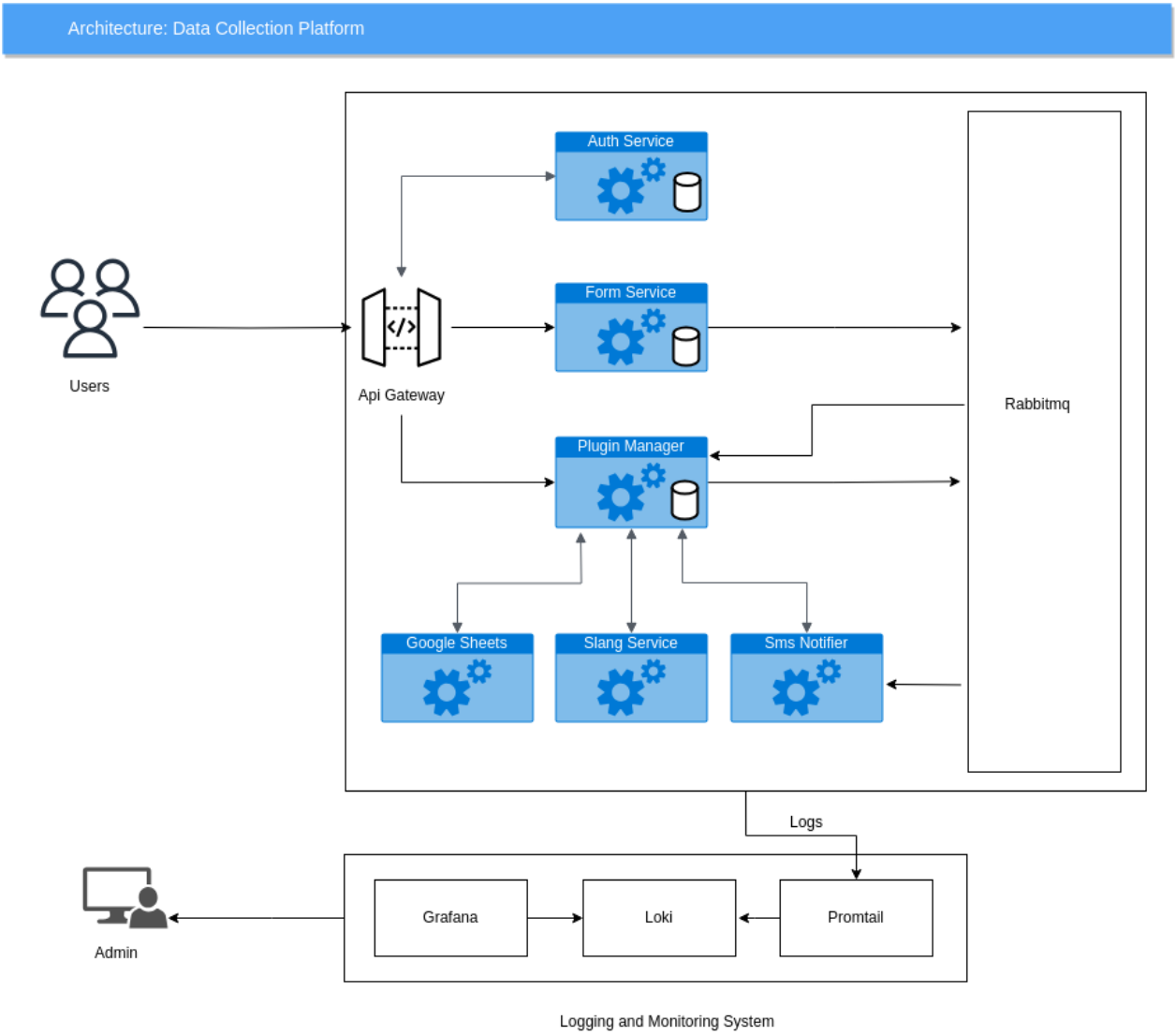
README

Design Document

Introduction and Overview

This data collection backend platform uses **microservices** architecture and has a easy and **modular plugin system** to enhance the post data ingestion functionalities.

