Formatting the JSON File

We detailed the process of formatting a JSON file to create a customized questionnaire form for the RTMIS system. The RTMIS system leverages the Akvo Form Service for generating initial JSON form structures. We explored the basic structure and components of the form JSON, as documented in the Akvo React Form's README.

Additionally, we introduced specific customizations required for RTMIS, including the addition of **submission_types** at the form level and **three new parameters** at the question level: default_value, disabled, and hidden. These custom parameters are defined as objects based on the submission_type, which is specified as an enumeration.

We provided a detailed example of the JSON structure incorporating these customizations and outlined the manual steps needed to add these custom parameters after generating the initial JSON form. By following this process, users can effectively format their JSON files to meet the requirements of RTMIS customized questionnaire forms.

Overview

The RTMIS system uses a JSON file to build a questionnaire form. This JSON file can be generated using an internal library called **Akvo Form Service**. For more information and to access the editor, visit the following link: https://form-service.akvotest.org/forms

In general, all components and formats in the form JSON are documented in the Akvo React Form's README file. You can find the documentation here: https://github.com/akvo/akvo-react-form/blob/main/README.md

However, for this project, we have added customizations at the form-level definition and at the question level. These customizations include the addition of `submission_types` at the form level, and three additional parameters at the question level: `default_value`, `disabled`, and `hidden`. These parameters are defined as objects and depend on the `submission_type`. The `submission_type` itself is an enum value and is defined as a constant. You can view the definition here: https://github.com/akvo/rtmis/blob/main/backend/api/v1/v1_forms/constants.py#L37-L48.

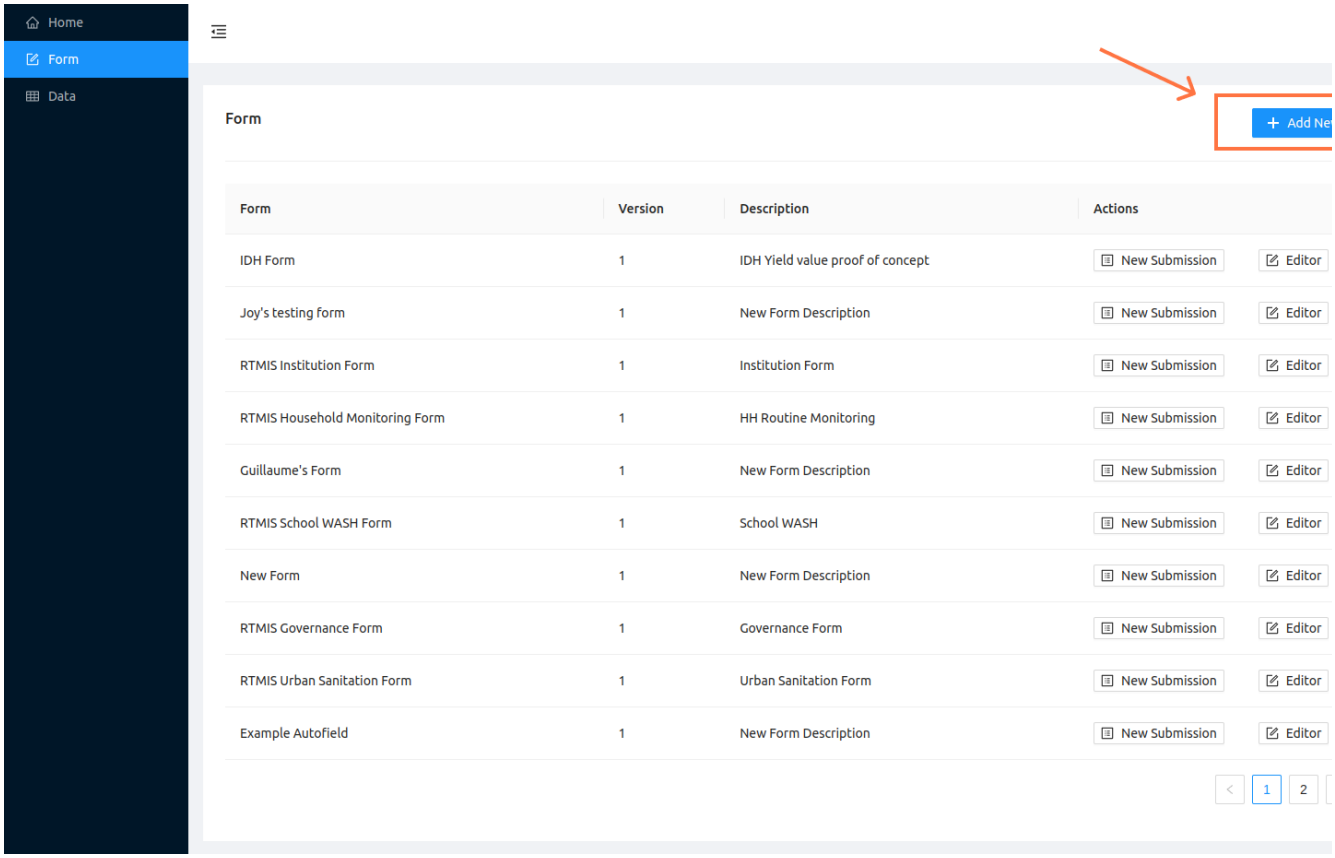
All the customizations will be added manually after the JSON form is generated with the **Akvo Form Service**.

