

API Reference Documentation

1. [Questionnaire](#)
 2. [User](#)
 3. [Workflow](#)
 4. [Archive](#)
-

Questionnaire

1. Get all questionnaires (returns a list of questionnaires)

[GET] /api/v1/questionnaire

2. Get questionnaire by id (returns one questionnaire)

[GET] /api/v1/questionnaire/{id}

3. Get form by assignedVendorId (returns a list of forms assigned to Vendor)

[GET] /api/v1/questionnaire/vendor/{assignedVendorId}

4. Get form by assignedAdminId (returns a list of forms assigned to Admin)

[GET] /api/v1/questionnaire/admin/{assignedAdminId}

5. Create questionnaire (returns one questionnaire)

[POST] /api/v1/questionnaire/create

[Example Request Body, MUST STRICTLY FOLLOW THIS FORMAT]

```
{
  "title" : "1st questionnaire",
  "assignedVendorId": "jack-1",
  "assignedAdminId": "not-jack-1",
  "assignedTo": "VENDOR", (must be VENDOR or ADMIN)
  "status": "SUBMITTED", (status must be: NOT_STARTED, SUBMITTED, ADMIN_APPROVED,
RETURNED, APPROVER_APPROVED)
  "submissionDeadline": "2023-03-22T09:16:28.866",
  "questionsAndAnswers": {
    "something": {
      "first thing" : "hello",
      "second thing": "omma"
    }
  },
  "feedback": []
}
```

6. Update questionnaire (returns one questionnaire) (WHENEVER U NEED TO UPDATE ANY FIELD JUST USE THIS METHOD)

[PUT] /api/v1/questionnaire/update

[Example Request Body]

```
{
  "id": "641a56ec5ad139669ecbce70",
  "title": "1st questionnaire",
  "assignedVendorId": "jack-1",
  "assignedAdminId": "not-jack-1",
  "assignedTo": "VENDOR",
  "status": "SUBMITTED",
  "submissionDate": "2023-03-22T09:16:28.866",
  "submissionDeadline": "2023-03-22T09:16:28.866",
  "questionsAndAnswers": {
    "something": {
      "first thing": "hello",
      "second thing": "omma"
    }
  },
  "feedback": [],
  "createdAt": "2023-03-22T09:16:28.866"
}
```

7. Delete questionnaire by id

[DELETE] /api/v1/questionnaire/delete/{id}/{userId}

8. Get questionnaires based on status (returns a list of questionnaires)

[GET] /api/v1/questionnaires/status/{formStatus}

User

1. Get all users and vendors

[GET] /api/v1/users

[GET] /api/v1/users/vendors

2. Get all users based on user type

- User Types `userType` include: `ADMIN`, `APPROVER`

NOTE: this is only to get admin and approver

[GET] /api/v1/users/type/{userType}

3. Get all vendors from 1 company

[GET] /api/v1/users/vendors/company/{companyName}

4. Get 1 user based on unique user id

- Regardless of the user type

[GET] /api/v1/users/{id}

[GET] /api/v1/users/vendors/{id}

5. Create a user and vendor

[POST] /api /v1/users/create

[Example post request body for ADMIN/APPROVER]

```
{
  id: "admin1",
  name: "Harry Styles",
  email: "harrystyles@quantum.com",
  password: "watermelonsugar",
  userType: "ADMIN",
}
```

[POST] /api /v1/users/vendors/create

[Example post request body for VENDOR]

```
{
  "id": "vendor1",
  "name": "Harry Styles",
  "email": "harrystyles@lovetour.com",
  "password": "watermelonsugar",
  "userType": "VENDOR",
  "companyName": "Love",
  "regNumber": "4567",
  "bizNature": "Love",
  "contactNum": "91234567",
  "gstnumber": "GST456",
  "country": "Singapore"
}
```

