Lua Workshop

By Paul Le

Overview

1. Introduction
2. Error Handling in Lua
3. Logging in Lua
4. Preventing memory leaks in Lua
5. Walk-through
6. Q & A

What to Expect

1. What is it
2. Why is it important
3. Real world application

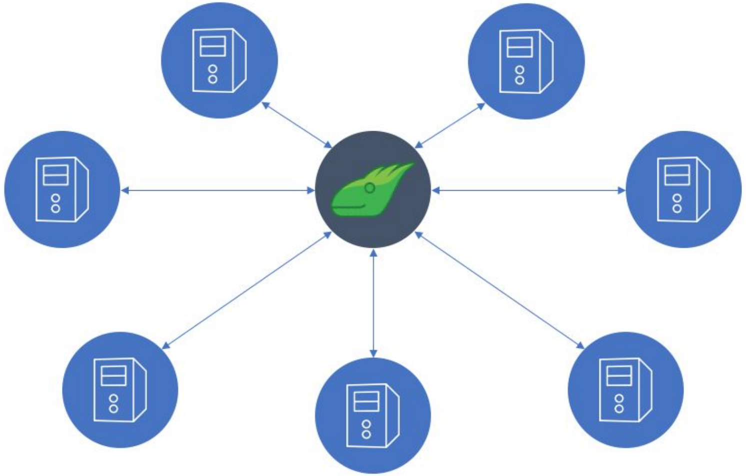
Introduction

Who Are We

- Integration company with a 20+ year history of excellence
- We develop the Iguana middleware integration engine

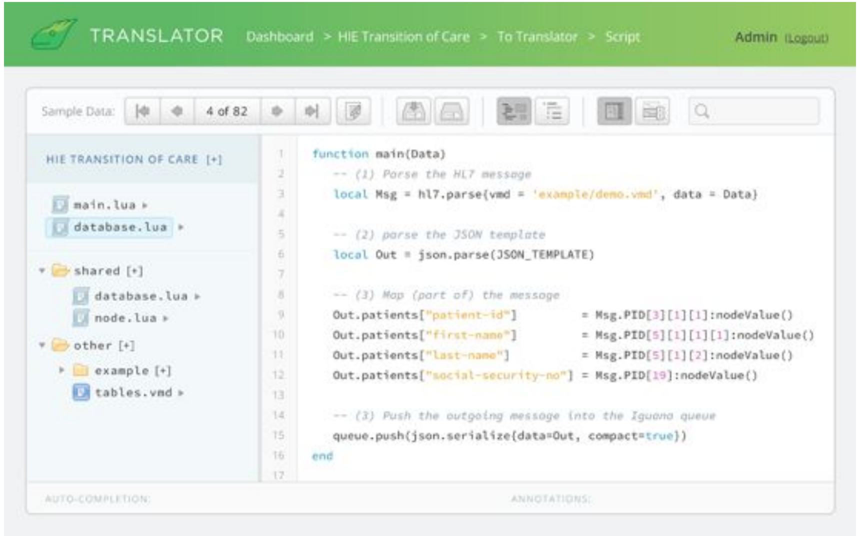
What We Do

- Work with healthcare providers and software vendors
- We provide rapid, reliable, and scalable interoperability solutions
- Move data from point A to B



The Iguana Translator

- Patented development environment
- Provides a Lua scripting environment
- Perform customizable filtering and message transformation

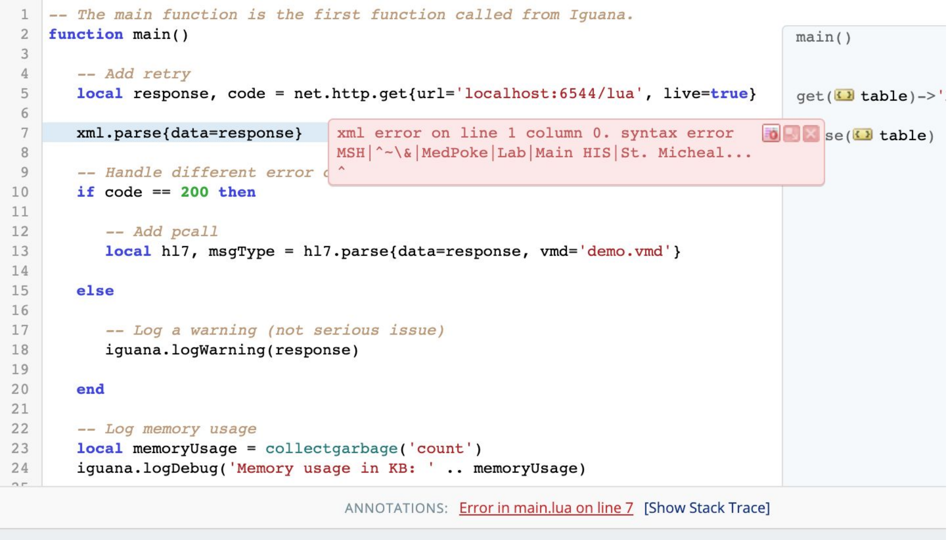


```lua
function main(Data)
   -- (1) Parse the HL7 message
   local Msg = hl7.parse{vmd = 'example/demo.vmd', data = Data}

   -- (2) parse the JSON template
   local Out = json.parse(JSON_TEMPLATE)

   -- (3) Map (part of) the message
   Out.patients["patient-id"]          = Msg.PID[3][1][1]:nodeValue()
   Out.patients["first-name"]          = Msg.PID[5][1][1][1]:nodeValue()
   Out.patients["last-name"]           = Msg.PID[5][1][2]:nodeValue()
   Out.patients["social-security-no"] = Msg.PID[19]:nodeValue()

   -- (3) Push the outgoing message into the Iguana queue
   queue.push(json.serialize{data=Out, compact=true})
end
```

Error Handling in Lua

Error Handling in Lua

- Lua script errors will cause a channel to stop
- Errors should be accounted for and gracefully handled
- Behaviour should not be unexpected



