

Forward-Looking Statements



This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Data-to-Everything, D2E and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners. ©2021 Splunk, Inc. All rights reserved.



Jason Conger

Principal Solutions Engineer | Splunk, Inc.






Tom Martin

Director, Field Solutions Engineering | Splunk, Inc.



Agenda

How to build on Splunk like an internal Splunker

-  Configuration User Interface
-  Debugging
-  Using repositories for code
-  Add-on conversion
-  Automatic workflows

Add-on Builder



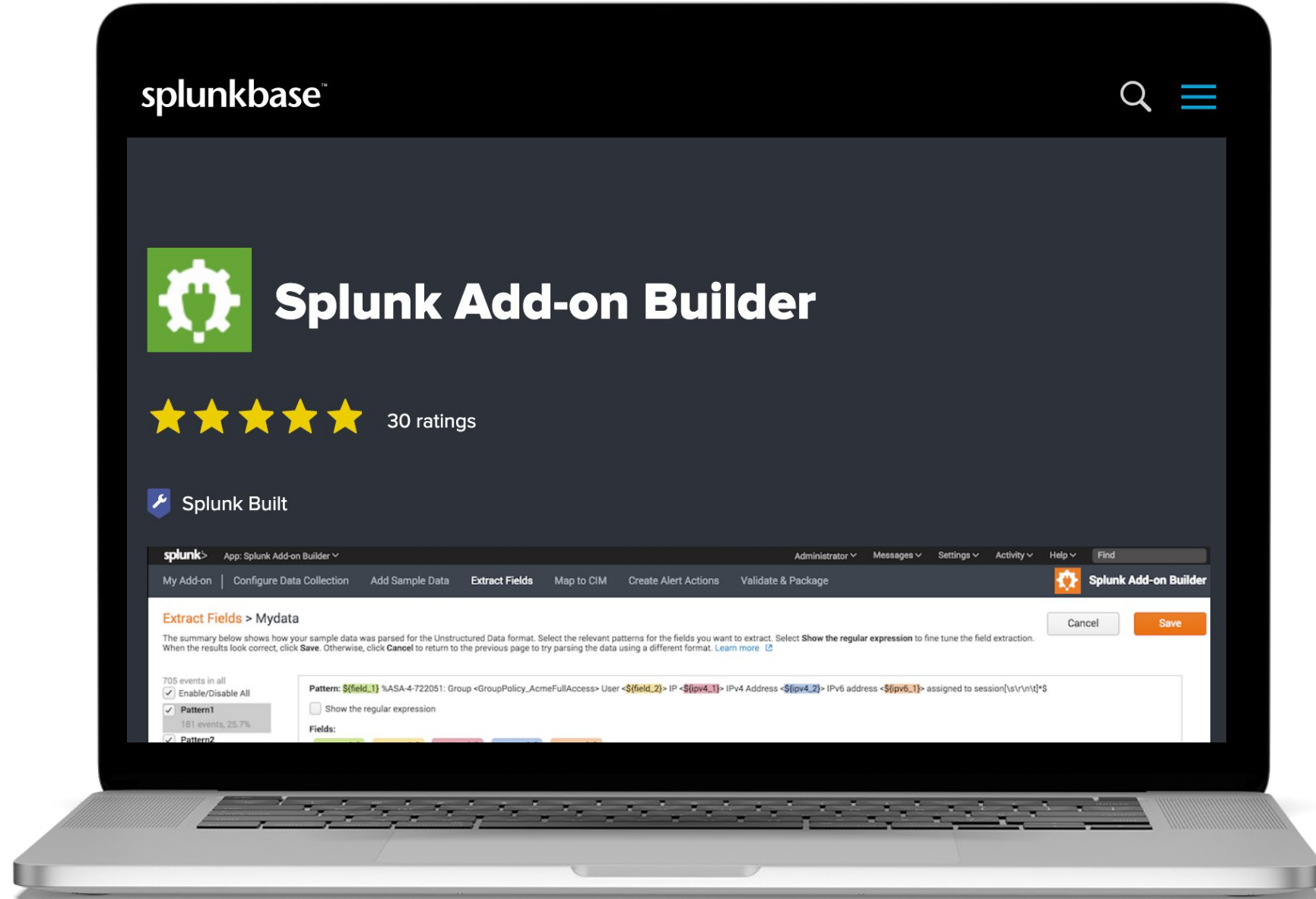
 Wires up files

 Helpers

 Packaging

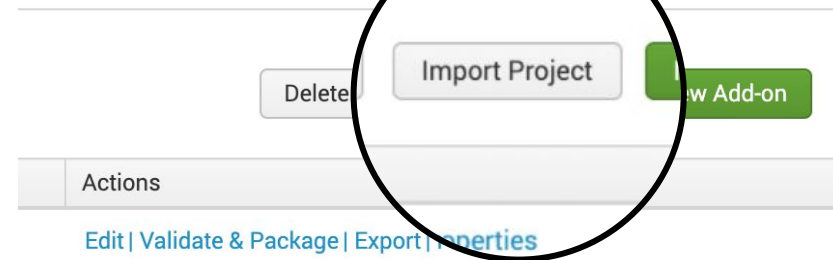
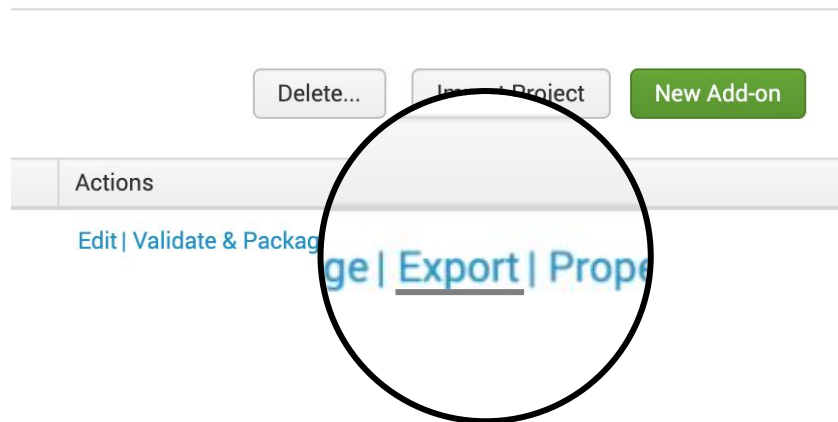
 Validation

 User Interface



Shared Development

Limitations





Splunk Add-on Builder

Splunk Add-on Builder Features



Modern Development Functionality

splunk> 





Get your cameras ready!



