



SHAPE API Quick Overview

Table of Contents

Architecture.....	1
Application Overview	3
Swagger Documentation	3
YAML Definition	3

Architecture

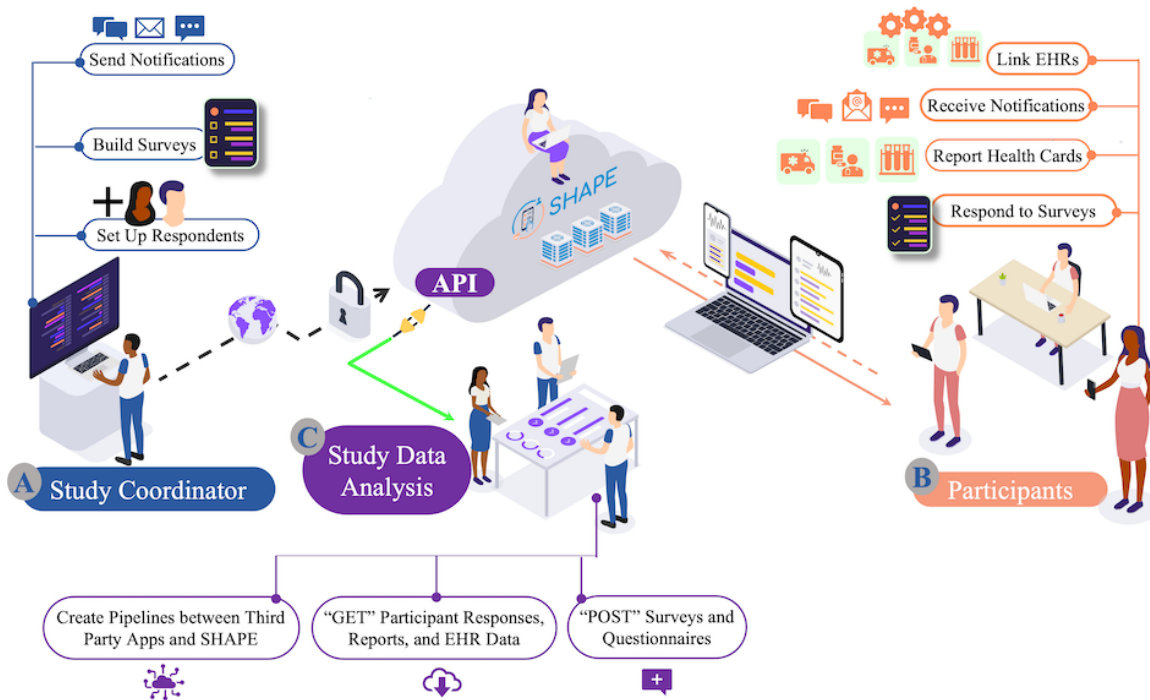
FDA’s Survey of Health and Patient Experience (SHAPE) Application is a mobile application that is available in the Apple App Store, Google Play Store, and on the web. The SHAPE Application has been designed and developed as a flexible and scalable application for the design and deployment of patient experience surveys. The SHAPE Administrator Application is where study administrators can configure and share surveys with SHAPE App users. User, survey, response, and metadata information is stored in Google’s Cloud Firestore database and Cloud Storage.

The SHAPE Platform consists of:

- SHAPE Mobile Application
 - o Framework: React, Ionic and Node.js
 - o Features include:
 - Surveys
 - Questionnaires
 - Self-Reports
 - Message Center
 - Profile
 - EHR Integration
- SHAPE Administrator Application
 - o Framework: React, Ionic and Node.js
 - o Features include:
 - Web-based configuration portal for survey administrators

- Survey Skip Logic
- Message Center for email, SMS and in-app notification communications
- Automated functionality to send notifications, as well as add users to follow-up questionnaires based on the completion of a previous questionnaire or responses to a question
- Response dashboards
- Study population curation center
- Question development portal
 - Single Line Text, Radio Button, Date Time, Checkbox, Dropdown, Slider, Text Area, Range, Information Only type questions
- SHAPE API
 - Framework: React, Express and Node.js
- Secure Storage Environment
 - Google Firebase
 - Google Authentication
 - Google Cloud Firestore
 - Google Storage
 - Google Functions
 - Google Hosting