Generate the JSON Form

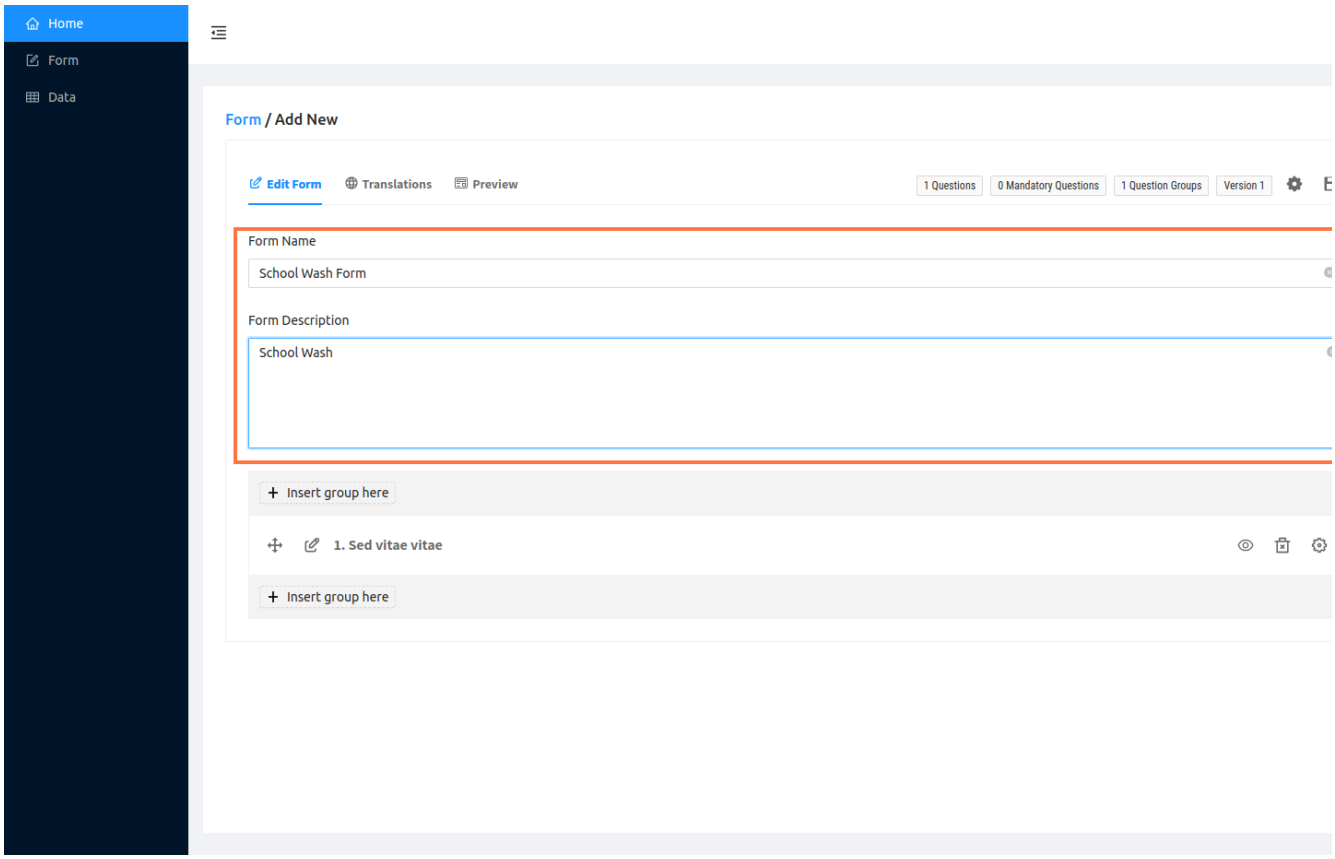
Create a New Form

Go to Akvo Form Service: Open your browser and navigate to https://form-service.akvotest.org/

