# Table of Contents

**Introduction**

Overview of the Splunk Add-on Builder ............................................................... 1
Support and resources for the Splunk Add-on Builder ........................................... 3

**Get started**

Install the Add-on Builder .................................................................................... 4
What to know before you build add-ons .............................................................. 19

**Use the Add-on Builder**

Use the Splunk Add-on Builder ............................................................................ 21
Import and export an add-on project ................................................................... 22
Create an add-on .................................................................................................. 24
Configure data collection using a REST API call ................................................. 27
Configure data collection using a shell command ................................................ 37
Configure data collection using your Python code .............................................. 41
Create a setup page ............................................................................................. 46
Manage source types ......................................................................................... 48
Extract fields ........................................................................................................ 51
Map to data model ............................................................................................... 58
Create alert actions ............................................................................................. 60
Validate and package .......................................................................................... 68

**Use the Add-ons**

Use the add-on .................................................................................................... 73

**Advanced**

Create a new data model ..................................................................................... 76
Modify conf files directly ..................................................................................... 78
Python helper functions ....................................................................................... 78

**Release Notes**

What's new .......................................................................................................... 84
Known issues for Splunk Add-on Builder ............................................................. 87
Fixed issues for Splunk Add-on Builder ............................................................... 88
Third-party software credits for Splunk Add-on Builder ..................................... 89
Introduction

Overview of the Splunk Add-on Builder

The Splunk Add-on Builder is a Splunk app that helps you build and validate technology add-ons for your Splunk platform deployment.

What is an add-on?

An add-on is a reusable Splunk component, much like an app, but is dedicated to a function such as getting a specific system’s data in and out of Splunk Enterprise. Add-ons can include any combination of custom configurations, scripts, data inputs, custom reports or views, and themes that can change the look and feel of Splunk Enterprise. A single add-on can be used in multiple apps, suites, or solutions.

Technology add-ons are specialized add-ons that help to collect, transform, and normalize data feeds from specific sources in your Splunk environment. Technology add-ons typically include:

- Knowledge management components, such as field extractions, transforms, and lookups, that make the data easy to use.
- Knowledge mapping components, such as event types and tags, that normalize the data to the Common Information Model.
- Configurations and/or tools to gather data from a source.

For more about apps and add-ons, see Develop Splunk Apps on the Splunk Developer Portal.

Why create a technology add-on?

Technology add-ons provide knowledge mapping, making it easier to do data transformations on unstructured data and extract value from raw event data. Many add-ons help collect the data from data sources, reducing the time spent to reach value. And, add-ons are particularly useful when you need to get data into the Splunk platform and the data is not in one of the native input formats.
Why use the Splunk Add-on Builder?

The Splunk Add-on Builder is intended to guide you through the process of creating a technology add-on without you having to know everything there is to know about the Splunk platform.

The goals of the Splunk Add-on Builder are to:

- Guide you through all of the necessary steps of creating an add-on
- Build alert actions and adaptive response actions for Splunk Enterprise Security
- Reduce development and testing time
- Follow best practices and naming conventions
- Maintain data model (including CIM) compliance
- Maintain quality of add-ons
- Validate and test the add-on, helping you to check for release readiness and to identify any limitations such as compatibilities and dependencies
- Maintain a consistent look and feel while still making it easy for you to add branding
- Package the add-on and helps you get ready to publish the add-on

Who is the Splunk Add-on Builder for?

The Splunk Add-on Builder is for:

- Splunk admins who would like to onboard additional data into the Splunk platform.
- Developers who are looking for a tool to help them build and validate a Splunk add-on.

DISCLAIMER

The Splunk Add-on Builder is intended for on-premises customers and developers only. It is intended for those interested in developing Splunk Add-ons and should not be used in a production environment. If you are using the Splunk Add-on Builder with any third-party add-on, or component thereof, you are responsible for ensuring that your actions comply with the applicable third-party license terms.

- Splunk Add-on Builder is not compatible with search head clusters or Splunk Cloud. As a best practice, use Splunk Add-on Builder in a development environment with a single instance on-premises Splunk
Enterprise to produce add-ons. After you package an add-on, test it on other deployment environments before running it in production.
- Using the Splunk Add-on Builder requires you to be a member of the Admin role.

Support and resources for the Splunk Add-on Builder

The following sections provide additional information and resources for the Splunk Add-on Builder.

Walkthrough

Use the Splunk Add-on Builder walkthrough on the Splunk Developer Portal.

Support

Get support for the Splunk Add-on Builder:

- Ask questions and get answers through community support at Splunk Answers.
- If you have a support contract, log a case using the Splunk Support Portal.
- If you have a support contract, contact customer support.

Resources

Access additional resources for the Splunk Add-on Builder:

- Read the Known issues for Splunk Add-on Builder in this manual.
- Read the Troubleshooting Manual for Splunk Enterprise.
- Join the Splunk user group Slack channel.
- Start a training or certification track on Splunk Education.
- Access more community resources on the Splunk Community page.
Get started

Install the Add-on Builder

The Splunk Add-on Builder is only intended for use on a single-instance development environment, such as a laptop.

Do not install the Splunk Add-on Builder app in a production Splunk environment with real users.

Do not install the Splunk Add-on Builder app in a search head cluster or index cluster environment.

Requirements

Requirements for the Splunk Add-on Builder:

- **Splunk Enterprise**
  
  Splunk Enterprise 6.5 or later is required, running on a single-instance development environment such as a laptop.

  Splunk Light is not supported.

  Add-on Builder is not supported in a search head cluster or index cluster environment.

  The Add-on Builder does not require a paid Enterprise license. You can use the Free license. However, consider getting a Splunk Developer License to work with more data.

  For more about Splunk Enterprise system requirements, see System Requirements in the Splunk Enterprise *Installation Manual*.

- **Mozilla Firefox, Google Chrome, or Microsoft Internet Explorer 11**
- **Admin privileges on Splunk Enterprise**
  
  Without admin privileges, you cannot create an add-on project, or create and run modular inputs for data collection.

- **The Splunk Common Information Model (CIM) add-on**
The CIM add-on version 4.6 or later is required for CIM mapping and creating adaptive response alerts.

- Download the Splunk Common Information Model add-on from Splunkbase.
- For details about installation, see Install the Splunk Common Information Model Add-on.

The CIM add-on 4.6 or later is also required for end users of your add-on to run adaptive response alerts.

**Performance tips**

- While developing data inputs, consider disabling those inputs that you do not need. Multiple data inputs scheduled to run at frequent intervals can consume system resources.
- Running more than 10 add-ons in your development environment might negatively impact Add-on Builder's performance.