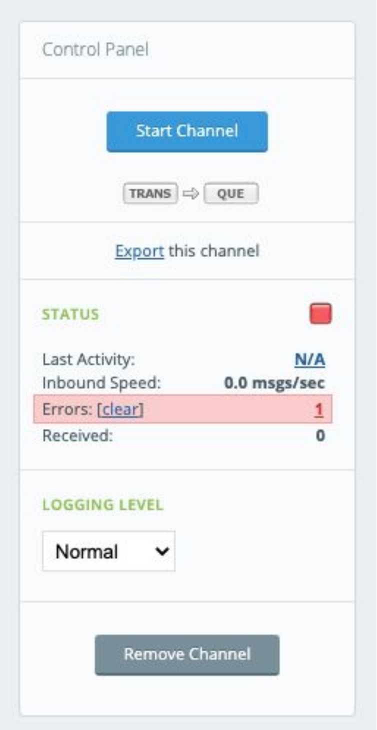
```lua
 1  -- The main function is the first function called from Iguana.
 2  function main()
 3
 4     -- Add retry
 5     local response, code = net.http.get{url='localhost:6544/lua', live=true}
 6
 7     xml.parse{data=response}
 8
 9     -- Handle different error c
10     if code == 200 then
11
12        -- Add pcall
13        local hl7, msgType = hl7.parse{data=response, vmd='demo.vmd'}
14
15     else
16
17        -- Log a warning (not serious issue)
18        iguana.logWarning(response)
19
20     end
21
22     -- Log memory usage
23     local memoryUsage = collectgarbage('count')
24     iguana.logDebug('Memory usage in KB: ' .. memoryUsage)
```

```
xml error on line 1 column 0. syntax error
MSH|^~\&|MedPoke|Lab|Main HIS|St. Micheal...
^
```

ANNOTATIONS: Error in main.lua on line 7 [Show Stack Trace]

Error Handling in Lua

- Lua script errors will cause a channel to stop
- Errors should be accounted for and gracefully handled
- Behaviour should not be unexpected



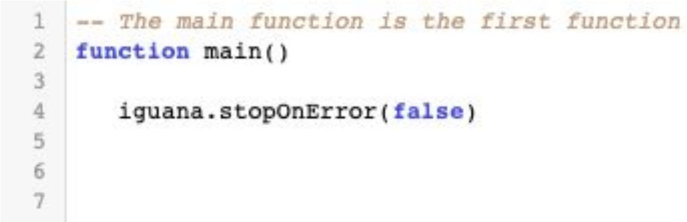
Last Activity: N/A
Inbound Speed: 0.0 msgs/sec
Errors: [clear] 1
Received: 0

LOGGING LEVEL: Normal

Gracefully Capture Errors

iguana.stopOnError

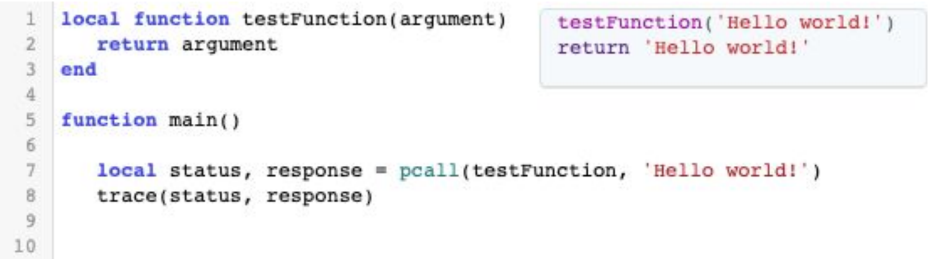
- Enables/disables stopping the channel when errors occur
- Generally not recommended
- Sometimes stopping a channel on error is desired



```
-- The main function is the first function
function main()

   iguana.stopOnError(false)



```

pcall

- Protected call
- Catches errors and returns a status code
- Allows for complete control on error handling



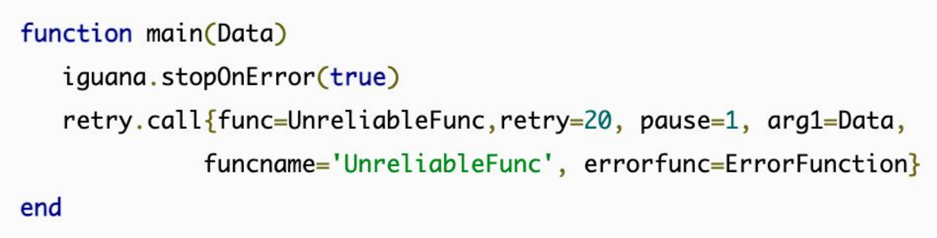
```lua
local function testFunction(argument)
   return argument
end

function main()

   local status, response = pcall(testFunction, 'Hello world!')
   trace(status, response)


```

```
testFunction('Hello world!')
return 'Hello world!'
```

Retrying Intermittent Issues

retry.lua

- Retry operations that are prone to periodic failure
- Specify timeout time, pause time, retry count
- Module includes pcall embedded inside

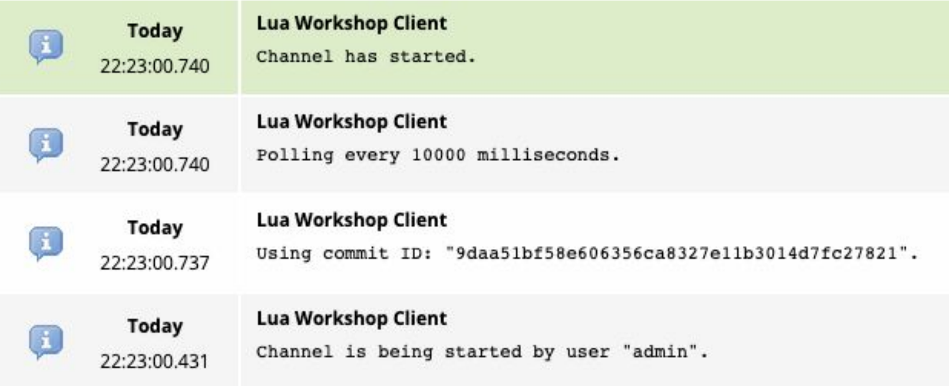


```lua
function main(Data)
   iguana.stopOnError(true)
   retry.call{func=UnreliableFunc,retry=20, pause=1, arg1=Data,
             funcname='UnreliableFunc', errorfunc=ErrorFunction}
end
```

