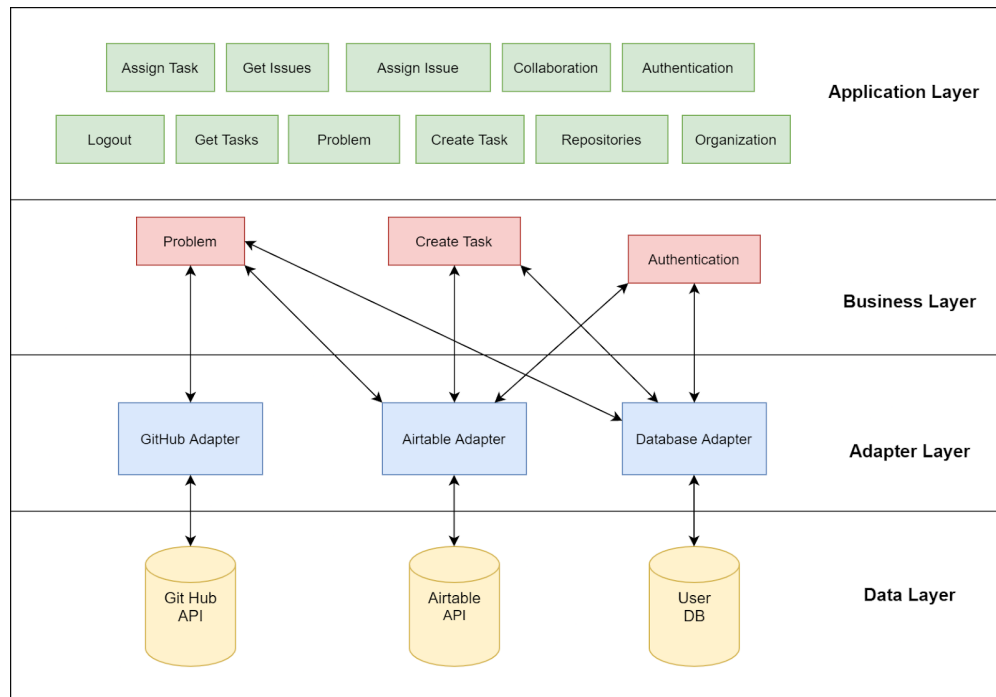




E-Agle Trento Racing Team Telegram Bot API



Overview

The division of the teams about the software development part is divided into control, sensors, and telemetry.

Each group has a project manager who takes care of achieving the goals and coordinating according to the goals selected together with the supervisors of the macro-areas.

The development of a Telegram Bot arises from the need of some of the supervisors and project managers of the sensors and telemetry area to have a tool with which it is easy to view the latest tasks added and create new ones. The tasks are based on the latest messages that are sent on the group without having to log in to in other applications to create tasks or issues in their respective repositories or kanban.

The project in question also made it possible to have the foundation of an infrastructure that can be used later by other team members to create new functionalities and automate other processes. The programming language used is Javascript with Node.JS and as a database MongoDB. The frameworks used are already part of some of the solutions developed by the members, this choice is to make the maintenance easier in the future.

The division of the application is based on several services that communicate with each other.

Telegram is used as an interface through a Bot that communicates with the endpoints of action service used to communicate directly with the other services.



The services developed are the following:

- **Telegram** as an interface to the various services.
- **Actions** which deals with managing the various calls to endpoints and returning the data to be displayed to the Telegram interface.
- **Github** to access the organization's repositories, topics, and issues.
- **Airtable** to view the information of the members access the tasks
- **User Database** to save the information of users who use the application and be able to return tokens to access the various APIs.

Telegram

The **Telegram** service communicates with other services through the action's service. From here it is possible to call various commands or actions thanks to an interface composed of buttons that facilitate the execution of the activities. The Telegram service offers parameters used in other services such as the values to make requests and the chat ID used to authenticate the user.

Git Hub

The **Github** service is used to collect information from the various groups with their repositories and issues. The endpoints expose various features including the ability to create issues for a certain repository, assign it to a user, invite the user to collaborate on a project, invite the user to join the *eagletrt* organization, and view the repositories by the group.

Airtable

The **Airtable** service collects the information of the members that are already on the cloud collaboration service and the related tasks that can be software or generic such as contact a sponsor or finish the test of a hardware component. This service is called to verify the user's groups and other information added through a form or directly added by the project manager. In the case of the telemetry group, the groups are *telemetry* and *steering wheel*. This service can create, view, and assign tasks based on the task already assigned to the group member.

User Database

The **User Database** service is used to maintain the session of the users who use the application, the users who intend to register generally have higher permissions than the other members of the group, in fact through their chat ID it is possible to carry out most of the operations. The service is called to create, update, and delete user data and verify that the user is allowed thanks to the token previously stored. The tokens are mandatory to call the various API described below.

Actions

The **Action** service takes care of the communication between the Telegram interface and the other services, enabling authentication through the userID and automating requests such as the creation of issues both on the GitHub and on Airtable.