# High-level documentation for API

A high-level documentation on how to combine endpoint to create a client application. A client application must provide the following functionalities:

* Define an administration process, which is a sequence of phase.
* Define responsible users for every phase in process.
* Define a structure of a form, which are sections and fields.
* Authenticated user can instantiate an instance from the form.
* Instance can be transit through phases.
* Reconstruct instance at any phase in the commit history.

These functionalities are break down in correspond sections below.

We use HTTP methods and the path in the URL to denote an endpoint, for example: POST /forms. Documentation on how to create a request and get back the response is provided in low-level documentation as file openapi.json. In this high-level documentation, we only describe how to combine endpoints to get the proper functionalities of the designed system.

## Define a sequence of phases (administration process)

Steps for create an administration process:

1. Create form
2. Create phases
3. Create transitions

To create a sequence of phase, first we must create a form. We use the endpoint POST /forms to create a form. An example of the response body in JSON format is follow:

**{**

**"id": 1,**

**"created\_at": "2022-03-21T14:46:04.566020+01:00",**

**"name": "Proposal topic for thesis",**

**"creator\_id": 1,**

**"public": false,**

**"obsolete": false**

**}**

In the response, we have the id of the new form, from which we can use to create the phases for this form.

To create a phase for the form, we use the endpoint POST /phases, in which we provide the id of the form in the field form\_id of request body. Each request returns the new phase’s information including the id of the phase, which we use to create transitions of process. An example of a response body is as follow:

**{**

**"id": 1,**

**"created\_at": "2022-03-21T14:06:37.162Z",**

**"form\_id": 1,**

**"name": "Phase for student",**

**"description": "string",**

**"position\_id": 11,**

**"phase\_type": "begin",**

**"order": 1**

**}**

*Note: the creation of phase includes the definition responsible user for this phase. Please refer to the next subsection.*

To create a transition, we use the endpoint POST /transitions. In which we provide the id of the source phase in field from\_phase\_id, and the id of the target phase in field to\_phase\_id of the request body.

Best practice show that we should plan all phases and its transitions in the administration process and before creating it.

We can show all phases and transitions of the form through endpoint GET /forms/{form\_id}/phases and GET /forms/{form\_id}/transitions.

*Note: Only admin can create an administration process.*

## Define responsible users for every phase in administration process

We know that for each phase there is only one director and for each section of this phase there only one handler. Responsible users are director and handlers. To define director and handlers for a phase, we use position to indicate that only user, who held the correspond position can become director or handlers of the phase. We say: a position is designated for a phase and a position is assigned for a section.

First, we must get all positions in the systems by using endpoint GET /positions. An example of a response body from this endpoint is as follow:

**[**

**...**

**{**

**"id": 11,**

**"created\_at": "2022-03-20T12:27:58.917751+01:00",**

**"name": "Student of 2-CIT-C",**

**"group\_id": 7,**

**"role\_id": 8,**

**"creator\_id": 1**

**},**

**...**

**]**

A position id is used to provide in the field positions\_id of the request body of endpoint POST /phases to designate a position for this new phase. An example of request body for creating a phase is follow:

**{**

**"form\_id": 1,**

**"name": "Phase for student",**

**"description": null,**

**"position\_id": 11,**

**"phase\_type": "begin",**

**"order": 1**

}

Similarly, we provide a position id in field position\_id of the request body of endpoint POST /sections to assign a position for this new section. An example of request body for creating a section is follow:

**{**

**"name": "Section for student of 2-CIT-C",**

**"phase\_id": 1,**

**"position\_id": 11,**

**"order": 1**

**}**

*Note: only admin can designate and assign positions for phases and sections of an administration process.*

## Define a structure of form

A structure of a form includes sections and fields. Each phase of a form includes sections, and each section includes fields.

To create section, first we must know which phase the new section belongs to. The phase id then is used to provide as value of the field phase\_id in the request body of endpoint POST /sections. An example of the response body from this endpoint is as follow:

**{**

**"id": 1,**

**"name": "Section for student of 2-CIT-C",**

**"created\_at": "2022-03-20T12:28:00.367400+01:00",**

**"phase\_id": 1,**

**"position\_id": 11,**

**"order": 1**

**}**

*Note: the creation of section includes the definition responsible user for this section of a phase. Please refer to the previous subsection.*

To create a field, we use endpoint POST /fields, in which we provide section id as a value of the field section\_id to indicate that the new field belongs to this section. An example of the response from this endpoint is as follow:

**{**

**"id": 1,**

**"created\_at": "2022-03-20T12:28:00.367400+01:00",**

**"name": "Field for student of 2-CIT-C",**

**"section\_id": 1,**

**"order": 1**

**}**

*Note: Only admin can define structure of the form.*

## Create instance from defined form

To create an instance from a form, we use the endpoint POST /instances, in which we provide form id as a value of the field form\_id in the request body. Only user with position designated for the begin phase of the form can instantiate an instance. An example of a request body of this endpoint is as follow:

**{**

**"form\_id": 1**

**}**

An example of the response body is as follow:

**{**

**"id": 1,**

**"created\_at": "2022-03-20T12:28:01.652561+01:00",**

**"updated\_at": null,**

**"form\_id": 1,**

**"current\_phase\_id": 1,**

**"creator\_id": 1,**

**"current\_state": "initialized"**

**}**

When instance was created, the creator of the instance become the director of the begin phase and become the handler of all begin sections, which are assigned for creator’s position. All contents of these sections are auto created with the value of an empty string. This is necessary for creating the hash of the envelope.