Figure 1. SHAPE Platform

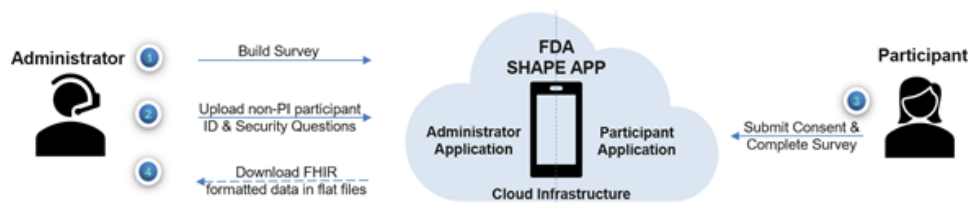


Application Overview

Figure 2 illustrates the overall survey development and deployment process. The SHAPE Platform is hosted in a cloud environment and consists of two separate applications: the SHAPE Admin application and the SHAPE App (the participant side mobile application).

The SHAPE API App is the portion of SHAPE that interfaces with the SHAPE Admin application to give the SHAPE Admin application access to the Firebase database. The API layer also allows authorized third parties to read, write, and update records in the SHAPE database without using the SHAPE Admin user interface.

Figure 2. Overall Administrator Process



This document provides the API documentation for SHAPE API endpoints. The base URL should be adjusted for the developer's firebase functions web api endpoint.

Swagger Documentation

SHAPE API's API definition can be found at the endpoint if a user is logged into their Firebase project account: <https://<firebase-function--base-url>.cloudfunctions.net/webApi/swagger>

YAML Definition

A copy of the YAML definition can be found at `/src/swagger/api-docs.yaml` and is also referenced below.

```
openapi: 3.0.3
info:
  title: SHAPE-API
  description: Swagger OpenAPI 3.0 endpoint defintions for SHAPE-API. Access to this API requires a valid Firebase bearer token.
  version: '1.0'
paths:
  /auth/:
```

```
post:
  tags:
    - authentication
  summary: authenticate using username and password
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Auth'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DatabaseError'

/ehr/:
  get:
    tags:
      - ehr
    summary: get all ehr records
```

```

responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
/ehr/{respondentId}/{participantName}/{dob}:
get:
  tags:
    - ehr
  summary: get ehr record by url path variables
  parameters:
    - name: respondentId
      in: path
      description: id of respondent
      required: true
      schema:
        type: string
    - name: participantName
      in: path
      description: name of individual associated with ehr record
      required: true
      schema:

```

```

    type: string
  - name: dob
    in: path
    description: date of birth of respondent
    required: true
    schema:
      type: string
      example: 'mmddyyyy'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DatabaseError'
  /ehr/get/:
    post:
      tags:
        - ehr
      summary: get ehr record by filepath
      requestBody:
        required: true
        content:
          application/json:

```

```
    schema:
      $ref: '#/components/schemas/Path'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DatabaseError'
```

```
/export/diary/:
  post:
    tags:
      - export
    summary: export diaries in json format
    requestBody:
      required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Query'
  responses:
    '200':
```

```

description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
/export/diary/fhir/:
post:
tags:
- export
summary: export diaries in fhir format
requestBody:
required: true
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Query'
responses:
'200':
description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':

```



```

description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
/export/questionnaire/:
post:
  tags:
    - export
  summary: export questionnaires in json format
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Query'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':

```

```

description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
/export/questionnaire/fhir/:
post:
tags:
  - export
summary: export questionnaires in fhir format
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Query'
responses:
'200':
description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':

```

description: Invalid parameters
content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
 application/json:
 schema:
 \$ref: '#/components/schemas/DatabaseError'