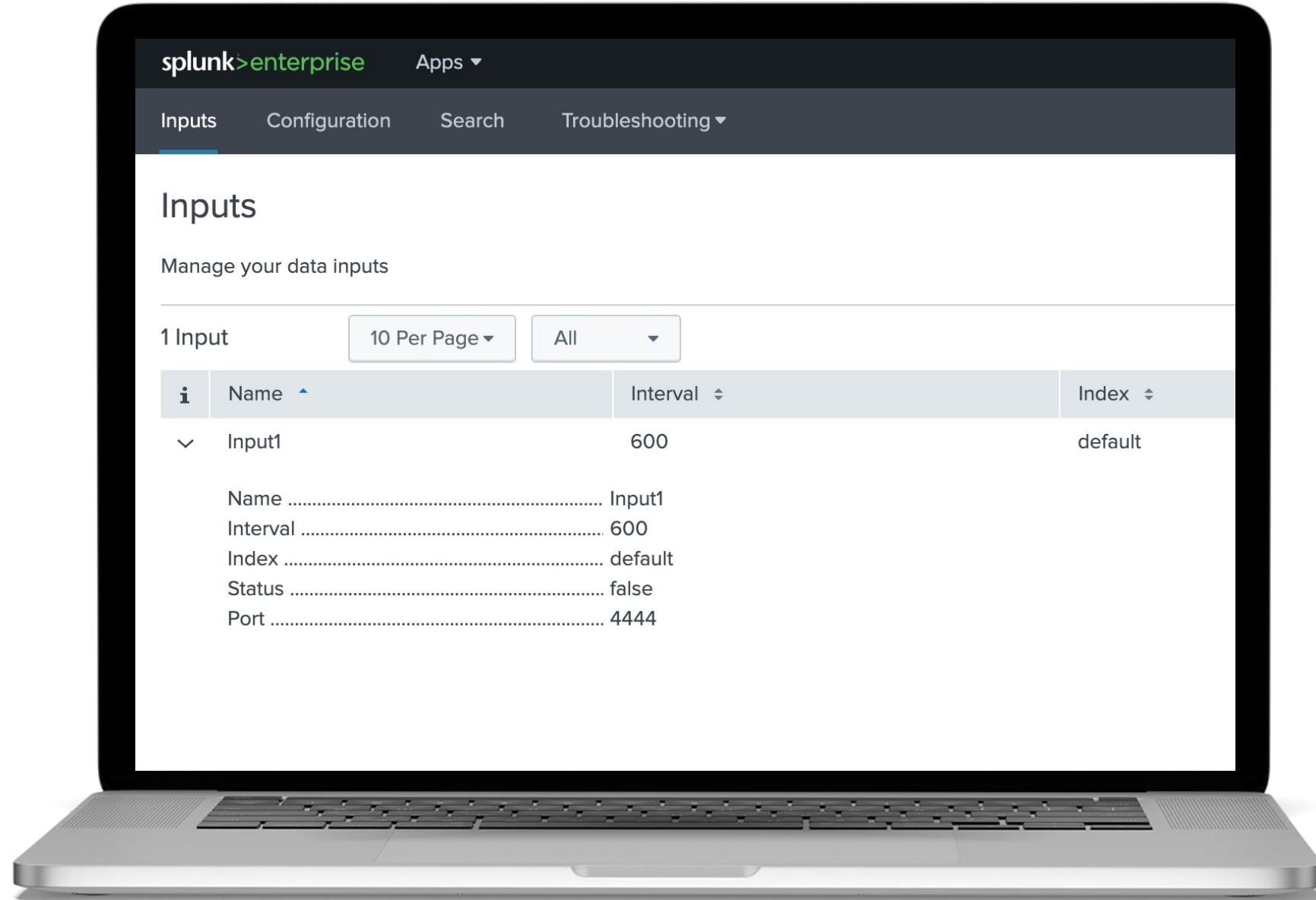
A lot of QR codes are coming your way!

Universal Configuration Console

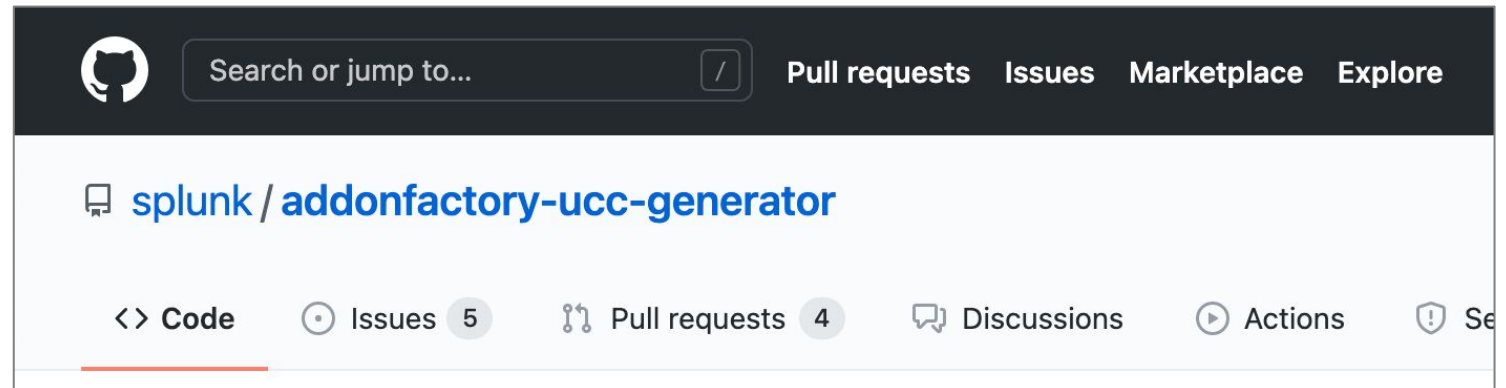
(a.k.a. UCC)



User Interface for apps built with Splunk Add-on Builder



UCC can be used outside of Splunk Add-on Builder!