**Download and install the Splunk Add-on Builder app**

1. Download the Splunk Add-on Builder from Splunkbase.
2. Log into Splunk Web.
3. Click the Manage Apps icon.
4. On the Apps page, click Install app from file.
5. Click Choose File, navigate to and select the Splunk Add-on Builder package file, then click Open.
6. Click Upload.

To install the Splunk Add-on Builder app from the command line:

- At the command line, enter:

  splunk install app <path>/<packagefilename>

- Alternatively, unpack the file and then copy the /splunk_app_addon-builder directory to $SPLUNK_HOME/etc/apps.

Ensure that the /splunk_app_addon-builder directory and its files have proper permissions and ownership so that Splunk Enterprise can read and write to them.
Upgrade the Splunk Add-on Builder app

When upgrading the Add-on Builder app, use Splunk Web.

**Note** Do not delete the $SPLUNK_HOME/etc/apps/splunk_app_addon-builder folder directly before upgrading the app, or else your previously-created Add-on Builder projects will no longer be configurable within the Add-on Builder.

1. Download the Splunk Add-on Builder from Splunkbase.
2. Log into Splunk Web.
3. Click the Manage Apps icon.
4. On the Apps page, click **Install app from file**.
5. Click **Choose File**, navigate to and select the Splunk Add-on Builder package file, then click **Open**.
6. Select **Upgrade app**.
7. Click **Upload**.

Share data in Splunk Add-on Builder

The first time you run Add-on Builder, a message is displayed requesting you to opt in to send anonymized usage data to Splunk to help improve the app in future releases.

*How data is collected*

If you opt in, the app enables an internal library to track basic usage and crash information. The library uses browser cookies to track uniqueness and sessions of app users, then sends events to Splunk using XHR in JSON format.

*Usage data that is collected*

Fields common to all events

The data that the Add-on Builder sends to Splunk includes the following fields that are common to all events.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
<th>Example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiKey</td>
<td>string</td>
<td>The MINT API key for the Splunk Add-on</td>
<td>&quot;4t2fk73n&quot;</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
<td>Value</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>appRunningState</td>
<td>string</td>
<td>Not used.</td>
<td>&quot;NA &quot;</td>
</tr>
<tr>
<td>appVersionCode</td>
<td>string</td>
<td>Not used.</td>
<td>&quot;NA&quot;</td>
</tr>
<tr>
<td>appVersionName</td>
<td>string</td>
<td>The version of the Splunk Add-on Builder.</td>
<td>&quot;2.1.0&quot;</td>
</tr>
<tr>
<td>browser</td>
<td>string</td>
<td>The browser name.</td>
<td>&quot;chrome&quot;</td>
</tr>
<tr>
<td>browserVersion</td>
<td>string</td>
<td>The browser version.</td>
<td>&quot;47.0.2526.111&quot;</td>
</tr>
<tr>
<td>carrier</td>
<td>string</td>
<td>Not used.</td>
<td>&quot;NA&quot;</td>
</tr>
<tr>
<td>connection</td>
<td>string</td>
<td>Not used.</td>
<td>&quot;NA&quot;</td>
</tr>
<tr>
<td>device</td>
<td>string</td>
<td>The type of device used.</td>
<td>&quot;MacIntel&quot;</td>
</tr>
<tr>
<td>extraData</td>
<td>object</td>
<td>The version of Splunk Enterprise.</td>
<td>&quot;splunk_version&quot;: &quot;6.3.2&quot;</td>
</tr>
<tr>
<td>locale</td>
<td>string</td>
<td>The user locale set in the browser.</td>
<td>&quot;en-US&quot;</td>
</tr>
<tr>
<td>osVersion</td>
<td>string</td>
<td>The version of the operating system.</td>
<td>&quot;OS X 10.11.2&quot;</td>
</tr>
<tr>
<td>packageName</td>
<td>string</td>
<td>The package name of the Splunk Add-on Builder.</td>
<td>&quot;splunk_app_addon-builder&quot;</td>
</tr>
<tr>
<td>platform</td>
<td>string</td>
<td>The platform.</td>
<td>&quot;web&quot;</td>
</tr>
<tr>
<td>remoteIP</td>
<td>string</td>
<td>Not used.</td>
<td>&quot;NA&quot;</td>
</tr>
<tr>
<td>screenOrientation</td>
<td>string</td>
<td>Not used.</td>
<td>&quot;NA&quot;</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td>Data (along with common fields)</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>sdkVersion</td>
<td>string</td>
<td>The version of the internal library.</td>
<td>&quot;4.3&quot;</td>
</tr>
<tr>
<td>session_id</td>
<td>string</td>
<td>A unique session identifier.</td>
<td>&quot;a5026251&quot;</td>
</tr>
<tr>
<td>state</td>
<td>string</td>
<td>Indicates whether the browser is online.</td>
<td>&quot;CONNECTED&quot; or &quot;DISCONNECTED&quot;</td>
</tr>
<tr>
<td>userIdentifier</td>
<td>string</td>
<td>Not used.</td>
<td>&quot;NA&quot;</td>
</tr>
<tr>
<td>uuid</td>
<td>UUID</td>
<td>A random identifier that tracks the user uniqueness.</td>
<td>&quot;b1da8edd-6eb5-4620-8fed-6f2e01e2800f&quot;</td>
</tr>
</tbody>
</table>

**Additional fields**

In addition to **fields common to all events**, the Splunk Add-on Builder sends the following fields to Splunk when usage data is enabled.

<table>
<thead>
<tr>
<th>Event</th>
<th>Source Type</th>
<th>Description</th>
<th>Data (along with common fields)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session start</td>
<td>mint:ping</td>
<td>Each ping event indicates that a new session has started.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>fsEncrypted</td>
<td>N/A Not used, always &quot;NA&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rooted</td>
<td>N/A Not used, always false</td>
</tr>
<tr>
<td>Session end</td>
<td>mint:gnip</td>
<td>Each gnip event indicates that a session has ended.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ses_duration</td>
<td>int How long the session lasted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>current</td>
<td>string The URL of the current web page, without the hostname.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>currentView</td>
<td>string Not used. Hardcoded</td>
</tr>
</tbody>
</table>