/image/:
post:
 tags:
 - image
 summary: create image
 requestBody:
 required: true
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Image'
 responses:
 '200':
 description: Success
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/SuccessResponse'
 '400':
 description: Insufficient parameters
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InsufficientParameters'
 '403':
 description: unauthorized
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Unauthorized'
 '422':
 description: Invalid parameters
 content:
 application/json:
 schema:

```

    $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
  get:
    tags:
      - image
    summary: get all images
    responses:
      '200':
        description: Success
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SuccessResponse'
      '400':
        description: Insufficient parameters
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InsufficientParameters'
      '403':
        description: unauthorized
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Unauthorized'
      '422':
        description: Invalid parameters
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InvalidParameters'
      '500':
        description: Database error
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/DatabaseError'
  /image/{documentId}:
    get:
      tags:
        - image
      summary: get image by Id
      parameters:

```

- name: documentId
in: path
description: id of document in database
required: true
schema:
 type: string
responses:
 '200':
 description: Success
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/SuccessResponse'
 '400':
 description: Insufficient parameters
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InsufficientParameters'
 '403':
 description: unauthorized
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Unauthorized'
 '422':
 description: Invalid parameters
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InvalidParameters'
 '500':
 description: Database error
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/DatabaseError'

/inbox/:
post:
tags:
 - inbox
summary: create inbox
requestBody:
 required: true
 content:
 application/json:

```

    schema:
      $ref: '#/components/schemas/Inbox'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DatabaseError'
  get:
    tags:
      - inbox
    summary: get all inboxes
    description: get all inboxes OR get inbox by participantId if query param is included
    parameters:
      - name: participantId
        in: query
        description: participantId associated with the inbox ***Optional
        schema:
          type: string
    responses:
      '200':
        description: Success
        content:

```

```

    application/json:
      schema:
        $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
/inbox/{documentId}:
  get:
    tags:
      - inbox
    summary: get inbox by Id
    parameters:
      - name: documentId
        in: path
        description: id of document in database
        required: true
        schema:
          type: string
    responses:
      '200':
        description: Success
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SuccessResponse'
      '400':
        description: Insufficient parameters

```

```
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
put:
  tags:
    - inbox
  summary: update inbox by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Inbox'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
```



```
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
delete:
  tags:
    - inbox
  summary: delete inbox by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
```

```
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
```

/informed-consent/:

```
post:
  tags:
    - informed consent
  summary: create informed consent
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InformedConsent'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
```

```

'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
get:
  tags:
    - informed consent
  summary: get all informed consents
  description: get all informed consents OR get informed consent(s) by surveyId if query param is
included
  parameters:
    - name: surveyId
      in: query
      description: surveyId associated with the informed consent(s) ***Optional
      schema:
        type: string
        example: survey1234
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:

```

```

    schema:
      $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
/informed-consent/{documentId}:
get:
  tags:
    - informed consent
  summary: get informed consent by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:

```

```
application/json:
  schema:
    $ref: '#/components/schemas/DatabaseError'
```

```
/message/:
```

```
  post:
```

```
    tags:
```

```
      - message
```

```
    summary: create message
```

```
    requestBody:
```

```
      required: true
```

```
      content:
```

```
        application/json:
```

```
          schema:
```

```
            $ref: '#/components/schemas/Message'
```

```
    responses:
```

```
      '200':
```

```
        description: Success
```

```
        content:
```

```
          application/json:
```

```
            schema:
```

```
              $ref: '#/components/schemas/SuccessResponse'
```

```
      '400':
```

```
        description: Insufficient parameters
```

```
        content:
```

```
          application/json:
```

```
            schema:
```

```
              $ref: '#/components/schemas/InsufficientParameters'
```

```
      '403':
```

```
        description: unauthorized
```

```
        content:
```

```
          application/json:
```

```
            schema:
```

```
              $ref: '#/components/schemas/Unauthorized'
```

```
      '422':
```

```
        description: Invalid parameters
```

```
        content:
```

```
          application/json:
```

```
            schema:
```

```
              $ref: '#/components/schemas/InvalidParameters'
```

```
      '500':
```

```
        description: Database error
```

```
        content:
```

```
          application/json:
```

```
            schema:
```

```
              $ref: '#/components/schemas/DatabaseError'
```

```
    get:
```

```

tags:
  - message
summary: get all messages
description: get all messages OR get message(s) by participantId if query param is included
parameters:
  - name: participantId
    in: query
    description: participantId associated with the message(s) ***Optional
    schema:
      type: string
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
/message/{documentId}:
get:
  tags:
    - message
  summary: get message by Id
  parameters:
    - name: documentId