Architecture Overview



API Request Flow

1. **User sends a request to the API Gateway.**

2. **API Gateway Routes:**
    - The API Gateway directs the request to either the Form Service, Auth Service, or Plugin Manager Service based on the request type.
3. **Authorization Check (for Form Service):**
    - If the request is for the Form Service, the API Gateway ensures user authorization by forwarding the request to the Auth Service.
    - Based on the validation result, the API Gateway proceeds to route the request to the Form Service.
4. **Event Processing:**
    - Events emitted by the Form Service are sent to RabbitMQ for efficient event queuing.
5. **Plugin Management:**
    - RabbitMQ forwards the events to the Plugin Manager Service, which efficiently routes them to the appropriate plugin for processing.
6. **Plugin Actions:**
    - Teams can perform actions on plugins, and requests for these actions are managed and forwarded by the Plugin Manager Service.

Technology Stack

- **Microservices** - Golang Gin Framework
- **Database** - Postgresql
- **Message queue** - RabbitMQ
- **Log aggregation** - Loki
- **Alerts and Monitoring** - Grafana

Microservices Design

1. **API Gateway**
    - Routes API requests to respective services with necessary authorization.
2. **Form Service**
    - Manages form-related requests: **creation**, **submission**, and **event emission** (e.g., response-submission).
3. **Auth Service**
    - Handles authentication-related requests: **login**, **registration**, and **validation**.

4. **Plugin Manager Service**
    - Manages plugins, configurations, actions, and event routing.
    - Routes events from core services to the appropriate plugin queue.
    - Monitors plugin health by polling the `/health` endpoint.
5. **Plugins**
    - **Google Sheets**
        - Enables form **export** functionality.
    - **SMS**
        - Sends response submission messages to users.
    - **Slang**
        - Facilitates local language slang searches.

Plugin Architecture

Each plugin, treated as a microservice, exhibits the following functionalities:

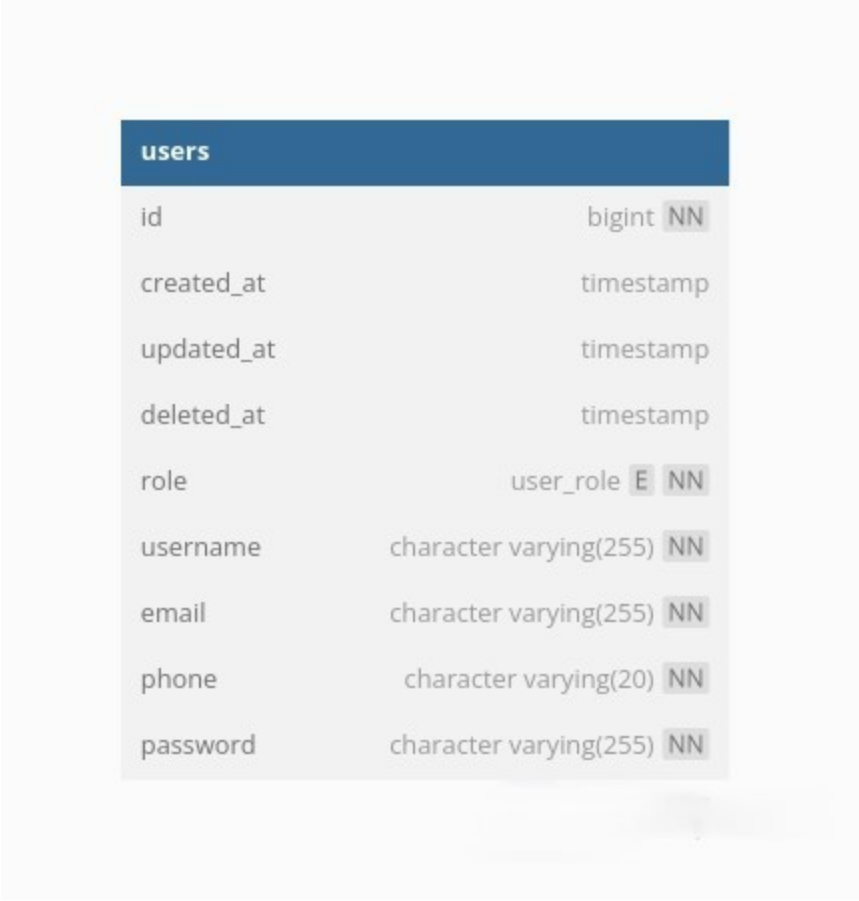
- Listens to **events** from the core platform and executes specific functionality in response.
- Receives **actions** from form management teams via the `/action` endpoint to perform designated actions.
- **Registers** itself with the manager service to confirm its operational status.
- Offers a `/health` endpoint to provide its status.
- Provides a `/configure` endpoint for plugin specific configuration.