**Page views**

mint:view Triggered once per page view in the app.
| **App performance and configuration** | mint:log | Usage and performance logs for the Add-on Builder that track dashboard memory usage, dashboard loading times, the number of accounts, inputs, and regions configured in the app, and non-sensitive input configuration parameters. | **level** | int | Log level. For example, 60 means "error". |
| | | | | log_name | any | Log content. See examples below. |
| **API calls** | mint:network | XMLHttpRequest calls, usually HTTP | **failed** | boolean | Indicates whether the |
API calls from client side (browser) to the Splunk server.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>latency</td>
<td>int</td>
<td>Time spent before the response was received.</td>
</tr>
<tr>
<td>protocol</td>
<td>string</td>
<td>Network protocol (http or https).</td>
</tr>
<tr>
<td>requestLength</td>
<td>string</td>
<td>N/A. Not used.</td>
</tr>
<tr>
<td>responseLength</td>
<td>int</td>
<td>The size of the response.</td>
</tr>
<tr>
<td>statusCode</td>
<td>string</td>
<td>HTTP response code.</td>
</tr>
<tr>
<td>url</td>
<td>string</td>
<td>The request URL, without the hostname.</td>
</tr>
</tbody>
</table>

Example app performance and configuration events

The Add-on Builder sends performance and configuration information using the `log_name` field in the `mint:log` source type. This `log_name` field contains two sub-fields, `name`, which indicates which type of logs are being transmitted, and `data`, the content of the tracking log.

The following examples demonstrate what data the Add-on Builder sends for each type of event.

<table>
<thead>
<tr>
<th>Log name</th>
<th>Example JSON object</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>track_performance</code></td>
<td>{</td>
</tr>
<tr>
<td>Tracks dashboard performance for memory usage and loading times.</td>
<td>&quot;memory&quot;:{</td>
</tr>
<tr>
<td></td>
<td>&quot;totalJSHeapSize&quot;:72200000,</td>
</tr>
<tr>
<td></td>
<td>&quot;usedJSHeapSize&quot;:39600000,</td>
</tr>
<tr>
<td></td>
<td>&quot;jsHeapSizeLimit&quot;:1620000000</td>
</tr>
<tr>
<td></td>
<td>},</td>
</tr>
</tbody>
</table>
"timing": {
  "navigationStart": 1453273923766,
  "unloadEventStart": 1453273923929,
  "unloadEventEnd": 1453273923930,
  "redirectStart": 0,
  "redirectEnd": 0,
  "fetchStart": 1453273923766,
  "domainLookupStart": 1453273923766,
  "domainLookupEnd": 1453273923766,
  "connectStart": 1453273923766,
  "connectEnd": 1453273923766,
  "secureConnectionStart": 0,
  "requestStart": 1453273923773,
  "responseStart": 1453273923927,
  "responseEnd": 1453273923929,
  "domLoading": 1453273923939,
  "domInteractive": 1453273923975,
  "domContentLoadedEventStart": 1453273923975,
  "domContentLoadedEventEnd": 1453273923975,
  "domComplete": 1453273926985,
  "loadEventStart": 1453273926985,
  "loadEventEnd": 1453273926987
}

### track_addon_builder_summary

Tracks information about add-ons created by the Add-on Builder.

```
[
  {
    "author": "splunk",
    "id": "Splunk_TA_test_panda",
    "is_validated": "not validated",
    "last_modified": "2016/10/26",
    "name": "test panda",
    "version": "1.0.0",
    "visible": true
  }
]
```

### track_addon_summary

Tracks information about one add-on, such as data inputs, alert actions, event types, CIM mappings, source types, and so forth.

```
{
  "sourcetype": [
    {
      "data_format": "unstructured_data",
      "event_count": "4584",
      "has_cimmapping": false,
      "has_field_extraction": false,
      "input_name": "dsfsdafdsa",
      "input_type": "rest",
      "sourcetype": "fdsafdsafdsafdsa"
    }
  ],
  "modular_alert": [
    {
      "description": "Send HipChat room notifications",
      "short_name": "hipchat",
      "label": "HipChat",
      "active_response": {
```
"group": [
    "information gathering",
    "information conve"
],
"task": [
    "block"
],
"technology": [
    {
        "product": "endpoint",
        "version": [
            "1.0"
        ],
        "vendor": "symantec"
    },
    "subject": [
        "router",
        "firewall"
    ]
],
"alert_props": {
    "payload_format": "json",
    "is_custom": 1
},
"parameters": [
    {
        "default_value": "message",
        "name": "notification_type",
        "required": true,
        "help_string": "Choose style of HipChat notification.",
        "possible_values": {
            "Application Card": "card",
            "Message": "message"
        },
        "label": "Notification Style",
        "format_type": "dropdownlist"
    },
    {
        "label": "Card Attributes",
        "required": false,
        "help_string": "",
        "name": "card_attributes",
        "format_type": "text"
    },
    {
        "label": "Message Format",
        "required": false,
        "possible_values": {
            "Html": "html",
            "Plain Text": "plain"
        }
    }
]
```
{
    "name": "message_format",
    "format_type": "radio"
},
{
    "default_value": "red",
    "name": "color",
    "required": false,
    "help_string": "",
    "possible_values": {
        "None": "",
        "Purple": "purple",
        "random": "random",
        "Grey": "grey",
        "Yellow": "yellow",
        "Green": "green",
        "Red": "red"
    },
    "label": "Message Color",
    "format_type": "dropdownlist"
},
{
    "label": "Notify users in the room",
    "required": false,
    "name": "notify",
    "format_type": "checkbox"
}
}
```

---

**track_package**
Tracks whether the user has downloaded the add-on package.

```json
{
    "app_name": 'test addon'
}
```

**track_validation**
Tracks validation performance and information.

```json
{
    "timeStart": 1477547657730,
    "timeEnd": 1477547658745,
    "score": 90,
```
track_field_extraction
Tracks field extraction performance and information.

track_step_view
Tracks which steps the user has accessed, and possibly the amount of time spent on each step.

track_step_view_wizard
Tracks which steps the user has accessed in a wizard, and possibly the amount of time spent on each step.

track_creation
Tracks the creation of add-on items, such as data inputs and alert actions.