To retrieve all directors and receivers of the instance, we use endpoint GET /instances/{instance\_id}/directors and GET /instances/{instance\_id}/receivers.

To retrieve all contents of the instance, we use endpoint GET /instances/{instance\_id}/instances\_fields.

*Note: Only participants of the instance can get related resource of instance resource.*

## Transit instance through administration process

To transit (redirect) instance to the next phase, we use endpoint PATCH /instances/{instance\_id}. In the request body of this endpoint we must specified:

* The id of the next phase as a value of field current\_phase\_id.
* The id of the user, who will become a director of the next phase, as a value of field director\_id.
* The id of sections and id of the users, who will become handlers of sections of the next phase, as key/value pairs in the field receivers.
* The message for the commit, which auto created when the instance is transited to the next phase, as value in field message.

An example of request body is as follow:

**{**

**"transit": {**

**"current\_phase\_id": 2,**

**"receivers": {**

**"3": 5**

**},**

**"director\_id": 5,**

**"message": "The proposal topic for final thesis of Pham Si Anh"**

**}**

**}**

*Note: Only director of the current phase can transit (redirect) the instance to the next phase. Director must be specified but not all section’s handlers must be specified.*

User with designated positions is called phase’s potential director. User with position assigned for a section is called section’s potential receiver (or handler). We can get all potential directors of a phase by using endpoint GET phases/{phase\_id}/potential\_directors and can get all potential receivers of a section, using endpoint GET /sections/{section\_id}/potential\_receivers or GET /sections/{section\_id}/potential\_handlers.

The receiver can handle the instance by using the same endpoint as transiting the instance, PATCH /instances/{instance\_id}. The request body however is different. Receiver only mark the field handle to true. An example is shown below:

**{**

**"handle": true**

**}**

When instance was handled by the receiver, the receiver become handler of the sections and all content of these sections assigned for this receiver will be created. These contents can only be edited by its handler.

## How to recover instance at any phase.

The commit is auto created by the system when the director redirects the instance to the next phase. The participants of the instance can retrieve all commits of the instance by using the endpoint GET /instances/{instance-id}/commits. An example of the history of commits is shown below, where contain only 1 commit.

**[**

**{**

**"hash\_commit": "f92892e32870eac834ba245a553997440226dbc9b41407a3cda7ba94c065607e",**

**"prev\_hash\_commit": null,**

**"hash\_tree": "fd5d2c21ed200ca30d8d92ec5afffe465888741a3da64d740fc9c9b535c7bfdb",**

**"creator\_id": 1,**

**"instance\_id": 1,**

**"created\_at": "2022-04-06T12:07:01.354591+02:00",**

**"current\_phase\_id": 1,**

**"message": "****The proposal topic for final thesis of Pham Si Anh"**

**}**

**]**

To reconstruct the instance at any phase of the history of commits, first we must retrieve all current contents of the instance, then we must retrieve all envelopes of instance’s commit. The client application can use the content value of these envelopes with the structure of the form to inject to content of the corresponding fields. We use the end point GET  /instances/{instance\_id}/instances\_fields to get all the current contents of the instance and the endpoint GET /commits/{hash\_commit}/envelopes to retrieve all envelopes of the instance’s commit. An example of respond body for retrieving all contents of the instances is shown below:

**[**

**{**

**"id": 1,**

**"created\_at": "2022-04-06T12:00:53.909711+02:00",**

**"updated\_at": "2022-04-06T12:01:50.279872+02:00",**

**"instance\_id": 1,**

**"field\_id": 1,**

**"creator\_id": 1,**

**"value": "Bc. Pham Si Anh",**

**"resolved": true**

**},**

**...**

**]**

An example of respond body to retrieve all envelopes of a commit is shown below.

**[**

**{**

**"hash\_envelope": "eea27c6235e8866d1184cf7f3d18dae3efc3dd24e33b80db2f6bb0f4bbf780f8",**

**"instance\_field\_id": "1",**

**"creator\_id": 1,**

**"content\_value": "Bc. Pham Si Anh",**

**"updated\_at": "2022-04-06T12:01:50.279872+02:00"**

**},**

**...**

**]**