<https://github.com/splunk/addonfactory-ucc-generator>



UCC-GEN Repository Structure

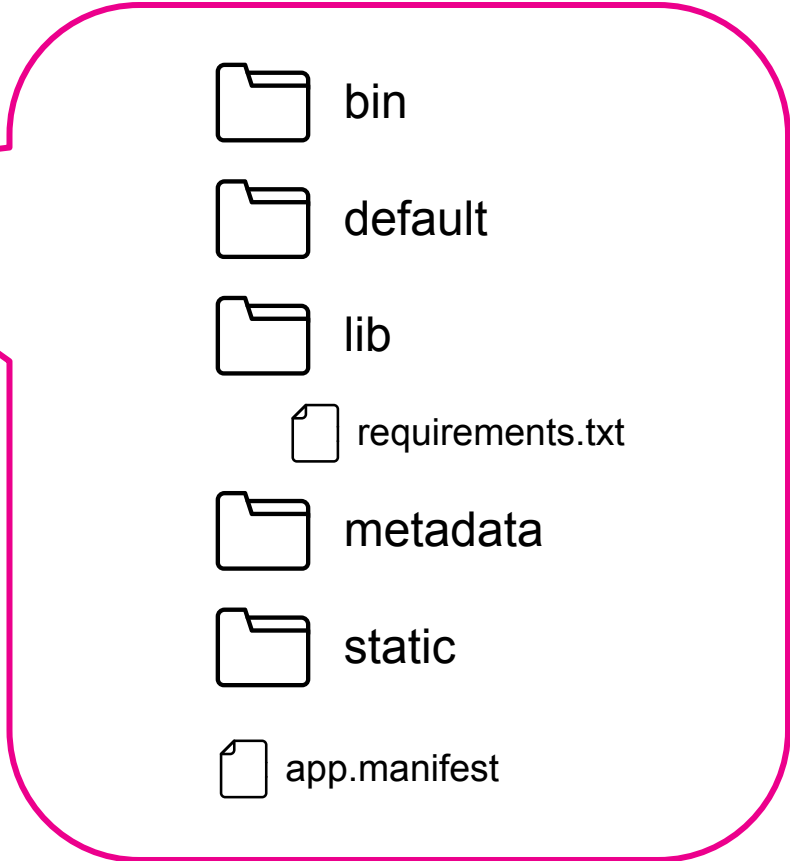
What goes where?

 package (required)	Splunk App Files
 globalConfig.json (required)	Defines User Interface
 .gitignore (recommended)	git - don't look at me
 tests (optional)	Unit Tests

UCC-GEN Repository Structure

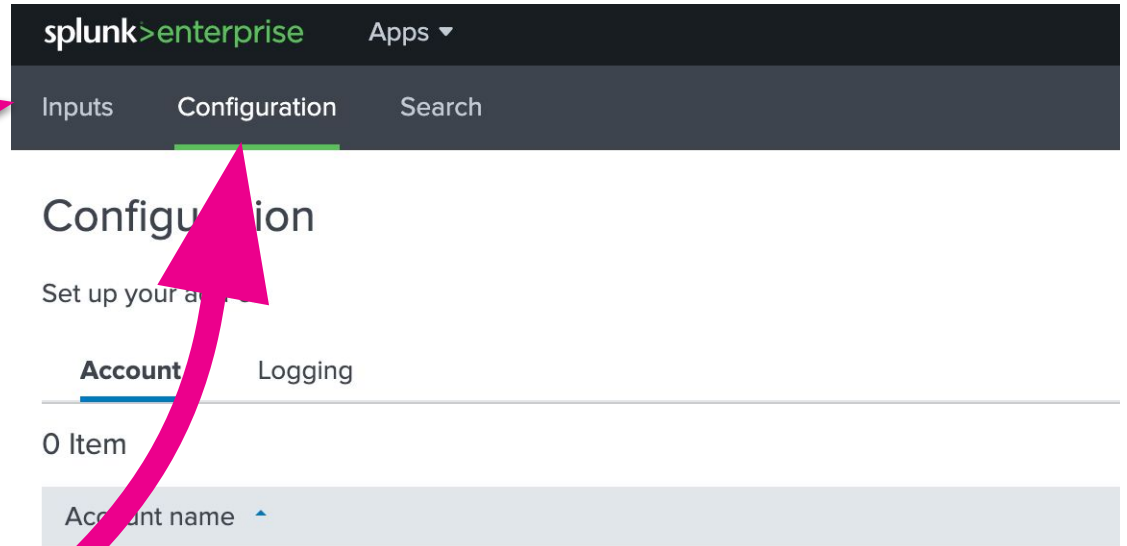
What goes where?