```

in: path
description: id of document in database
required: true
schema:
 type: string
responses:
 '200':
 description: Success
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/SuccessResponse'
 '400':
 description: Insufficient parameters
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InsufficientParameters'
 '403':
 description: unauthorized
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Unauthorized'
 '422':
 description: Invalid parameters
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InvalidParameters'
 '500':
 description: Database error
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/DatabaseError'

/participant-diary/:

post:
 tags:
 - participant diary
 summary: create participant diary
 requestBody:
 required: true
 content:
 application/json:
 schema:

```

    $ref: '#/components/schemas/ParticipantDiary'
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
get:
  tags:
    - participant diary
  summary: get all participant diaries
  description: get all participant diaries OR get participant diaries by participantId if query param is
included
  parameters:
    - name: participantId
      in: query
      description: participantId associated with the participant diaries ***Optional
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:

```



```

    application/json:
      schema:
        $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
/participant-diary/{documentId}:
  get:
    tags:
      - participant diary
    summary: get participant diary by Id
    parameters:
      - name: documentId
        in: path
        description: id of document in database
        required: true
        schema:
          type: string
    responses:
      '200':
        description: Success
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SuccessResponse'
      '400':
        description: Insufficient parameters

```

```

content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
put:
  tags:
    - participant diary
  summary: update participant diary by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ParticipantDiary'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters

```

```

content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
delete:
tags:
  - participant diary
summary: delete participant diary by Id
parameters:
  - name: documentId
    in: path
    description: id of document in database
    required: true
    schema:
      type: string
responses:
'200':
description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized

```

```

content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
/participant-diary/query/:
post:
  tags:
    - participant diary
  summary: query for participant diaries
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Query'
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters

```

```
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
```

/participant-response/:

```
post:
  tags:
    - participant response
  summary: create participant response
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ParticipantResponse'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
```

```

'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
get:
  tags:
    - participant response
  summary: get all participant responses
  description: get all participant responses OR get participant responses(s) by participantId if query
param is included
  parameters:
    - name: participantId
      in: query
      description: participantId associated with the participant response(s) ***Optional
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:
        application/json:
          schema:

```

```

    $ref: '#/components/schemas/DatabaseError'
  /participant-response/{documentId}:
  get:
    tags:
      - participant response
    summary: get participant response by Id
    parameters:
      - name: documentId
        in: path
        description: id of document in database
        required: true
        schema:
          type: string
    responses:
      '200':
        description: Success
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SuccessResponse'
      '400':
        description: Insufficient parameters
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InsufficientParameters'
      '403':
        description: unauthorized
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Unauthorized'
      '422':
        description: Invalid parameters
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InvalidParameters'
      '500':
        description: Database error
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/DatabaseError'
  put:
    tags:
      - participant response
    summary: update participant response by Id

```

```
parameters:
- name: documentId
  in: path
  description: id of document in database
  required: true
  schema:
    type: string
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/ParticipantResponse'
responses:
'200':
  description: Success
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/SuccessResponse'
'400':
  description: Insufficient parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
delete:
  tags:
  - participant response
  summary: delete participant response by Id
```



```

parameters:
  - name: documentId
    in: path
    description: id of document in database
    required: true
    schema:
      type: string
responses:
  '200':
    description: Success
  '400':
    description: Insufficient parameters
  '405':
    description: Not allowed
  '422':
    description: Invalid parameters
  '500':
    description: Database error
/participant-response/query/:
post:
  tags:
    - participant response
  summary: query for participant response(s)
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Query'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'

```

'422':
description: Invalid parameters
content:
application/json:
schema:
\$ref: '#/components/schemas/InvalidParameters'

'500':
description: Database error
content:
application/json:
schema:
\$ref: '#/components/schemas/DatabaseError'

/participant/:

post:

tags:

- participant

summary: create participant

requestBody:

content:

application/json:

schema:

\$ref: '#/components/schemas/Participant'

responses:

'200':

description: Success

content:

application/json:

schema:

\$ref: '#/components/schemas/SuccessResponse'

'400':

description: Insufficient parameters

content:

application/json:

schema:

\$ref: '#/components/schemas/InsufficientParameters'

'403':

description: unauthorized

content:

application/json:

schema:

\$ref: '#/components/schemas/Unauthorized'

'422':

description: Invalid parameters

content:

application/json:

schema:

```

    $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
  get:
    tags:
      - participant
    summary: get all participants
    description: get all participants OR get participant(s) by participantId if query param is included
    parameters:
      - name: participantId
        in: query
        description: participantId associated with the participant(s) ***Optional
        schema:
          type: string
    responses:
      '200':
        description: Success
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SuccessResponse'
      '400':
        description: Insufficient parameters
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InsufficientParameters'
      '403':
        description: unauthorized
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Unauthorized'
      '422':
        description: Invalid parameters
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InvalidParameters'
      '500':
        description: Database error
        content:
          application/json:
            schema:

```

```

    $ref: '#/components/schemas/DatabaseError'
/participant/{documentId}:
get:
  tags:
    - participant
  summary: get participant by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DatabaseError'
  put:
    tags:
      - participant
    summary: update participant by Id

```

```
parameters:
  - name: documentId
    in: path
    description: id of document in database
    required: true
    schema:
      type: string
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Participant'
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
delete:
  tags:
    - participant
  summary: delete participant by Id
```

```
parameters:
- name: documentId
  in: path
  description: id of document in database
  required: true
  schema:
    type: string
responses:
'200':
  description: Success
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/SuccessResponse'
'400':
  description: Insufficient parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
```

/questionnaire/:

```
post:
  tags:
  - questionnaire
  summary: create questionnaire
  requestBody:
    required: true
  content:
```

```

application/json:
  schema:
    $ref: '#/components/schemas/Questionnaire'
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
get:
  tags:
    - questionnaire
  summary: get all questionnaires
  description: get all questionnaires OR get questionnaire(s) by participantId if query param is
included
  parameters:
    - name: participantId
      in: query
      description: participantId associated with the questionnaire(s) ***Optional
      schema:
        type: string
  responses:
    '200':

```

```

description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
/questionnaire/{documentId}:
get:
  tags:
    - questionnaire
  summary: get questionnaire by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'

```



```

'400':
  description: Insufficient parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
put:
  tags:
    - questionnaire
  summary: update questionnaire by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Questionnaire'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'

```

```

'400':
  description: Insufficient parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
delete:
  tags:
    - questionnaire
  summary: delete questionnaire by Id
  parameters:
    - name: documentId
      in: path
      description: id of document in database
      required: true
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'

```

```

'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
/questionnaire/query/:
  post:
    tags:
      - questionnaire
    summary: query for questionnaire(s)
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Query'
    responses:
      '200':
        description: Success
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/SuccessResponse'
      '400':
        description: Insufficient parameters
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InsufficientParameters'
      '403':
        description: unauthorized
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Unauthorized'

```

'422':
description: Invalid parameters
content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InvalidParameters'

'500':
description: Database error
content:
 application/json:
 schema:
 \$ref: '#/components/schemas/DatabaseError'

/question/:
post:
 tags:
 - question
 summary: create question
 requestBody:
 required: true
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Question'

responses:
'200':
 description: Success
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/SuccessResponse'

'400':
description: Insufficient parameters
content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InsufficientParameters'

'403':
description: unauthorized
content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Unauthorized'

'422':
description: Invalid parameters
content:
 application/json:

```

    schema:
      $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
get:
tags:
  - question
summary: get all questions
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
/question/{documentId}:
get:
tags:
  - question
summary: get question by Id

```

```
parameters:
- name: documentId
  in: path
  description: id of document in database
  required: true
  schema:
    type: string
responses:
'200':
  description: Success
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/SuccessResponse'
'400':
  description: Insufficient parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
put:
  tags:
  - question
  summary: update question by Id
  parameters:
  - name: documentId
    in: path
    description: id of document in database
    required: true
    schema:
```

```

    type: string
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Question'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
      description: Database error
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DatabaseError'
  delete:
    tags:
      - question
    summary: delete question by Id
    parameters:
      - name: documentId
        in: path
        description: id of document in database
        required: true
        schema:

```

```
    type: string
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
```

```
/survey/:
  post:
    tags:
      - survey
    summary: create survey
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Survey'
    responses:
      '200':
        description: Success
```



```

content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
  description: Insufficient parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InsufficientParameters'
'403':
  description: unauthorized
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Unauthorized'
'422':
  description: Invalid parameters
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
get:
  tags:
    - survey
  summary: get all surveys
  description: get all surveys OR get survey(s) by participantId if query param is included
  parameters:
    - name: participantId
      in: query
      description: participantId associated with the survey(s) ***Optional
      schema:
        type: string
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters

```

```

content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
/survey/{documentId}:
get:
tags:
- survey
summary: get survey by Id
parameters:
- name: documentId
  in: path
  description: id of document in database
  required: true
  schema:
    type: string
responses:
'200':
description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':

```

```

description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
put:
tags:
  - survey
summary: update survey by Id
parameters:
  - name: documentId
    in: path
    description: id of document in database
    required: true
    schema:
      type: string
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Survey'
responses:
'200':
description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':

```

```

description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
description: Database error
content:
  application/json:
    schema:
      $ref: '#/components/schemas/DatabaseError'
delete:
tags:
  - survey
summary: delete survey by Id
parameters:
  - name: documentId
    in: path
    description: id of document in database
    required: true
    schema:
      type: string
responses:
'200':
description: Success
content:
  application/json:
    schema:
      $ref: '#/components/schemas/SuccessResponse'
'400':
description: Insufficient parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InsufficientParameters'
'403':
description: unauthorized
content:
  application/json:
    schema:
      $ref: '#/components/schemas/Unauthorized'
'422':

```

```
description: Invalid parameters
content:
  application/json:
    schema:
      $ref: '#/components/schemas/InvalidParameters'
'500':
  description: Database error
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/DatabaseError'
/survey/query/:
  post:
    tags:
      - survey
    summary: query for survey(s)
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Query'
  responses:
    '200':
      description: Success
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SuccessResponse'
    '400':
      description: Insufficient parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InsufficientParameters'
    '403':
      description: unauthorized
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Unauthorized'
    '422':
      description: Invalid parameters
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/InvalidParameters'
    '500':
```

description: Database error
content:
application/json:
schema:
\$ref: '#/components/schemas/DatabaseError'

/user/:

get:

tags:

- user

summary: get all users

description: Get all users OR get users by participantId if query param is included

parameters:

- name: participantId

in: query

description: participantId associated with the user(s) ***Leave blank to get all users

schema:

type: string

responses:

'200':

description: Success

content:

application/json:

schema:

\$ref: '#/components/schemas/SuccessResponse'

'400':

description: Insufficient parameters

content:

application/json:

schema:

\$ref: '#/components/schemas/InsufficientParameters'

'403':

description: unauthorized

content:

application/json:

schema:

\$ref: '#/components/schemas/Unauthorized'

'422':

description: Invalid parameters

content:

application/json:

schema:

\$ref: '#/components/schemas/InvalidParameters'

'500':

description: Database error

content:

application/json:

```

    schema:
      $ref: '#/components/schemas/DatabaseError'
  /user/{documentId}:
    get:
      tags:
        - user
      summary: get user by id
      parameters:
        - name: documentId
          in: path
          description: id of document in database
          required: true
          schema:
            type: string
      responses:
        '200':
          description: Success
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/SuccessResponse'
        '400':
          description: Insufficient parameters
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/InsufficientParameters'
        '403':
          description: unauthorized
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Unauthorized'
        '422':
          description: Invalid parameters
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/InvalidParameters'
        '500':
          description: Database error
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DatabaseError'
    put:
      tags:
        - user