Logging in Lua

Logging in Lua

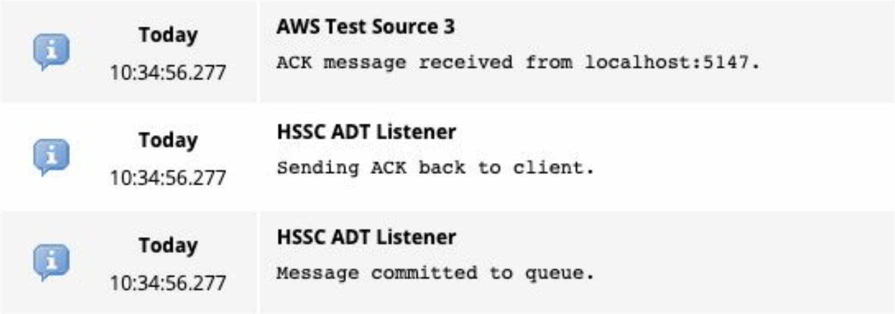
- Not limited to default logging provided by a channel
- Able to add custom logging in Lua script
- Four types of logging available



```
Today          Lua Workshop Client
22:23:00.740   Channel has started.

Today          Lua Workshop Client
22:23:00.740   Polling every 10000 milliseconds.

Today          Lua Workshop Client
22:23:00.737   Using commit ID: "9daa51bf58e606356ca8327e11b3014d7fc27821".

Today          Lua Workshop Client
22:23:00.431   Channel is being started by user "admin".
```

iguana.logInfo

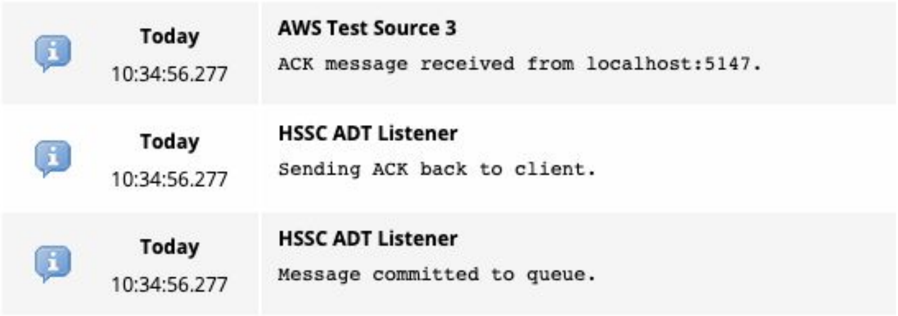
- Trigger a regular log message
- Great for auditing or informational purposes



ℹ	Today 10:34:56.277	**AWS Test Source 3** `ACK message received from localhost:5147.`
ℹ	Today 10:34:56.277	**HSSC ADT Listener** `Sending ACK back to client.`
ℹ	Today 10:34:56.277	**HSSC ADT Listener** `Message committed to queue.`

iguana.logInfo Best Practices

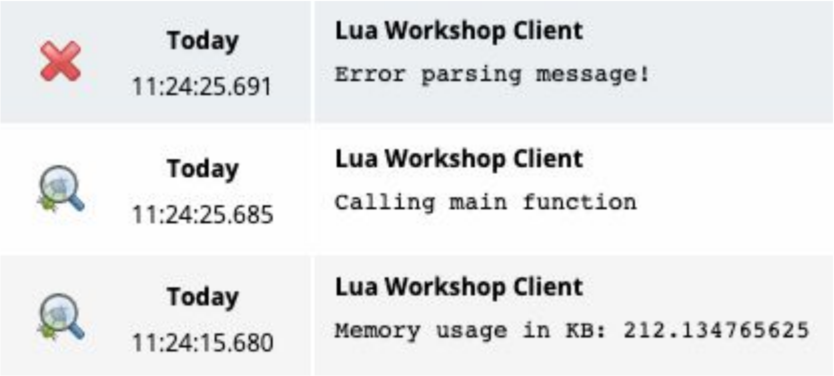
- Useful things to log:
  - API responses
  - Data read from files
  - All file operations



```
Today          AWS Test Source 3
10:34:56.277   ACK message received from localhost:5147.

Today          HSSC ADT Listener
10:34:56.277   Sending ACK back to client.

Today          HSSC ADT Listener
10:34:56.277   Message committed to queue.
```

iguana.logError

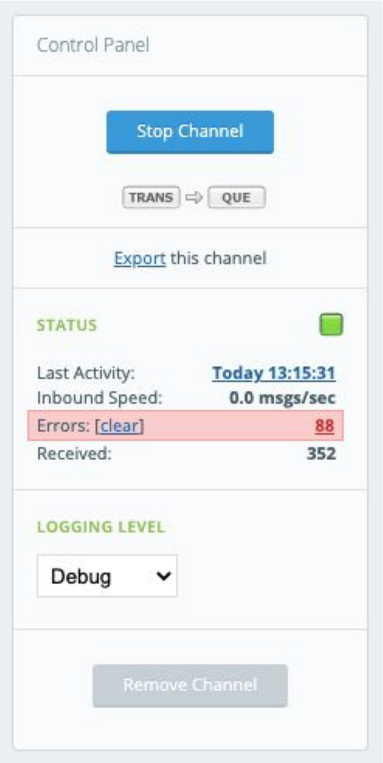
- Trigger an error log message without stopping channel
- This will increase error log count on the Dashboard
- Great for alerting



```
✖   Today         Lua Workshop Client
    11:24:25.691  Error parsing message!

🔍  Today         Lua Workshop Client
    11:24:25.685  Calling main function

🔍  Today         Lua Workshop Client
    11:24:15.680  Memory usage in KB: 212.134765625
```

iguana.logError

- Trigger an error log message without stopping channel
- This will increase error log count on the Dashboard
- Great for alerting



