Luis E Perez   READ ME
Elixir 4 CSP

iex -S mix   to start module

When the requests function is executed the request actor is registered.
Upon being sent {:ok} the actor generates a request
Upon being sent {:wait, time} the actor stops generating requests for that X time

Works the same way as the CSP, froma user perspective

Function Flow

Rider.start :   it will start the agents

Rider.menu : lets you just press numbers to trigger the options (make requests, select one,
.   reset requests and exit)

Rider.register :   it will allow you to register an app with a communications channel. In order to
.   register more than 1 app, run the function as many times as apps you want
.   to add. Any amount of apps can be added, as of now

Rider.unregister({“<app>”,”<comm>”}): it is necessary to enter the name and channel in the
.   form shown above for it to actually unregister it.

Rider.requests : if apps are empty it redirects you to it. Makes and prints 1 request every 5 seconds.

Rider.menu : it lets you select the following menu options.
--------- -- -- --- -- -
1. Make requests
2. Select a service
3. Exit
4. Reset requests
5. Stop requests

Simply type the number and the function will be done or ask for for the expected parameter.

2. Lets you select a service created within the last 90 seconds that is on the list, if the index is not on the list, the service will not be selected.

4. Reset Requests Empties all requests from the list, valid and old.

5. Stop requests, stops the process but it does not wipe the request list. The created requests will remain in the list. If the request process is already stopped, then it doesn't stop it again.

Rider.reset_apps
Rider.reset_count
Rider.reset_select
Rider.reset_reqs

These functions return the Agents to their original State

Rider.currrent_apps
Rider.current_count
Rider.current_select
Rider.current_reqs

These functions return the current state of the agent in its name