To simplify the creation of plugin services, a dedicated `plugin-server` folder contains the necessary packages.

Database Design

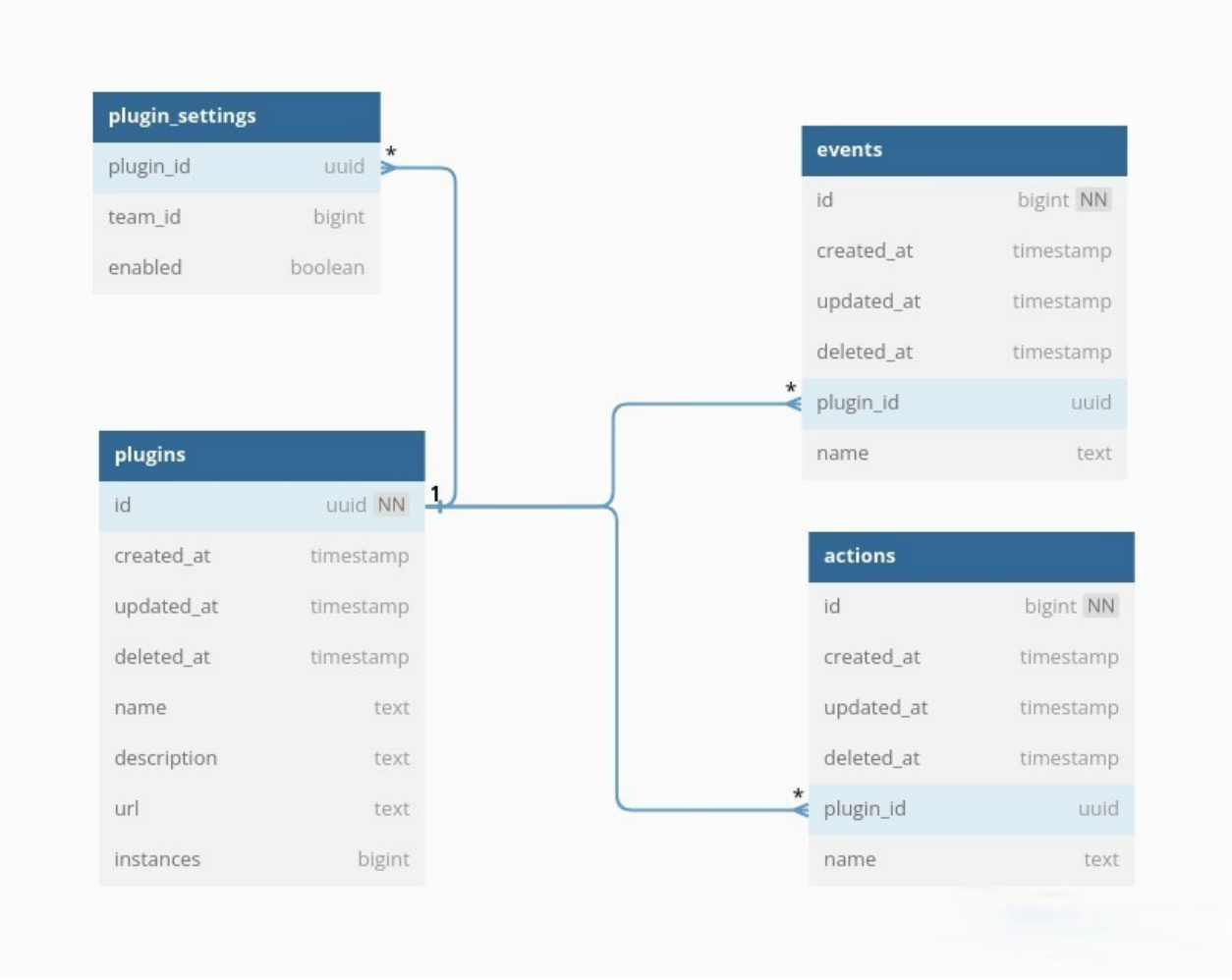
There are three main database for the core platform :

Auth Database



Form Database

Plugin Manager Database



API Design

- **Form Service API Endpoint**

  External Endpoints

  - **Create form**

    ```
    POST /api/v1/form/
    ```

    Description: This API endpoint is used to create forms by team.