-  package (required)
-  globalConfig.json (required)
-  .gitignore (recommended)
-  tests (optional)

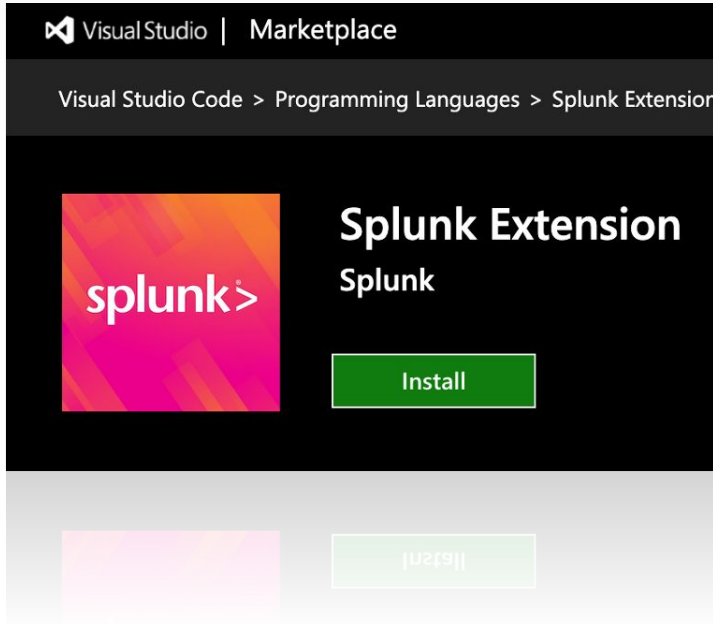


globalConfig.json

```
{  
  "meta": {  
    "name": "TA-conf21",  
    "displayName": "Splunk .conf21 Example App",  
    "version": "1.0.0",  
    "apiVersion": "3.2.0",  
    "restRoot": "TA_conf21",  
    "schemaVersion": "0.0.2"  
  },  
  "pages": {  
    "inputs": {  
      "title": "Inputs",  
      "description": "Manage your data inputs",  
      "table": { ... },  
      "services": [ ... ]  
    },  
    "configuration": {  
      "title": "Configuration",  
      "description": "Set up your add-on",  
      "tabs": [ ... ]  
    }  
  }  
}
```



← globalConfig.json schema



Pro Tip!

<https://code.visualstudio.com>

Use Microsoft Visual Studio Code + the Splunk Extension



<https://marketplace.visualstudio.com/items?itemName=Splunk.splunk>

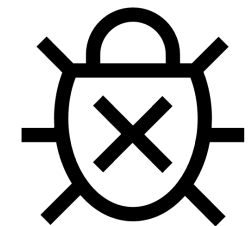
Shameless Plug

Debugging Splunk Apps, Add-ons

and configurations with Microsoft Visual
Studio Code

Jason Conger

Principal Solutions Engineer | Splunk, Inc.

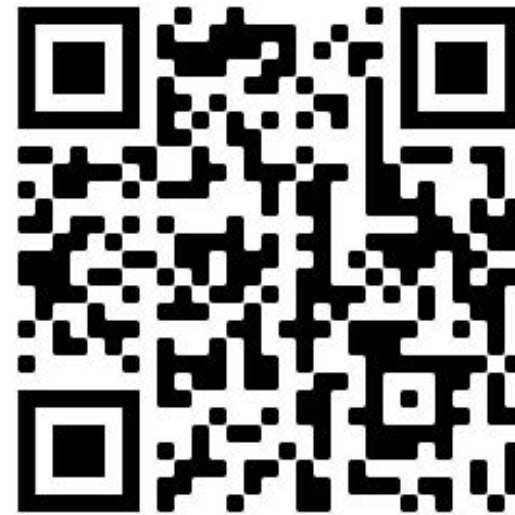


<https://conf.splunk.com/watch/conf-online.html?search=dev1188c#/>

splunk> .conf21

github.com/JasonConger/splunk-conf21

Follow along at home...



Bare Bones Template

Converting Add-on Builder to UCC

I want to open source my TA, but I used Add-on Builder :(

Sounds great, but how much time and effort is this going to take?

- 1) Build your TA using AOB so you get all the goodies.
- 2) Convert your TA using the ucc migration utility¹
- 3) Run ucc-generator² on your newly converted package
- 4) Test your new UCC-based TA
- 5) Publish to Splunkbase and GitHub
- 6) Accept pull requests from others to make your TA even better!

Let's see it in action!



¹ ucc migration utility: https://github.com/tmartin14/ucc_migration_test

² ucc-generator: <https://github.com/splunk/addonfactory-ucc-generator>

1) Clone the Repo & expand your Add-on

```
tomm-mbp-e42el:ucc_migration_test tomm$ ls -al
total 12560
drwxr-xr-x@ 15 tomm  staff    480 Jul 26 16:37 .
drwxr-xr-x@ 29 tomm  staff    928 Jul 26 16:37 ..
-rw-r--r--@  1 tomm  staff   6148 Jul 26 12:34 .DS_Store
drwxr-xr-x@ 18 tomm  staff    576 Jul 26 11:30 .git
drwxr-xr-x@  3 tomm  staff     96 Apr 21 15:27 .vscode
-rw-r--r--  1 tomm  staff   3048 Jul 26 11:30 README.md
drwxr-xr-x@ 13 tomm  staff    416 Jul 26 12:36 Splunk_TA_AppDynamics
-rw-r--r--@  1 tomm  staff  6214540 Jul 23 17:18 Splunk_TA_AppDynamics-1.9.0.tgz
-rwxr-xr-x@  1 tomm  staff   24430 Jul 26 16:12 convert.sh
-rw-r--r--@  1 tomm  staff   39118 Jul 26 16:34 globalConfig.json
-rw-r--r--  1 tomm  staff    361 Apr 16 09:37 imports.py
drwxr-xr-x@ 14 tomm  staff    448 Jul 26 16:37 package
-rwxr-xr-x@  1 tomm  staff   1097 Apr 27 11:00 package_app.sh
-rwxr-xr-x@  1 tomm  staff   1607 Apr 22 14:49 refresh_docker.sh
-rwxr-xr-x  1 tomm  staff   4368 Apr 27 11:00 submit_appInspect.sh
tomm-mbp-e42el:ucc_migration_test tomm$
```



2) Convert It

```
[tomm-mbp-e42e1:ucc_migration_test tomm$ ./convert.sh Splunk_TA_AppDynamics

Converting Splunk_TA_AppDynamics...

Creating a new ./package directory

Updating globalConfig.json. Adding "template":"input_with_helper" to each "service"

-----
      Creating lib/requirements.txt file
-----

      Adding datetime to requirements.txt
      Adding html to requirements.txt
      Adding urllib to requirements.txt
      Updating base64 to pybase64 in requirements.txt

Please check ./package/lib/requirements.txt file for additional libraries you *may* need to include for your
modular inputs to work. This script collects libraries that may or may not be needed and comments them all in
./package/lib/requirements.txt:
splunktaucclib>=4.1.0
#
# The following list *may* also be required for your modular inputs to work.
# Uncomment those that are required.
#
#HTMLParser
#pybase64
```

2) Convert It



The script checks for mismatches between tabs and spaces.