\{ type: 'add-on',
  data: {
    "friendlyName": "panda",
    "projectAuthor": "splunk",
    "projectDescription": "a test add-on",
    "projectName": "panda",
    "projectNamePrefix": "Splunk_TA_",
    "projectVersion": "1.0.0",
    "themeColor": "#65A637",
    "visible": 1
  }
}\n
//or
\{ type: 'modular-input',
  data: {
    "type": "rest",
    "sourcetype": "test:rest",
    "name": "test_rest",
    "options": {
      "serviceName": "test_rest",
      "url": "http://localhost:8089/services/test_rest",
      "fields": ["field1", "field2"]
    }
  }
\}
"title": "test rest",
"description": "test rest description",
"interval": "30",
"data_inputs_options": [{
   "name": "_rest_api_url",
   "rest_header": false,
   "description": "url",
   "value": "http://www.sina.com.cn"
}, {
   "name": "_rest_api_method",
   "rest_header": false,
   "description": "method",
   "value": "GET"
}, {
   "type": "customized_var",
   "name": "string_label",
   "title": "Text",
   "description": ",",
   "required_on_edit": true,
   "required_on_create": true
}],
"customized_options": [{
   "name": "string_label",
   "value": "123"
}]
}

//or

{"type": 'modular-input'.
 data: {
   "type": "command",
   "sourcetype": "test:command",
   "name": "test_command",
   "title": "test command",
   "description": "this is a test",
   "interval": "30",
   "data_inputs_options": [{
      "name": "_command",
      "description": "command",
      "value": "ls -l"
   }, {
      "type": "customized_var",
      "name": "string_label",
      "title": "Text",
      "description": "",
      "required_on_edit": true,
      "required_on_create": true
   }],
   "customized_options": [{
      "name": "string_label",
      "value": "123"
   }]
}
//or
{
  type: 'modular-input',
  data: {
    "type": "customized",
    "sourcetype": "test:code",
    "name": "test_code",
    "title": "test code",
    "description": "test code description",
    "interval": "30",
    "data_inputs_options": [{
      "type": "customized_var",
      "name": "string_label",
      "title": "Text",
      "description": "",
      "required_on_edit": true,
      "required_on_create": true
    }],
    "customized_options": [{
      "name": "string_label",
      "value": ""
    }]
  }
}

//or
{
  "type": "sourcetype",
  "data": {
    "sourcetype": "test:st:abc",
    "key_values": "({"CHARSET":"UTF-8","SHOULD_LINEMERGE":"0","pulldown_type":"1","category":"Splunk App Add-on Builder"})",
    "from_splunk": false
  }
}

//or
{
  "type": "event-type",
  "data": {
    "eventtype": "test_event_type",
    "sourcetype": "test:st:abc",
    "searchstr": ""
  }
}
// or
{
  "type": "cim-mapping",
  "data": {
    "event_type": "test_eventtype",
    "event_field": "date_year",
    "eval": "date_year",
    "cim": "Alerts",
    "cim_field": "dest"
  }
}
// or
{
  "type": "modular-alert",
  "data": {
    "parameters": [{
      "required": false,
      "default_value": ",",
      "name": "password",
      "label": "Password",
      "help_string": "",
      "format_type": "password"
    }],
    "short_name": "test_alert",
    "label": "test alert"
  }
}

track_deletion
Tracks deletion of add-on items.

{
  type: 'add-on',
  data: {
    "app_name": "Splunk_TA_panda"
  }
}

// or
{
  type: 'modular-input'.
  data: {
    "type": "customized",
    "name": "test_code"
  }
}

// or
{
  "type": "sourcetype",
  "data": {
    "sourcetype": "test:st:abc"
  }
}
track_error
Tracks errors that were triggered in the Add-on Builder.

```
{
  err_code: '6099',
  err_args: ['123']
  message: 'Cannot create a modular input with name "123". Input name must start with an alphabetic character, following by alphanumeric characters.'
}
```

Data that is not collected

The following kinds of data are not collected:

- Sensitive data, such as usernames or passwords.
- Identifying information, such as addresses, phone numbers, IP addresses, or hostnames.
- Indexed data in your Splunk platform instance.

How to opt in or out of data collection

The Add-on Builder displays an opt-in request the first time you run Add-on Builder. You can change your settings at any time on the Configuration > Make Add-on Builder Better tab, which is available from the Splunk Add-on Builder home page.

For more information about how Splunk collects and uses data, please refer to the Splunk Privacy Policy.
What to know before you build add-ons

Before you begin to build an add-on:

- Be familiar with your data and know what you want to extract from it.
- Consider how you intend to gather your data. If you are going to use file monitors, network listeners, or the HTTP Event Collector, you do not need to build a modular input and can skip the input options requirement.
- If you plan to create a modular input, have sample data and/or a test account for the system that the modular will contact. Know the input options that are required to access your data.
- Know which parts of the data model you want to map data to.

Building an add-on can include creating modular inputs to send data to Splunk, getting your data into Splunk and extracting the fields you need, mapping your data to the data model, and creating alert actions. Understanding these concepts will help you create your add-ons.

Understanding modular inputs

Modular inputs, which are typically written in Python or Java, allow you to programmatically create a way to get your data into Splunk when the native Splunk data collection inputs do not meet your needs. For example, use a modular input if you need to query a third-party API to get to your data or if you have a type of data that is not a native Splunk input (such as HEC, syslog, or file monitor).

The Splunk Add-on Builder can create simple modular inputs for you and generate the basic Python code. For more complex cases, you can write your own Python code for the data input and provide the input arguments. You can validate each data input in the Splunk Add-on Builder UI by running the input.

For more about modular inputs, see Modular inputs overview in the Splunk Enterprise Developing Views and Apps for Splunk Web manual.

Understanding field extractions

When Splunk Enterprise indexes data, it parses the data stream into a series of events. As part of this process, Splunk adds a number of fields to the event data. These fields include default fields that are added automatically and any custom fields that you specify. The Splunk Add-on Builder lets you add custom fields to your data and perform field mapping at index and/or search time.
For more about fields and field extraction, see:

- Create custom fields at index time in the Splunk Enterprise *Getting Data In* manual.
- Extract fields from files with structured data in the Splunk Enterprise *Getting Data In* manual

**Understanding the data model**

From version 2.2.0, Splunk Add-on Builder supports normalize your data from different sources or vendors to match to generic data models, including common information model (CIM).

- The CIM has 22 predefined data models. If you want your data to match these standards you can add data model mapping to your data, which is applied at search time. For more information about the CIM, see Overview of the Splunk Common Information Model in the *Common Information Model Add-on Manual*.  
- If you want to define your own data model, see create a new data model.

**Understanding alert actions**

Alerts monitor for and respond to specific events, in real time or on a schedule. When certain conditions are met, alerts are triggered and run alert actions. Splunk Enterprise includes a number of pre-configured alert actions, such as running a script, sending an email, or logging an event in response to an alert. Using Add-on Builder, you can create your own alert actions that are available in Splunk Enterprise to users of your add-on. Alert actions might define third-party integrations, add custom functionality, or include adaptive response actions for Splunk Enterprise Security.

For more about creating alert actions, see Custom alert actions overview in the *Developing Views and Apps for Splunk Web* manual.