    Request Format:

    ```
    {
      "title": "Health and Lifestyle Survey",
      "description": "Survey to know health",
      "questions": [
        {
          "type": "text",
          "text": "What is your full name?",
    ```

```
      "required": true
    },
    {
      "type": "text",
      "text": "How old are you?",
      "required": true
    },
    {
      "type": "radio",
      "text": "Do you exercise regularly?",
      "options": ["Yes", "No"]
    },
    {
      "type": "text",
      "text": "How many hours of sleep do you typically get per nig
      "required": true
    },
    {
      "type": "checkbox",
      "text": "Select your dietary preferences:",
      "options": ["Vegetarian", "Vegan", "Omnivore", "Other"]
    },
    {
      "type": "text",
      "text": "Do you have any known allergies or medical condition
    },
    {
      "type": "text",
      "text": "On average, how many glasses of water do you drink p
      "required": true
    }
  ]
}
```

Response Format:

```
{
  "form_id": 11,
  "message": "Form created successfully",
  "status": "success"
}
```

Headers:

- `Content-Type: application/json`

- `Authorization: Bearer <token>`

Notes:

- Access: team

- **View form**

```
GET /api/v1/form/<form-id>
```

Description: This API endpoint is used to view forms.

Response Format:


```
{
  "description": "Survey to know health",
  "id": 1,
  "questions": [
    {
      "id": 1,
      "required": true,
      "text": "What is your full name?",
      "type": "text"
    },
    {
      "id": 2,
      "required": true,
      "text": "How old are you?",
      "type": "text"
    },
    {
      "id": 3,
      "options": ["Yes", "No"],
      "text": "Do you exercise regularly?",
      "type": "radio"
    },
    {
      "id": 4,
      "required": true,
      "text": "How many hours of sleep do you typically get per nig
      "type": "text"
    },
    {
      "id": 5,
      "options": ["Vegetarian", "Vegan", "Omnivore", "Other"],
      "text": "Select your dietary preferences:",
      "type": "checkbox"
    },
```

```
      {
        "id": 6,
        "text": "Do you have any known allergies or medical condition
        "type": "text"
      },
      {
        "id": 7,
        "required": true,
        "text": "On average, how many glasses of water do you drink p
        "type": "text"
      }
    ],
    "title": "Health and Lifestyle Survey"
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: user, team.

- **Submit response**

```
POST /api/v1/form/responses
```

Description: This API endpoint is used to submit response by user for a form.

Request Format:

```
{
  "form_id": 1,
  "answers": [
    {
      "question_id": 1,
      "answer": {
        "type": "text",
        "value": "Alice Smith"
      }
    },
    {
      "question_id": 2,
      "answer": {
        "type": "text",
        "value": "28"
```

```
      }
    },
    {
      "question_id": 3,
      "answer": {
        "type": "radio",
        "value": 1
      }
    },
    {
      "question_id": 4,
      "answer": {
        "type": "text",
        "value": "7 hours"
      }
    },
    {
      "question_id": 5,
      "answer": {
        "type": "checkbox",
        "value": [0, 1]
      }
    },
    {
      "question_id": 6,
      "answer": {
        "type": "text",
        "value": "None"
      }
    },
    {
      "question_id": 7,
      "answer": {
        "type": "text",
        "value": "8 glasses"
      }
    }
  ]
}
```

Response Format:

```
{
  "message": "Response submitted successfully",
  "response_id": 63,
```

```
  "status": "success"
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: user

- **View response**

```
GET /api/v1/form/responses/<response-id>
```

Description: This API endpoint is used to view an individual response for a form.

Response Format:

```
{
  "form": {
    "description": "Survey to know health",
    "id": 1,
    "questions": [
      {
        "answer": "Alice Smith",
        "id": 1,
        "text": "What is your full name?",
        "type": "text"
      },
      {
        "answer": "28",
        "id": 2,
        "text": "How old are you?",
        "type": "text"
      },
      {
        "answer": "No",
        "id": 3,
        "options": ["Yes", "No"],
        "text": "Do you exercise regularly?",
        "type": "radio"
      },
      {
        "answer": "7 hours",
```

```
            "id": 4,
            "text": "How many hours of sleep do you typically get per n
            "type": "text"
          },
          {
            "answer": ["Vegetarian", "Vegan"],
            "id": 5,
            "options": ["Vegetarian", "Vegan", "Omnivore", "Other"],
            "text": "Select your dietary preferences:",
            "type": "checkbox"
          },
          {
            "answer": "None",
            "id": 6,
            "text": "Do you have any known allergies or medical conditi
            "type": "text"
          },
          {
            "answer": "8 glasses",
            "id": 7,
            "text": "On average, how many glasses of water do you drink
            "type": "text"
          }
        ],
        "title": "Health and Lifestyle Survey"
      },
      "id": 1,
      "submission_time": "2023-09-07T17:08:48.833847Z",
      "user_id": 9
    }
```


Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: user, team

- **View all responses**

```
GET /api/v1/form/<form-id>/responses
```

Description: This API endpoint is used to view all responses for a form.