Python 2 code indented with a mixture of tabs and spaces should be converted to using spaces exclusively.

```
tomm-mbp-e42e1:ucc_migration_test tomm$ ./convert.sh Splunk_TA_AppDynamics

Converting Splunk_TA_AppDynamics...

removing existing ./package directory
Creating a new ./package directory

Updating globalConfig.json. Adding "template":"input_with_helper" to each "service"

-----
Creating lib/requirements.txt file
-----
Adding datetime to requirements.txt
Adding html to requirements.txt
Adding urllib to requirements.txt
Updating base64 to pybase64 in requirements.txt

Please check ./package/lib/requirements.txt file for additional libraries you *may* need to include for your
modular inputs to work. This script collects libraries that may or may not be needed and comments them all in
./package/lib/requirements.txt:
splunktaucclib>=4.1.0
#
# The following list *may* also be required for your modular inputs to work.
# Uncomment those that are required.
#
#HTMLParser
#pybase64
#json
#requests
#urllib
#urlparse
#datetime
#html
#urllib

-----
Processing Alert Actions...
-----
Finished with Alert Actions

-----
Processing Modular Inputs...
-----

Checking for potential issues with tabs and/or indentation. (python3 doesn't like mixing tabs and spaces)
*****
Issues Found. These must be fixed before running ucc-gen.
*****
input_module_appdynamics_api.py:124:           # this value is a list object which means we are processing 'application_performance' or 'business_transactions', etc.
input_module_appdynamics_api.py:127:         for values in metrics['metricValues']:
input_module_appdynamics_api.py:128:             # strip off the very last part of the metric name to insert in the Splunk event
input_module_appdynamics_api.py:138:         for values in metrics['metricValues']:
input_module_appdynamics_api.py:139:             # strip off the last 2 part of the metric name to insert in the Splunk event
input_module_appdynamics_summary.py:123:     # this value is a list object which means we are processing 'application_performance' or 'business_transactions', etc.
input_module_appdynamics_summary.py:126:     for values in metrics['metricValues']:
input_module_appdynamics_summary.py:127:         # strip off the very last part of the metric name to insert in the Splunk event
input_module_appdynamics_summary.py:130:         # and now grab the value attribute from metricValues and add it to this metric
input_module_appdynamics_summary.py:131:         #mystr = "%s:%s" % (metricName, values['value'])
input_module_appdynamics_summary.py:132:
input_module_appdynamics_summary.py:134:     for values in metrics['metricValues']:
input_module_appdynamics_summary.py:137:         # strip off the last 2 part of the metric name to insert in the Splunk event
input_module_appdynamics_summary.py:138:
*****
Exiting
tomm-mbp-e42e1:ucc_migration_test tomm$
```



2a) Fix any identified code errors

```

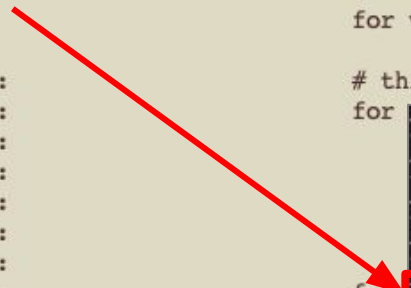
-----
Processing Alert Actions...
-----
Finished with Alert Actions

-----
Processing Modular Inputs...
-----

Checking for potential issues with tabs and/or indentation. (python3 doesn't like mixing tabs and spaces)
*****
Issues Found. These must be fixed before running ucc-gen.
*****
input_module_appdynamics_api.py:124:          # this value is a list object which means we are processing 'application_performance' or 'business_transactions', etc.
input_module_appdynamics_api.py:127:          for values in metrics['metricValues']:
input_module_appdynamics_api.py:128:              # strip off the very last part of the metric name to insert in the Splunk event
input_module_appdynamics_api.py:138:          for values in metrics['metricValues']:
input_module_appdynamics_api.py:139:              # strip off the last 2 part of the metric name to insert in the Splunk event
input_module_appdynamics_summary.py:123:      # this value is a list object which means we are processing 'application_performance' or 'business_transactions',
input_module_appdynamics_summary.py:126:      for
119          def parse_piped_naming(d):
120              # let's parse out the metricPath from component parts. This w
121              for key, val in list(d.items()):
122                  if isinstance(val, list):
123                      # this value is a list object which means we are process
124                      if (key == "application_performance" or key == "end_use
125                      for metrics in val:
126                      for values in metrics['metricValues']:
127                      # strip off the very last part of the metr
*****
Exiting

```

The script checks for mismatches between tabs and spaces. Python 3 code does not allow a mixture.



```

119          def parse_piped_naming(d):
120              # let's parse out the metricPath from component parts. This w
121              for key, val in list(d.items()):
122                  if isinstance(val, list):
123                      # this value is a list object which means we are process
124                      if (key == "application_performance" or key == "end_use
125                      for metrics in val:
126                      for values in metrics['metricValues']:
127                      # strip off the very last part of the metr

```

2b) Re-run the script

2c) Fix any version mismatches



```
-----  
Version Checking  
  appserver/static/js/build/globalConfig.json --> 1.8.0  
  app.manifest --> 1.9.0  
  default/app.conf --> 1.9.0  
  
ERROR:  Versions are out of synch.  Please edit so that all 3 files reflect the same version number  
Exiting now
```

(Yes, it's in 3 places)

