

splunk>

.conf2017

© 2017 SPLUNK INC.

Friction Free Splunk App Certification: A How to Guide For Approaching The Process

Tedd Hellmann
Andy Nortrup
Logan Knecht

How Does the Splunk App Certification Process Benefit Me?

Tedd Hellmann
Product Manager - Data Ecosystem

What is App Certification?

And why should I care?

App Certification:

- ▶ Ensures your app meets our best practices
 - Includes customer documentation
- ▶ Checks for security vulnerabilities
- ▶ Checks for operational issues
- ▶ *Gives your app more prominent listing on Splunkbase!*



Splunk Certified

Learn more: <http://dev.splunk.com/view/app-cert/SP-CAAEE2S>

What is Cloud Vetting?

And why should I care?

Cloud Vetting:

- ▶ Checks for security vulnerabilities
- ▶ Checks for operational issues
- ▶ *Is required before your app can be installed on Splunk Cloud!*

COMPATIBILITY

Products: [Splunk Cloud](#), [Splunk Enterprise](#)

Learn more: <http://dev.splunk.com/view/app-cert/SP-CAAEE2S>

How can I make my life easier?

CLI output

Bad End

```
thellmann-mbp15:apps thellmann$ splunk-appinspect inspect --included-tags splunk_appinspect ./Hello_World
Validating: Hello_World Version: 0.1
.FF...F.F...FFF.....
.....F.....
..

Application Content Structure Standards
  Check that static/appIcon_2x is 72x72px or less
  FAILURES: static/appIcon_2x.png does not exist
  Check that static/appIcon_2x is a png file
  FAILURES: static/appIcon_2x.png does not exist
  Check that static/appIcon is 36x36px or less
  FAILURES: static/appIcon.png does not exist
  Check that static/appIcon is a png file
  FAILURES: static/appIcon.png does not exist

Directory Structure Standards
  Check that the 'local' directory does not exist. All configuration
  should be in the 'default' directory.
  FAILURES: local directory exists in the app
  Check that the file 'local.meta' does not exist. All metadata
  permissions should be set in 'default.meta'.
  FAILURES: Do not supply a 'local.meta' file - put all settings in
  default.meta
  Check that when decompressed the Splunk App directory name matches the
  app.conf [package] stanza's id property.
  FAILURES: The 'app.conf' [package] stanza was not detected.
  Please add this stanza.

Documentation standards
  Check that the app has a /README file that includes version support,
  system requirements, installation, configuration, troubleshooting and
  running of the app, or a link to online documentation.
  FAILURES: No README was found.

Hello_World Report Summary:

  skipped: 0
  success: 61
  manual_check: 21
  failure: 4
  warning: 0
  error: 0
  not_applicable: 72
-----
  Total: 162
```

Good End

```
thellmann-mbp15:apps thellmann$ splunk-appinspect inspect --included-tags splunk_appinspect ./Hello_World_Final
Validating: Hello_World_Final Version: 0.1
.....
.....
..

Hello_World_Final Report Summary:

  skipped: 0
  success: 75
  manual_check: 22
  failure: 4
  warning: 0
  error: 0
  not_applicable: 65
-----
  Total: 162
```

How can I make my life easier?

API output - HTML

App Cert Report

Hello

My first app!

Author

Version

Hash

Run Time

Execution

App Cert

Totals:	
Status	Count
Failures	
Errors	

Lines of Code

[*not_applicable*] Check that setup.xml will work in a distributed topology.

```
File: default/setup.xml was not found.
```

[**success**] Check that apps with app.manifest are valid or apps without an app.manifest can generate one.

Directory Structure Standards

[**failure**] Check that when decompressed the Splunk App directory name matches the app.conf [package] stanza's id property.

```
The `app.conf` [package] stanza was not detected. Please add this stanza.
```

[**success**] Check that local/passwords.conf does not exist. Password files are not transferable between instances.

Run Parameters:

Field

AppInspect Version

User ID

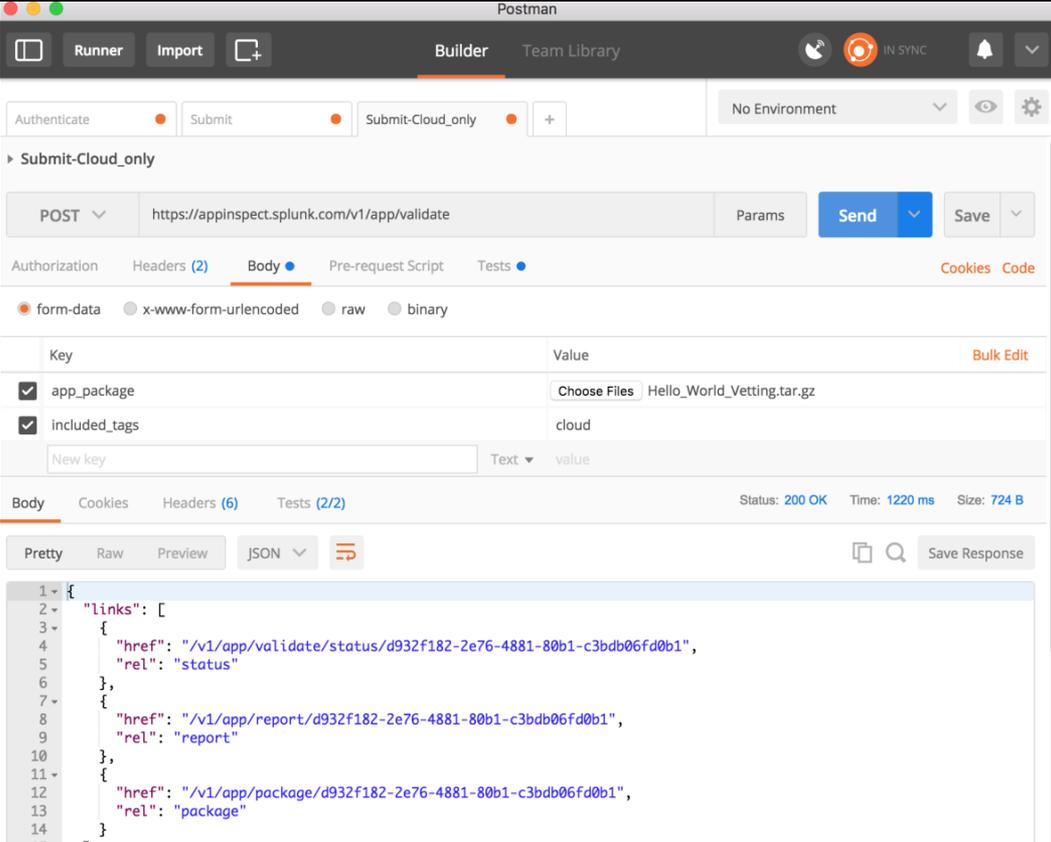
conf2017

How can I make my life easier?

API with Postman

Step 2:

- ▶ Upload app to validate endpoint
- ▶ <https://appinspect.splunk.com/v1/app/validate>
- ▶ Postman can store the request id



The screenshot shows the Postman interface for a POST request to `https://appinspect.splunk.com/v1/app/validate`. The request body is set to `form-data` and contains two fields: `app_package` (with a file named `Hello_World_Vetting.tar.gz`) and `included_tags` (with the value `cloud`). The response is displayed in JSON format, showing a `links` array with three entries: `status`, `report`, and `package`.

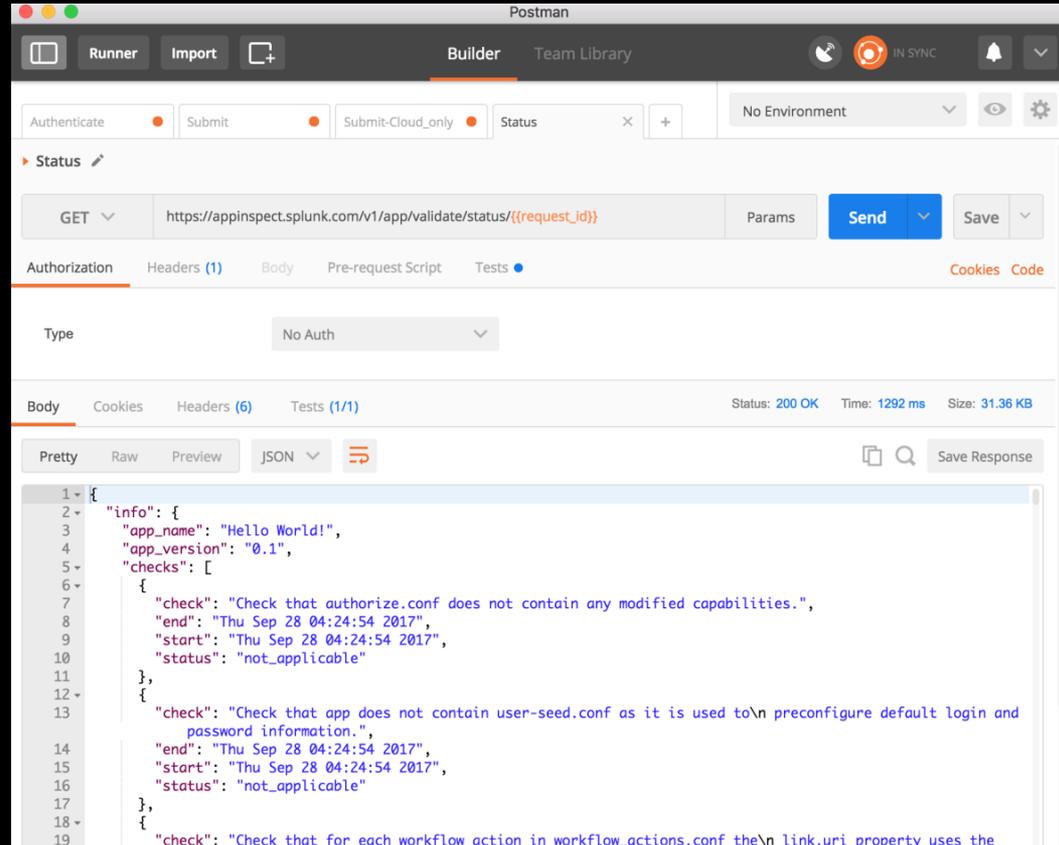
```
1 {
2   "links": [
3     {
4       "href": "/v1/app/validate/status/d932f182-2e76-4881-80b1-c3bdb06fd0b1",
5       "rel": "status"
6     },
7     {
8       "href": "/v1/app/report/d932f182-2e76-4881-80b1-c3bdb06fd0b1",
9       "rel": "report"
10    },
11    {
12      "href": "/v1/app/package/d932f182-2e76-4881-80b1-c3bdb06fd0b1",
13      "rel": "package"
14    }
15  ]
16 }
```