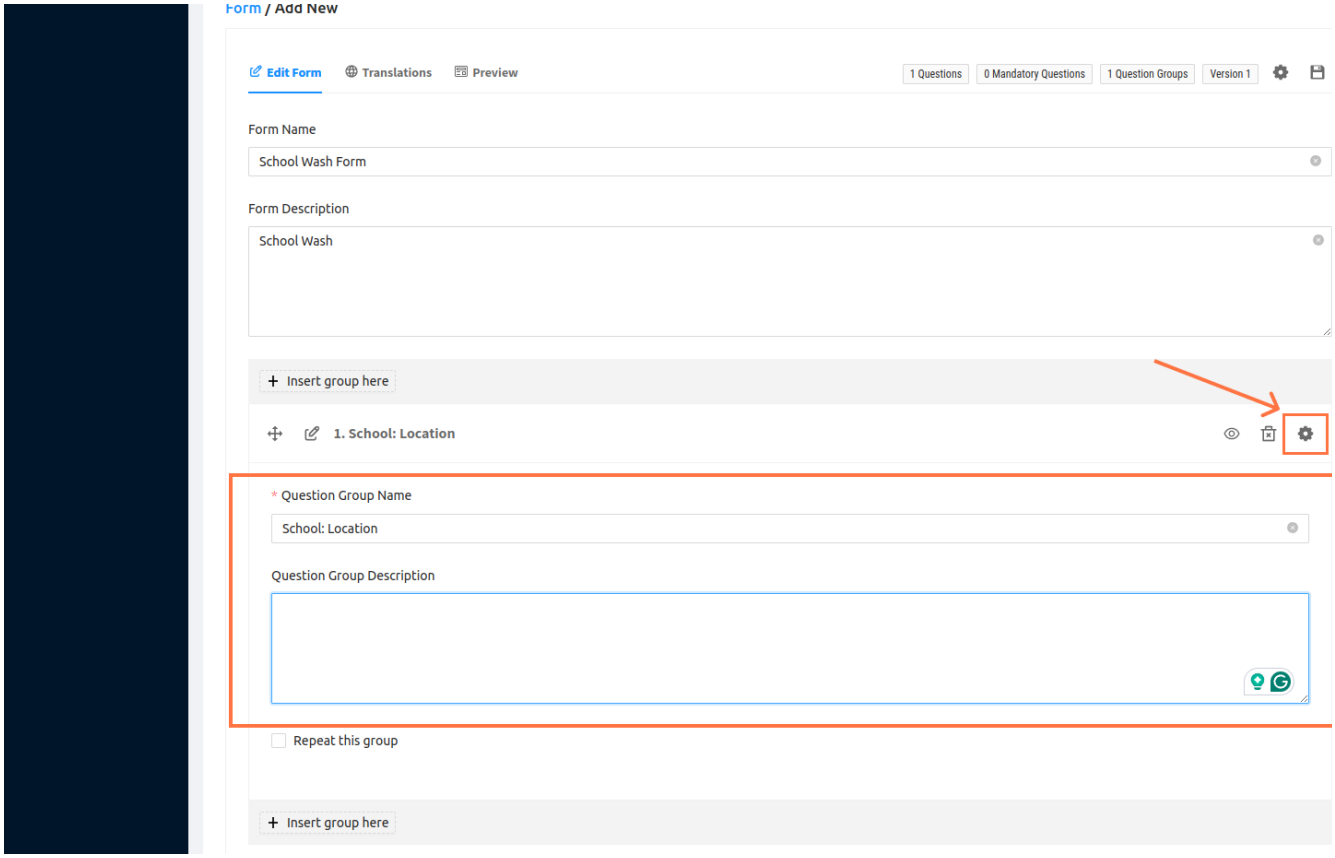
Access the Forms Menu: Click on the "Forms" menu and Click the "New" button to create a new form.