STATUS 🟩

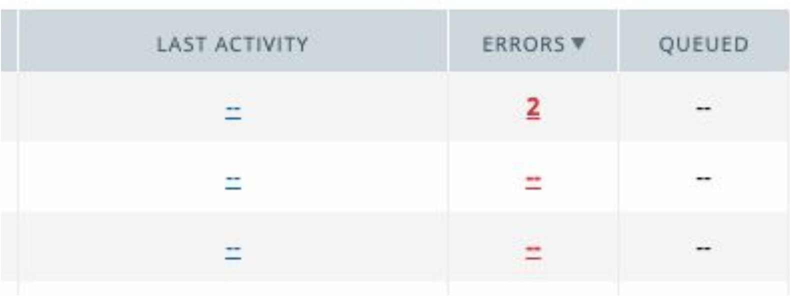
Last Activity: Today 13:15:31
Inbound Speed: 0.0 msgs/sec
Errors: [clear] 88
Received: 352

LOGGING LEVEL

Debug

iguana.logError

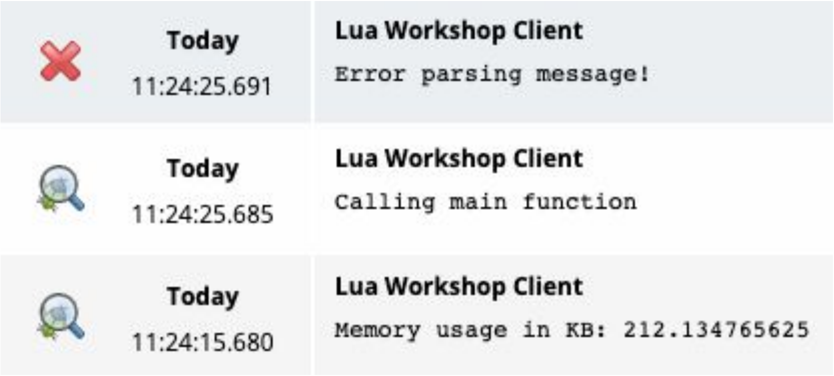
- Trigger an error log message without stopping channel
- This will increase error log count on the Dashboard
- Great for alerting



	LAST ACTIVITY	ERRORS ▼	QUEUED
	--	2	--
	--	--	--
	--	--	--

iguana.logError Best Practices

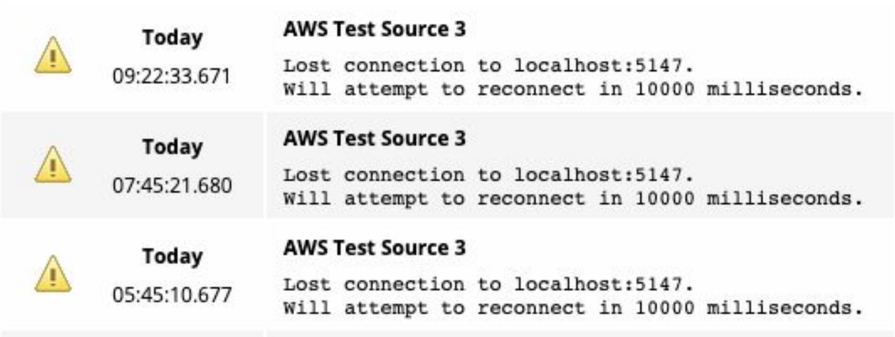
- Useful things to log:
  - Issues with external systems



```
Today          Lua Workshop Client
11:24:25.691   Error parsing message!

Today          Lua Workshop Client
11:24:25.685   Calling main function

Today          Lua Workshop Client
11:24:15.680   Memory usage in KB: 212.134765625
```

iguana.logWarning

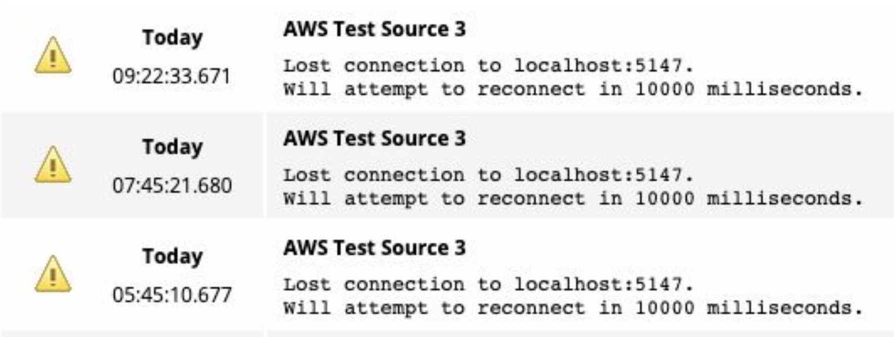
- Trigger a warning log message
- Great for warning



```
⚠  Today          AWS Test Source 3
   09:22:33.671   Lost connection to localhost:5147.
                  Will attempt to reconnect in 10000 milliseconds.

⚠  Today          AWS Test Source 3
   07:45:21.680   Lost connection to localhost:5147.
                  Will attempt to reconnect in 10000 milliseconds.

⚠  Today          AWS Test Source 3
   05:45:10.677   Lost connection to localhost:5147.
                  Will attempt to reconnect in 10000 milliseconds.
```