How can I make my life easier?

API with Postman

Step 3a:

- ▶ Check for completion
- ▶ `https://appinspect.splunk.com/va/app/validate/status/{{request_id}}`



Postman interface showing a successful GET request to the endpoint `https://appinspect.splunk.com/v1/app/validate/status/{{request_id}}`. The response is a JSON object with the following structure:

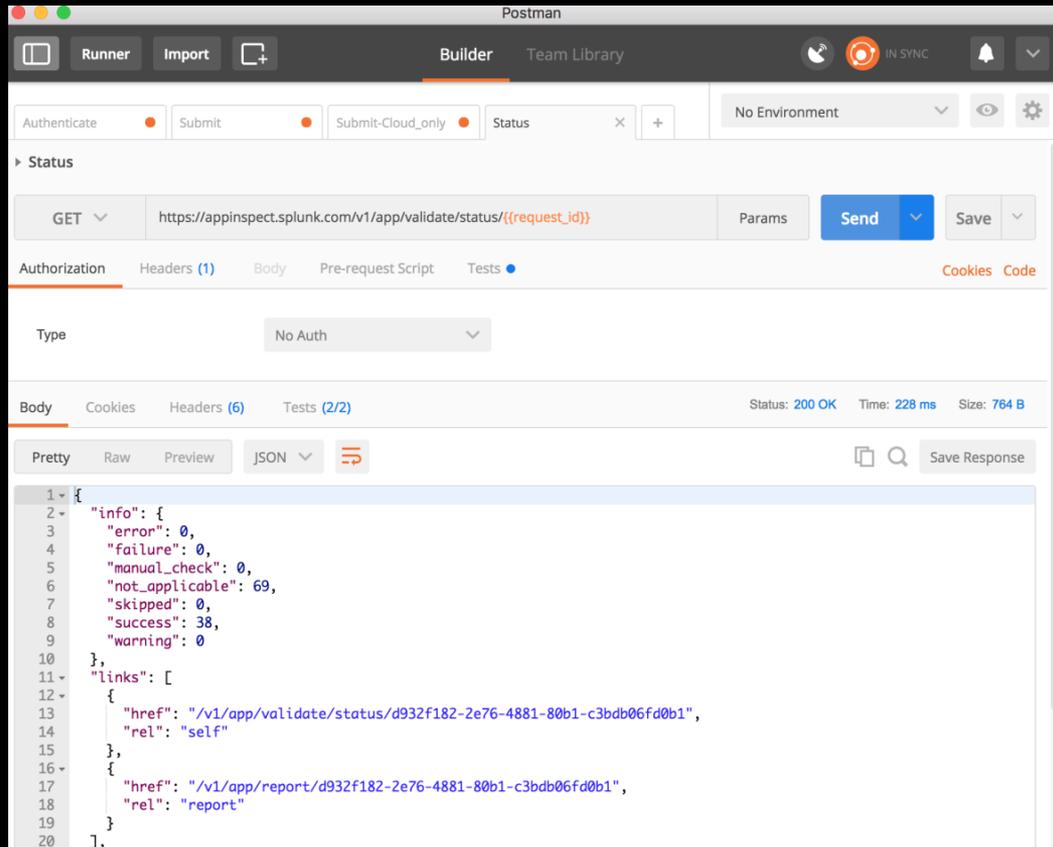
```
1- {
2-   "info": {
3-     "app_name": "Hello World!",
4-     "app_version": "0.1",
5-     "checks": [
6-       {
7-         "check": "Check that authorize.conf does not contain any modified capabilities.",
8-         "end": "Thu Sep 28 04:24:54 2017",
9-         "start": "Thu Sep 28 04:24:54 2017",
10-        "status": "not_applicable"
11-      },
12-      {
13-        "check": "Check that app does not contain user-seed.conf as it is used to\n preconfigure default login and
14-        password information.",
15-        "end": "Thu Sep 28 04:24:54 2017",
16-        "start": "Thu Sep 28 04:24:54 2017",
17-        "status": "not_applicable"
18-      },
19-      {
20-        "check": "Check that for each workflow action in workflow_actions.conf the\n link.uri property uses the
```

How can I make my life easier?

API with Postman

Step 3b:

- ▶ When the report is ready, you'll see a summary of the results instead of individual checks



The screenshot shows the Postman interface with a GET request to `https://appspect.splunk.com/v1/app/validate/status/{(request_id)}`. The response is a JSON object with the following structure:

```
1 - {
2 -   "info": {
3 -     "error": 0,
4 -     "failure": 0,
5 -     "manual_check": 0,
6 -     "not_applicable": 69,
7 -     "skipped": 0,
8 -     "success": 38,
9 -     "warning": 0
10 -  },
11 -   "links": [
12 -     {
13 -       "href": "/v1/app/validate/status/d932f182-2e76-4881-80b1-c3bdb06fd0b1",
14 -       "rel": "self"
15 -     },
16 -     {
17 -       "href": "/v1/app/report/d932f182-2e76-4881-80b1-c3bdb06fd0b1",
18 -       "rel": "report"
19 -     }
20 -   ],
21 - }
```

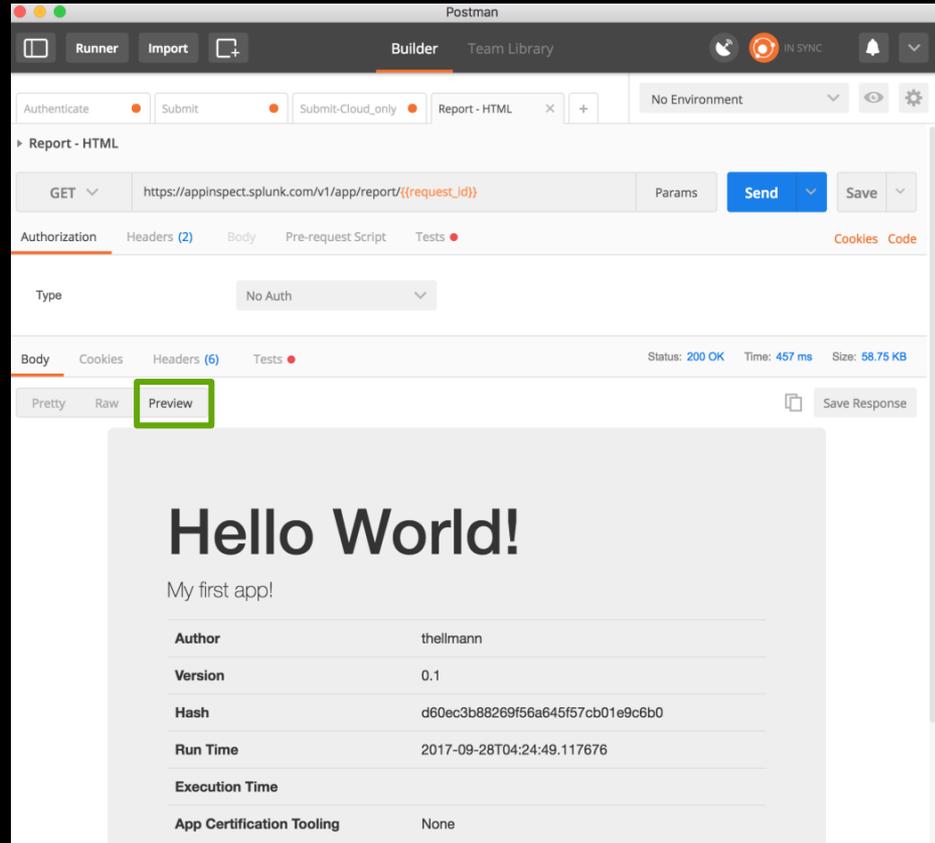
The response status is 200 OK, with a time of 228 ms and a size of 764 B. The response is displayed in the Pretty view.

How can I make my life easier?