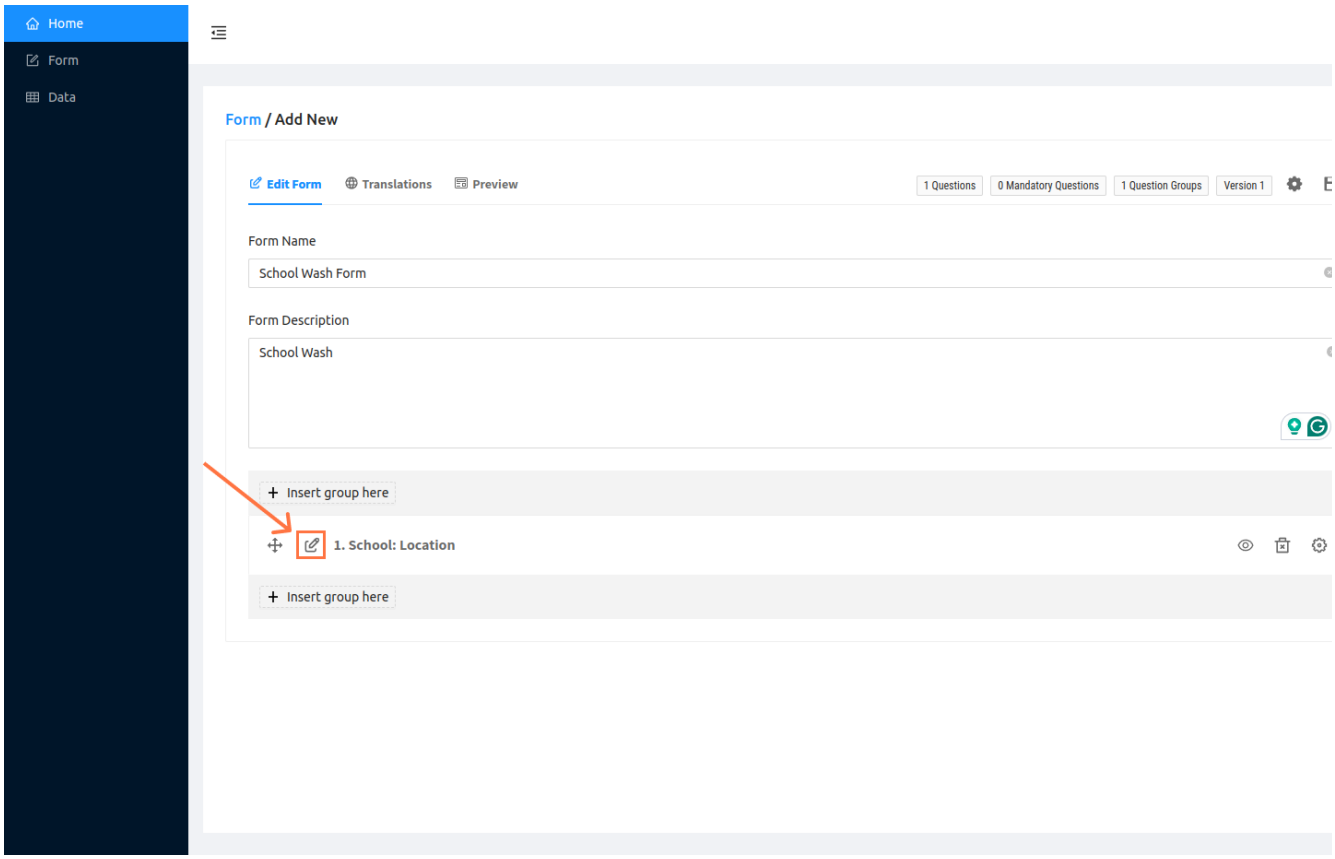
Update Title and Description: Modify the default title and description as needed.



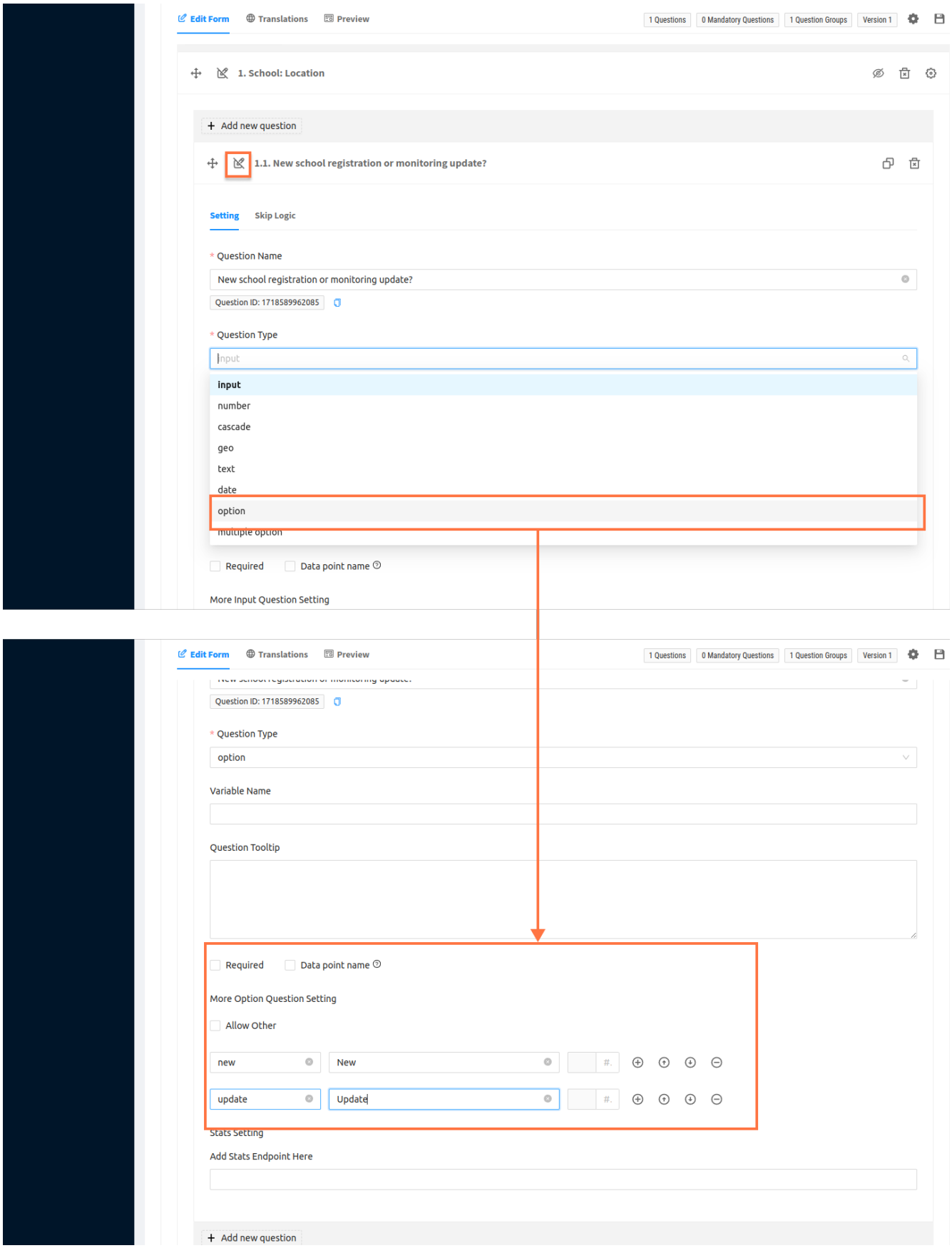
Edit Default Group Question: Click the gear icon on the right side to update the default group question. Once done, click the gear icon again to exit edit mode.