iguana.logWarning Best Practices

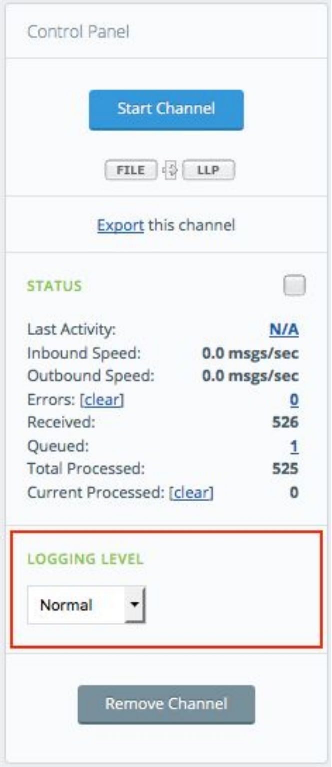
- Useful things to log:
  - Skipped messages
  - Intermittent issues



```
⚠  Today          AWS Test Source 3
   09:22:33.671   Lost connection to localhost:5147.
                  Will attempt to reconnect in 10000 milliseconds.
⚠  Today          AWS Test Source 3
   07:45:21.680   Lost connection to localhost:5147.
                  Will attempt to reconnect in 10000 milliseconds.
⚠  Today          AWS Test Source 3
   05:45:10.677   Lost connection to localhost:5147.
                  Will attempt to reconnect in 10000 milliseconds.
```

iguana.logDebug

- Used for debugging when needed
- Does not bloat the logs when not needed
- Great for testing and debugging



Control Panel

Export this channel

STATUS

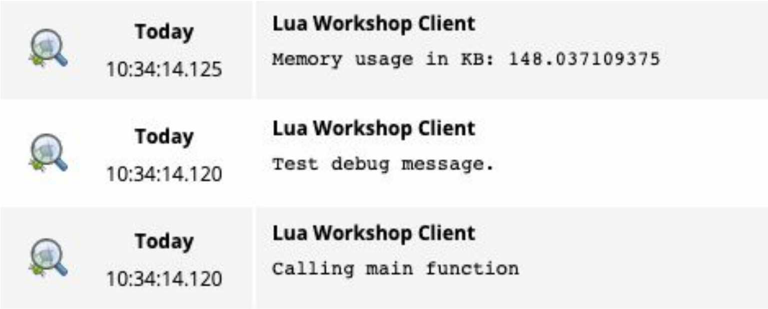
Last Activity: N/A
Inbound Speed: 0.0 msgs/sec
Outbound Speed: 0.0 msgs/sec
Errors: [clear] 0
Received: 526
Queued: 1
Total Processed: 525
Current Processed: [clear] 0

LOGGING LEVEL

Normal

iguana.logDebug

- Used for debugging when needed
- Does not bloat the logs when not needed
- Great for testing and debugging



```
Today 10:34:14.125   Lua Workshop Client
                     Memory usage in KB: 148.037109375
Today 10:34:14.120   Lua Workshop Client
                     Test debug message.
Today 10:34:14.120   Lua Workshop Client
                     Calling main function
```

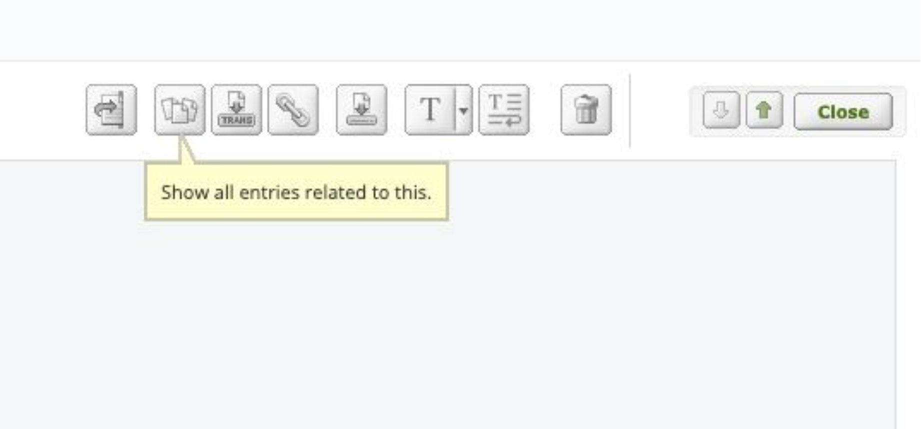
iguana.logDebug
Best Practices

- Useful things to log:
  - SQL statements
  - Responses from databases
  - Memory usage
  - Function calls and responses