For more about creating alert actions that are compatible with adaptive response for Enterprise Security, see:

- Use the common action model to build custom alert actions in the *Common Information Model Add-on Manual*.  
- Create an adaptive response action on the Splunk Developer Portal
Use the Add-on Builder

Use the Splunk Add-on Builder

The home page of the Splunk Add-on Builder displays projects for all of the add-ons that were created using the Splunk Add-on Builder along with apps and add-ons that are installed on your instance of Splunk Enterprise. An Add-on Builder project is an editable add-on along with its meta data. You can export projects for use on other instances of Add-on Builder.

Work with other apps and add-ons

Using Add-on Builder, you can work with other apps and add-ons that are installed on your instance of Splunk Enterprise, even if the app or add-on was not created using Add-on Builder.

- **Validate&Package**: Validate the app or add-on against best practices and other rules, and determine whether your app is ready for Splunk App Certification. You can also download a package file for the app or add-on.
- **Manage Source Type**: Create or import a source type and add sample data for the app or add-on.
- **Extract Fields**: Parse the data in your source types to create field extractions for the app or add-on.
- **Map to Data Model**: Map fields from the app or add-on to the data models.
- **Create Alert Actions**: Create and configure alert actions for the app or add-on.

If an existing app or add-on already has configuration files in the app's `/default` directory, you are prompted to allow Add-on Builder to move these configuration
files to the app's \texttt{/local} directory and merge them with any existing configuration files. Typically these configuration files include \texttt{props.conf}, \texttt{eventtypes.conf}, and \texttt{tags.conf}, which are used for data model mapping.

**To use Add-on Builder to work with other apps and add-ons**

1. On the Splunk Add-on Builder home page, click the \textbf{Other apps and add-ons} tab.
2. On the app or add-on, click the link for the activity you want: \texttt{Validate \\ Package}, \texttt{Manage Source Type}, \texttt{Manage Source Type}, \texttt{Extract Fields}, \texttt{Map to Data Model} or \texttt{Create Alert Actions}.

**End-to-end walkthroughs**

For end-to-end walkthroughs, see the \textbf{Splunk Developer Portal}.

**Import and export an add-on project**

Each add-on you create in Splunk Add-on Builder has a corresponding \textit{project} that contains the configuration settings for the add-on. The only way you can modify certain parts of your add-on that are built by add-on builder, such as data inputs or field extractions, is by having access to the add-on project.

If you want to modify your add-on on a different computer from the one you used for developing it, or if you want to share your add-on project, you can export the add-on project as a TGZ file. Then, the project file can be imported to a different instance of Splunk Enterprise running Add-on Builder.

There are two ways to import and export add-on project, Splunk Web or using migration tool.

- Using Splunk web is very straightforward and usually recommended, the exported file is gzipped.
- If you want to automate the process or for code repository (such as git) check-in, you can also use the migration tool provided by Splunk Add-on builder. The exported file is an extracted folder.

You can only import and export projects for add-ons that were created in Splunk Add-on Builder.
To import an add-on project using Splunk web

1. On the Splunk Add-on Builder home page on the Created with Add-on Builder tab, click Import Project.
2. Navigate to the add-on TGZ project file and click Open.

To export an add-on project using Splunk web

• On the Splunk Add-on Builder home page on the Created with Add-on Builder tab, click Export for the add-on.

To import or export project using migration tool

Run shell command migration tool project_migration_tool under
$SPLUNK_HOME$/etc/apps/splunk_app_addon-builder/bin/aob/aob_tools.

Arguments of the migration tool

-h, --help  Show the help of the migration tool and then exit
-l,--log_level  {DEBUG,INFO,WARNING,ERROR}  Log level of the migration tool. Default:INFO
-o, --output_dir  The output directory to export the project to.
-t, --ta_name  The project name of which you want to import/export.
-u, --user  Splunk user name. Default:admin
-p, --password  Splunk password. Default:changeme
-s, --splunk_web  Splunk platform URL. Default: http://localhost:8000
-a, --action {export,import}  The operation you want to do. Default:export
-i, --input_directory  The source directory of the project.
-v, --version  Show the version of the migration tool.

**Example:** Import the ServiceNow add-on on github to Splunk add-on builder under $YOUR_FOLDER/ta_servicenow.

Run the following command of the migration tool:

```
./project_migration_tool -a import -i $YOUR_FOLDER/ta_servicenow/TA-ServiceNow -s http://localhost:8000 -u admin -p changeme
```

**Example:** Export the ServiceNow add-on on your local Splunk instance to github.

1. Run the following command:

   ```
   ./project_migration_tool -a export -t TA-ServiceNow -o /tmp/ta_servicenow/ -s http://localhost:8000 -u admin -p changeme
   ```

2. Copy the exported content to git repo directory.

   ```
   cp -r /tmp/ta_servicenow/TA-ServiceNow/* $YOUR_FOLDER/ta_servicenow/TA-ServiceNow
   ```

3. Check in the changes.

**Create an add-on**

Click **Create an add-on** on the Add-on Builder home page, then specify the basic properties for your add-on. These settings also determine how your add-on appears in Splunk Web.
Set the following properties for your project:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add-on Name</td>
<td>The name of the add-on.</td>
</tr>
<tr>
<td>Author</td>
<td>The author name, which appears with the add-on in Splunk Web.</td>
</tr>
<tr>
<td>Add-on Folder Name</td>
<td>The folder name for the add-on under $SPLUNK_HOME/etc/apps/.</td>
</tr>
<tr>
<td></td>
<td>Add-on folder names have a prefix of &quot;TA-&quot;, which is a requirement for Splunk Enterprise Security add-ons. For add-ons created by Splunk, the folder name prefix is &quot;Splunk_TA_&quot;. To change the folder name, click Edit.</td>
</tr>
<tr>
<td>Version</td>
<td>The version number of the add-on, which appears with the add-on in Splunk Web.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of your add-on.</td>
</tr>
<tr>
<td>Visible</td>
<td>Indicates whether you want the add-on to be visible in the Splunk Web app list. By default, add-ons are not visible. However, if you create a setup page for your add-on, the Visible property will be enabled automatically.</td>
</tr>
<tr>
<td>Icon</td>
<td>Upload an image file to use as the icon for your add-on. The Add-on Builder creates a large version (72x72 pixels) and a small version (36x36 pixels) of the icon from this image.</td>
</tr>
<tr>
<td>Theme Color</td>
<td>The color to use for your add-on.</td>
</tr>
</tbody>
</table>
After you click **Create** to save your settings, the Splunk Add-on Builder displays a home page for your new add-on with the workflow for building it, including a summary of the objects you have configured.