3) Success!

```
tomm-mbp-e42e1:ucc_migration_test tomm$ ./convert.sh Splunk_TA_AppDynamics

Converting Splunk_TA_AppDynamics...

removing existing ./package directory
Creating a new ./package directory

Updating globalConfig.json. Adding "template":"input_with_helper" to each

-----
Creating lib/requirements.txt file
-----
Adding datetime to requirements.txt
Adding html to requirements.txt
Adding urllib to requirements.txt
Updating base64 to pybase64 in requirements.txt

Please check ./package/lib/requirements.txt file for additional libraries
modular inputs to work. This script collects libraries that may or may not
./package/lib/requirements.txt:
splunktaucclib>=4.1.0
#
# The following list *may* also be required for your modular inputs to
# Uncomment those that are required.
#
#HTMLParser
#pybase64
#json
#requests
#urllib
#urlparse
#datetime
#html
#urllib

-----
Processing Alert Actions...
-----
Finished with Alert Actions

-----
Processing Modular Inputs...
-----

Checking for potential issues with tabs and/or indentation. (python3 doesn't
no tab/indentation issues found.

Processing input named: appdynamics_analytics_api.py
Done.
Processing input named: appdynamics_api.py
Done.
Processing input named: appdynamics_summary.py
Done.
Finished with Modular Inputs
```

Finished with Modular Inputs

Cleaning Up... Removing files that are no longer needed

Done.

Version Checking

```
appserver/static/js/build/globalConfig.json --> 1.9.0
app.manifest --> 1.9.0
default/app.conf --> 1.9.0
```

all version settings match

Incrementing version number from 1.9.0 to 1.10.0.

VERSIONING NOTE

When running ucc-gen to generate your new TA, ucc-gen will attempt
to create a version number from the 'tag' in the github repo.

If you prefer to explicitly set the version, please use this flag
when running ucc-gen:

```
ucc-gen --ta-version 1.10.0
```

```
The current version of your TA is:      1.9.0
Recommended version for this update is: 1.10.0
```

Finished.

Checking for potential issues with tabs and/or indentation. (python3 doesn't like mixing tabs and spaces)
no tab/indentation issues found.

Reminder: If you need to uncomment entries in package/lib/requirements.txt, do NOT run ucc-gen at this point.
Uncomment what you need and then run ucc-gen.

Would you like to run ucc-gen --ta-version 1.10.0 now?

4) Generate the new UCC-based Add-on

```
Would you like to run ucc-gen --ta-version 1.10.0 now? y
Some functions will not be available outside of a splunk hosted process
2021-07-26 17:27:40,985 [UCC] INFO: Cleaning out directory /Users/tomm/Documents/Documents - tomm-mbp-e42e1/GitHub/ucc_migration_test/output
2021-07-26 17:27:40,985 [UCC] INFO: Cleaned out directory /Users/tomm/Documents/Documents - tomm-mbp-e42e1/GitHub/ucc_migration_test/output
2021-07-26 17:27:41,008 [UCC] INFO: Config is valid
2021-07-26 17:27:41,012 [UCC] INFO: Addon Version : 1.10.0
2021-07-26 17:27:41,012 [UCC] INFO: Package ID is Splunk_TA_AppDynamics
2021-07-26 17:27:41,013 [UCC] INFO: Copy UCC template directory
2021-07-26 17:27:41,045 [UCC] INFO: Copy globalConfig to output
2021-07-26 17:27:41,045 [UCC] INFO: Install Addon Requirements into /Users/tomm/Documents/Documents - tomm-mbp-e42e1/GitHub/ucc_migration_test/output/Splunk_TA_AppDynamics/lib from package
WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/issues/9617>
distutils: /Library/Frameworks/Python.framework/Versions/3.7/include/python3.7m/UNKNOWN
sysconfig: /Library/Frameworks/Python.framework/Versions/3.7/include/python3.7m
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
Requirement already satisfied: pip in /Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages (21.1)
Collecting pip
```


5) Package the Add-on & AppInspect It

```
Successfully installed certifi-2021.5.30 charset-normalizer-2.0.3 defusedxml-0.7.1 httplib2-0.19.1 idna-3.2 pyparsing-2.4.7 requests-2.26.0 sol
nlib-4.2.0 sortedcontainers-2.4.0 splunk-sdk-1.6.16 splunktalib-2.1.0 splunktaucclib-5.0.4 urllib3-1.26.6
2021-07-26 17:42:19,630 [UCC] INFO: Replace tokens in views
INFO:UCC:Replace tokens in views
2021-07-26 17:42:19,656 [UCC] INFO: Copy package directory
INFO:UCC:Copy package directory

Would you like to package this app now? y

Your new UCC_based app can be found here: ./output/Splunk_TA_AppDynamics_1.10.0.tgz ←

Would you like to submit this app for appInspect now? y
running ./submit_appInspect.sh ./output/Splunk_TA_AppDynamics_1.10.0.tgz

Enter your Splunkbase login: tomm@splunk.com
Enter your Splunkbase password:
Mon Jul 26 17:42:48 EDT 2021    Authenticating...
Mon Jul 26 17:42:53 EDT 2021    Successfully Athenticated...

Mon Jul 26 17:42:53 EDT 2021    Submitting AppInspect request with Splunk Cloud compliance...
```

Yeah! My new, open source ready Add-on

Create a GitHub repo and start accepting pull requests!



