Architecture Document

Retrieving and Analyzing GitHub Repositories and Dependencies

### Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Author |
| 1.0 | 3/15/2023 | Initial architecture document | Lokesh Agnihotri |
| 1.2 | 3/16/2023 | Updated the document |

## Abstract:

This architecture document outlines a project that retrieves all dependencies and sub-dependencies for a given GitHub user, processes the data in Python to create a JSON file, and sends the data to a Splunk Cloud instance for visualization. The project consists of three main components - Data Retrieval, Data Processing, and Data Visualization - each with their own sub-components. The document describes the workflow and technologies used in the project and includes a version table. This document serves as a guide for understanding the architecture of the project and its various components, as well as the reason for the technology choices.

Contents

[Version 1](#_Toc129868676)

[Abstract: 1](#_Toc129868677)

[1. Introduction 3](#_Toc129868678)

[2. Architecture Overview 4](#_Toc129868679)

[3. Components 4](#_Toc129868680)

[3.1 GitHub API 4](#_Toc129868681)

[3.2 Python Script 4](#_Toc129868682)

[3.3 Splunk 5](#_Toc129868683)

[5. Why Splunk and Python? 5](#_Toc129868684)

[4. Workflow 6](#_Toc129868685)

[5. Technologies Used 6](#_Toc129868686)

[6. Version Table 7](#_Toc129868687)

[Component Version 7](#_Toc129868688)

[7. Conclusion 7](#_Toc129868689)

# 1. Introduction

This document describes the architecture for a system that retrieves a list of GitHub repositories and their dependencies for a given user and analyzes the data. The system is designed to automate the process of collecting and analyzing data, making it easy to keep track of the dependencies in multiple repositories and ensure they are up to date.

The architecture consists of multiple components that work together to retrieve the repositories and dependencies, analyze the data, and store the results. The components include the GitHub API, Splunk, and a custom Python script.

# 2. Architecture Overview

The system is designed to retrieve a list of GitHub repositories and their dependencies for a given user and send the data to Splunk using HTTP Event Collector. The system consists of three main components:

GitHub API: The GitHub API is used to retrieve information about the user's repositories and their dependencies.

Python script: A Python script is used to call the GitHub API and extract the required data. The script also uses the Splunk HTTP Event Collector library to send the data to Splunk.

Splunk: Splunk is used as the central data store for the data retrieved from GitHub.

# 3. Components

## 3.1 GitHub API

The GitHub API is a RESTful API that can be used to retrieve information about repositories and their contents. In this system, we are using the API to retrieve information about a user's repositories and the contents of their requirements.txt file.To use the GitHub API, we need to authenticate with GitHub using our credentials. In this system, we are using Basic Authentication with a username and a personal access token (PAT). The PAT is stored as an encrypted environment variable in the Python script to ensure security.

## 3.2 Python Script

The Python script is responsible for calling the GitHub API, extracting the required data, and sending it to Splunk. The script uses the requests library to make API requests and the Splunk HTTP Event Collector library to send data to Splunk. The script takes a single command-line argument, which is the GitHub username for which the repositories need to be retrieved. The username is passed to the API endpoint to retrieve the list of repositories. For each repository, the script makes another API request to retrieve the contents of the requirements.txt file. If the file exists, the script extracts the dependencies and adds them to a list along with the repository name. This list is then sent to Splunk as a JSON object.

## 3.3 Splunk

Splunk is a powerful data analytics and visualization tool that is widely used for log management, security, and IT operations. In this system, we are using Splunk as the central data store for the data retrieved from GitHub.

The data sent to Splunk is stored in an index named "main" and is tagged with a **source type** of "github\_repositories" and a source of "github". This allows us to easily search and filter the data in Splunk.

# 5. Why Splunk and Python?

GitHub API: The GitHub API is a well-documented and widely used API that provides easy access to information about repositories and their contents. It is also highly customizable, allowing us to retrieve exactly the data we need.

Python: Python is a popular programming language that is easy to learn and has a wide range of libraries and frameworks available. In this system, we are using the requests library for making API requests and the Splunk HTTP Event Collector library for sending data to Splunk. These libraries are well-documented and easy to use.

Splunk: Splunk is a powerful tool for log management and data analytics. It provides a centralized location for storing and analyzing data, making it easy to search and filter data from multiple sources. It also provides powerful visualization tools for creating dashboards and reports. Also, Splunk is capable to index huge amount of data, and of many types, be it machine date or logs or Json file, csv. Thinking about future I think Splunk is the best choice in case company wants to expand the dashboards and visualizations by including codebase data, Splunk will be easily able to cater to that need.

# 4. Workflow

The workflow for this project is as follows:

The Data Retrieval component retrieves all dependencies and sub-dependencies for a given GitHub user.

The Data Processing component processes the retrieved data to create a JSON file.

The Data Visualization component sends the JSON file to a Splunk Cloud instance for visualization.

# 5. Technologies Used

The following technologies are used in this project:

Python: Used for data processing and JSON creation.

Github API: Used for data retrieval.

Splunk Cloud: Used for data visualization.

# 6. Version Table

The following table shows the versions of the components used in this project:

## Component Version

* Python 3.10
* GitHub API v3
* Splunk Cloud 8.2.1

# 7. Conclusion

This document has outlined the architecture for a project that retrieves all dependencies and sub-dependencies for a given GitHub user, processes the data to create a JSON file, and sends the data to a Splunk Cloud instance for visualization. The project involves three main components - Data Retrieval, Data Processing, and Data Visualization - each with their own sub-components. The workflow and technologies used in the project have also been described.