IPO Chart

Component: Orchestration

Module Name: Service Broker

Author: Wyatt Radican
Reviewers:

Inputs	Process	Outputs
Service to be called Plist(input1)	Determines what Service is being called, finds the address of the Module, calls it, then passes the Parameters.	Outputs a string of the Parameters given to the Service Broker and sent to the called Module Plist(output1, output2, …)
A string of Parameters Plist(input2, input3, ...)		

Inputs:
A string containing the Module (Service) that is to be called and the Parameters to be passed to the Service.
Example:

Tax 2021 H 10000

In this example the "Tax" Module is trying to be called and the parameters: "2021", "H", and "10000" are being sent to it. In this case it is the year, status, and the gross income.

Process Description
Calls on desired Service by taking the first Parameter to determine what Module is trying to be called, opening the Service.txt file to determine if the Service exists and what is the address for it, and then passing the rest of the Parameters to said Module. If the Service trying to be called does not exist then an Error code is passed (703 - Service Not Found) as a Parameter. If no parameters are passed to the Service Broker an Error code is passed (512 - No Parameters Passed).

Outputs
Outputs a string containing the Parameters that is passed to the called Module (Service)
Example

2021 H 10000

In this example the Parameters listed above are being sent to a Module.

APIs/Objects:
Service.txt - Contains the list of Services and their addresses.
Tax.py - Called to find the amount of tax needed to pay.
Translate.py - Called to find a translation of an English word.
Error.py - Called to find the name of the language file to open for errors.
TextBroker.py- Called to read from text files and return the appropriate data from them.

IPO Chart

Pseudocode:

```
-Gets the Service that is being called along with the Parameters.
-Open and read the “Service.txt” file to determine if the service is valid and what is the address.
-Opens the Module and passes the Parameters to it.
-If error occurs, call Error.py to determine the error message from the Error code.
-After the last Module is called the output is printed to the screen (Terminal).
```