API with Postman

Step 4:

- ▶ Get the full report from the report endpoint
- ▶ Click “preview” for a human-readable view
- ▶ `https://appinspect.splunk.com/v1/app/report/{{request_id}}`



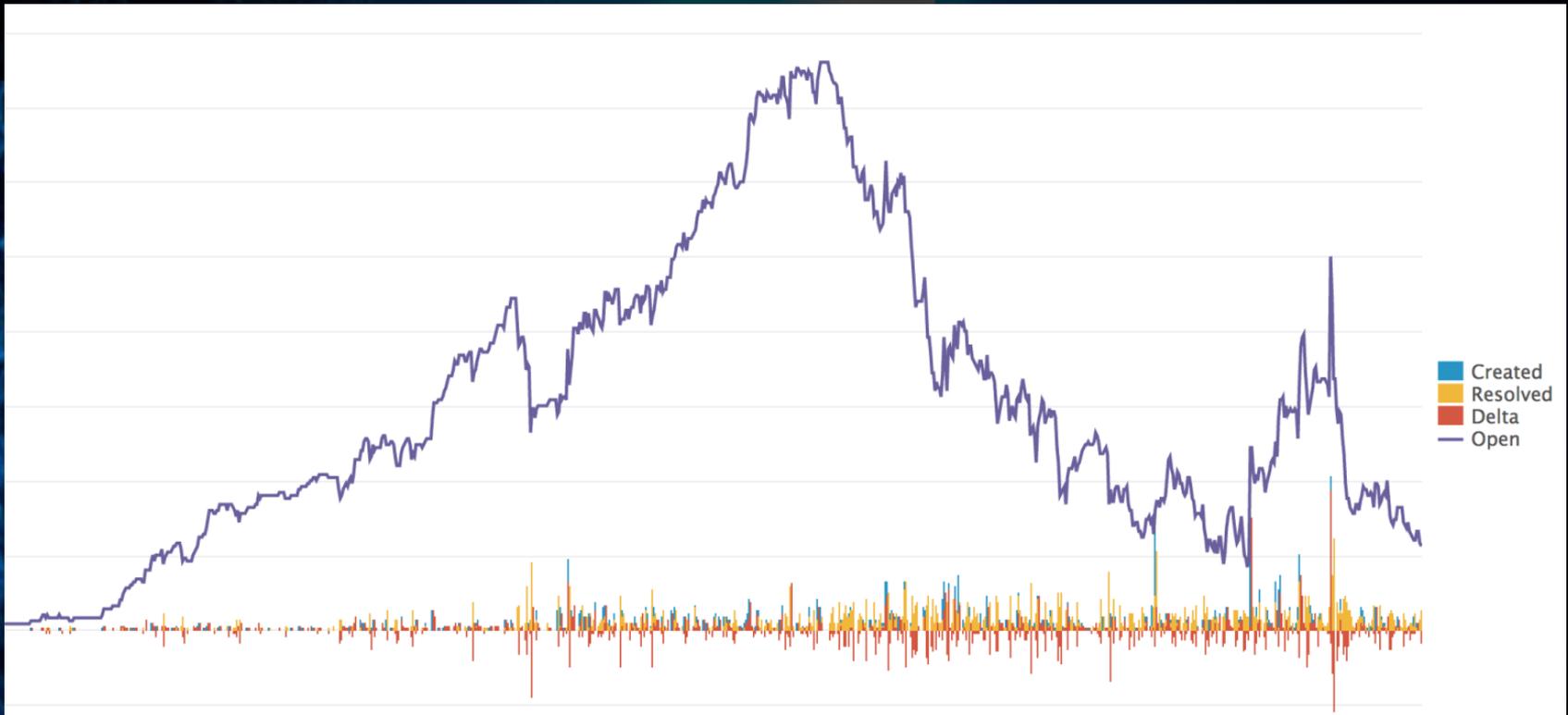
The screenshot shows the Postman interface with a GET request to `https://appinspect.splunk.com/v1/app/report/{{request_id}}` executed. The response is displayed in a human-readable HTML format. The response status is 200 OK, with a response time of 457 ms and a size of 58.75 KB. The response body contains the following HTML:

```

<h1>Hello World!</h1>
<p>My first app!</p>
<table>
<tr>
<td>Author</td>
<td>thellmann</td>
</tr>
<tr>
<td>Version</td>
<td>0.1</td>
</tr>
<tr>
<td>Hash</td>
<td>d60ec3b88269f56a645f57cb01e9c6b0</td>
</tr>
<tr>
<td>Run Time</td>
<td>2017-09-28T04:24:49.117676</td>
</tr>
<tr>
<td>Execution Time</td>
<td></td>
</tr>
<tr>
<td>App Certification Tooling</td>
<td>None</td>
</tr>
</table>

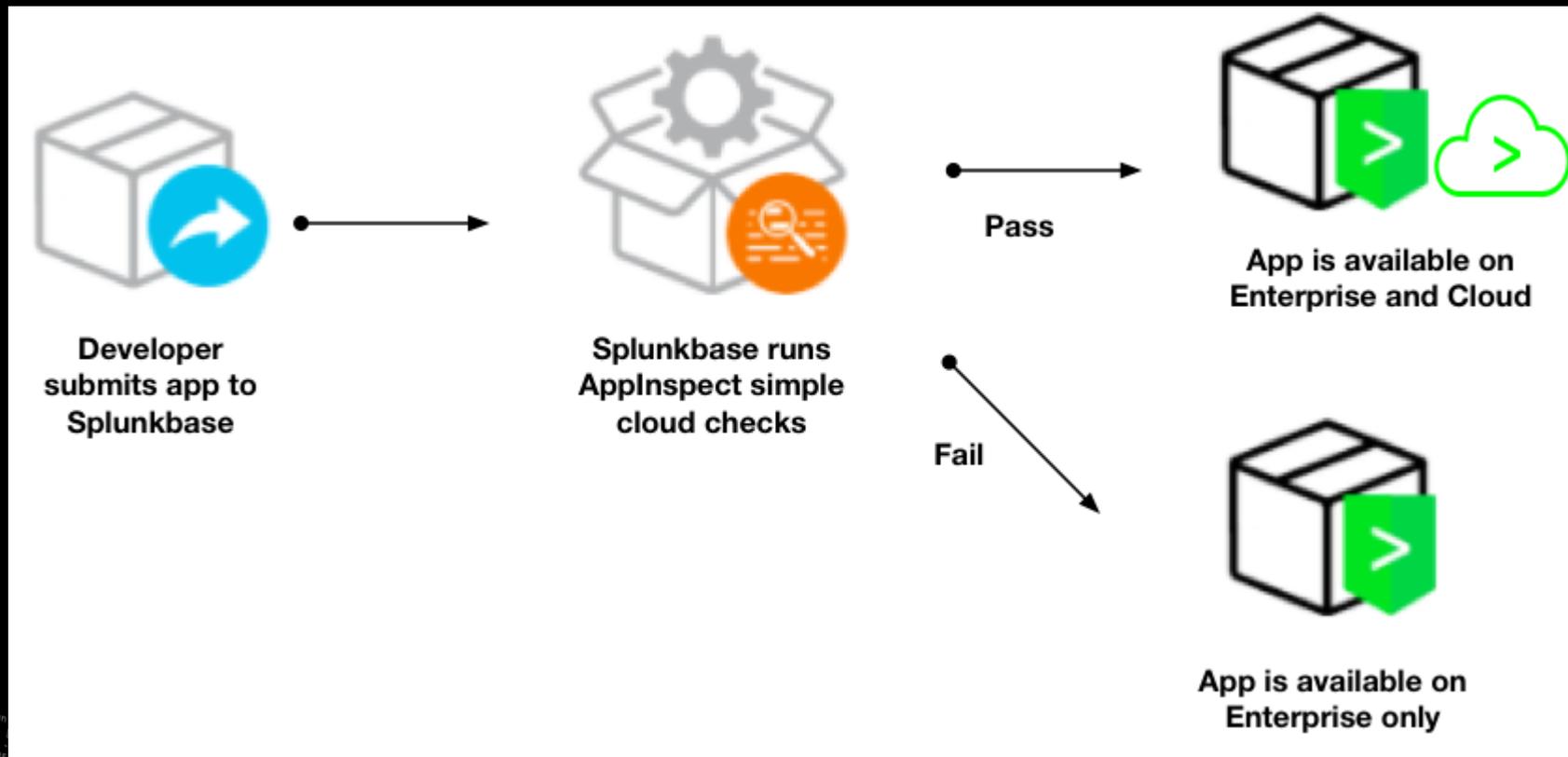
```

How much has AppInspect helped?



How does Cloud Vetting work?

Path 1 - on Splunkbase upload



How does Cloud Vetting work?