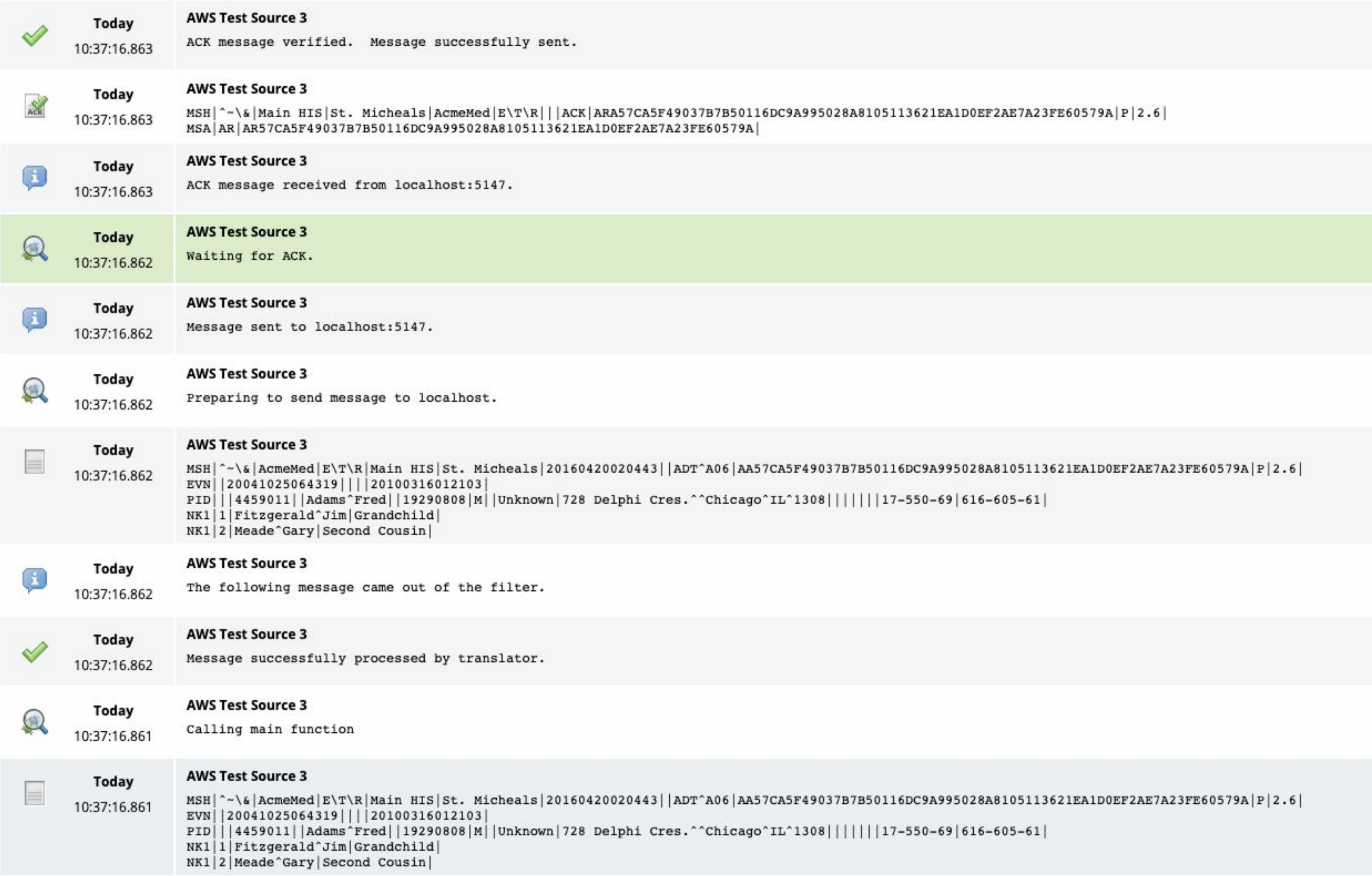
Troubleshooting with Logging

Troubleshooting Production Issues

- All log messages associated with a message are related
- Need to balance between extra log messages and better visibility
- Use iguana.logDebug for troubleshooting



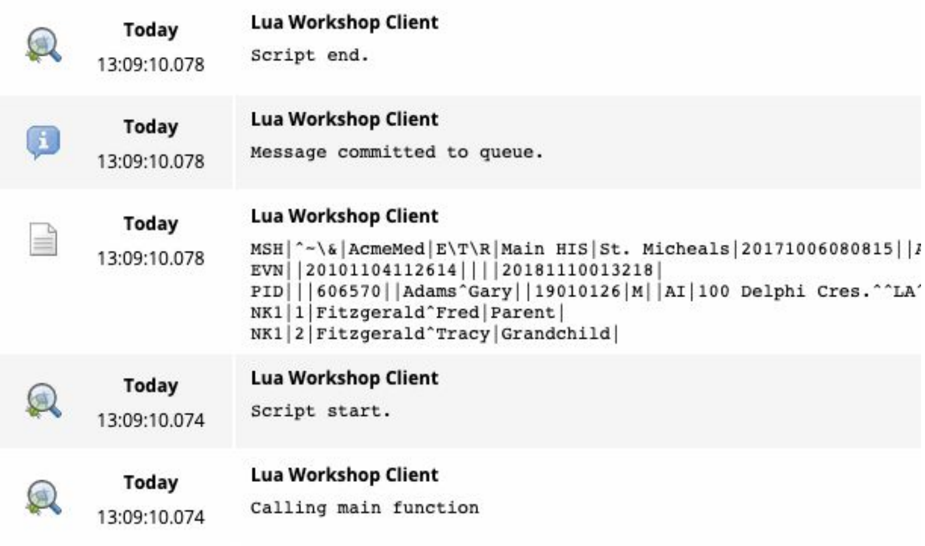
Show all entries related to this.



```
✓   Today          AWS Test Source 3
    10:37:16.863   ACK message verified.  Message successfully sent.

    Today          AWS Test Source 3
    10:37:16.863   MSH|^~\&|Main HIS|St. Micheals|AcmeMed|E\T\R|||ACK|ARA57CA5F49037B7B50116DC9A995028A8105113621EA1D0EF2AE7A23FE60579A|P|2.6|
                   MSA|AR|AR57CA5F49037B7B50116DC9A995028A8105113621EA1D0EF2AE7A23FE60579A|

ℹ   Today          AWS Test Source 3
    10:37:16.863   ACK message received from localhost:5147.

    Today          AWS Test Source 3
    10:37:16.862   Waiting for ACK.

ℹ   Today          AWS Test Source 3
    10:37:16.862   Message sent to localhost:5147.

    Today          AWS Test Source 3
    10:37:16.862   Preparing to send message to localhost.

    Today          AWS Test Source 3
    10:37:16.862   MSH|^~\&|AcmeMed|E\T\R|Main HIS|St. Micheals|20160420020443||ADT^A06|AA57CA5F49037B7B50116DC9A995028A8105113621EA1D0EF2AE7A23FE60579A|P|2.6|
                   EVN||20041025064319||||20100316012103|
                   PID|||4459011||Adams^Fred||19290808|M||Unknown|728 Delphi Cres.^^Chicago^IL^1308|||||||17-550-69|616-605-61|
                   NK1|1|Fitzgerald^Jim|Grandchild|
                   NK1|2|Meade^Gary|Second Cousin|

ℹ   Today          AWS Test Source 3
    10:37:16.862   The following message came out of the filter.

✓   Today          AWS Test Source 3
    10:37:16.862   Message successfully processed by translator.

    Today          AWS Test Source 3
    10:37:16.861   Calling main function

    Today          AWS Test Source 3
    10:37:16.861   MSH|^~\&|AcmeMed|E\T\R|Main HIS|St. Micheals|20160420020443||ADT^A06|AA57CA5F49037B7B50116DC9A995028A8105113621EA1D0EF2AE7A23FE60579A|P|2.6|
                   EVN||20041025064319||||20100316012103|
                   PID|||4459011||Adams^Fred||19290808|M||Unknown|728 Delphi Cres.^^Chicago^IL^1308|||||||17-550-69|616-605-61|
                   NK1|1|Fitzgerald^Jim|Grandchild|
                   NK1|2|Meade^Gary|Second Cousin|
```

