API Documentation

# Authentication & User

## POST /signup

* - \*\*Description:\*\* Register a new user.
* - \*\*Request Body:\*\*

{  
 "email": "string (email)",  
 "password": "string (min 6 chars)",  
 "name": "string"  
}

* - \*\*Response:\*\*

{  
 "id": "string (uuid)",  
 "email": "string",  
 "name": "string",  
 "createdAt": "string (ISO date)",  
 "updatedAt": "string (ISO date)"  
}

* - \*\*Status Codes:\*\* 201 Created

## POST /signin

* - \*\*Description:\*\* Login with credentials.
* - \*\*Request Body:\*\*

{  
 "email": "string (email)",  
 "password": "string (min 6 chars)"  
}

* - \*\*Response:\*\*

{  
 "access\_token": "string"  
}

* - \*\*Status Codes:\*\* 200 OK

## GET /me

* - \*\*Description:\*\* Get current user profile.
* - \*\*Auth:\*\* Bearer JWT
* - \*\*Response:\*\*

{  
 "id": "string (uuid)",  
 "email": "string",  
 "name": "string",  
 "createdAt": "string (ISO date)",  
 "updatedAt": "string (ISO date)"  
}

* - \*\*Status Codes:\*\* 200 OK

## DELETE /me

* - \*\*Description:\*\* Delete current user.
* - \*\*Auth:\*\* Bearer JWT
* - \*\*Response:\*\*

{  
 "success": true  
}

* - \*\*Status Codes:\*\* 201 Created, 401 Unauthorized if not found

# Questionnaire

## POST /questionnaire

* - \*\*Description:\*\* Create a new questionnaire.
* - \*\*Auth:\*\* Bearer JWT
* - \*\*Request Body:\*\*

{  
 "title": "string",  
 "description": "string"  
}

* - \*\*Response:\*\*

{  
 "id": "string (uuid)",  
 "title": "string",  
 "description": "string",  
 "status": "draft | queued | completed | failed",  
 "userId": "string (uuid)",  
 "questions": [ /\* Array of Question (see below) \*/ ],  
 "createdAt": "string (ISO date)",  
 "updatedAt": "string (ISO date)",  
 "prompt": "string (optional)"  
}

* - \*\*Status Codes:\*\* 201 Created

## PATCH /questionnaire/:id

* - \*\*Description:\*\* Update a questionnaire.
* - \*\*Auth:\*\* Bearer JWT
* - \*\*Request Body:\*\*

{  
 "title": "string (optional)",  
 "description": "string (optional)",  
 "status": "draft | queued | completed | failed (optional)"  
}

* - \*\*Response:\*\* Same as POST /questionnaire
* - \*\*Status Codes:\*\* 200 OK

## POST /questionnaire/:id/question

* - \*\*Description:\*\* Add a question to a questionnaire.
* - \*\*Auth:\*\* Bearer JWT
* - \*\*Request Body:\*\*

{  
 "description": "string (min 10 chars)",  
 "imageUrl": "string (optional)",  
 "videoUrl": "string (optional)",  
 "type": "text | number | date | select | checkbox | radio",  
 "options": ["string", "..."] // optional  
}

* - \*\*Response:\*\* Updated Questionnaire
* - \*\*Status Codes:\*\* 201 Created

## Question (object)

{  
 "id": "string (uuid)",  
 "description": "string",  
 "imageUrl": "string (optional)",  
 "videoUrl": "string (optional)",  
 "type": "text | number | date | select | checkbox | radio",  
 "options": ["string", "..."], // optional  
 "questionnaireId": "string (uuid)",  
 "createdBy": "User"  
}

## POST /questionnaire/UsingAI

* - \*\*Description:\*\* Create a questionnaire using AI.
* - \*\*Auth:\*\* Bearer JWT
* - \*\*Request Body:\*\*