```

```
summary: update user by id
parameters:
  - name: documentId
    in: path
    description: id of document in database
    required: true
    schema:
      type: string
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/User'
responses:
  '200':
    description: Success
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SuccessResponse'
  '400':
    description: Insufficient parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InsufficientParameters'
  '403':
    description: unauthorized
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Unauthorized'
  '422':
    description: Invalid parameters
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/InvalidParameters'
  '500':
    description: Database error
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/DatabaseError'
/user/query/:
  post:
  tags:
```


- user
summary: query for user(s)
requestBody:
 required: true
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Query'
responses:
'200':
 description: Success
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/SuccessResponse'
'400':
 description: Insufficient parameters
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InsufficientParameters'
'403':
 description: unauthorized
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/Unauthorized'
'422':
 description: Invalid parameters
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/InvalidParameters'
'500':
 description: Database error
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/DatabaseError'

components:
 schemas:
 EHR:
 type: object
 properties:
 id:
 type: string

receipts:
type: array
items:
 \$ref: '#/components/schemas/Receipt'
participantId:
type: string
timestamp:
type: string
org:
type: string

Image:
type: object
properties:
 id:
 type: string
 fileName:
 type: string
 data:
 type: string
 org:
 type: string

Inbox:
type: object
properties:
 id:
 type: string
 participantId:
 type: string
 messages:
 type: array
 items:
 \$ref: '#/components/schemas/Msg'
 org:
 type: string

InformedConsent:
type: object
properties:
 id:
 type: string
 surveyId:
 type: string
 participantId:

type: string
emailSent:
type: string
dateAgreed:
type: string
org:
type: string

Message:
type: object
properties:
id:
type: string
emailRecipients:
type: array
items:
type: string
pushRecipients:
type: array
items:
type: string
smsRecipients:
type: array
items:
type: string
message:
type: string
subject:
type: string
timestamp:
type: string
org:
type: string

ParticipantDiary:
type: object
properties:
id:
type: string
profileDOB:
type: string
profileName:
type: string
surveyId:
type: string
participantId:

type: string
formType:
type: string
dateWritten:
type: string
Health Event
healthEvent:
type: number
healthEventSpecification:
type: string
onsetDate:
type: string
endDate:
type: string
ongoingStatus:
type: number
descriptionData:
type: string
outcome:
type: number
outcomeSpecification:
type: string
eventTreatment:
type: number
postEventTreatment:
type: number
Dr Visit
assessers:
type: array
items:
type: number
assessorText:
type: string
GMFCScore:
type: string
GMFCType:
type: number
eventDate:
type: string
redirect:
type: boolean
visitReason:
type: string
prescription:
type: number
device:
type: array
items:

type: number
Withdrawl
withdrawalDate:
type: string
withdrawalReason:
type: string
org:
type: string

ParticipantResponse:
type: object
properties:
id:
type: string
surveyId:
type: string
questionnaireId:
type: string
profile:
type: string
profileDOB:
type: string
participantId:
type: string
dateWritten:
type: string
complete:
type: boolean
responses:
type: array
items:
\$ref: '#/components/schemas/Response'
org:
type: string

Participant:
type: object
properties:
id:
type: string
securityQuestions:
type: array
items:
\$ref: '#/components/schemas/SecurityQuestion'
public:
type: boolean

participantId:
type: string
org:
type: string

Questionnaire:
type: object
properties:
id:
type: string
surveyId:
type: string
name:
type: string
shortDescription:
type: string
description:
type: string
dateCreated:
type: string
open:
type: boolean
archived:
type: boolean
locked:
type: boolean
participants:
type: array
items:
type: string
questions:
type: array
items:
\$ref: '#/components/schemas/Question'
org:
type: string

Question:
type: object
properties:
id:
type: string
title:
type: string
variable:
type: string

type:
 type: string
text:
 type: string
required:
 type: boolean
requiredMessage:
 type: string
placeholder:
 type: string
max:
 type: string
min:
 type: string
format:
 type: string
options:
 \$ref: '#/components/schemas/Options'
choices:
 type: array
 items:
 \$ref: '#/components/schemas/Choice'
rules:
 type: array
 items:
 \$ref: '#/components/schemas/Rule'
sections:
 type: array
 items:
 type: object
org:
 type: string

