Base URL: `https://api.nexgate.co/api/v1/onboarding/secondary`

Short Description: Secondary onboarding completes a user's profile after the primary onboarding step (name, phone, birth date). It is a step-machine — each endpoint returns the next missing step and a refreshed access token. The flow is complete when `stepsRemaining` reaches `0` and `nextMissing` is `null`.

Hints:

- All endpoints require a valid `Bearer` access token issued after primary onboarding or login.
- Every response includes a fresh `accessToken` with updated onboarding flags embedded in the JWT claims — replace the stored token after each step.
- Steps can be completed in any order. The `nextMissing` field signals the next recommended step; it does not enforce ordering.
- Email linking has two independent paths: **custom** (user-provided email + OTP) and **Google** (Google ID token). Only one path needs to be completed.
- Profile picture upload expects `multipart/form-data`, not JSON.

Standard Response Format

All API responses follow a consistent structure using our Globe Response Builder pattern:

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2025-09-23T10:30:45",
```

```
  "action": "COLLECT_EMAIL",
  "data": {}
}
```

Error Response Structure

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Error description",
  "action_time": "2025-09-23T10:30:45",
  "data": "Error description"
}
```

Standard Response Fields

Field	Type	Description
`success`	boolean	`true` for successful operations, `false` for errors
`httpStatus`	string	HTTP status name (OK, BAD_REQUEST, etc.)
`message`	string	Human-readable result description
`action_time`	string	ISO 8601 timestamp of the response
`action`	string	Next frontend action to perform (see action codes below)
`data`	object/string	Response payload or error detail

Action Codes

Code	Meaning
`COLLECT_USERNAME`	Username step is next
`COLLECT_EMAIL`	Email step is next
`COLLECT_PROFILE_PIC`	Profile picture step is next
`COLLECT_INTERESTS`	Interests step is next
`COLLECT_BIO`	Bio step is next
`PROCEED`	All steps complete — onboarding is done

Standard Secondary Onboarding Response Fields

Most endpoints return a `SecondaryOnboardingResponse`. Its fields are:

Field	Type	Description
`accessToken`	string	Fresh JWT — store and use this for all subsequent requests
`onboarding`	object	Current completion state of all onboarding steps (see below)
`nextMissing`	string	Key name of the next incomplete step, or `null` if all done
`stepsRemaining`	integer	Number of steps still pending

`onboarding` Object Fields

Field	Type	Description
`primaryComplete`	boolean	Primary onboarding (name/phone/DOB) is done
`username`	boolean	Username has been set
`email`	boolean	Email has been linked and verified
`profilePic`	boolean	Profile picture has been uploaded
`interests`	boolean	Interests have been selected
`bio`	boolean	Bio has been written

Endpoints

1. Get Username Suggestions

Purpose: Returns up to 5 AI-generated username suggestions based on the user's first name, last name, and birth date.

Endpoint: `GET` `{base_url}/username/suggestions`

Access Level: □□ Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Username suggestions",
  "action_time": "2026-04-27T10:30:45",
  "data": {
    "suggestions": [
      "john_sakweli",
      "johnsakweli99",
      "j_sakweli",
      "johnsak2004",
      "jsakweli_"
    ]
  }
}
```

Success Response Fields:

Field	Description
`suggestions`	Array of up to 5 available username strings

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Account not found",
  "action_time": "2026-04-27T10:30:45",
  "data": "Account not found"
}
```

2. Set Username

Purpose: Sets a unique username for the account.

Endpoint: POST `{base_url}/username`

Access Level: Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`
`Content-Type`	string	Yes	`application/json`

Request JSON Sample:

```
{
  "username": "john_sakweli"
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
`username`	string	Yes	Desired username	Min: 3, Max: 30 characters. Must start with a letter. Only letters, numbers, and underscores allowed.

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Username set successfully",
  "action_time": "2026-04-27T10:30:45",
  "action": "COLLECT_EMAIL",
  "data": {
    "accessToken": "eyJhbGciOiJSUzI1NiJ9...",
```

```
    "onboarding": {
      "primaryComplete": true,
      "username": true,
      "email": false,
      "profilePic": false,
      "interests": false,
      "bio": false
    },
    "nextMissing": "email",
    "stepsRemaining": 4
  }
}
```

Success Response Fields:

Field	Description
`accessToken`	Fresh JWT with updated onboarding claims — replace stored token
`onboarding.username`	Now `true`
`nextMissing`	Next recommended step key
`stepsRemaining`	Steps left to complete

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Username is already taken",
  "action_time": "2026-04-27T10:30:45",
  "data": "Username is already taken"
}
```

Standard Error Types:

- `400 BAD_REQUEST`: Username already taken or invalid format
- `401 UNAUTHORIZED`: Missing or invalid token
- `422 UNPROCESSABLE_ENTITY`: Validation failure (length, pattern)

---

3. Set Bio

Purpose: Saves a short bio to the user's profile.

Endpoint: `POST` `{base_url}/bio`

Access Level: □□ Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`
`Content-Type`	string	Yes	`application/json`

Request JSON Sample:

```
{
  "bio": "Event enthusiast, live music lover, always at the front row."
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
`bio`	string	Yes	User's short profile bio	Max: 160 characters. Cannot be blank.

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Bio saved",
  "action_time": "2026-04-27T10:30:45",
  "action": "PROCEED",
  "data": {
    "accessToken": "eyJhbGciOiJSUzI1NiJ9...",
    "onboarding": {
      "primaryComplete": true,
      "username": true,
      "email": true,
      "profilePic": true,
      "interests": true,
```

```
      "bio": true
    },
    "nextMissing": null,
    "stepsRemaining": 0
  }
}
```

Standard Error Types:

- `400 BAD_REQUEST`: Bio is blank
- `401 UNAUTHORIZED`: Missing or invalid token
- `422 UNPROCESSABLE_ENTITY`: Exceeds 160 characters

---

4. Set Interests

Purpose: Saves the user's selected interest categories (minimum 3 required).

Endpoint: **POST** `{base_url}/interests`

Access Level: □□ Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`
`Content-Type`	string	Yes	`application/json`

Request JSON Sample:

```
{
  "interestIds": [
    "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "7a1234bc-1234-4321-abcd-1234567890ab",
    "9c87654d-4321-1234-dcba-0987654321cd"
  ]
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
`interestIds`	array of UUID	Yes	IDs of selected interest categories	Min: 3 items. Must not be empty. Use the interests listing endpoint to get valid IDs.

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Interests saved",
  "action_time": "2026-04-27T10:30:45",
  "action": "COLLECT_BIO",
  "data": {
    "accessToken": "eyJhbGciOiJSUzI1NiJ9...",
    "onboarding": {
      "primaryComplete": true,
      "username": true,
      "email": true,
      "profilePic": true,
      "interests": true,
      "bio": false
    },
    "nextMissing": "bio",
    "stepsRemaining": 1
  }
}
```


Standard Error Types:

- `400 BAD_REQUEST`: Interest IDs not found
- `401 UNAUTHORIZED`: Missing or invalid token
- `422 UNPROCESSABLE_ENTITY`: Fewer than 3 interests selected

---

5. Upload Profile Picture

Purpose: Uploads and stores the user's profile picture.

Endpoint: **POST** `{base_url}/profile-pic`

Access Level: □□ Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`
`Content-Type`	string	Yes	`multipart/form-data`

Query Parameters:

Parameter	Type	Required	Description	Validation	Default
`file`	file (form field)	Yes	Image file to upload	Image formats: JPEG, PNG, WEBP	—

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Profile picture uploaded",
  "action_time": "2026-04-27T10:30:45",
  "action": "COLLECT_INTERESTS",
  "data": {
    "accessToken": "eyJhbGciOiJSUzI1NiJ9...",
    "onboarding": {
      "primaryComplete": true,
      "username": true,
      "email": true,
      "profilePic": true,
      "interests": false,
      "bio": false
    },
    "nextMissing": "interests",
    "stepsRemaining": 2
  }
}
```


Standard Error Types:

- `400 BAD_REQUEST`: File is missing, empty, or unsupported format

- `401 UNAUTHORIZED`: Missing or invalid token
- `500 INTERNAL_SERVER_ERROR`: Storage failure

---

6. Initiate Email Linking (Custom)

Purpose: Sends a 6-digit OTP to the provided email address and returns a `tempToken` required for the verify step.