{  
 "title": "string",  
 "description": "string",  
 "prompt": "string"  
}

* - \*\*Response:\*\* Questionnaire
* - \*\*Status Codes:\*\* 201 Created

# Survey

## POST /survey

* - \*\*Description:\*\* Create a new survey.
* - \*\*Auth:\*\* Bearer JWT
* - \*\*Request Body:\*\*

{  
 "questionnaireId": "string (uuid)",  
 "populationId": "string (uuid)",  
 "deliveryModes": ["email" | "whatsapp"],  
 "expiresAt": "string (ISO date)"  
}

* - \*\*Response:\*\*

{  
 "id": "string (uuid)",  
 "questionnaireId": "string (uuid)",  
 "populationId": "string (uuid)",  
 "userId": "string (uuid)",  
 "deliveryModes": ["email" | "whatsapp"],  
 "expiresAt": "string (ISO date)",  
 "analyzed": "boolean",  
 "sendJobStatus": "NotStarted | Queued | Processing | Completed | Failed",  
 "sendJobId": "string (optional)",  
 "sendProgress": "number",  
 "sendJobStartedAt": "string (ISO date, optional)",  
 "sendJobCompletedAt": "string (ISO date, optional)",  
 "createdAt": "string (ISO date)",  
 "updatedAt": "string (ISO date)"  
}

* - \*\*Status Codes:\*\* 201 Created

## POST /survey/submit

* - \*\*Description:\*\* Submit survey answers.
* - \*\*Request Body:\*\*

{  
 "jwt": "string",  
 "answers": [  
 {  
 "questionId": "string (uuid)",  
 "value": ["string", "..."]  
 }  
 ]  
}

* - \*\*Response:\*\* void
* - \*\*Status Codes:\*\* 201 Created

## Answer (object)

{  
 "id": "string (uuid)",  
 "surveyId": "string (uuid)",  
 "personId": "string (uuid)",  
 "questionId": "string (uuid)",  
 "value": ["string", "..."],  
 "createdAt": "string (ISO date)"  
}

# Population

## Population (object)

{  
 "id": "string (uuid)",  
 "name": "string",  
 "userId": "string (uuid)",  
 "parentPopulation": "Population (optional)",  
 "children": [ "Population", ... ],  
 "persons": [ "Person", ... ],  
 "status": "completed | queued | working | failed",  
 "jobId": "string (uuid, optional)",  
 "createdAt": "string (ISO date)",  
 "updatedAt": "string (ISO date)"  
}

## addPersonDto (POST /population/:populationId/addPerson)

{  
 "email": "string (email)",  
 "phone": "string (optional)",  
 "name": "string (optional)",  
 "customFields": { "key": "string | number | boolean | Date" }  
}

## UpdatePersonDto (PUT /population/:populationId/person/:personId)

{  
 "email": "string (email, optional)",  
 "phone": "string (optional)",  
 "name": "string (optional)",  
 "customFields": { "key": "string | number | boolean | Date" }  
}

## Person (object)

{  
 "id": "string (uuid)",  
 "email": "string",  
 "name": "string (optional)",  
 "phone": "string (optional)",  
 "customFields": { "key": "any" },  
 "populations": [ "Population", ... ],  
 "createdAt": "string (ISO date)",  
 "updatedAt": "string (ISO date)"  
}

# Status Codes Used

* - 200 OK: Successful GET, DELETE, or PATCH requests.
* - 201 Created: Successful resource creation.
* - 202 Accepted: Request accepted for processing (async jobs).
* - 400 Bad Request: Invalid input.
* - 401 Unauthorized: Invalid/missing authentication.
* - 404 Not Found: Resource/job not found.
* - 409 Conflict: Job is active and cannot be removed.

# Notes

* - All endpoints requiring authentication expect a Bearer JWT in the Authorization header.
* - For endpoints returning arrays, pagination may be supported via `page` and `limit` query parameters.
* - For file uploads, use `multipart/form-data` with a `file` field