Troubleshooting Performance Issues

- All Iguana logs are timestamped
- Use iguana.logDebug to measure execution time



```
Today          Lua Workshop Client
13:09:10.078   Script end.

Today          Lua Workshop Client
13:09:10.078   Message committed to queue.

Today          Lua Workshop Client
13:09:10.078   MSH|^~\&|AcmeMed|E\T\R|Main HIS|St. Micheals|20171006080815||
               EVN||20101104112614||||20181110013218|
               PID|||606570||Adams^Gary||19010126|M||AI|100 Delphi Cres.^^LA
               NK1|1|Fitzgerald^Fred|Parent|
               NK1|2|Fitzgerald^Tracy|Grandchild|

Today          Lua Workshop Client
13:09:10.074   Script start.

Today          Lua Workshop Client
13:09:10.074   Calling main function
```

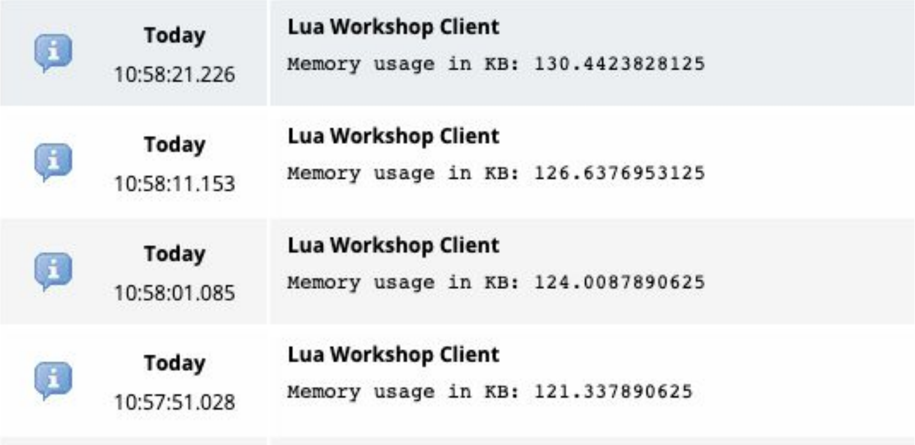
Preventing memory leaks in Lua

Preventing memory leaks in Lua

- Lua typically is good at managing memory
- Best practice is to define all variables as local
- Only use global variables if you know what you’re doing

How to troubleshoot memory leaks

- Use global.collectgarbage('count') to measure memory usage
- Log memory usage to monitor memory usage
- Gradually increasing memory usage indicates memory leak

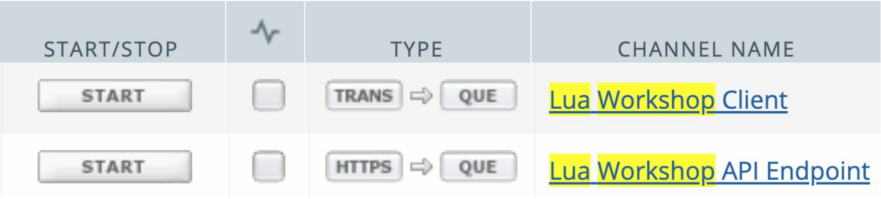


```
Today          Lua Workshop Client
10:58:21.226   Memory usage in KB: 130.4423828125

Today          Lua Workshop Client
10:58:11.153   Memory usage in KB: 126.6376953125

Today          Lua Workshop Client
10:58:01.085   Memory usage in KB: 124.0087890625

Today          Lua Workshop Client
10:57:51.028   Memory usage in KB: 121.337890625
```

Walk-through

Walk-through

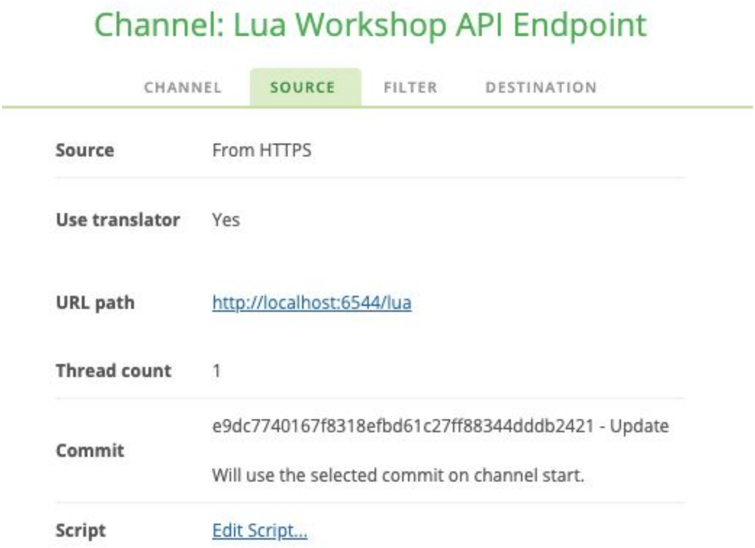
- Make API call to endpoint to get HL7 messages
- Goal is to create channel that is able to make API call and handle possible errors cases



START/STOP		TYPE	CHANNEL NAME
START	□	TRANS ⇨ QUE	Lua Workshop Client
START	□	HTTPS ⇨ QUE	Lua Workshop API Endpoint

Lua Workshop API Endpoint

- Treat as a black box that returns a random HL7 message
- Randomly returns errors or bad HL7 messages