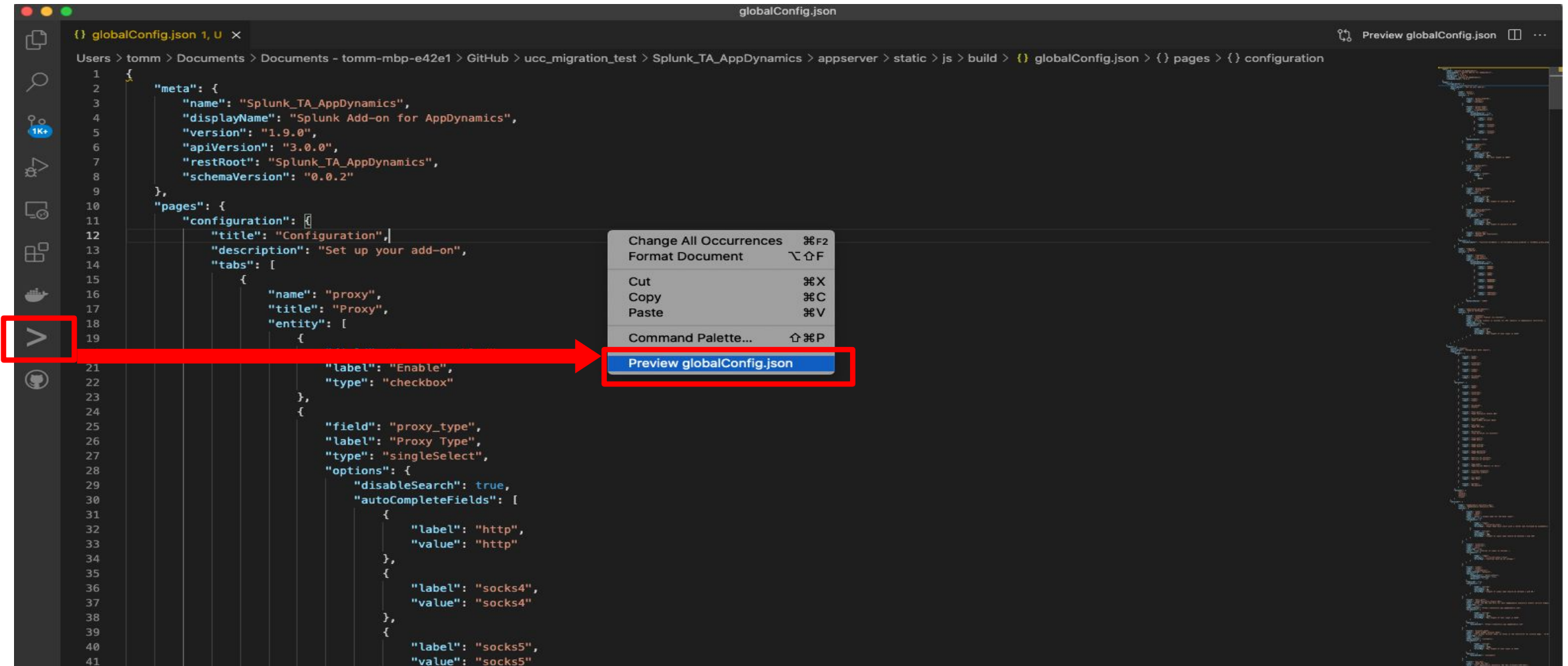
```
Your new UCC_based app can be found here: ./output/Splunk_TA_AppDynamics_1.10.0.tgz
```

But, I REALLY Liked AOB's WYSIWYG Editor



Install the Splunk Extension for VS-Code!

globalConfig.json



...And We're Back!

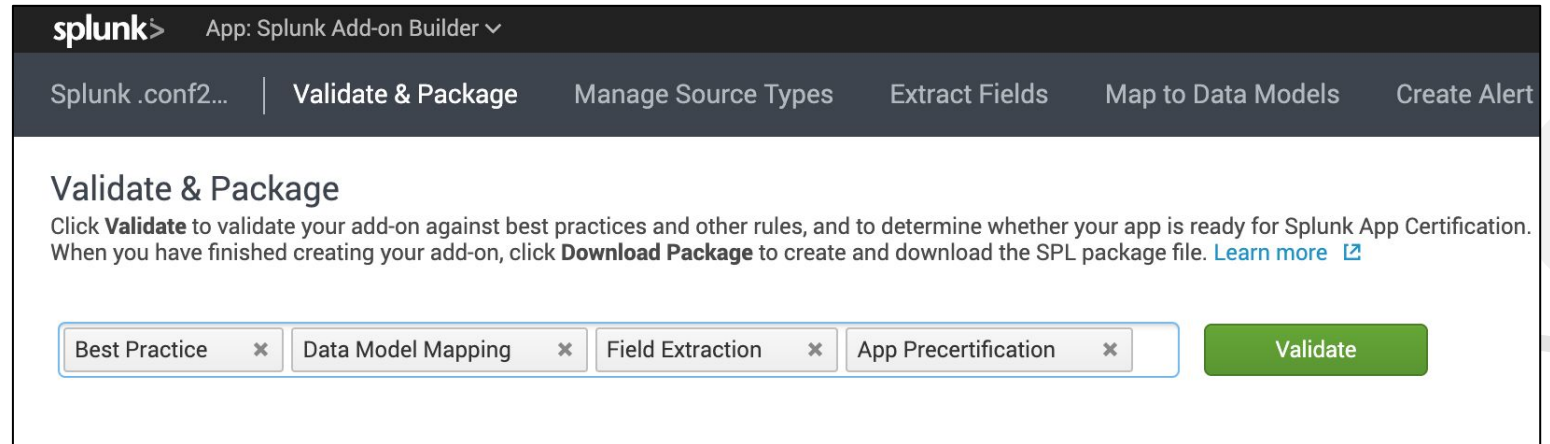
Never Fear, the VS-Code extension is here!

The image shows a side-by-side comparison of the Splunk configuration process. On the left, a VS Code editor displays the `globalConfig.json` file for the `Splunk_TA_AppDynamics` app. The JSON structure includes a `meta` section with app details and a `pages` section for configuration. The `configuration` page has a `tabs` array with a `proxy` tab containing fields for `proxy_enabled` (checkbox) and `proxy_type` (select). The `proxy_type` field has options for `http`, `socks4`, and `socks5`.