**Add-on Builder workflow**

The following diagram shows the basic workflow in creating an add-on.
<table>
<thead>
<tr>
<th>Workflow action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a data input</td>
<td>Get data into your add-on by configuring data inputs.</td>
</tr>
<tr>
<td>Configure data collection</td>
<td>Configure a data input using a REST API call, using a shell command, or writing Python code.</td>
</tr>
<tr>
<td>Create a setup page</td>
<td>Create a setup page for your add-on when you need to prompt users for information to run your add-on, including authentication.</td>
</tr>
<tr>
<td>Manage source types</td>
<td>Add or import a source type and add sample data to your add-on so that you can add knowledge objects from it. You can import data from an existing source type or upload sample files.</td>
</tr>
<tr>
<td>Extract fields</td>
<td>Create field extractions from the data inputs you configured.</td>
</tr>
<tr>
<td>Map to data model</td>
<td>Map the fields from the field extractions to the data model you want to use.</td>
</tr>
<tr>
<td>Create alert actions</td>
<td>Create custom alert actions, including running an adaptive response action for Splunk Enterprise Security.</td>
</tr>
<tr>
<td>Validate the add-on</td>
<td>Validate the add-on for best practices, and test the modular/scripted inputs, field extractions, and data model mappings. The Splunk Add-on Builder shows you any errors or warnings, along with recommendations about how to address them.</td>
</tr>
<tr>
<td>Package the add-on</td>
<td>Package the add-on when you are ready to install your add-on and use it. The result is a compressed file that contains a directory with the necessary configuration files.</td>
</tr>
</tbody>
</table>

### Configure data collection using a REST API call

In **Configure Data Collection**, configure the data inputs and setup page for your add-on. This page displays a list of data inputs that you have configured for your add-on. Data inputs determine how you will get data into your add-on:

- Use a REST API call, described in this topic
- Run a shell command script
- Create a modular input from your own Python code
Once you have configured data inputs for your add-on, you can manage them from the Configure Data Collection page in Add-on Builder. Data inputs you configure for your add-on also appear on the Inputs page of the add-on.

Data inputs must be enabled so that Splunk Enterprise can start indexing the data from them. You'll need indexed data if you want to perform field extractions and data model mapping. Data inputs you configure are enabled by default.

Users of your add-on should create data inputs from your configuration within the add-on. Do not direct users to use the Settings > Data inputs page in Splunk Enterprise. When users open your add-on, the Inputs page provides the option to create a new input.

**Add a data input using a REST API**

Build the data collection for your add-on to gather data from a REST API. A REST data input uses JSON as a data type, and supports basic authentication and API-based authentication. For advanced data collection, create a modular input by writing your own Python code.

1. On your add-on homepage, click the Configure Data Collection icon.
2. On the Configure Data Collection page, click New Input.
   The Create Data Input wizard starts.
3. On the Choose Input Method page, click Modular input using a REST API.
4. On the Data Input Properties tab:
   - Enter a name for the source type to create for this data input.
   - Enter an input display name, internal name, and description.
   - Specify a collection interval indicating how often to retrieve data using this input, in seconds.
5. Optionally, click the **Data Input Parameters** tab to define any parameters that are required for users to configure your data input. For each parameter, drag a field from the Component Library to the center panel, then specify its properties in the Property Editor.

6. Optionally, click the **Add-on Setup Parameters** tab to define any parameters that are required for users to configure your add-on.
   - Select predefined options to prompt for account, proxy, or logging settings.
   - Add custom inputs by dragging fields from the Component Library to the center panel panel. Select an input to specify its properties in the Property Editor.

   For more about setup pages, see Create a setup page.

8. Click **Next**.

9. On the **Data Input Definition** tab under **REST settings**:
   - Specify a **REST URL** for a REST endpoint.
   - Select the **REST method** (GET or POST).
   - If you are using the GET method, specify the names and values for the **REST URL parameters**. You can add parameters individually to the REST header or body, or add them directly to the URL. For details, see Pass values from data input parameters and Pass values from setup parameters.
   - Optionally, specify the names and values for **REST request headers**.
   - Optionally, if you are using the POST method, specify the names and values for the **REST request body**.
   - If you added a **Global Account** field to your data input parameters, select **Enable basic authentication** if you want to include the username and password values in the HTTP header using HTTP Basic Authentication.

10. On the **Data Input Definition** tab under **Data input parameters**, enter a
sample value for each data input parameter for testing purposes. To use a value from a data input parameter in the REST URL, use the format `${internal_name}`, using the internal name defined for the parameter in the previous step. For details, see Pass values from data input parameters and Pass values from setup parameters.

11. On the **Data Input Definition** tab under **Event extraction settings**, optionally specify how to break the JSON payload into individual events. Use this setting when the results from the API return a JSON array of events, and you would like to break them into individual events before indexing them. Specify the JSON path to that array within the payload to use for event breaking. When the JSON path matches the array in the payload, the extracted events are highlighted in the test output. For more, see JSON path formats.

12. On the **Data Input Definition** tab under **Checkpoint settings**, if you want to set a checkpoint select **Enable checkpointing**. A checkpoint allows you to read and index only the data that is new since the last time data was received. For more about checkpoints, including a sample configuration, see Use checkpoints for incremental data collection.

Specify the checkpoint settings:

♣ For **Checkpoint parameter name**, enter the REST parameter name to use as a checkpoint. Make sure you use the `${param_name}` format to replace the checkpoint value in the REST URL definition.
♣ For **Checkpoint field path**, enter the JSON path to the field in the response payload to use as the checkpoint. For more about specifying the checkpoint field path, see JSON path formats.
♣ For **Checkpoint initial value**, enter the initial value to use for the checkpoint. Users can modify this value when creating a data input.
♣ If the timestamp format of the request is different from the response, specify values for **Response timestamp format** and **Request timestamp format**. If both timestamps use the same
format, leave these fields empty. For more, see Normalize timestamp formats.

13. If you created a setup page, click the Add-on Setup Parameters tab to enter sample values for setup parameters for testing purposes.
14. Click the Data Input Definition tab, then click Test to run the REST command and test the data input configuration. The Output section indicates whether the REST command succeeded (sample data is displayed) or failed (the error message is displayed).

15. Optionally, if you want to collect data using SSL certificate, use send_http_request function. See Python helper functions for more details.
16. Click Save to save the data input, then click Finish.

The Add-on Builder displays a summary of the new data input, along with suggested next steps.