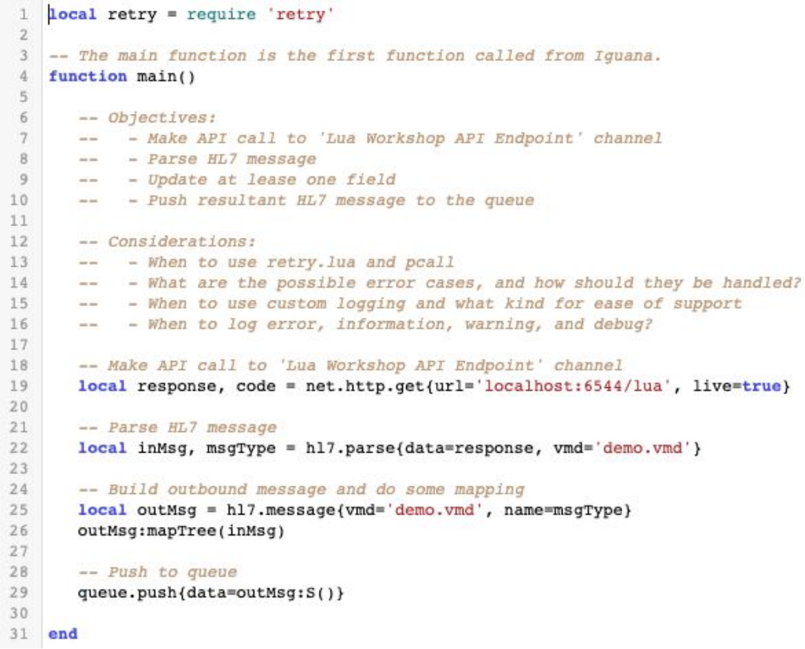


Channel: Lua Workshop API Endpoint

Source	From HTTPS
Use translator	Yes
URL path	http://localhost:6544/lua
Thread count	1
Commit	e9dc7740167f8318efbd61c27ff88344dddb2421 - Update Will use the selected commit on channel start.
Script	Edit Script...

Lua Workshop Client

- Update Lua script to handle error cases
- Use logging to improve supportability
- Goal is to write Lua script that can handle all error cases and run for at least an hour



```lua
local retry = require 'retry'

-- The main function is the first function called from Iguana.
function main()

   -- Objectives:
   --    - Make API call to 'Lua Workshop API Endpoint' channel
   --    - Parse HL7 message
   --    - Update at lease one field
   --    - Push resultant HL7 message to the queue

   -- Considerations:
   --    - When to use retry.lua and pcall
   --    - What are the possible error cases, and how should they be handled?
   --    - When to use custom logging and what kind for ease of support
   --    - When to log error, information, warning, and debug?

   -- Make API call to 'Lua Workshop API Endpoint' channel
   local response, code = net.http.get{url='localhost:6544/lua', live=true}

   -- Parse HL7 message
   local inMsg, msgType = hl7.parse{data=response, vmd='demo.vmd'}

   -- Build outbound message and do some mapping
   local outMsg = hl7.message{vmd='demo.vmd', name=msgType}
   outMsg:mapTree(inMsg)

   -- Push to queue
   queue.push{data=outMsg:S()}

end
```

Questions

What is the latest/greatest version of the retry module?

- The latest is the one you import from the **Retry periodic failure** channel from the **Builtin: Iguana Tools** repository
- Has not been changed much in the past few years
- Reference: https://help.interfaceware.com/v6/retry-example

What kind of Channels utilize "Show all entries related to this"?

- The “Show all entries related to this” button shows up for any log entries related to a message: https://help.interfaceware.com/v6/using-the-logs#related
- Custom log entries will be related to messages that are passed through Filter and To Trans components
- From Trans and From HTTPS channel components only show custom log entries in sequential order when viewing logs (i.e. custom log entries are not related to messages)

Do global Lua variables persist between poll time executions

- Global Lua variables persist between poll time executions
- Examples of when this is desired:
  - Persistent database connections: https://help.interfaceware.com/v6/database-connection
  - Caching lookup values in memory: https://help.interfaceware.com/v6/database-query-caching

A mechanism to lock a database while Iguana is updating a record?

- No mechanism in Iguana
- Typically something that the database team would be able to implement if possible for the database

How to prevent duplicate messages from being passed through Iguana?

- Query the Iguana logs via the log query API to check for duplicate messages: https://help.interfaceware.com/v6/http-api-reference#api_query
- For API requests that timeout, use the **retry.lua** module
- We have a more efficient solution for preventing duplicate messages that also uses the log query API: https://help.interfaceware.com/v6/duplicate-filter

Can I store data retrieved from database in memory?

- We have a module for storing lookup values in memory: https://help.interfaceware.com/v6/database-query-caching

Have More Questions?

Don’t Hesitate to Ask Questions

Email us your questions:

Webinar@Interfaceware.com

Thank You!

iNTERFACEWARE | integration made easy

References

1. https://help.interfaceware.com/api/#iguana_stopOnError
2. https://help.interfaceware.com/api/#global_pcall
3. https://help.interfaceware.com/v6/working-with-pcall
4. https://help.interfaceware.com/v6/trap-errors-pcall
5. https://help.interfaceware.com/v6/retry-lua-module
6. https://help.interfaceware.com/v6/retry-example
7. https://help.interfaceware.com/v6/retrying-unreliable-external-resources
8. https://help.interfaceware.com/v6/retry-database-connection
9. https://help.interfaceware.com/v6/retry-a-web-service
10. https://help.interfaceware.com/v6/log-features#types
11. https://help.interfaceware.com/api/#iguana_logInfo
12. https://help.interfaceware.com/api/#iguana_logError
13. https://help.interfaceware.com/api/#iguana_logWarning
14. https://help.interfaceware.com/api/#iguana_logDebug
15. https://help.interfaceware.com/v6/how-to-troubleshoot-iguana-memory-leak