Response Format:

```
{
  "form_id": 1,
  "title": "Health and Lifestyle Survey",
  "description": "Survey to know health",
  "questions": [
    {
      "id": 1,
      "value": "What is your full name?"
    },
    {
      "id": 2,
      "value": "How old are you?"
    },
    {
      "id": 3,
      "value": "Do you exercise regularly?"
    },
    {
      "id": 4,
      "value": "How many hours of sleep do you typically get per ni
    },
    {
      "id": 5,
      "value": "Select your dietary preferences:"
    },
    {
      "id": 6,
      "value": "Do you have any known allergies or medical conditio
    },
    {
      "id": 7,
      "value": "On average, how many glasses of water do you drink
    }
  ],
  "responses": [
    {
      "user_id": 4,
      "answers": [
        {
          "question_id": 1,
          "value": "Sarah Smith"
        },
        {
          "question_id": 2,
          "value": "30"
        },
        {
```

```
        "question_id": 3,
        "value": "No"
      },
      {
        "question_id": 4,
        "value": "8 hours"
      },
      {
        "question_id": 5,
        "value": ["Omnivore"]
      },
      {
        "question_id": 6,
        "value": "None"
      },
      {
        "question_id": 7,
        "value": "10 glasses"
      }
    ]
  },
  {
    "user_id": 5,
    "answers": [
      {
        "question_id": 1,
        "value": "Mike Brown"
      },
      {
        "question_id": 2,
        "value": "35"
      },
      {
        "question_id": 3,
        "value": "No"
      },
      {
        "question_id": 4,
        "value": "6 hours"
      },
      {
        "question_id": 5,
        "value": ["Other"]
      },
      {
        "question_id": 6,
        "value": "Seasonal allergies"
      },
```

```
      {
        "question_id": 7,
        "value": "6 glasses"
      }
    ]
  },
  {
    "user_id": 6,
    "answers": [
      {
        "question_id": 1,
        "value": "Emily Jones"
      },
      {
        "question_id": 2,
        "value": "25"
      },
      {
        "question_id": 3,
        "value": "Yes"
      },
      {
        "question_id": 4,
        "value": "7.5 hours"
      },
      {
        "question_id": 5,
        "value": ["Vegetarian", "Vegan"]
      },
      {
        "question_id": 6,
        "value": "None"
      },
      {
        "question_id": 7,
        "value": "9 glasses"
      }
    ]
  },
  {
    "user_id": 7,
    "answers": [
      {
        "question_id": 1,
        "value": "Alex Wilson"
      },
      {
        "question_id": 2,
```

```
            "value": "40"
          },
          {
            "question_id": 3,
            "value": "No"
          },
          {
            "question_id": 4,
            "value": "7 hours"
          },
          {
            "question_id": 5,
            "value": ["Omnivore"]
          },
          {
            "question_id": 6,
            "value": "High blood pressure"
          },
          {
            "question_id": 7,
            "value": "7 glasses"
          }
        ]
      }
    ]
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: team

Internal Endpoints

It doesn't need any authorisation and are only called by internal services.

- **View all responses**

  `GET /<form-id>/responses`

  Description:

This API endpoint is used to view all responses for a form. This is internally used by google sheets service to export all the responses.

Everything same as the previous request.

- **Get answer for a question**

```
GET /<form-id>/responses?question_id=<question-id>&response_id=
```

Description: This API endpoint is used to get a particular text answer for a question id and a response id. This is used by the slang plugin service internally.

Response Format

```
{
  "value": "8 glasses"
}
```

Notes:

- Only works for text answer

- **Auth Service API**

  External Endpoints

  - **Register**

```
POST /api/v1/auth/register
```

Description: This API endpoint is used to register a user or team.

Request Format:

```
{
  "username": "gavin_hooly",
  "password": "secret123",
  "email": "gavin@hooly.com",
  "phone": "1234567890",
  "role": "user"
}
```

Response Format:

```
{
  "message": "User registered successfully",
  "user": {
    "ID": 14,
    "CreatedAt": "2023-09-16T18:31:55.581952702Z",
    "UpdatedAt": "2023-09-16T18:31:55.581952702Z",
    "DeletedAt": null,
    "role": "user",
    "username": "gavin_hooly",
    "email": "gavin@hooly.com",
    "phone": "1234567890"
  }
}
```

Headers:

- `Content-Type: application/json`

- **Login**

```
POST /api/v1/auth/login
```

Description: This API endpoint is used to login a user or team.

Request Format:

```
{
  "username": "gavin_hooly",
  "password": "secret123"
}
```

Response Format:

```
{
  "message": "Login successful",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlYXQiOjE2OTQ5Nz
  "user": {
    "ID": 14,
    "CreatedAt": "2023-09-16T18:31:55.581952Z",
    "UpdatedAt": "2023-09-16T18:31:55.581952Z",
    "DeletedAt": null,
    "role": "user",
    "username": "gavin_hooly",
    "email": "gavin@hooly.com",
    "phone": "1234567890"
```

```
  }
}
```

Headers:

- `Content-Type: application/json`

- **Validate**

```
GET /api/v1/auth/validate
```

Description: This API endpoint is used to validate the auth token.

Response Format:

```
{
  "claims": {
    "eat": 1694975624,
    "iat": 1694889224,
    "id": 14,
    "role": "user"
  },
  "message": "Token is valid"
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Response Headers:

- `X-Id: <user-id>`
- `X-Role: <user-role>`

Internal Endpoints

- **Get user details**

```
GET /users/<user-id>
```

Description: This API endpoint is used to user details. This is internally used by sms service to get the users phone number for sending sms.

Response Format:

```
{
  "ID": 10,
  "CreatedAt": "2023-09-07T14:52:45.37813Z",
  "UpdatedAt": "2023-09-07T14:52:45.37813Z",
  "DeletedAt": null,
  "role": "team",
  "username": "sophieclark44",
  "email": "sophieclark44@example.com",
  "phone": "+2223334444"
}
```


Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

- **Plugin Manager Service API**

  External Endpoint

  - **Get all plugin details**

    ```
    GET /api/v1/plugins/
    ```

    Description: This API endpoint is used to get all the plugin details.

    Response Format:


```
[
  {
    "id": "38a9f989-bb24-590b-ad3b-ed047a1065d7",
    "name": "google-sheets-exporter",
    "description": "Google Sheets Exporter",
    "events": null,
    "actions": ["export"]
  },
  {
    "id": "00635020-9d4b-5f0b-a2c1-a2dcf5fe2935",
    "name": "Sms notifier",
    "description": "Sms notifier on correct data ingestion",
    "events": ["response-submission"],
    "actions": null
  },
  {
    "id": "61f2a431-af54-50d9-8024-83f9fd6b60fe",
    "name": "Slang Finder",
```

```
    "description": "Finds slang for an answer",
    "events": null,
    "actions": ["slang"]
  }
]
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: team.

- **Get a plugin's details**

```
GET /api/v1/plugins/<plugin-id>
```

Description: This API endpoint is used to get a particular the plugin details.

Response Format:

```
{
  "id": "38a9f989-bb24-590b-ad3b-ed047a1065d7",
  "name": "google-sheets-exporter",
  "description": "Google Sheets Exporter",
  "events": null,
  "actions": ["export"]
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: team.

- **Get a plugin's settings**

```
GET /api/v1/plugins/<plugin-id>/settings
```

Description: This API endpoint is used to get a particular the plugin settings.

Response Format:

```
{
  "plugin_id": "38a9f989-bb24-590b-ad3b-ed047a1065d7",
  "team_id": 10,
  "enabled": true
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: team.

- **Update the plugin status**

```
POST /api/v1/plugins/<plugin-id>/status
```

Description: This API endpoint is used to update the plugin status.

Request Format:

```
{
  "enabled": true
}
```

Response Format:

```
{
  "message": "Plugin status updated",
  "plugin": {
    "plugin_id": "61f2a431-af54-50d9-8024-83f9fd6b60fe",
    "team_id": 12,
    "enabled": true
  }
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: team.

- **Configure the plugin specific setting**

```
POST /api/v1/plugins/<plugin-id>/configure
```

Description: This API endpoint is used to update the plugin specific setting for that team.