**Add authentication to a data input**

When you need to use authentication for a REST input, add a Global Setup input field to your data input parameters form:
The **Global account settings** option is automatically selected on your setup page.

When users first open your add-on, they set up their accounts on the add-on’s **Configuration** page:

Then, users configure new inputs on the add-on’s **Inputs** page, which displays the data input parameters you configured:

---

**Use checkpoints for incremental data collection**

When creating your REST API data input, you can use checkpoints to mark where data has already been read and indexed, indicating where to resume reading data the next time it is collected, thereby avoiding collecting the same data again. Checkpoints are useful for REST APIs that return a number of events per call. The checkpoint field is typically a timestamp, but you can specify any
field. You do not need to use checkpoints with a REST API that returns a point-in-time value, such as metrics.

When using checkpoints, the REST API should do the following:

- Allow you to sort fields by the one you want to use for checkpoints. Otherwise, that field can't be used as a marker that indicates where new data begins.
- Be able to query data by the checkpoint field, starting with the initial checkpoint value.
- Allow timestamps in increments at least as small as the collection interval. For example, if your collection interval is 30 seconds, a timestamp in days would not be a useful checkpoint.

For an end-to-end walkthrough, see Walkthrough: Configure a data input with checkpoints on the Splunk Developer Portal.

**Pass values from data input parameters**

You can pass parameter values to a variable in a REST URL, a REST header, a REST body, and in a shell command. Following the Mako syntax, use the format \${varname} for the variable. Then, define a modular input variable for varname.

For example, let's say you want to use a URL such as http://localhost:5111/api/task/<taskID>, where <taskID> is a variable value.

When you define the REST API input, on the Inputs & Parameters step, click the Data Input Parameters tab and add an input variable for Task ID with the internal name "taskid":

33
Next, on the **Define & Test** step when you specify the REST URL, use "http://localhost:5111/api/${taskid}". To test it, use an initial value, such as "t123". The resulting REST URL is "http://localhost:5111/api/task/t123".

When users open your add-on, they can create instances of this data input on the **Inputs** page. The form displays the **Task ID** field as one of the data input parameters you configured:

**Pass values from setup parameters**

When referring to setup parameters, you must include a namespace string before the parameter name to get the value from the parameter. The following shows an example, where `param_name` is the parameter name:

```plaintext
${__settings__.additional_parameters.param_name}
```

For example, let's say you have a text input on your setup page that prompts the user for an API token, with the internal name "api_token". To pass the value of the API token to the REST call, use the following format:

```plaintext
api-key=${__settings__.additional_parameters.api_token}
```

We recommend you do not set account, proxy, or logging fields directly. You can access the global account values as follows:
JSON path formats

Use the following guidelines to form JSON paths when creating REST inputs.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>The root element to query.</td>
</tr>
<tr>
<td>*</td>
<td>Wildcard. Available anywhere a name or number is required.</td>
</tr>
<tr>
<td>..</td>
<td>Deep scan. Available anywhere a name is required.</td>
</tr>
<tr>
<td>.&lt;name&gt;</td>
<td>Dot-notated child.</td>
</tr>
<tr>
<td>['&lt;name&gt;']</td>
<td>Bracket-notated child.</td>
</tr>
<tr>
<td>[&lt;number&gt;]</td>
<td>Array index or indexes.</td>
</tr>
</tbody>
</table>

Example

Here is an example JSON response:

```json
{
    "count": 251,
    "results": [
        {
            "isocode": "MZ",
            "country": "Mozambique"
        },
        {
            "isocode": "PG",
            "country": "Papua New Guinea"
        }
    ]
}
```

The following table shows different ways of accessing data in the JSON response.

<table>
<thead>
<tr>
<th>JSON path</th>
<th>Returned events</th>
</tr>
</thead>
<tbody>
<tr>
<td>$.count</td>
<td>251</td>
</tr>
<tr>
<td>$.results[*] or $.results</td>
<td>{&quot;isocode&quot;: &quot;MZ&quot;, &quot;country&quot;: &quot;Mozambique&quot;}</td>
</tr>
<tr>
<td>JSON Path</td>
<td>Value</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>$.results[0]</td>
<td>{&quot;isocode&quot;: &quot;PG&quot;,&quot;country&quot;: &quot;Papua New Guinea&quot;}</td>
</tr>
<tr>
<td>$.results[-1]</td>
<td>{&quot;isocode&quot;: &quot;PG&quot;,&quot;country&quot;: &quot;Papua New Guinea&quot;}</td>
</tr>
<tr>
<td>$.results[0].country</td>
<td>Mozambique</td>
</tr>
<tr>
<td>$['results'][0]['isocode']</td>
<td>MZ</td>
</tr>
<tr>
<td>$.results[0]</td>
<td>MZ Mozambique</td>
</tr>
<tr>
<td>$..isocode</td>
<td>MZ PG</td>
</tr>
</tbody>
</table>

When the JSON path matches a field in the payload, the extracted value is highlighted in the test output.

### Normalize timestamp formats

When you set a checkpoint, you only need to specify timestamp formats when the format of the request timestamp is different from the response timestamp. If these formats are different, use the following format for timestamps:

```
%Y-%m-%dT%H:%M:%S.%f+08:00
```

Additional notes:

- "0" left padding is added to months, days, hours, minutes, and seconds. To remove left padding, use "-". For example, use `%m`.  
- "0" right padding is added to milliseconds.  
- Add-on Builder supports the time format as described in the Python function `time_str2str` for transforming a time string to a different format.  
- The `%Nf` format is supported for `to_format` only.
Due to a limitation in Python 2.x, the date-time formatting codes \%z for "time zone name" and \%Z for "time zone offset from GMT" are not supported.

**End-to-end walkthroughs**

For end-to-end walkthroughs, see the **Splunk Developer Portal**.

**Learn more**

For more information, see the following Splunk Enterprise documentation:

- How timestamp assignment works in the *Getting Data In* manual
- Source type naming conventions in the *Splunk Add-ons* manual
- Modular inputs overview in the *Developing Views and Apps for Splunk Web* manual
- How to create modular inputs using Splunk SDK for Python on the Splunk Developer Portal
- Create a setup page for a Splunk app on the Splunk Developer Portal

**Configure data collection using a shell command**

In **Configure Data Collection**, configure the data inputs and setup page for your add-on. This page displays a list of data inputs that you have configured for your add-on. Data inputs determine how you will get data into your add-on:

- Use a REST API call
- Run a shell command script, described in this topic
- Create a modular input from your own Python code
Once you have configured data inputs for your add-on, you can manage them from the Configure Data Collection page in Add-on Builder. Data inputs you configure for your add-on also appear on the Inputs page of the add-on.

