|  |  |  |  |
| --- | --- | --- | --- |
| Title: | Chassis 2023 Execution Types | Created: | 2023-09-12 |
| Filename: | CH23EXET.docx | Author: | Lion Kimbro |
| Status: | In Development (2023-09-12) | Related: | CH23CORE.docx |

# Intent

This collects details about Chassis 2023 “Execution Types” – the unchanging primary mode of operation for a Chassis.

# Progress

|  |  |
| --- | --- |
| 2023-09-12 | started document; wrote up intent |
|  |  |
|  |  |
|  |  |
|  |  |

# Location

Source file: package.json

Source key: D[“CHASSIS2023”][“EXECUTIONTYPE”]

# Possible Values

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *date\** | *value* | *title* | *status* | *cli\_collect\_data ref* | *uses HOST, PORT* |  |
| *found* | CLITOOL |  | imaginary |  |  |  |
| *found* | WEBSERVER |  | imaginary |  |  |  |
| 2023-09-12 | JSONWEBSERVICE | JSON Web Service | implementing | X | X |  |
| 2023-09-12 | TCPSERVICE |  | imaginary | X | X |  |
| *found* | FILETALKSERVER |  | imaginary |  |  |  |
| *found* | TKINTERGUI |  | imaginary |  |  |  |
| *found* | INTERACTIVEMENU |  | imaginary |  |  |  |
| *found* | LIBRARY |  | imaginary |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

*(\*) “found” dates – these were entered before 2023-09-12*

# Execution Type: JSONWEBSERVICE

|  |  |
| --- | --- |
| Title: | JSON Web Service |
| Code: | JSONWEBSERVICE |
| Status: | In Development (2023-09-12) |
| Projects: | chromeurltitle (P#0103) |

## Use

As in all execution types, have a module \_\_main\_\_.py, that calls: chassis2023.chassis23.run\_package(\_\_package\_\_)

Then, also have a module handler.py in your package. The chassis2023 will find it, and call handle(data), whenever JSON data is received.

That function receives the data from the JSON, and then should return data that will be sent back as JSON.

## Development Notes

### 2023-09-12

Abstract: I was trying to figure out how to handle logging, and what logging configuration will demand of the system.

Stream of Consciousness: There’s a question of logging – I hate the logging output, typically, and just want to suppress it. So there’s the concept of a “logging” configuration – which I’d detailed how to do in P#0103, as presently implemented. It does successfully suppress log output, and it took a lot of work to figure out how to get it to shut up. How would that be specified in the code? Locations of interest: In the package.json, in the (presently non-existent) user’s configuration – which is called the user’s “session file,” or in a command line argument as an override. I would think that if I went the route of package.json, I’d just be supplying the default value, for the user’s session file. There’d be a default for the user’s session file, specifiable, w.r.t. the logging output location. OK, so, that’s the answer: the configuration of logging is kept in the session file. I haven’t written anything about the session file, yet, and I’m not going to right now: I’m focused on getting a basic program running right now.

Condensation:

* The session file will contain whether logging is suppressed or not, and if it is not, what file it will be output to. This will be overridable by the CLI. This will be a general part of all chassis 2023 servers.
* For the time being, I am hard coding nothing special about logging.