Path 1 - on Splunkbase upload

```
appinspect — lknecht@lknecht-mbr15 — ..es/appinspect — -zsh — 130x40  
(venv) lknecht@lknecht-mbr15 ~/Repositories/appinspect <development>  
$ splunk-appinspect inspect --included-tags cloud ~/Desktop/exampl_splunk_app.tgz
```

Postman

Builder Team Library

lknecht - Splunk.com Login Validate Local Machine

Validate Examples (0)

POST https://appinspect.splunk.com/v1/app/validate Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Cookies Code

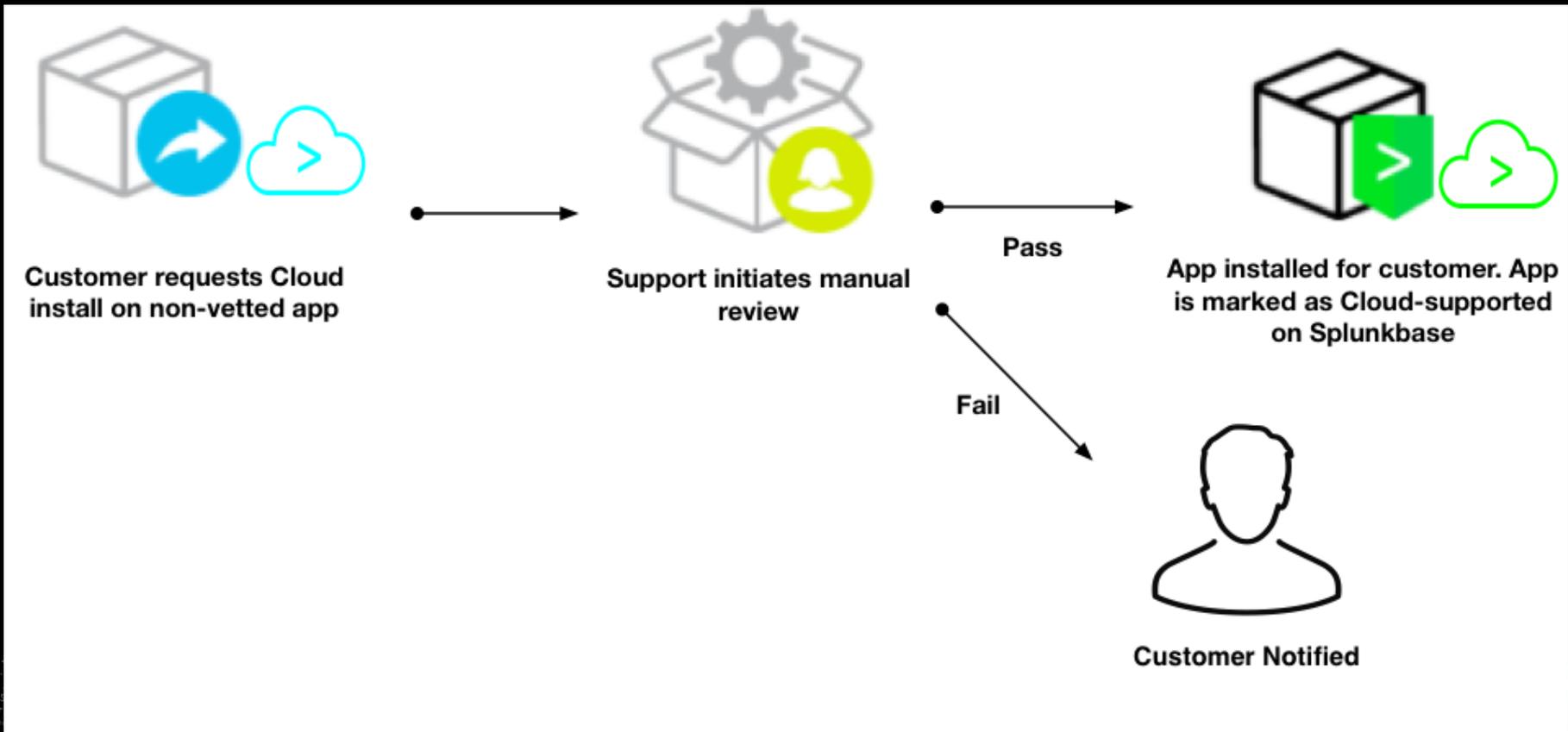
form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> app_package	Choose Files meta-woot_204.tgz	
<input checked="" type="checkbox"/> included-tags	cloud	
New key	Text Value	Description

Body Cookies (1) Headers (6) Tests (2/2) Status: 200 OK Time: 1960 ms Size: 724 B

How does Cloud Vetting work?

Path 3 - on Cloud install request

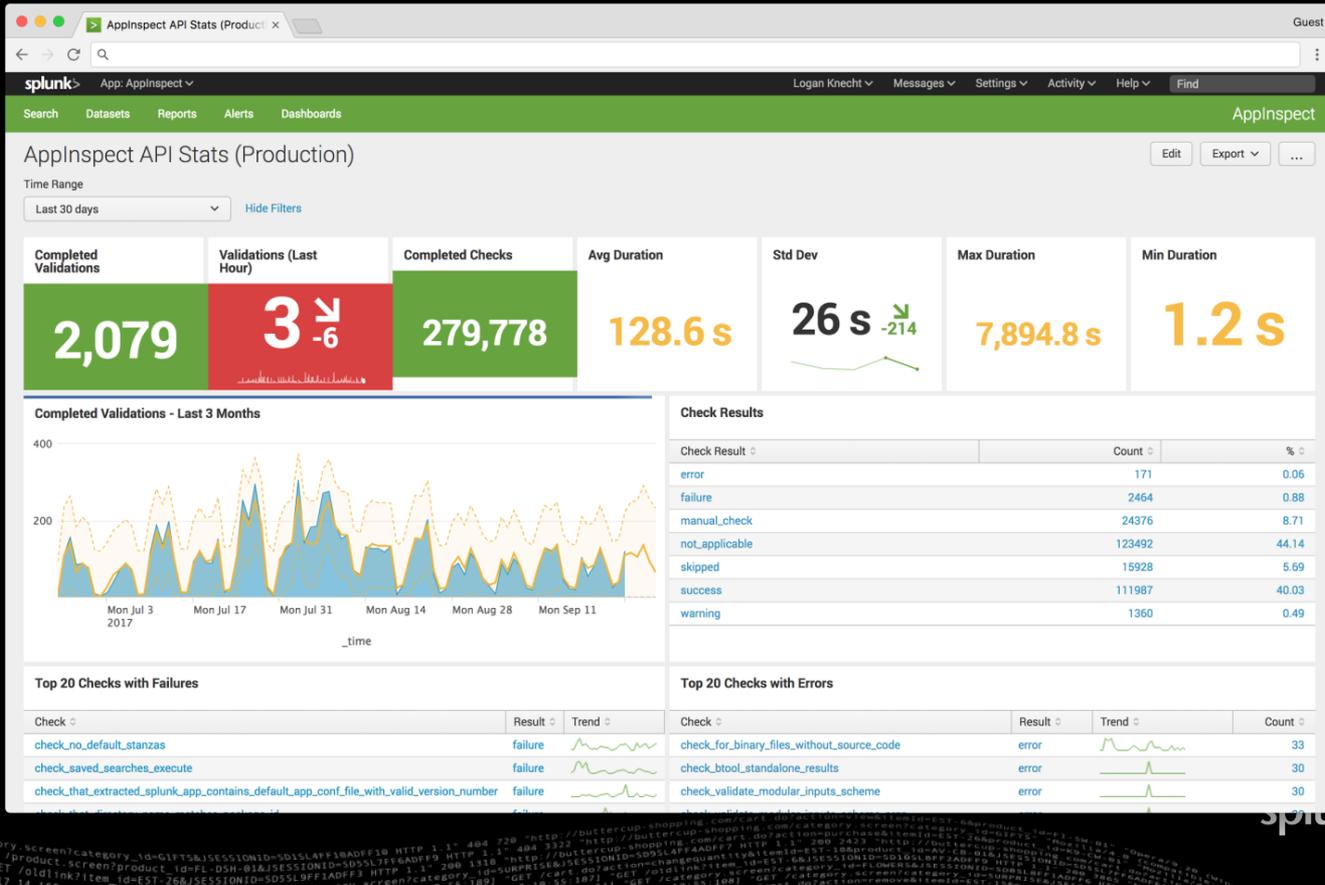




Instrumentation For Your Platform

Andy Nortrup
Product Area Owner Splunk Cloud

General Overview



Top Check Failures

June 2017

Config File Parsing

Packaging Toolkit Validation

btool

How to Speed Up Certification Review Time

Logan Knecht
Software Engineer - Data Ecosystem



Storing Splunk App settings in a custom .conf file

The screenshot shows the Splunk web interface. On the left, a 'FOLDERS' sidebar lists the directory structure: 'developer_guidance_setup_view' (selected), 'appserver', 'default', 'local' (selected), 'app.conf', 'passwords.conf', 'setup_view_example.conf' (selected), 'metadata', and 'static'. On the right, the 'setup_view_example.conf' file is open, showing a configuration stanza:

```
[example_stanza]
api_url=https://example_url.com
```

BAD IDEA



Storing Splunk App settings in Splunk Core specific .conf files

The screenshot shows the Splunk web interface. On the left, a file explorer displays the directory structure under 'splunk'. A red box highlights the 'launcher' folder (labeled '2'), and another red box highlights the 'inputs.conf' file within the 'local' subdirectory (labeled '3'). A blue box highlights a 'REDACTED' entry in the file list (labeled '1'). An arrow labeled '4' points from the 'inputs.conf' file to the configuration editor on the right.