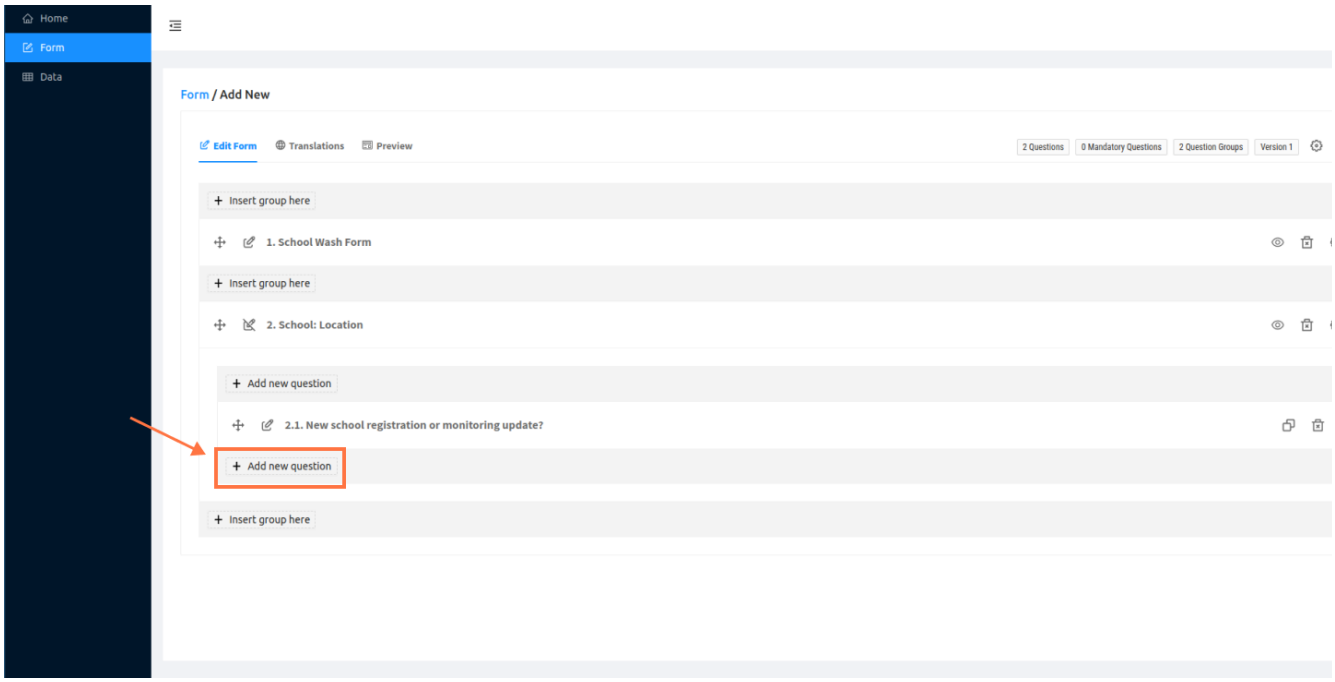
Edit questions: Click the pencil icon on the group question to edit or add questions.