Survey:
 type: object
 properties:
 id:
 type: string
 name:
 type: string
 description:
 type: string
 shortDescription:
 type: string
 informedConsent:
 type: string
 open:

type: boolean
locked:
type: boolean
archieved:
type: boolean
dateCreated:
type: string
participants:
type: array
items:
type: string
org:
type: string

User:

type: object
properties:
id:
type: string
firstName:
type: string
lastName:
type: string
userName:
type: string
active:
type: boolean
emailEnabled:
type: boolean
smsEnabled:
type: boolean
agreedToTerms:
type: boolean
dateCreated:
type: string
agreedToTermsDate:
type: string
phoneNumber:
type: string
participantId:
type: string
profiles:
type: array
items:
\$ref: '#/components/schemas/Profile'
securityQuestions:
type: array


```
items:
  $ref: '#/components/schemas/SecurityQuestion'
org:
  type: string
```

subtypes

```
Profile:
  type: object
  properties:
    id:
      type: string
    dob:
      type: string
    gender:
      type: string
    isNew:
      type: boolean
    name:
      type: string
```

```
SecurityQuestion:
  type: object
  properties:
    question:
      type: string
    answer:
      type: string
```

```
Response:
  type: object
  properties:
    question:
      type: string
    response:
      type: string
```

```
Msg:
  type: object
  properties:
    id:
      type: string
    message:
      type: string
    read:
      type: boolean
    subject:
      type: string
```

timestamp:
type: string

Receipt:
type: object
properties:
ehr:
 \$ref: '#/components/schemas/EHRData'
path:
 type: string
profile:
 \$ref: '#/components/schemas/Profile'

EHRData:
type: object
properties:
id:
 type: string
ehrType:
 type: string
logo:
 type: string
name:
 type: string

Question subtypes

Options:
type: object
properties:
max:
 type: string
min:
 type: string
pin:
 type: boolean
step:
 type: string
ticks:
 type: boolean

Choice:
type: object
properties:
index:
 type: string
order:
 type: number

text:
 type: string
value:
 type: string

Rule:
 type: object
 properties:
 skipTo:
 type: string
 ruleType:
 type: string
 id:
 type: string
 expression:
 \$ref: '#/components/schemas/RuleExpression'

RuleExpression:
 type: object
 properties:
 entries:
 type: array
 items:
 \$ref: '#/components/schemas/RuleEntry'

RuleEntry:
 type: object
 properties:
 value:
 type: string
 fact:
 type: string
 operator:
 type: string

/query/ types

Query:
 type: object
 properties:
 query:
 type: array
 items:
 \$ref: '#/components/schemas/QueryParam'

QueryParam:
 type: object
 properties:

key:
 type: string
 example: 'Id'
operator:
 type: string
 example: '=='
value:
 type: string
 example: 'id1234'

auth body type

Auth:
 type: object
 properties:
 username:
 type: string
 password:
 type: string

path type

Path:
 type: object
 properties:
 path:
 type: string
 example:
 '849500/John/06112021/PatientId-e467435623rfg86f-healthrecord-10859271.json'

Response types

SuccessResponse:
 type: object
 properties:
 STATUS:
 type: string
 example: 'SUCCESS'
 MESSAGE:
 type: string
 example: 'message'
 DATA:
 type: object
 example: {"example data": 'your response data will be here'}

#400

InsufficientParameters:
 type: object
 properties:
 STATUS:
 type: string
 example: 'FAILURE'

MESSAGE:
type: string
example: 'Insufficient parameters'
DATA:
type: object
example: {"example error": "missing an expected field"}

#422

InvalidParameters:
type: object
properties:
STATUS:
type: string
example: 'FAILURE'
MESSAGE:
type: string
example: 'invalid parameters'
DATA:
type: object
example: {"example error": "field is of wrong data type or value"}

#500

DatabaseError:
type: object
properties:
STATUS:
type: string
example: 'FAILURE'
MESSAGE:
type: string
example: 'databse error'
DATA:
type: object
example: {"example error": "not found"}

#403

Unauthorized:
type: string
example: unauthorized