The configuration editor on the right shows the contents of 'inputs.conf':

```

1 [REDACTED://example_stanza]
2 broker_host = localhost
3 clean_session = 0
4 hec_batch_mode = 0
5 hec_https = 0
6 output_type = stdout
7 password = plain_text_password
8 qos = 0
9 sourcetype = syslog
10 topic_name = example_topic
11 use_ssl = 0
12 username = example_username
13

```


BAD IDEA



Ignoring secure communication techniques and features at every possible chance

```

FOLDERS
  REDACTED
  bin
  default
  metadata
  README
  static
  README.md

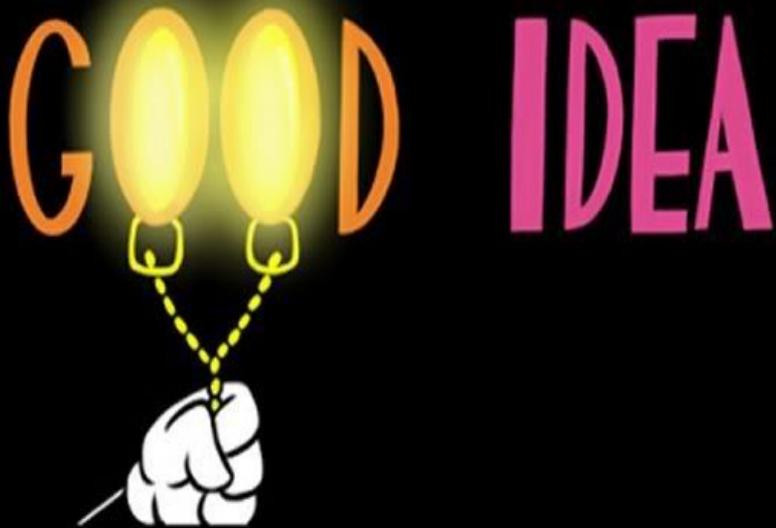
nx_logger.py
90 def send(self, address, port, session, pa
91     header = {'Content-Type': 'applicatio
92             REDACTED': 'sessi
93             'Cookie': 'REDACTED'
94
95     url = "https://{}:{:}{}".format(address
96
97     resp = requests.put(url,
98                       headers=header,
99                       data=json.dumps(p
100                      verify=False)
101
102     return resp.status_code
103
104 def on_connect(self, address, port, sessi
105     self.info('Sending statistics data to
106
107     if self.product is None or self.stati
108         self.error('Invalid product name
109         self.error('Statistics collection
110     return None
  
```

```

FOLDERS
  REDACTED
  appserver
  bin
  default
  data
  /* app.conf
  /* indexes.conf
  /* inputs.conf
  /* props.conf
  /* savedsearches.conf
  /* transforms.conf
  lookups
  metadata
  static
  README.md

inputs.conf
1 [tcp://514]
2 connection_host = dns
3 index = REDACTED
4 sourcetype = syslog
5 disabled = 1
6
  
```

130.60.4 - [07/Jan 18:30:57:153] "GET /category_screen?category_id=GLFTS&SESSIONID=SD5SL9FF1ADFF3 HTTP/1.1" 404 720 "http://buttercup.shopp
128.341.228.02 - [07/Jan 18:10:57:123] "GET /product_screen?product_id=FLD5H-8&SESSIONID=SD9SL7FF6ADFF9 HTTP/1.1" 404 322 "http://buttercu
- 317.27.168.0.0 - [07/Jan 18:10:56:156] "GET /oidlink?item_id=EST-256/SESSIONID=SD5SL9FF1ADFF3 HTTP/1.1" 200 1310 "http://buttercup.shoppin
//buttercup-106product_id=RP-LI-02" 488 125 17 "http://buttercup.com/can
shopping.com/shopping.com/oidlink?item_id=EST-256/SESSIONID=SD5SL9FF1ADFF3 HTTP/1.1" 200 1310 "http://buttercup.com/can
/buttercup.com/can



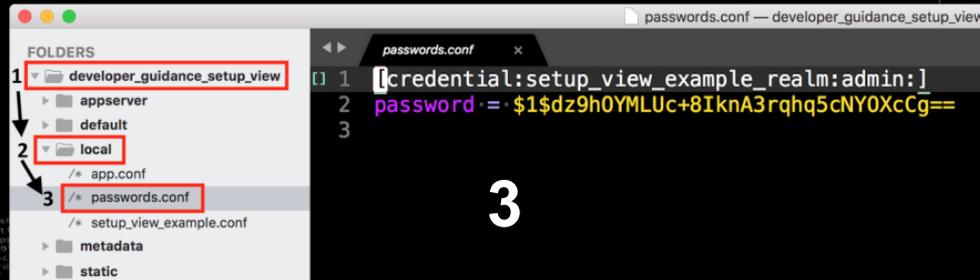
Using the
storage/passwords
endpoint to encrypt
sensitive data

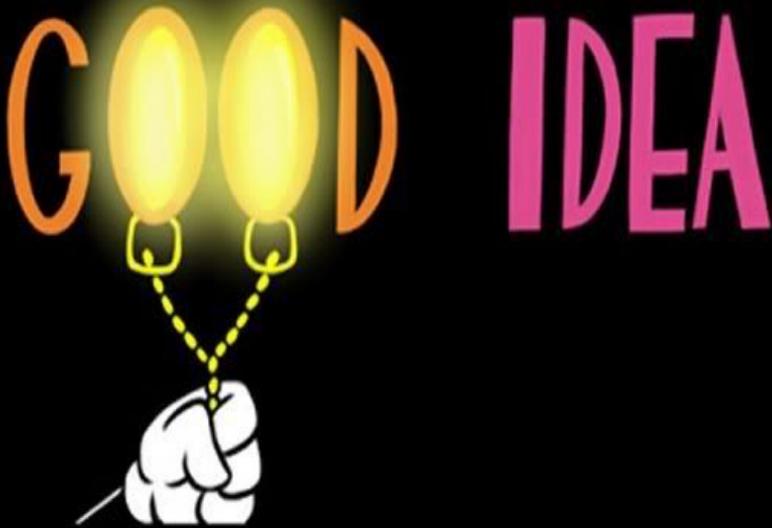
```
encrypt_api_key: async function encrypt_api_key(  
  ... splunk_js_sdk_service,  
  ... api_key,  
) {  
  1
```

```
var does_storage_password_exist = this.does_storage_password_exist(  
  ... storage_passwords_accessor,  
  ... realm,  
  ... username,  
);  
2
```

```
if (does_storage_password_exist) {  
  ... await this.delete_storage_password(  
    ... storage_passwords_accessor,  
    ... realm,  
    ... username,  
  );  
}
```

```
await storage_passwords_accessor.fetch();  
  
await this.create_storage_password_stanza(  
  ... storage_passwords_accessor,  
  ... realm,  
  ... username,  
  ... api_key,  
);
```





Logging to

`$(SPLUNK_HOME)/var/log/<SPLUNK_APP_NAME>/*.log`

```

get_splunk_apps.py — fabrix
FOLDERS
  ▾ fabrix
    ▾ appserver
    ▾ bin
    ▾ /* get_splunk_apps.py
    ▾ default
    ▾ lookups
    ▾ metadata
    ▾ static
    ▾ README

get_splunk_apps.py x
28 rundir = '/var/run/REDACTED'
29 vardir = '/var/log/REDACTED'
30 xmldir = 'default/data/ui/views'
31 xmllocaldir = '/local/data/ui/views'
32 output_file = 'splunk_apps.csv'
33 httproot = ''
34
35 exten = 'default.xml'
36 list_files = []
37 list_apps = []
38
39 # Setup the logger
40 LOG_NAME = "get_splunk_apps"
41
42
43 def setup_logger(splunk_home, log_file):
44     LOG_FILE = splunk_home + vardir + "/" + log_file + ".log"
45     LOG_MAX_BYTES = 10 * (1024 * 1024) # 10MB
46     LOG_COUNT = 3
47     logger = logging.getLogger(__name__)
48     logger.propagate = False # Prevent the log messages from
49     logger.setLevel(logging.DEBUG)
50
51 # file_handler = logging.handlers.RotatingFileHandler(make_s
52     file_handler = logging.handlers.RotatingFileHandler(LOG FI
53     formatter = logging.Formatter('%(asctime)s %(levelname)s %
54     file_handler.setFormatter(formatter)
55
56     logger.addHandler(file_handler)
57
58     return logger
59
60
61 def get_pid(splunk_home):
62     PID_FILE = splunk_home + rundir + "/" + LOG_NAME + ".pid"
63     LOGGER.info("...CRE...%s..." % (PID_FILE))
64     pid = str(os.getpid())
65     if os.path.isfile(PID_FILE):
  
```


BAD IDEA



To achieve this, a local `inputs.conf` should be manually configured or deployed via a Deployment Server to monitor REDACTED.log file which default location is `/var/log/REDACTED/REDACTED.log`

In the `__$SPLUNK_HOME/etc/apps/REDACTED/src/config.xml__` file, enter the tenant name, base download URL and token, for example:

- Asking users to manually edit `.conf` files
- Zero configurability
- Hard-coding opinionated settings

Quick Tangent

Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as setup.xml pages
Setup views are NOT the same as set



```
130.60.4 - [07/Jan 18:30:57:153] "GET /category.screen?category_3d=61f75&SESSIONID=5D95L9FF1ADFF10 HTTP/1.1" 404 720 "http://buttercup-shopping.com/cart.do?action=yView&itemId=EST-88&product_id=EST-88"
128.241.228.82 - [07/Jan 18:30:57:123] "GET /product.screen?product_id=FL-DSH-0&SESSIONID=5D95L9FF1ADFF3 HTTP/1.1" 200 1310 "http://buttercup-shopping.com/cart.do?action=purchase&itemId=EST-88&product_id=EST-88"
317.17.168.0 - [07/Jan 18:30:57:123] "GET /product.screen?product_id=FL-DSH-0&SESSIONID=5D95L9FF1ADFF3 HTTP/1.1" 200 1310 "http://buttercup-shopping.com/cart.do?action=purchase&itemId=EST-88&product_id=EST-88"
107.1.1.9V1: NET CLR 1.1.4322.1 488 125 17 "GET /oidlink?item_id=EST-26&SESSIONID=5D95L9FF1ADFF3 HTTP/1.1" 200 187 "GET /cart.do?action=changequantity&itemId=EST-38&product_id=EST-38"
buttercup-shopping.com/product_id=RP-LI-02" 488 125 17 "GET /category.screen?category_3d=61f75&SESSIONID=5D95L9FF1ADFF3 HTTP/1.1" 404 720 "http://buttercup-shopping.com/cart.do?action=yView&itemId=EST-88&product_id=EST-88"
shopping.com/purchase&itemId=EST-88"
buttercup-shopping.com/cart.do?action=purchase&itemId=EST-88"
buttercup-shopping.com/product_id=RP-LI-02" 488 125 17 "GET /category.screen?category_3d=61f75&SESSIONID=5D95L9FF1ADFF3 HTTP/1.1" 404 720 "http://buttercup-shopping.com/cart.do?action=yView&itemId=EST-88&product_id=EST-88"
```


Developer Guidance - Setup View Example For Splunk



splunkbase

Search App by keyword, technology...