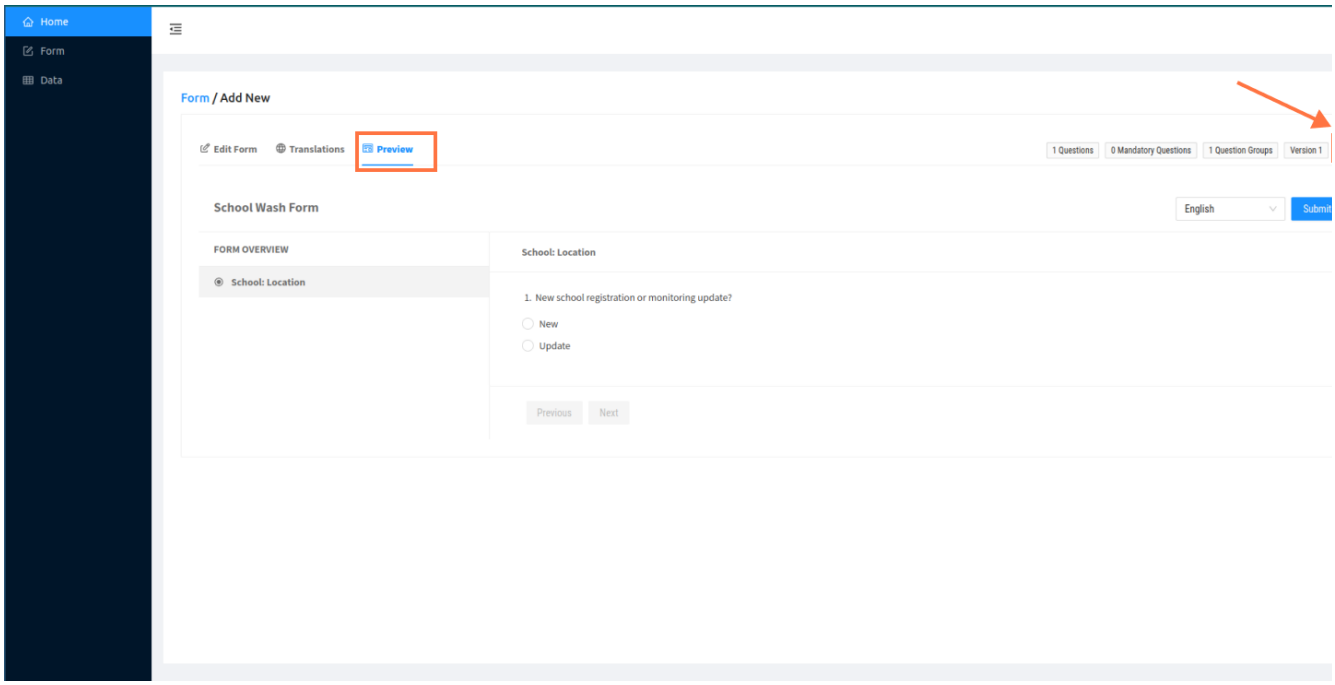
Edit questions: For the default question, click the pencil icon and update the question type (e.g., option), fill in all necessary options, and click the pencil icon again to collapse the question.



Add Questions: To add a new question, click "Add New Question" at the bottom to insert a new question **after the current one**, or click "Add New Question" at the top to insert before.

