|  |  |
| --- | --- |
| Valid HRHomeStationsRequest for single existing device | |
| Description: | This tests that the resolution server will correctly respond to a request for a single device that is in the database.  The request sent to the server was: {“HRHomeStationsRequest”:[{“StationDID”:”mac”}]} |
| Expected Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”mac”,”StationIP”:”10.250.1.128”}]} |
| Actual Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”mac”,”StationIP”:”10.250.1.128”}]} |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Valid HRHomeStationsRequest for single non-existing device | |
| Description: | This tests that the resolution server will correctly respond to a request for a single device that is not in the database.  The request sent to the server was: {“HRHomeStationsRequest”:[{“StationDID”:”imaginary”}]} |
| Expected Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”imaginary”,”StationIP”:null}]} |
| Actual Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”imaginary”,”StationIP”:null}]} |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Valid HRHomeStationsRequest for multiple existing devices | |
| Description: | This tests that the resolution server will correctly respond to a request for multiple devices that are in the database.  The request sent to the server was: {“HRHomeStationsRequest”:[{“StationDID”:”mac”},{“StationDID”:”hello”}]} |
| Expected Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”mac”,”StationIP”:”10.250.1.128”}, {“StationDID”:”hello”,”StationIP”:”10.250.1.128”}]} |
| Actual Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”mac”,”StationIP”:”10.250.1.128”}, {“StationDID”:”hello”,”StationIP”:”10.250.1.128”}]} |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Valid HRHomeStationsRequest for existing and non-existing devices | |
| Description: | This tests that the resolution server will correctly respond to a request for multiple devicse that are not all in the database.  The request sent to the server was: {“HRHomeStationsRequest”:[{“StationDID”:”mac”},{“StationDID”:”imaginary”}]} |
| Expected Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”mac”,”StationIP”:”10.250.1.128”}, {“StationDID”:”imaginary”,”StationIP”:null}]} |
| Actual Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”mac”,”StationIP”:”10.250.1.128”}, {“StationDID”:”imaginary”,”StationIP”:null}]} |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Valid HRHomeStationsRequest for multiple non-existing devices | |
| Description: | This tests that the resolution server will correctly respond to a request for multiple devices that are not in the database.  The request sent to the server was: {“HRHomeStationsRequest”:[{“StationDID”:”imaginary”},{“StationDID”:”fake”}]} |
| Expected Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”imaginary”,”StationIP”:null}, {“StationDID”:”fake”,”StationIP”:null}]} |
| Actual Result: | {“HRHomeStationsReplyt”:[{“StationDID”:”mac”,”StationIP”:”10.250.1.128”}, {“StationDID”:”mac”,”StationIP”:”10.250.1.128”}]} |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Invalid JSON | |
| Description: | This test ensures that receiving bad JSON will not crash the server.  The request sent to the server was:  {“HRHomeStationsRequest”:{“StationDID”:”mac”}]} |
| Expected Result: | none |
| Actual Result: | none |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Invalid request type | |
| Description: | This test ensures that receiving an unrecognized request type will not crash the server.  The request sent to the server was:  {“HRImaginaryRequest”:{“StationDID”:”mac”}]} |
| Expected Result: | none |
| Actual Result: | none |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| HRHomeStationsRequest with no devices given | |
| Description: | This test ensures that receiving an HRHomeStationsRequest with an empty device list will not crash the server.  The request sent to the server was:  {“HRHomeStationsRequest”:[]} |
| Expected Result: | none |
| Actual Result: | none |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| HRHomeStationsRequest with no id given | |
| Description: | This test ensures that receiving an HRHomeStationsRequest without an id will not crash the server.  The request sent to the server was:  {“HRImaginaryRequest”:{“StationDID”:”}]} |
| Expected Result: | {“HRHomeStationsReplyt”:[{“StationDID”:””,”StationIP”:null}]} |
| Actual Result: | {“HRHomeStationsReplyt”:[{“StationDID”:””,”StationIP”:null}]} |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Valid HRHomeStationUpdate request | |
| Description: | This test ensures that the server will correctly respond to a valid HRHomeStationUpdate.  The request sent to the server was:  {“HRHomStationUpdate”:[{“StationDID”:”mac”}]} |
| Expected Result: | none |
| Actual Result: | none |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| HRHomeStationUpdate request with no device given | |
| Description: | This test ensures that the server will correctly respond to an HRHomeStationUpdate without a device given.  The request sent to the server was:  {“HRHomStationUpdate”:[]} |
| Expected Result: | none |
| Actual Result: | none |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| HRHomeStationUpdate request with no device id given | |
| Description: | This test ensures that the server will correctly respond to an HRHomeStationUpdate with an empty device id.  The request sent to the server was:  {“HRHomStationUpdate”:[{“StationDID”:”]} |
| Expected Result: | none |
| Actual Result: | none |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Finder.find with device in database | |
| Description: | This tests if the Finder class correctly retrieves the ip address for a device in the database.  The argument passed to Finder.find was “mac” |
| Expected Result: | “10.250.1.128” |
| Actual Result: | “10.250.1.128” |
| Pass/Fail: | Pass |
| Finder.find with device not in database | |
| Description: | This tests if the Finder class correctly retrieves the ip address for a device not in the database.  The argument passed to Finder.find was “imaginary” |
| Expected Result: | None |
| Actual Result: | None |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Finder.find with invalid device id | |
| Description: | This tests if the Finder class correctly retrieves the ip address for an invalid device id.  The argument passed to Finder.find were None |
| Expected Result: | None |
| Actual Result: | None |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Finder.update with existing device id and up to date IP | |
| Description: | This tests if the Finder class correctly handles an update request with an existing device with an up to date IP address.  The argument passed to Finder.find were “mac” and “10.250.1.128” |
| Expected Result: | Do nothing |
| Actual Result: | Do nothing |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Finder.update with existing device id and new IP | |
| Description: | This tests if the Finder class correctly handles an update request with an existing device with a new IP address.  The argument passed to Finder.find were “hello” and “10.250.1.128” |
| Expected Result: | Update database with new IP address |
| Actual Result: | Updated database with new IP address |
| Pass/Fail: | Pass |

|  |  |
| --- | --- |
| Finder.update with existing new id and new IP | |
| Description: | This tests if the Finder class correctly handles an update request with a new device  The argument passed to Finder.find were “aNewID” and “10.250.1.128” |
| Expected Result: | Update database with new device and IP address |
| Actual Result: | Updated database with new device and IP address |
| Pass/Fail: | Pass |