My Account Support & Services

Developer Guidance - Setup View Example For Splunk



0 rating Splunk Certified

Overview Details

The Developer Guidance: Setup View Example For Splunk is an example Splunk App to demonstrate how to use setup views that adhere to Splunk Application Certification standards.

9 Installs 20 Downloads

LOGIN TO DOWNLOAD

Release Notes

Version 1.0.0 Sept. 25, 2017

VERSION 1.0.0

BUILT BY Splunk Works





splunk> enterprise

Username	Password	Sign in
----------	----------	---------



Home | Splunk 6.6.2 x Guest

localhost:8000/en-US/app/launcher/home

splunk> Administrator Messages Settings Activity Help Find

Apps

Search & Reporting

Developer Guidance - Setup View

Explore Splunk Enterprise



Choose a home dashboard

```

130.60.4 - [07/Jan 18:30:57:153] "GET /category.screen?category_id=61f75&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 728
128.241.228.82 - [07/Jan 18:30:57:123] "GET /product.screen?product_id=FL-DSH-81&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 3322
317.27.168.0 - [07/Jan 18:10:57:0] "GET /product.screen?product_id=FL-DSH-81&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 3322
130.60.4 - [07/Jan 18:30:57:153] "GET /category.screen?category_id=61f75&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 728
128.241.228.82 - [07/Jan 18:30:57:123] "GET /product.screen?product_id=FL-DSH-81&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 3322
317.27.168.0 - [07/Jan 18:10:57:0] "GET /product.screen?product_id=FL-DSH-81&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 3322
130.60.4 - [07/Jan 18:30:57:153] "GET /category.screen?category_id=61f75&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 728
128.241.228.82 - [07/Jan 18:30:57:123] "GET /product.screen?product_id=FL-DSH-81&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 3322
317.27.168.0 - [07/Jan 18:10:57:0] "GET /product.screen?product_id=FL-DSH-81&SESSIONID=5D55L9FF1ADFF3 HTTP/1.1" 404 3322

```

Manager - Splunk x Guest

localhost:8000/en-US/app/developer_guidance_setup_view/search

splunk> App: Search & Reporting v Administrator v Messages v Settings v Activity v Help v Find

App configuration

The "Developer Guidance - Setup View" app has not been fully configured yet.

This app has configuration properties that can be customized for this Splunk instance. Depending on the app, these properties may or may not be required.

[Continue to app setup page](#)

About Support File a Bug Documentation Privacy Policy © 2005-2017 Splunk Inc. All rights reserved.

How?

```
app.conf
[install]
is_configured = false
```

setup_view_dashboard | Splunk x Guest

localhost:8000/en-US/app/developer_guidance_setup_view/setup_view_dashboard

splunk> App: Developer Guidance - Setup View Administrator Messages Settings Activity Help Find

Search Datasets Reports Alerts Dashboards Developer Guidance - Setup View

setup_view_dashboard Edit Export ...

Welcome to Your Setup Page!

Overview

This setup view will create/modify three files in the local directory of this Splunk App.

Splunk App Directory Path: '\$SPLUNK_HOME/etc/apps/developer_guidance_setup_view/local'

- app.conf
 - Sets the [install] stanza's 'is_configured' property to 'true'
- passwords.conf
 - Creates and encrypts the API key, resulting in a new passwords.conf stanza.
- setup_view_example.conf
 - Creates a custom conf file to manage Splunk App specific settings
 - Creates the stanza [example_stanza]
 - Creates the property 'api_url' in the [example_stanza] and assigns the 'API URL' to it

Setup Properties

API URL:

Please specify the url that will be used for API requests.

API Key:

Please specify the API Key that will be used to authenticate to the API.

Complete the Setup

Please press the 'Perform Setup' button below to complete the Splunk App setup.

Implementation Details

Overview

This is a setup view that is intended to serve as a point of reference for Splunk Developers. This has been created using Application Certification approved best practices.

Splunk Techniques Used

- Splunk Dashboards
 - [API Documentation](#) (docs.splunk.com)
- Splunk Setup View
 - [app.conf Specification](#)
- Splunk Web Framework
 - [API Documentation](#) (docs.splunk.com)
 - [Main Website](#) (dev.splunk.com)

Technology Used

- CSS
- HTML
- JavaScript
 - Backbone JS
 - [Main Website](#)
 - [On GitHub](#)
 - [Views](#) are the only feature used
 - jQuery
 - [Main Website](#)
 - [On GitHub](#)
 - Splunk JavaScript Software Development Kit
 - [API Documentation](#) (docs.splunk.com)
 - [On GitHub](#)
 - [Main Website](#) (dev.splunk.com)

About Support File a Bug Documentation Privacy Policy

© 2005-2017 Splunk Inc. All rights reserved.

setup_view_dashboard | Splunk x Guest

localhost:8000/en-US/app/developer_guidance_setup_view/setup_view_dashboard

splunk> App: Developer Guidance - Setup View Administrator Messages Settings Activity Help Find

Search Datasets Reports Alerts Dashboards Developer Guidance - Setup View

setup_view_dashboard

Edit Export ...

Welcome to Your Setup Page!

Overview

This setup view will create/modify three files in the local directory of this Splunk App.

Splunk App Directory Path: '\$SPLUNK_HOME/etc/apps/developer_guidance_setup_view/local'

- app.conf
 - Sets the [install] stanza's 'is_configured' property to 'true'
- passwords.conf
 - Creates and encrypts the API key, resulting in a new passwords.conf stanza.
- setup_view_example.conf
 - Creates a custom conf file to manage Splunk App specific settings
 - Creates the stanza [example_stanza]
 - Creates the property 'api_url' in the [example_stanza] and assigns the 'API URL' to it

Setup Properties

API URL:
Please specify the url that will be used for API requests.

API Key:
Please specify the API Key that will be used to authenticate to the API.

Complete the Setup
Please press the 'Perform Setup' button below to complete the Splunk App setup.

localhost:8000/en-US/app/developer_guidance_setup_view/setup_view_dashboard

splunk> App: Developer Guidance - Setup View Administrator Messages Settings Activity Help Find

Search Datasets Reports Alerts Dashboards Developer Guidance - Setup View

setup_view_dashboard Edit Export ...

Welcome to Your Setup Page!

Overview

This setup view will create/modify three files in the local directory of this Splunk App.

Splunk App Directory Path: '\$SPLUNK_HOME/etc/apps/developer_guidance_setup_view/local'

- app.conf
 - Sets the [install] stanza's 'is_configured' property to 'true'
- passwords.conf
 - Creates and encrypts the API key, resulting in a new passwords.conf stanza.
- setup_view_example.conf
 - Creates a custom conf file to manage Splunk App specific settings
 - Creates the stanza [example_stanza]
 - Creates the property 'api_url' in the [example_stanza] and assigns the 'API URL' to it

Setup Properties

API URL:

Please specify the url that will be used for API requests.

API Key:

Please specify the API Key that will be used to authenticate to the API.

Complete the Setup

Please press the 'Perform Setup' button below to complete the Splunk App setup.

[Perform Setup](#)

- The 'API URL' specified was empty. Please provide a value.
- The 'API Key' specified was empty. Please provide a value.

setup_view_dashboard | Splunk x Guest

localhost:8000/en-US/app/developer_guidance_setup_view/setup_view_dashboard

splunk> App: Developer Guidance - Setup View Administrator Messages Settings Activity Help Find

Search Datasets Reports Alerts Dashboards Developer Guidance - Setup View

setup_view_dashboard Edit Export ...

Welcome to Your Setup Page!

Overview

This setup view will create/modify three files in the local directory of this Splunk App.