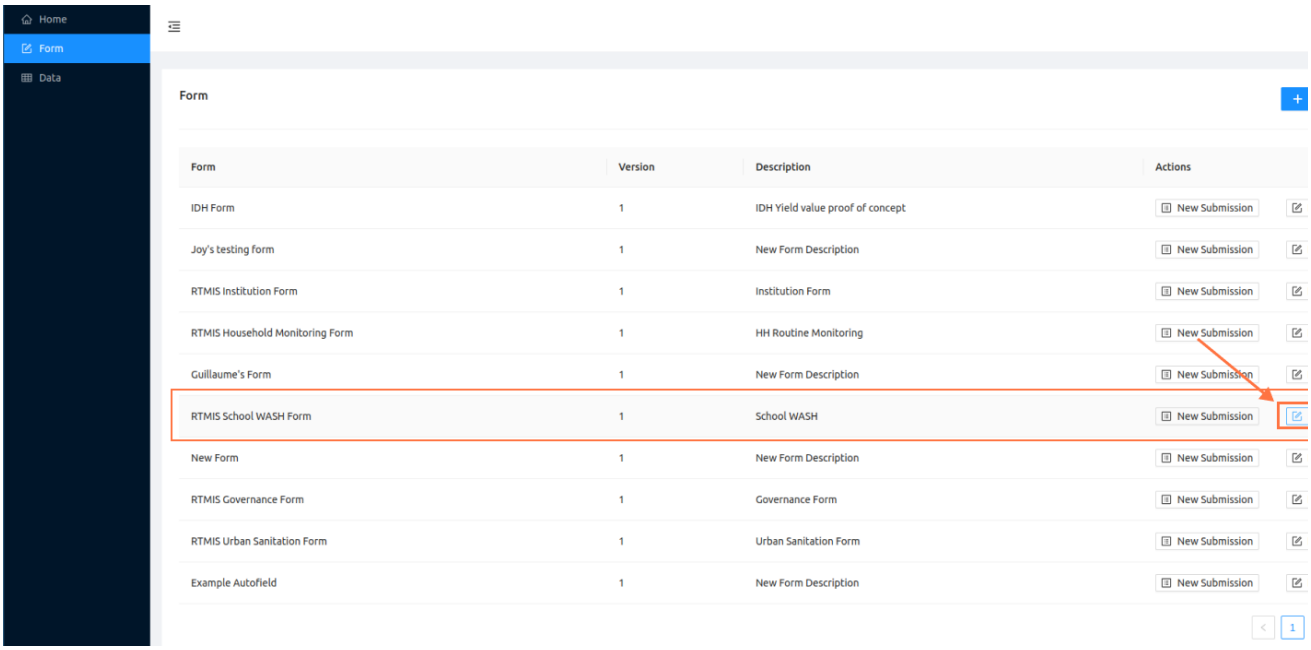


Preview and Save: Go to the "Preview" tab to review and evaluate the form settings. If everything is correct, click the "Save" button to store the current version of the form.

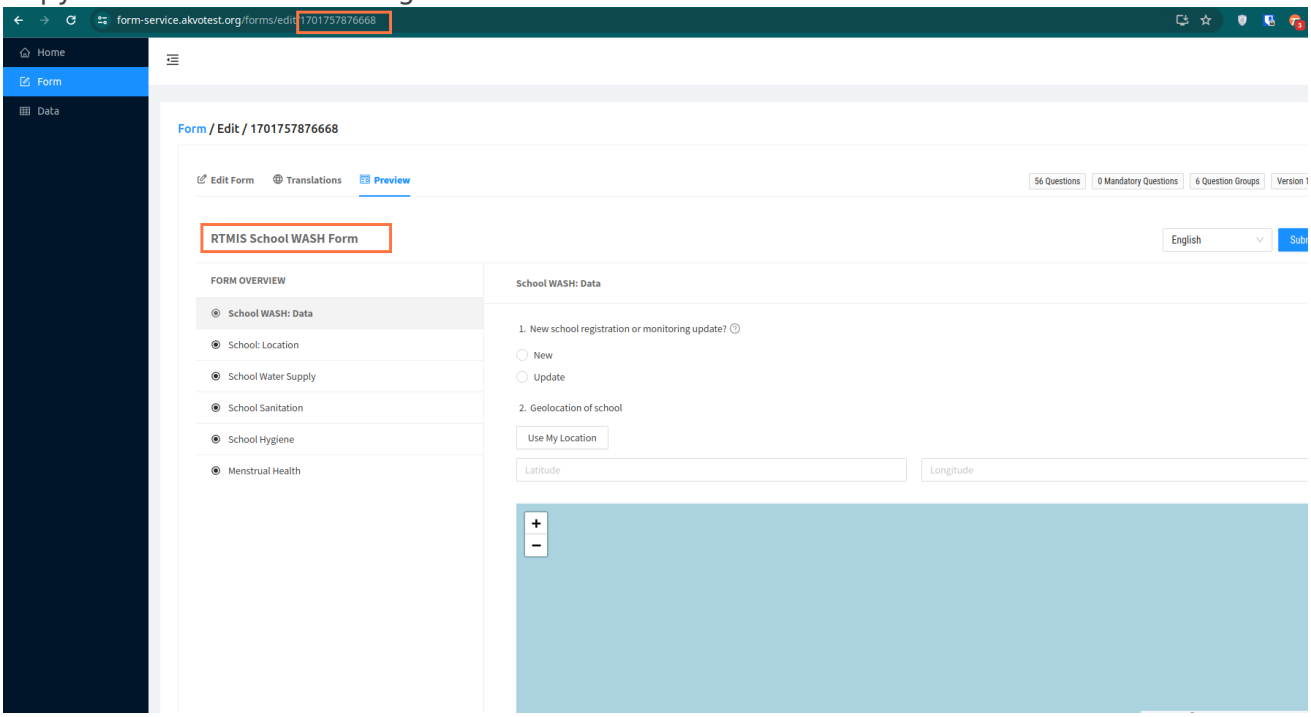


Download Akvo Form Service to RTMIS

- **Edit and Retrieve Form ID and Name**
  - On the form list, click the "Edit" button for the form you created (e.g., School Wash Form).



- Go to the "**Preview**" tab.
- Copy the ID from the last segment of the URL and the form name.