Endpoint: **POST** `{base_url}/email/custom/initiate`

Access Level: □□ Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`
`Content-Type`	string	Yes	`application/json`

Request JSON Sample:

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

Request Body Parameters:

Parameter	Type	Required	Description	Validation
`email`	string	Yes	Email address to link	Must be a valid email format. Cannot be blank.

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Verification code sent to your email",
  "action_time": "2026-04-27T10:30:45",
  "data": {
```

```
    "tempToken": "eyJhbGciOiJSUzI1NiJ9.temp...",
    "nextAction": "VERIFY_EMAIL"
  }
}
```

Success Response Fields:

Field	Description
`tempToken`	Short-lived token required as input to the verify step. Store this until OTP is submitted.
`nextAction`	Always `VERIFY_EMAIL` — signals frontend to show OTP input screen

Standard Error Types:

- `400 BAD_REQUEST`: Email already in use by another account
- `401 UNAUTHORIZED`: Missing or invalid token
- `422 UNPROCESSABLE_ENTITY`: Invalid email format

---

7. Verify Email (Custom)

Purpose: Confirms the OTP sent to the user's email and marks the email step as complete.

Endpoint: **POST** `{base_url}/email/custom/verify`

Access Level: □□ Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`
`Content-Type`	string	Yes	`application/json`

Request JSON Sample:

```
{
  "tempToken": "eyJhbGciOiJSUzI1NiJ9.temp...",
  "otp": "847291"
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
`tempToken`	string	Yes	Token received from the initiate step	Cannot be blank
`otp`	string	Yes	6-digit code sent to the user's email	Exactly 6 digits, numeric only

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Email verified",
  "action_time": "2026-04-27T10:30:45",
  "action": "COLLECT_PROFILE_PIC",
  "data": {
    "accessToken": "eyJhbGciOiJSUzI1NiJ9...",
    "onboarding": {
      "primaryComplete": true,
      "username": true,
      "email": true,
      "profilePic": false,
      "interests": false,
      "bio": false
    },
    "nextMissing": "profilePic",
    "stepsRemaining": 3
  }
}
```

Standard Error Types:

- `400 BAD_REQUEST`: OTP is incorrect or expired
- `401 UNAUTHORIZED`: `tempToken` is invalid or expired
- `422 UNPROCESSABLE_ENTITY`: OTP is not 6 numeric digits

---

8. Link Email via Google

Purpose: Links and verifies a Google-backed email using a Google ID token — skips the OTP step entirely.

Endpoint: `POST` `{base_url}/email/google`

Access Level: □□ Protected (Authenticated user)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
`Authorization`	string	Yes	`Bearer <access_token>`
`Content-Type`	string	Yes	`application/json`

Request JSON Sample:

```
{
  "idToken": "eyJhbGciOiJSUzI1NiIsImtpZCI6..."
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
`idToken`	string	Yes	Google ID token from the Google Sign-In SDK	Cannot be blank. Must be a valid Google-issued token.

Success Response JSON Sample:

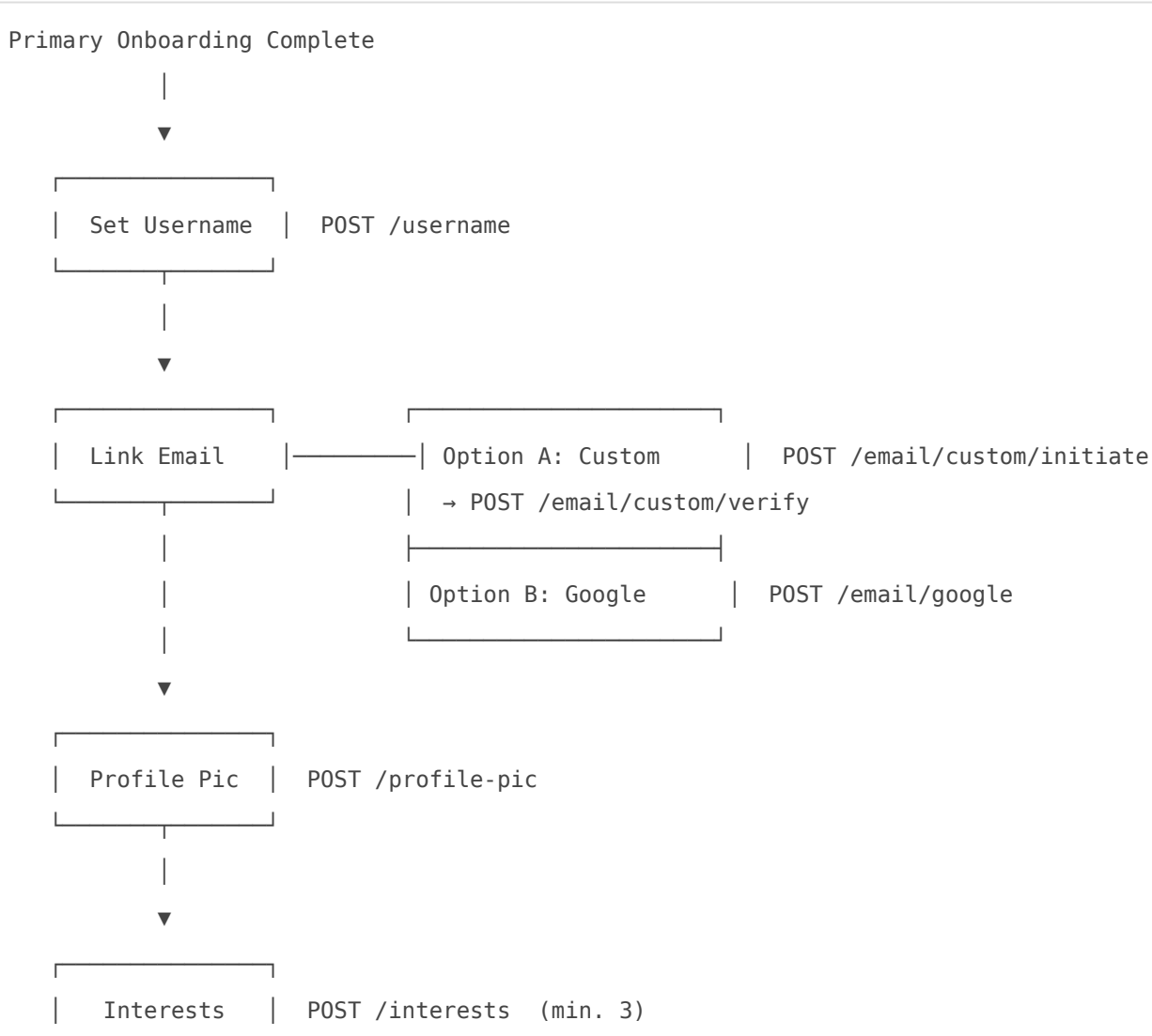
```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Email linked via Google",
  "action_time": "2026-04-27T10:30:45",
  "action": "COLLECT_PROFILE_PIC",
  "data": {
    "accessToken": "eyJhbGciOiJSUzI1NiJ9...",
    "onboarding": {
      "primaryComplete": true,
      "username": true,
      "email": true,
      "profilePic": false,
```

```
      "interests": false,
      "bio": false
    },
    "nextMissing": "profilePic",
    "stepsRemaining": 3
  }
}
```

Standard Error Types:

- `400 BAD_REQUEST`: Google token is invalid, expired, or email already linked to another account
- `401 UNAUTHORIZED`: Missing or invalid Bearer token
- `422 UNPROCESSABLE_ENTITY`: `idToken` is blank

Flow Overview



```
└──────────┘
      │
      ▼
┌──────────┐
│   Bio    │  POST /bio
└──────────┘
      │
      ▼
 action: PROCEED
(onboarding done)
```

> Steps can be completed out of order. The `nextMissing` field in each response always signals the next recommended incomplete step.

Quick Reference

#	Endpoint	Method	Auth	Purpose
1	`/username/suggestions`	GET	Bearer	Get suggested usernames
2	`/username`	POST	Bearer	Set username
3	`/bio`	POST	Bearer	Set bio
4	`/interests`	POST	Bearer	Set interests
5	`/profile-pic`	POST	Bearer	Upload profile picture
6	`/email/custom/initiate`	POST	Bearer	Send OTP to email
7	`/email/custom/verify`	POST	Bearer	Verify email with OTP
8	`/email/google`	POST	Bearer	Link email via Google token

Revision #2
Created 27 April 2026 15:11:12 by Admin Qbit
Updated 27 April 2026 15:31:07 by Admin Qbit