Splunk App Directory Path: '\$SPLUNK_HOME/etc/apps/developer_guidance_setup_view/local'

- app.conf
 - Sets the [install] stanza's 'is_configured' property to 'true'
- passwords.conf
 - Creates and encrypts the API key, resulting in a new passwords.conf stanza.
- setup_view_example.conf
 - Creates a custom conf file to manage Splunk App specific settings
 - Creates the stanza [example_stanza]
 - Creates the property 'api_url' in the [example_stanza] and assigns the 'API URL' to it

Setup Properties

API URL:

Please specify the url that will be used for API requests.

https://

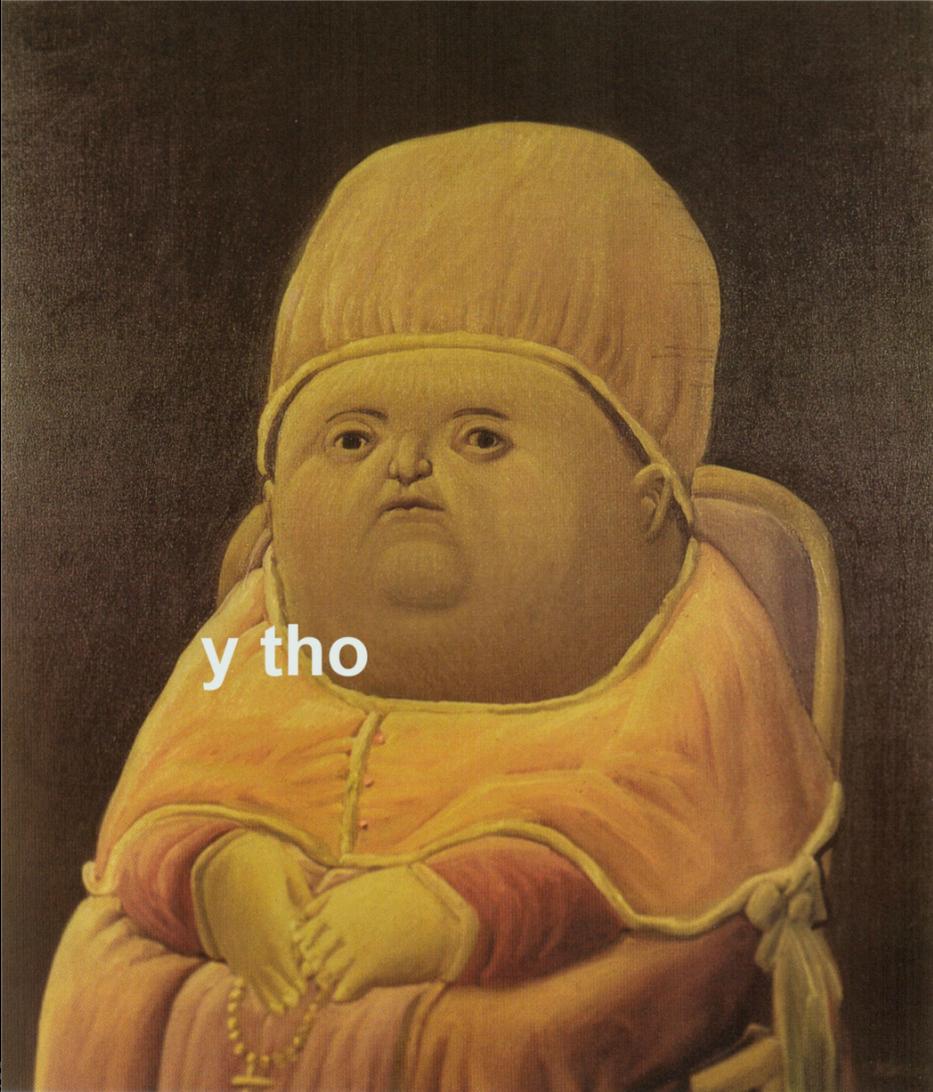
API Key:

Please specify the API Key that will be used to authenticate to the API.

Complete the Setup

Please press the 'Perform Setup' button below to complete the Splunk App setup.

- The 'API URL' specified is using 'http://' at the beginning of it. Please remove the 'http://' and enter the url with out it in 'API URL' field.



y tho

```
130.60.4 - [07/Jan 18:30:57:153] "GET /category.screen?category_3d=61f7s  
128.241.228.82 - [07/Jan 18:30:57:123] "GET /product.screen?product_id=  
317.37.168.0.0 - [07/Jan 18:30:56:156] "GET /oidlink?item_id=EST-256/  
/buttercup-10&product_id=RP-LI-02" 488 125 17  
shopping.com/purchase?shopping.com/n-LI-02" 488 125 17  
/buttercup-10
```


Setup View - app.conf

```
1 #
2 # Splunk app configuration file
3 #
4
5 [install]
6 is_configured = 0 1
7
8 [package]
9 id = developer_guidance_setup_view
10
11 [ui]
12 is_visible = 1
13 label = Developer Guidance -- Setup View
14 # Points to a file in:
15 # $SPLUNK_HOME/bin/apps/<SPLUNK_APP_NAME>/default/data/ui/views/*
16 # It does not need the extension of the file name provided
17 setup_view = setup_view_dashboard 2
18
19 [launcher]
20 author = Logan Knecht
21 description = This is an example Splunk App that demonstrates how to use a setup view to configure
22 version = 1.0.0
23
```

FOLDERS

- developer_guidance_setup_view_exam
- .git
- appserver
- default
 - data
 - ui
 - nav
 - views
 - setup_view_dashboard.xml 3

app.conf

- metadata
- screenshots
- static
- README.md

Setup View - XML Dashboard

setup_view_dashboard.xml — developer_guidance_setup_view_example

```
1 <!--
2 Both of the <dashboard> element's `styles` and `script` attribute search for
3 files in `${SPLUNK_HOME}/etc/<SPLUNK_APP_NAME>/appserver/static/*`
4
5 Documentation link:
6 http://docs.splunk.com/Documentation/Splunk/latest/Viz/PanelreferenceforSimplifiedXML#dashboard\_or\_form
7
8 <dashboard isDashboard="false"
9   script="javascript/setup_page.js"
10  stylesheet="styles/setup_page.css" >
11   <row >
12     <panel >
13       <html >
14         <div id="main_container"></div>
15       </html >
16     </panel >
17   </row >
18 </dashboard >
```

FOLDERS

- developer_guidance_set
- .git
- appserver
 - static
 - javascript
 - views
 - /* setup_page.js
 - /* setup_page.css
 - styles
 - default
 - data
 - ui
 - nav
 - views
- app.conf
- metadata
- screenshots
- static
- README.md

Setup View - Setup Page

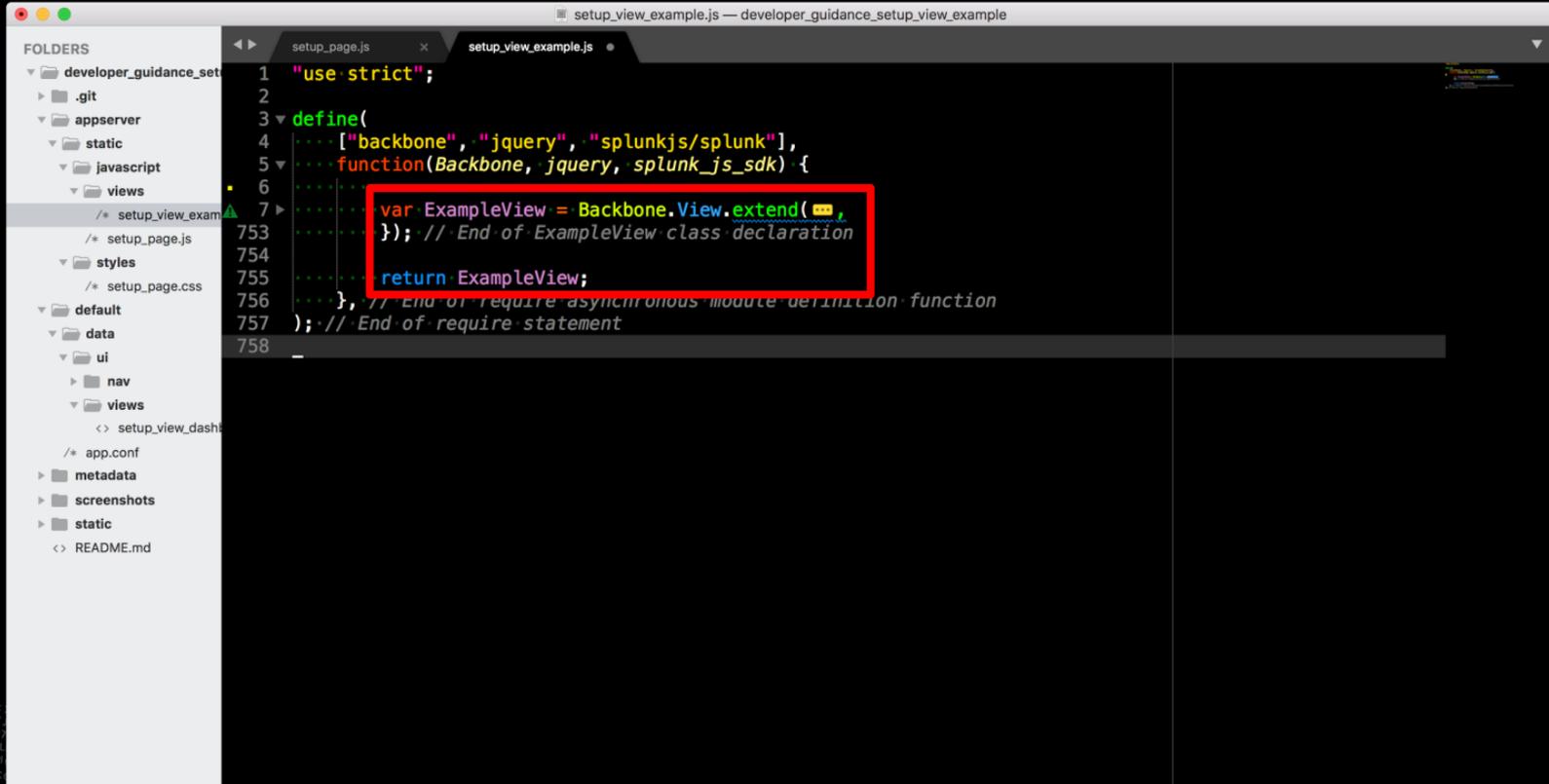
The image shows a code editor window titled "setup_page.js — developer_guidance_setup_view_example". The editor displays the following JavaScript code:

```
3 var app_name = "developer_guidance_setup_view";
4
5 // This is an undocumented Splunk solution for how to include javascript logic
6 // from other files.
7 // Declare a require.config() and provide a dictionary that has a `paths` keys.
8 // In the `paths` key provide another dictionary that declares custom classes.
9 // Each key should be the class name, and the value the path to the javascript file
10 // ../app/<extracted_folder_name_of_app>/<path to javascript file>(no extension)
11 // Then include those in the require method's array, and function.
12
13 // Common gotchas:
14 // 1) The path to the script automatically appends .js, so don't include it
15 // 2) This only provides support for JavaScript files, plain-text/html files won't work
16 // 3) The "../app/" is required as a prefix and your app name is required to follow it
17 // 4) After the app name, the path is provided as though it were from the
18 //    $SPLUNK_HOME/etc/apps/appserver/static/* directory
19 require.config({
20   paths: {
21     // $SPLUNK_HOME/etc/apps/SPLUNK_APP_NAME/appserver/static/javascript/views/setup_view_exam
22     SetupViewExample: "../app/" + app_name + "/javascript/views/setup_view_example",
23   },
24 });
25
26 require([
27   // Splunk Web Framework Provided files
28   "backbone", // From the SplunkJS stack
29   "jquery", // From the SplunkJS stack
30   // Custom files
31   "SetupViewExample",
32 ], function(Backbone, jquery, SetupViewExample) {
33   // new SetupViewExample(
34   // // Sets the element that will be used for rendering
35   // el: jquery("#main_container")
36   // });
37
38   example_setup_view.render();
39 });
```