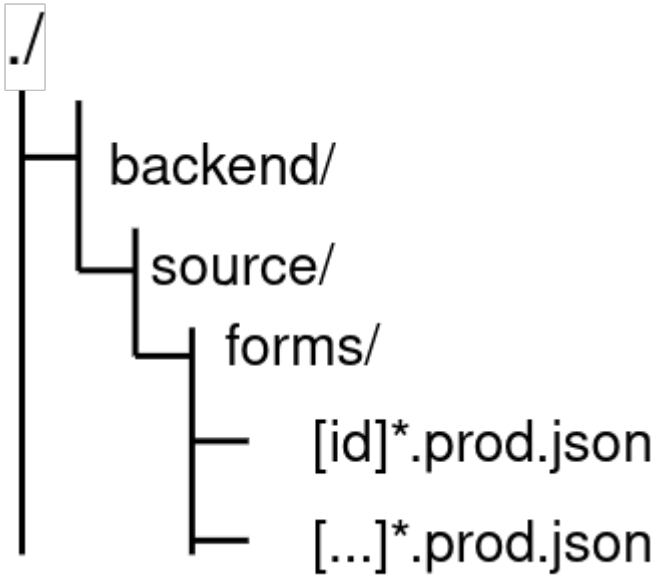
- Paste the ID and name into the designated file.
  backend/api/v1/v1_forms/management/commands/download_forms_from_afs.py#L8C1-L13C2

```
...
forms = [
    {
        "id": 1701757876668,
        "name": "RTMIS School WASH Form",
    },
```

```
]
...
```

- **Download the Form**
  - Run the command to download the form from Akvo Form Service into the RTMIS platform.

    ```
    ./dc-mobile.sh exec backend python manage.py download_forms_from_afs
    ```

  - Once the download is complete, navigate to the directory `backend/source/forms/`.
  - Locate the form by its ID with the suffix `.prod` and `.json` file extension (e.g., `1701757876668.prod.json`).
  - 

Customization Details

Form Level Customization

At the form level, we introduce a new parameter: `submission_types`. This parameter specifies the different types of submissions allowed for the form. The `submission_types` parameter is defined as an enumeration, providing a set of predefined submission types that the form can handle.

id	Submission Type	Description
1	registration	Utilized through the Registration form
2	monitoring	Utilized through the Monitoring form
3	verification	Utilized through the Grade Claim form
4	certification	Utilized through the Grade Certification form

Question Level Customization

At the question level, we introduce three new parameters:

1. **default_value**: Specifies the default value for the question. This value will be pre-filled when the form is loaded.

2. **disabled**: A boolean parameter that indicates whether the question should be disabled (i.e., not editable) when the form is displayed.

3. **hidden**: A boolean parameter that indicates whether the question should be hidden from view when the form is displayed.

These parameters are defined as objects and their values depend on the `**submission_type**`.

Example JSON Structure

Here is an example structure of the JSON file with the added customizations:


```
{
  "id": 123456,
  "form": "School WASH Form",
  "description": "School WASH",
  "defaultLanguage": "en",
  "languages": ["en"],
  "version": 1,
  "type": 1,
  "translations": null,
  "submission_types": [
    "registration",
    "monitoring",
    "verification",
    "certification"
  ],
  "question_groups": [
    {
      "id": 1699354006534,
      "order": 1,
      "name": "school_location_group_question",
      "label": "School: Location",
      "repeatable": false,
```

```
      "translations": null,
      "questions": [
        {
          "id": 1699354006535,
          "order": 1,
          "name": "new_school_registration_monitoring_update",
          "label": "New school registration or monitoring update?",
          "short_label": null,
          "type": "option",
          "tooltip": {
            "text": "Entry of school data in RTMIS (first time) or update of monitoring data
(existing school)"
          },
          "required": true,
          "meta": false,
          "options": [
            {
              "order": 1,
              "label": "New",
              "value": "new"
            },
            {
              "order": 2,
              "label": "Update",
              "value": "update"
            },
            {
              "order": 3,
              "label": "Verification",
              "value": "verification"
            },
            {
              "order": 4,
              "label": "Certification",
              "value": "certification"
            }
          ],
          "default_value": {
            "submission_type": {
              "monitoring": "update",
```

```
            "registration": "new",
            "verification": "verification",
            "certification": "certification"
          }
        }
      },
      {
        "id": 1699951210638,
        "order": 2,
        "name": "school_location",
        "label": "What is the location of the school?",
        "short_label": null,
        "type": "administration",
        "tooltip": {
          "text": "This question contains a list of possible school locations, starting with the government area or district, down to the community."
        },
        "required": true,
        "meta": true,
        "disabled": {
          "submission_type": ["monitoring", "verification", "certification"]
        }
      },
      {
        "id": 1716283778,
        "order": 33,
        "name": "schools_achieved_required_outcomes",
        "label": "Have 100% of school achieved the required outcomes for this grade?",
        "short_label": null,
        "type": "option",
        "required": true,
        "meta": false,
        "options": [
          {
            "order": 1,
            "label": "Yes",
            "value": "yes",
            "color": "green"
          },
          {
```

```
              "order": 2,
              "label": "No",
              "value": "no",
              "color": "red"
            }
          ],
          "hidden": {
            "submission_type": ["registration", "monitoring", "certification"]
          }
        }
      ]
    }
  ]
}
```

By following these steps, you can successfully format the JSON file to work with RTMIS as a customized questionnaire form.

Revision #7
Created 2024-06-14 12:14:52 UTC by Iwan Firmawan
Updated 2024-06-17 04:42:23 UTC by Iwan Firmawan