**Star Wars API Challenge**

**Requirements:**

Using the API at : <https://swapi.co/> , determine for a given distance how many resupply stops are required for all Starships. Output is a collection of all Starships and the number of stops required.

**Analysis:**

The result is stored in a dictionary of Starship object keys and values for the number of stops. As the ship data results are paginated, the request method needs to make multiple calls to the API. Each response is parsed for the next page until none are available. On each page, the ship data results are added to a results container. This is iterated over again, using each ships properties to compute the number of stops. The following properties are required for the calculation:

* MGLT : Speed in MGLT per hour.
* Consumables : Amount of time between resupply.

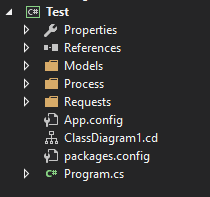
As the speed is per-hour and the time units can be in days, weeks, months or years, a mapping is used to convert before calculating the number of stops for given distance. This value-string is added to a dictionary with the ship as key. In cases where either property is unknown, a string “unknown” is stored as a value. The result object is returned to main and printed to the console.

**Unit Tests:**

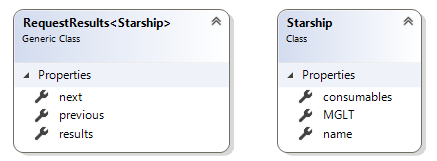
1. Assert that process method returns an object of the correct type.
2. Assert that three calculations are correctly performed for a given distance..

**Architecture**

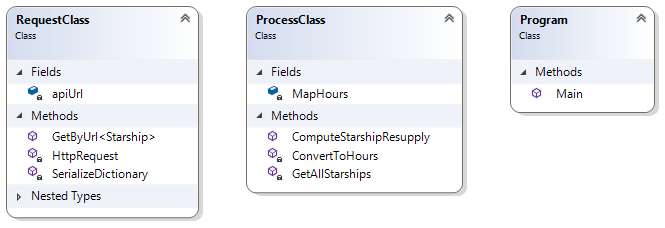
The core of the application is the Process class. It constructs the API URL’s and calls a method from the Request class. It then uses the response to compute resupply and create the result object which is passed to the Main program. Main then prints the results to the console.



**Data Models**



**Classes**



**Console UI:**