Request Format:

Depends on the particular plugin configuration. Right now there is no plugin using this endpoint.

Response Format:

```
{
  "message": "Plugin configured successfully"
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: team.

- **Do some action of the plugin**

```
POST /api/v1/plugins/<plugin-id>/actions/<action-name>
```

Description: This API endpoint is used to do some action provided by the plugin. For example, **google sheets** plugin has **export** action, **slang** service has **slang** action to search for the slangs.

Examples:

- **Google sheets plugin**

```
POST /api/v1/plugins/38a9f989-bb24-590b-ad3b-ed047a1065d7/actio
```

Request Format:

```
{
  "form_id": 1
}
```

Response Format:

```
{
  "message": "export executed successfully",
  "result": "https://docs.google.com/spreadsheets/d/1LEATkVcpmg
}
```

- **Slang service**

```
POST /api/v1/plugins/61f2a431-af54-50d9-8024-83f9fd6b60fe/actio
```

Request Format:

```
{
  "form_id": 12,
  "response_id": 76,
  "question_id_for_city": 47,
  "question_id_for_slang": 53
}
```

Response Format:

```
{
  "message": "slang executed successfully",
  "result": "ನಮ್ಮ ಪೀಳಿಗೆಯವರು ಬಳಸುವ ಕೆಲವು ಜನಪ್ರಿಯ ಗ್ರಾಮ್ಯ ಪದಗಳು/ಪದಗಳ
}
```

Headers:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`

Notes:

- Access: team.

Internal Endpoint

- **Register a plugin**

```
POST /register
```

Description: This API endpoint is used to register a plugin with the plugin manager.

Request Format:

```
{
  "id": "38a9f989-bb24-590b-ad3b-ed047a1065d7",
  "name": "google-sheets-exporter",
  "description": "Google Sheets Exporter",
  "url": "http://google-sheets-service",
  "events": null,
  "actions": ["export"]
}
```

Response Format:

```
{
  "message": "Plugin registered",
  "plugin": {
    "ID": "38a9f989-bb24-590b-ad3b-ed047a1065d7",
    "CreatedAt": "2022-09-04T08:13:55.000Z",
    "UpdatedAt": "2022-09-04T08:13:55.000Z",
    "DeletedAt": null,
    "name": "google-sheets-exporter",
    "description": "Google Sheets Exporter",
    "url": "http://google-sheets-service",
    "events": null,
    "actions": ["export"],
    "instances": 1
  }
}
```

Headers:

- `Content-Type: application/json`

Events Design

Event flow:

- The core service(right now only form service) emits an event `response-submission` on each user response submission. The data structure which is generalised is:

```
{
  "event": "response-submission",
  "team_id": 10,
  "data": {
    "form_id": 1,
    "user_id": 1,
    "title": "Health and Lifestyle Survey",
    "description": "Health and Lifestyle Survey"
  }
}
```

  Here, the `event`, `team_id` is the main keys for the event data which would be useful for routing it to the appropriate plugin.

- This event is sent to the rabbitmq from the form service to the exchange named `events` and routing key named `events`. This is consumed by the plugin manager service.

- Plugin manager gets the message packet and routes it according to two conditions:

  i. Which plugins are subscribed to this event.
  ii. Is this plugin enabled by the team id.

- The plugin manager sends it to the `manager` exchange and send to the routing key named `<plugin-id>`

Logging

This system uses **Promtail**, **Loki**, and **Grafana** for logging and log aggregation. This setup will enable efficient log collection, storage, querying, and visualization of Docker logs.

Components:

1. **Promtail:**
   - Promtail is an agent that collects and tail logs, sending them to Loki.
   - Configure Promtail to scrape Docker container logs.
2. **Loki:**
   - Loki is a horizontally-scalable, highly available log aggregation system.

- It stores logs in a way that allows for efficient querying while minimizing costs.

3. **Grafana:**

   - Grafana is a popular open-source platform for monitoring and observability.
   - Integrate Grafana with Loki to visualize logs and create dashboards.

Logging Flow:

1. Docker container logs are written to stdout/stderr within the container.
2. Promtail tail Docker logs and forwards them to Loki using the specified configuration.
3. Loki stores the logs in a format optimized for query performance and efficient storage.
4. Grafana queries Loki to display logs in visualizations and dashboards for monitoring and troubleshooting.