A multidisciplinary working system was followed in the process of creating the Knowledge Graph. According to this, the project aims to create knowledge graphs by pulling data from a specific GitHub repository using GitHub REST API v3 and Matplotlib libraries. The Knowledge Graph creation project includes the following main steps:

* Pulling data from repository using Github REST API v3
* Processing and analyzing fetched data
* Visualization using Matplotlib library
* Reporting the emerging knowledge graph

**Techniques and Libraries Used**

**GitHub REST API v3:** The project used GitHub REST API v3 to retrieve data from GitHub repositories. This API is at the center of the project and the data provided by the API is processed in other steps.

**Github Personal Access Token:** The project uses a Personal Access Token to access the GitHub API. This is a security measure used to provide authentication and manage access permissions.

**Python:** The project is based on the Python programming language for data extraction, processing and visualization.

**Matplotlib:** Used for data visualization to create knowledge graphs at the end of the project.

**Networkx:** Used package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks such as edges and nodes in the knowledge graph.

**Main Steps**

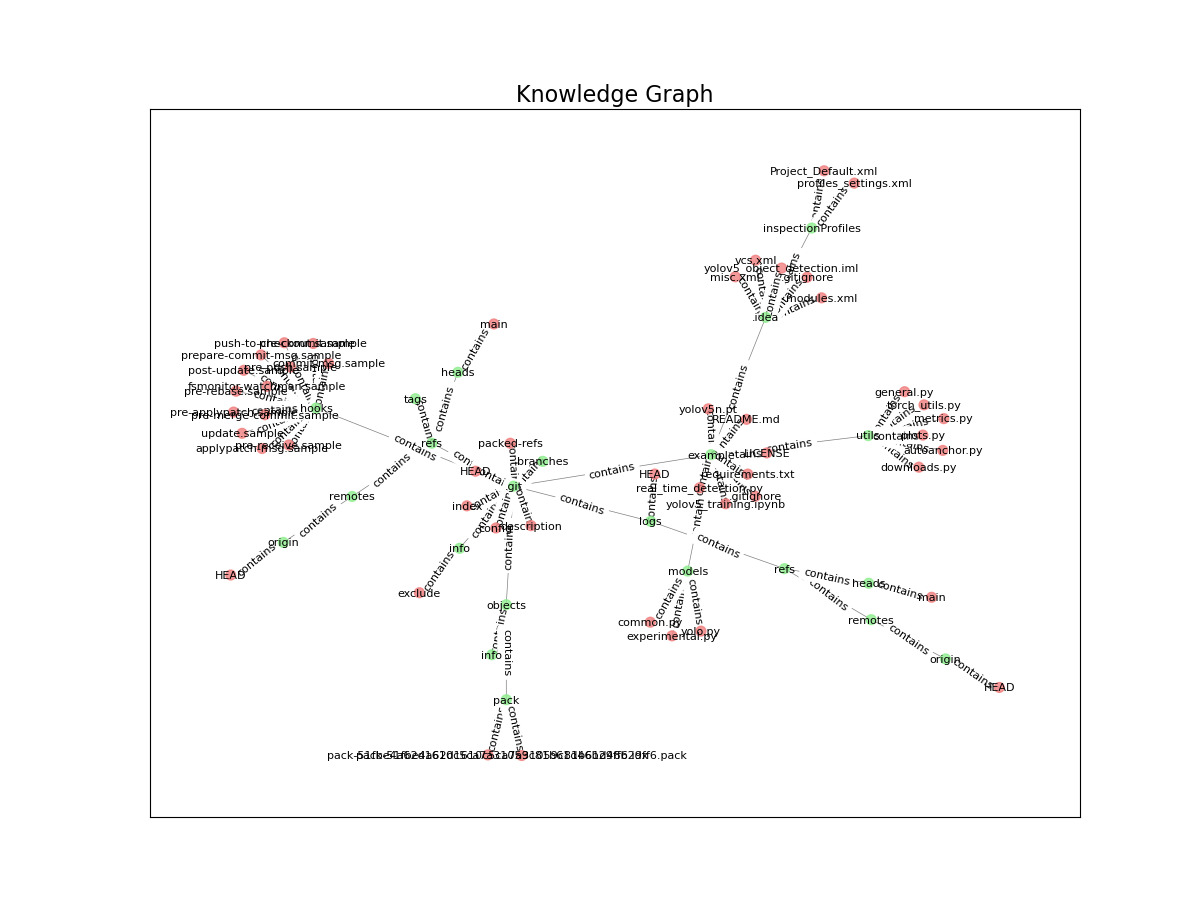
1. **Getting GitHub Repository Information:** At project startup, the GitHub API is used to pull basic information from the user-specified GitHub repository:
   * Files
   * Folders
   * Services
   * Backbone
2. **Data Analysis and Processing:** The captured data is analyzed and processed in Python. In particular, certain data may need to be selected and processed.
3. **Creating a Knowledge Graph with Matplotlib:** The processed data was represented as a graph using Matplotlib.
4. **Commissioning of Access Tools:** In order to run the main functions mentioned above, this last step aims to enter personal access tokens and other information and ensure successful data flow.

The **"akdenizz/yolov5\_object\_detection"** repository was chosen because it has various file types and folder structures of sufficient complexity for Knowledge Graph creation.

**Project Results**

The information graphic obtained as a result of the project visually represents important information about the warehouse. This chart; It has fulfilled the requirements by including how and where the tools, saved files and services used in the project were used.

**Output**



**Further Improvements**

The project has potential for future development. To further develop this project, the following improvements can be considered:

* Transferring the fetched data to a DBMS and making queries on the knowledge graph using tools such as Neo4j
* Adding the ability to pull and compare data from multiple GitHub repositories.
* Adding automatic report generation feature.

**Conclusion**

In this project, secure&private data extraction and knowledge graph creation from a GitHub repo was successfully carried out using GitHub REST API v3, Personal Access Token and Matplotlib. The project can be used to better understand and visualize the data of GitHub repositories.