6. Delete user and vendor by id

[DELETE] /api /v1/users/delete/{id}/{deleteId}

[DELETE] /api /v1/users/vendors/delete/{id}/{deleteId}

7. Update user and vendor

[PUT] /api /v1/users/update

[Example post request body for ADMIN/APPROVER]

```
{
  id: "admin1",
  name: "Harry Styles",
  email: "harrystyles@quantum.com",
  password: "treatpeoplewithkindness",
  userType: "ADMIN",
  createdAt: "2023-03-21T11:09:36.533"
}
```

[PUT] /api /v1/users/vendors/update

[Example post request body for VENDOR]

```
{
  "id": "vendor1",
  "name": "Harry Styles",
  "email": "harrystyles@lovetour.com",
  "password": "watermelonsugar",
  "userType": "VENDOR",
}
```

```
{
  "companyName": "Love",
  "regNumber": "4567",
  "bizNature": "Love",
  "contactNum": "91234567",
  "gstnumber": "GST456",
  "country": "Singapore"
}
```

8. Login for users and vendors

- EXTREMELY primitive login

[POST] /api /v1/users/vendors/login

[POST] /api /v1/users/login

=> please pass the user/vendor object in with id and password

- other fields will be treated as null

```
{
  "id": "vendor1",
  "password": "watermelonsugar",
}
```

will return the following output

- List of Strings
- format: userId and userType

["admin", "ADMIN"]

Workflow

1. Get all workflows

[GET] /api /v1/workflow

2. Get all assigned workflows

[GET] /api /v1/workflow/assigned

3. Get all assigned workflow with a specific status

[GET] /api /v1/workflow/assigned/status/{approverReviewStatus}

4. Find by ID

[GET] /api /v1/workflow/{id}

5. Find assigned workflow by ID

[GET] /api /v1/workflow/assigned/{id}

6. Find assigned by admin ID

[GET] /api /v1/workflow/assigned/admin/{id}

7. Find assigned by vendor ID

[GET] /api /v1/workflow/assigned/vendor/{id}

8. Create workflow

[POST] /api /v1/workfl ow/

```
{
  "id": "asodnoiefwe112312312",
  "workfl owName": "OOP is the best!",
  "workfl owDescription": "Best mod in SMU",
  "questi onnai reLi st": Li st<Stri ng>,
  "questi onnai res": Li st<Questi onnai re>,
  "createdAt": "2023-03-21T11: 09: 36. 533"
}
```

9. create assigned workflow

[POST] /api /v1/workfl ow/assi gned

approverReviewStatus must be in the following format: **INITIAL_DRAFT**, **REJECTED**, **FLAGGED**, **APPROVED**

```
{
  "id": "asodnoiefwe112312312",
  "workfl owName": "OOP is the best!",
  "workfl owDescription": "Best mod in SMU",
  "questi onnai reLi st": Li st<Stri ng>,
  "questi onnai res": Li st<Questi onnai re>,
  "createdAt": "2023-03-21T11: 09: 36. 533",
  "assi gnedVendorId": "vendor1",
  "assi gnedAdminId": "admin1",
  "approval RequestDate": "2023-03-22T11: 09: 36. 533",
  "approverRevi ewStatus": "APPROVED",
  "approvedAt": "2023-03-23T11: 09: 36. 533"
}
```

10. Update workflow

[PUT] /api /v1/workfl ow

11. Update assigned workflow

[PUT] /api /v1/workfl ow/assi gned

12. Delete workflow

[DELETE] /api /v1/workfl ow

13. Delete assigned workflow

[DELETE] /api /v1/workfl ow/assi gned

Archive

1. Get all archive document based on its collection (returns a list of archive document)

- Collection **col l ecti on** include: **WORFKLOWS**, **QUESTI ONNAI RES**, **USERS**, **VENDORS**

[GET] /api /v1/archi ve/col l ecti on/{col l ecti on}

2. restore archive document into its original collection

[DELETE] /api /v1/archi ve/restoreDocument/{i d}
