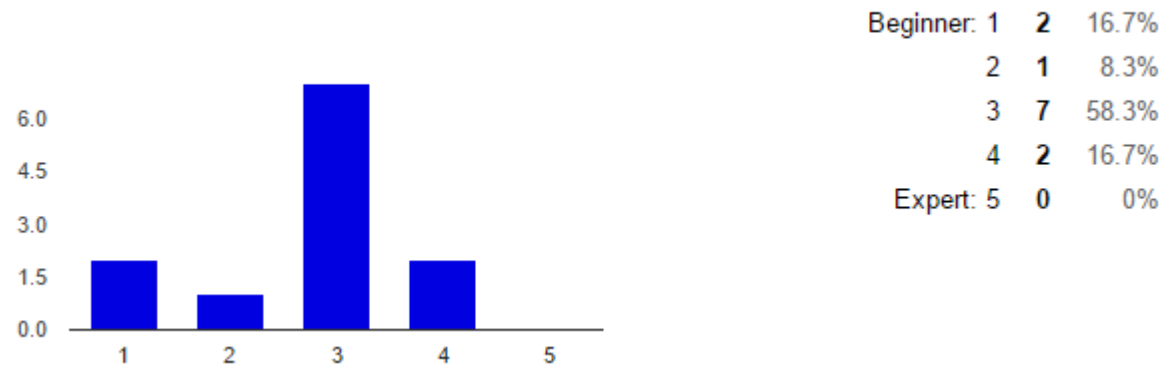


Extending the power of ILLiad through Server Addons

Better, Faster, Stronger

Addons & YOU

Rate your general understanding of addons



Which types of addons are you familiar with?



Client (shown as a form on the request form, main form, or user form)	10	100%
System addons (run within the client but are not shown)	4	40%
Server addons (run within the system manager)	5	50%

Not for today

Client & System addons (i.e. screen scraping & NCIP)

.NET programming

How to deal with your IT staff or your hosted service provider

Goals for today

- Bragging rights of knowing an 8.7 feature
- Not fall asleep
- Better understand how addons work
- Develop an addon in less than 3 hours

Housekeeping

Preconference Links

<http://my.illiad.rocks/preconf>

Extending ILLiad Slack Team

Invite: <http://goo.gl/forms/tnzpPkljsh>

Team URL: <https://extendingilliad.slack.com/>

Server
Client
System

Addons

...and what's the difference?

Server AddonExamples

PDF Reminders

Twilio SMS Notifications

Availability Routing

Addon Basics

XML configuration file

LUA code files

... extended with .NET libraries (sometimes)

Server Addon Basics

Customization Manager modifies it.

System Manager runs it.

Addon Interval

What's New?

Zip File Packages

Easier to
Distribute
Modify Settings
Maintain code

* Make sure the files are defined in the config

Breaking it down

Review a server addon
to see how it works

Best Practice #1

Write down requirements

...in detail

Be mindful about others

...and yourself when you
forget how it is *supposed* to
work

Today's Addon

Process textbook requests

automatically

Today's Addon

Process textbook requests

How?

- Check if the ISBN of ILLiad request is textbook
 - Route requests from faculty to a review queue
 - Cancel requests from users that are not Faculty
 - Add cancellation reason
 - Send notification
- If not, route it back to Awaiting Request Processing

Today's Addon

Process textbook requests

Whats missing?

- How does the addon know it's a textbook?
- What will trigger the addon to work?
 - What will make the routing rule *not* work after the addon processes it?

Before Starting: Resources

Addon Documentation

<https://prometheus.atlas-sys.com/display/ILLiadAddons/Addon+Technical+Overview>

<https://prometheus.atlas-sys.com/display/illiad/Command+Options+for+ILLiad+Addons>

ILLiad Database Tables

<https://prometheus.atlas-sys.com/display/illiad/Key+ILLiad+Data+Tables>

Extending ILLiad Slack Team

<http://goo.gl/forms/tnzpPkljsh>

Before Starting: Tools

Atom <https://atom.io/>

Text Editor

BareTail <http://baremetalsoft.com/baretail/>

Following log files

Notepad++ <https://notepad-plus-plus.org/>

Text Editor

Before Starting: Logging

1. `LogDebug('Message')`

Writes to File Log on Server

2. `ExecuteCommand("WriteSystemLog",
"Message");`

Writes to ILLiad EventLog database table

Can be viewed in Client

Exercise #1

Write an addon that monitors a queue and routes the request to Awaiting Request Processing & handles the ItemInfo1 requirement

Best Practice #2

You will make mistakes! It's ok. Version your addon with that in mind.



Update this when you make a new release and it has a major feature

Update this when you make a new release and it has a minor change

V 0.1.2 Update this when making a change within your development cycle

Exercise #2

Again, Update the addon to monitor a queue and route requests to Awaiting Request Processing.

Note

There are multiple ways to accomplish things when developing.

Exercise #3

Update the addon to make sure it only starts processing another iteration after it finished the last one.

Best Practice #3

This should be in every server addon that uses the `TimerElapsed` event handler

Exercise #4

Add a note when the request is a textbook and route to the faculty review queue.

Best Practice #5

Start simple!

Best Practice #6

Separate functions that you may want to re-use later or need to abstract.
With server addon packages, you can also put common functions into their own LUA file and re-use in other addons.

Exercise #5

Update the addon to only route if the user is faculty otherwise cancel the request.

Errors

Trap errors with LUA *pcall* when working with stuff that may fail

- Database connections
- Web services
- Files
- Your own functions that *error*

Best Practice #7

Love the pcall

LUA Error Handling and Exceptions

www.lua.org/pil/8.4.html

<http://pastebin.com/haWsg6TU>

You can & should make improvements

Exercise #6

Add the *real* check to see if the request is a textbook

```
local connection = CreateDatabaseConnection("Preconf.dbcx");  
connection.QueryString = "SELECT Count(*) as Count FROM Textbooks WHERE ISBN = @ISBN";  
connection:AddParameter('@ISBN', isbn, Types['System.Data.SqlDbType'].NVarChar);
```

```
connection:Connect();
```

```
local count = connection:ExecuteScalar();
```

```
LogDebug("Textbooks matching: " .. count);
```

```
connection:Dispose();
```

isbn is a
variable!

```
luanet.load_assembly("System.Data");
```

```
local Types = {};
```

```
Types['System.Data.SqlDbType'] =
```

```
luanet.import_type('System.Data.SqlDbType');
```

Questions?

Thanks!

Matt Calsada

mcalsada@atlas-sys.com

@mcal