On the right, the 'Global Config Preview' UI is shown. It features a dropdown menu set to 'AppDynamics Analytics API'. Below this are several configuration fields:

- Name:** A text input field with the instruction 'Enter a unique name for the data input'.
- Interval:** A text input field with the instruction 'Time interval of input in seconds.'
- Index:** A dropdown menu with the option 'Options will populate from data/indexes'.
- AppD Analytics Events URL:** A text input field with the instruction 'Enter the URL and Port for your AppDynamics Analytics Events service endpoint (i.e.: http://<events_service_endpoint>:9080)'. The placeholder text is `http://<events_service_endpoint>:9080`.
- AppD GLOBAL Account Name:** A text input field with the instruction 'The global account name, as shown in the Controller UI License page. (X-Events-API-AccountName)'.
- AppD API Key:** A text input field with the instruction 'Your AppDynamics Analytics API Key (X-Events-API-Key)'.
- Time Duration (in minutes):** A text input field.

Validation and Packaging?



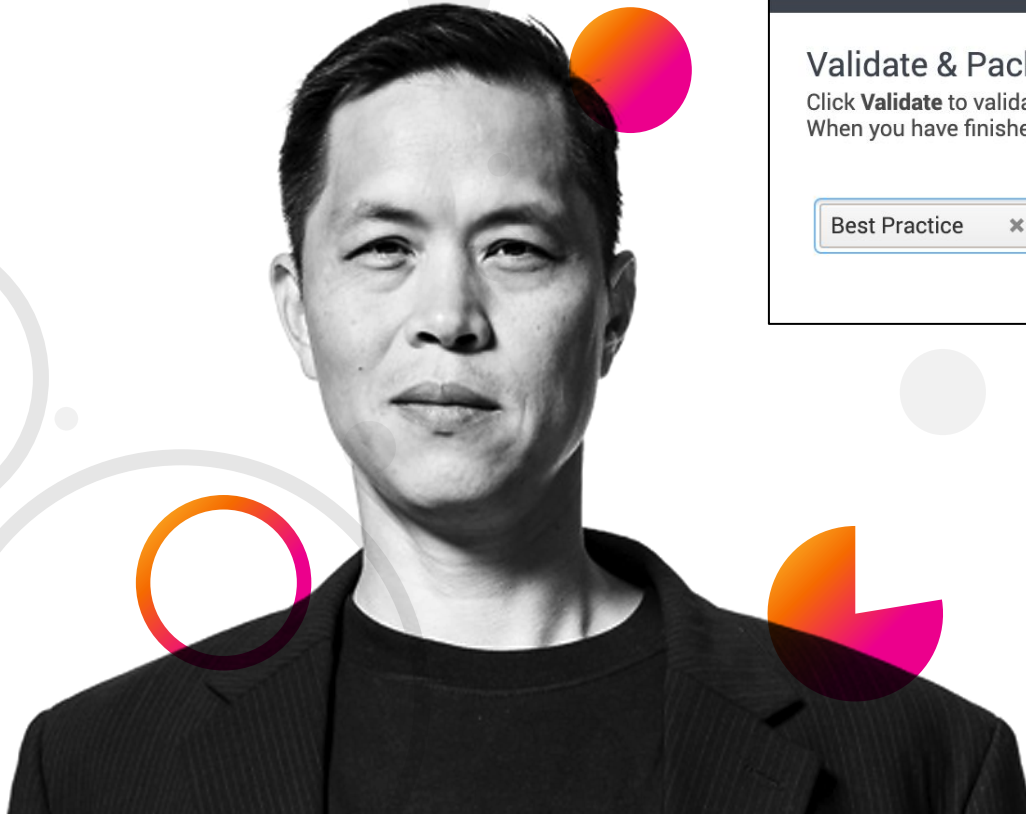
splunk> App: Splunk Add-on Builder ▾

Splunk .conf2... | **Validate & Package** | Manage Source Types | Extract Fields | Map to Data Models | Create Alert

Validate & Package

Click **Validate** to validate your add-on against best practices and other rules, and to determine whether your app is ready for Splunk App Certification. When you have finished creating your add-on, click **Download Package** to create and download the SPL package file. [Learn more](#) ↗

Best Practice × Data Model Mapping × Field Extraction × App Precertification × **Validate**





Search or jump to...

Pull requests Issues Marketplace Explore

JasonConger / splunk-conf21 Private

Code Issues Pull requests Actions Projects Security Insights Settings

Update ci.yml CI #54

Package

Summary

Jobs

- Lint Code Base
- Semgrep Scan
- Build
- AppInspect CLI (cloud)
- AppInspect CLI (appapproval)
- AppInspect CLI (deprecated_feature)
- AppInspect CLI (developer_guidance)
- AppInspect CLI (future)
- AppInspect CLI (self-service)
- AppInspect CLI (splunk_appinspect)

Check Code

Pushed via push 6 minutes ago	Status	Total duration	Billable time	Artifacts
JasonConger pushed -> 248359e main	Success	4m 40s	8m 39s	3

ci.yml

Lint Code Base 1m 31s

Semgrep Scan 16s

Build 48s

Matrix: Appinspect CLI

7 jobs completed

Show all jobs

App Inspect

Artifacts

Produced during runtime

Name

Size

github.com/JasonConger/splunk-conf21

Follow along at home...

[splunk-conf21](#) / [.github](#) / [workflows](#) / [ci.yml](#)



Bare Bones Template

Resources

- Comprehensive Example Repository
 - <https://github.com/splunk/splunk-add-on-for-ucc-example>
- .conf20 - Debugging Splunk Apps, Add-ons, and Configurations using Visual Studio Code
 - <https://conf.splunk.com/watch/conf-online.html?search=dev1188c#/>
- .conf20 - Fast, Off-the-Shelf Testing for Splunk Apps
 - <https://conf.splunk.com/watch/conf-online.html?search=dev1332c#/>

Related Sessions

DEV1453B - Touring the Software Factory: Get Visibility Into Your Pipeline Analytics to Ship Better Software Faster With Github

DEV1560B - Go From Zero-to-Automated CI/CD App Building and Testing In 30-Minutes (or Less)

DEV1783C - From "Create From Template..." to "App Installed Successfully" in Splunk Cloud Platform in Less Than 60 Minutes

Thank You

Please provide feedback via the

SESSION SURVEY