The file explorer on the left shows the following structure:

- developer_guidance_setup_view_exam
 - .git
 - appserver
 - static
 - javascript
 - views
 - setup_view_example.js
 - styles
 - setup_page.css
 - default
 - metadata
 - screenshots
 - static
 - README.md

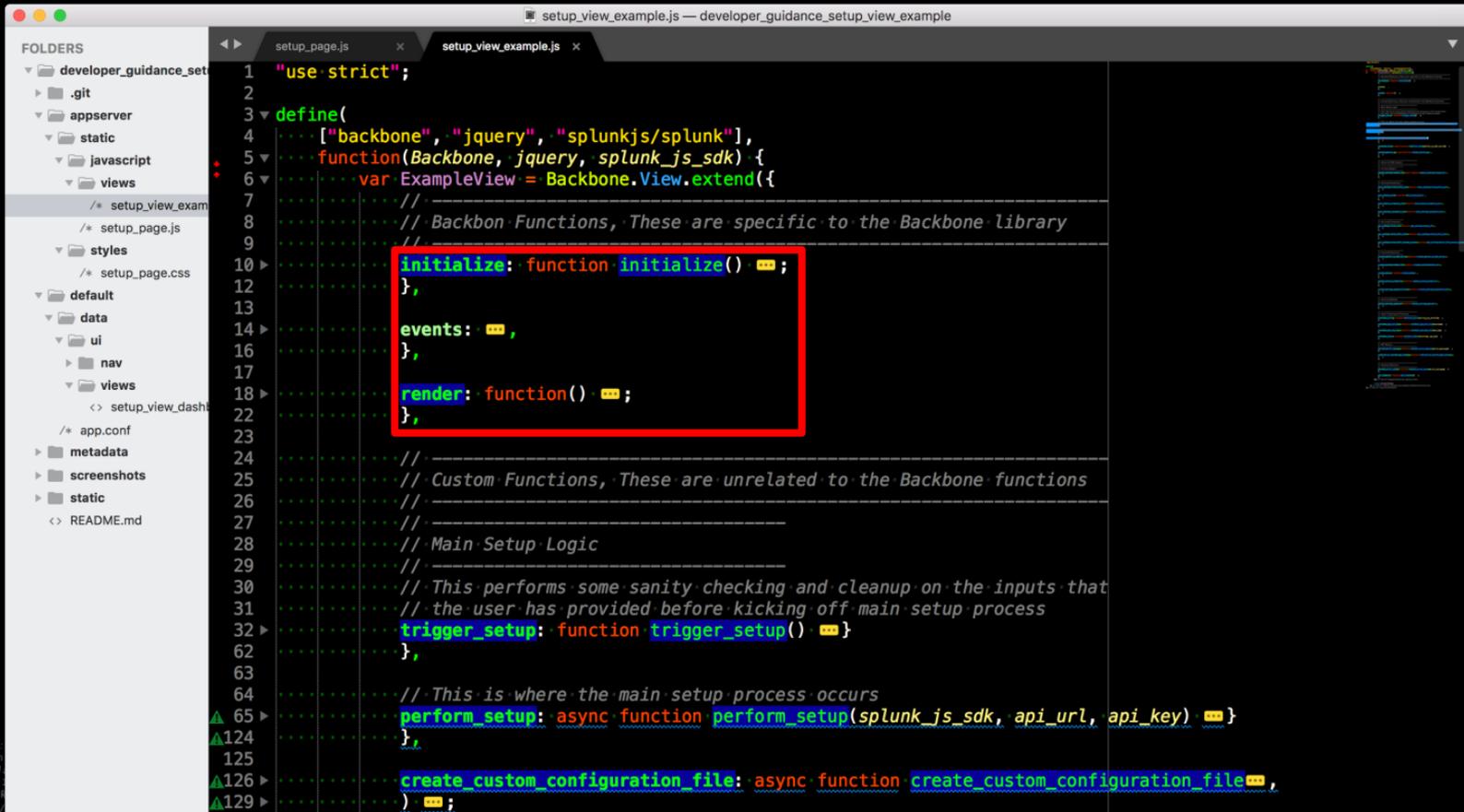
Setup View - Backbone View Page



```
1 "use strict";
2
3 define(
4   ["backbone", "jquery", "splunkjs/splunk"],
5   function(Backbone, jquery, splunk_js_sdk) {
6     var ExampleView = Backbone.View.extend({
7       // ...
8     }); // End of ExampleView class declaration
9     return ExampleView;
10  }, // End of require asynchronous module definition function
11 ); // End of require statement
12
```

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'developer_guidance_set', 'appserver', 'static', 'javascript', 'views', 'styles', 'default', 'data', 'ui', 'nav', 'views', 'app.conf', 'metadata', 'screenshots', 'static', and 'README.md'. The code editor shows the 'setup_page.js' file with the following code:

Setup View - Backbone View Page



```
1  "use strict";
2
3  define(
4    ["backbone", "jquery", "splunkjs/splunk"],
5    function(Backbone, jquery, splunk_js_sdk) {
6      var ExampleView = Backbone.View.extend({
7        //
8        // Backbone Functions, These are specific to the Backbone library
9        //
10     initialize: function initialize() {} ;
11   },
12   events: {},
13   render: function() {} ;
14 },
15 //
16 // Custom Functions, These are unrelated to the Backbone functions
17 //
18 // Main Setup Logic
19 //
20 // This performs some sanity checking and cleanup on the inputs that
21 // the user has provided before kicking off main setup process
22 trigger_setup: function trigger_setup() {}
23 },
24 // This is where the main setup process occurs
25 perform_setup: async function perform_setup(splunk_js_sdk, api_url, api_key) {}
26 },
27 create_custom_configuration_file: async function create_custom_configuration_file(),
28 ) ;
```

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for a project, including folders like 'developer_guidance_set', 'appserver', 'static', 'javascript', 'views', 'styles', 'default', 'data', 'ui', 'nav', 'views', 'metadata', 'screenshots', 'static', and 'README.md'. The code editor displays the contents of 'setup_view_example.js'. A red box highlights the 'initialize', 'events', and 'render' methods of the 'ExampleView' class. The code is written in a style with syntax highlighting and comments. The 'initialize' method is empty, 'events' is an empty object, and 'render' is empty. The 'perform_setup' and 'create_custom_configuration_file' methods are marked as 'async' and are currently empty.

Example Splunk App Resources

- **Example Splunk App Source Code**
 - https://github.com/splunk/Developer_Guidance_Setup_View_Example
- **Example Splunk App on Splunkbase**
 - <https://splunkbase.splunk.com/app/3728/>
- **Splunk Dashboard API**
 - <http://docs.splunk.com/Documentation/Splunk/latest/Viz/PanelreferenceforSimplifiedXML>
- **Splunk Web Framework**
 - **General** - <http://dev.splunk.com/webframework>
 - **API** - <http://docs.splunk.com/Documentation/WebFramework>
- **Backbone JS View**
 - <http://backbonejs.org/#View>