- Data inputs must be enabled so that Splunk Enterprise can start indexing the data from them. You’ll need indexed data if you want to perform field extractions and data model mappings. Data inputs you configure are enabled by default.

Users of your add-on should create data inputs from your configuration within the add-on. Do not direct users to use the Settings > Data inputs page in Splunk Enterprise. When users open your add-on, the Inputs page provides the option to create a new input.

**Add a data input using shell commands**

Get data into your add-on by using a script of shell commands.

1. On your add-on homepage, click the Configure Data Collection icon.
2. On the Configure Data Collection page, click Add Data.
   The Create Data Input wizard starts.
3. On the Choose Input Method page, select Modular input using shell commands.
4. On the Data Input Properties tab:
   - Enter a name for the source type to create for this data input.
   - Enter an input display name, internal name, and description.
   - Specify a collection interval indicating how often to retrieve data using this input, in seconds.
5. Optionally, click the Data Input Parameters tab to define any parameters that are required for users to configure your data input. For each parameter, drag a field from the Component Library to the center panel, then specify its properties in the Property Editor.
6. Optionally, click the Add-on Setup Parameters tab to define any parameters that are required for users to configure your add-on.
   - Select predefined options to prompt for account, proxy, or logging settings.
   - Add custom inputs by dragging fields from the Component Library to the center panel panel. Select an input to specify its properties in the Property Editor.

   For more about setup pages, see Create a setup page.

7. Click Next.
8. On the **Data Input Definition** tab:
   - Under **Command settings**, enter the shell commands.
   - Under **Data input parameters**, enter a sample value for each data input parameter for testing purposes.
   To use a value from a data input parameter in the REST URL, use the format `${internal_name}`, using the internal name defined for the parameter in the previous step. For more about passing parameter values to a REST call, see Pass values from data input parameters and Pass values from setup parameters.

9. If you created a setup page, click the **Add-on Setup Parameters** tab to enter sample values for setup parameters for testing purposes.

10. Click the **Data Input Definition** tab, then click **Test** to run the shell commands and test the data input configuration.
    The **Output** section indicates whether the shell commands succeeded (sample data is displayed) or failed (the error message is displayed).

11. Click **Save** to save the data input, then click **Finish**.

The Add-on Builder displays a summary of the new data input, along with suggested next steps.

**Pass values from data input parameters**

You can pass parameter values to a variable in a REST URL, a REST header, a REST body, and in a shell command. Following the Mako syntax, use the format `${varname}` for the variable. Then, define a modular input variable for varname.

For example, let's say you want to use a URL such as http://localhost:5111/api/task/<taskID>, where `<taskID>` is a variable value.

When you define the REST API input, on the **Inputs & Parameters** step, click the **Data Input Parameters** tab and add an input variable for Task ID with the internal name "taskid":
Next, on the **Define & Test** step when you specify the REST URL, use "http://localhost:5111/api/${taskid}". To test it, use an initial value, such as "t123". The resulting REST URL is "http://localhost:5111/api/task/t123".

When users open your add-on, they can create instances of this data input on the **Inputs** page. The form displays the **Task ID** field as one of the data input parameters you configured:

**Pass values from setup parameters**

When referring to setup parameters, you must include a namespace string before the parameter name to get the value from the parameter. The following shows an example, where `param_name` is the parameter name:
For example, let’s say you have a text input on your setup page that prompts the user for an API token, with the internal name "api_token". To pass the value of the API token to the REST call, use the following format:

api-key=${__settings__.additional_parameters.api_token}

We recommend you do not set account, proxy, or logging fields directly. You can access the global account values as follows:

${global_account.username}
{{global_account.username}}

${global_account.password}
{{global_account.password}}

Learn more

For more information, see the following Splunk Enterprise documentation:

- Source type naming conventions in the Splunk Add-ons manual.
- Modular inputs overview in the Developing Views and Apps for Splunk Web manual
- How to create modular inputs using Splunk SDK for Python on the Splunk Developer Portal
- Create a setup page for a Splunk app on the Splunk Developer Portal

Configure data collection using your Python code

In Configure Data Collection, configure the data inputs and setup page for your add-on. This page displays a list of data inputs that you have configured for your add-on. Data inputs determine how you will get data into your add-on:

- Use a REST API call
- Run a shell command script
- Create a modular input from your own Python code, described in this topic
Once you have configured data inputs for your add-on, you can manage them from the **Configure Data Collection** page in Add-on Builder. Data inputs you configure for your add-on also appear on the **Inputs** page of the add-on.

- Data inputs must be enabled so that Splunk Enterprise can start indexing the data from them. You’ll need indexed data if you want to perform field extractions and CIM mapping. Data inputs you configure are enabled by default.

Users of your add-on should create data inputs from your configuration within the add-on. Do not direct users to use the **Settings > Data inputs** page in Splunk Enterprise. When users open your add-on, the **Inputs** page provides the option to create a new input.

**Add a data input by writing your own code**

Use this option to implement your own data collection mechanism using Python code.

1. On your add-on homepage, click the **Configure Data Collection** icon.
2. On the **Configure Data Collection** page, click **Add Data**. The **Create Data Input** wizard starts.
3. On the **Choose Input Method** page, select **Modular input using your own Python code**.
4. On the **Data Input Properties** tab:
   - Enter a name for the source type to create for this data input.
   - Enter an input display name, internal name, and description.
   - Specify a collection interval indicating how often to retrieve data using this input, in seconds.
     - If this is a single-instance input, do not set a collection interval.
5. Click the **Data Input Parameters** tab to define any parameters that are required for users to configure your data input. At least one parameter is
required.
For each parameter, drag a field from the Component Library to the center panel, then specify its properties in the Property Editor.
6. Optionally, click the Add-on Setup Parameters tab to define any parameters that are required for users to configure your add-on.
   • Select predefined options to prompt for account, proxy, or logging settings.
   • Add custom inputs by dragging fields from the Component Library to the center panel panel. Select an input to specify its properties in the Property Editor.

   For more about setup pages, see Create a setup page.

7. Click Next.
8. On the Data Input Definition tab in the Code Editor panel, edit the Python code to create your modular input.
   • Add code under the validate_input function to provide validation logic.
   • Add code under the collect_events function to add data collection logic.
   • Add code under the send_http_request function to set SSL certification validation.
   • Use the Python helper functions, which are dynamically generated as commented code, as a guide to working with the input parameters you defined in the previous step.
     The autogenerated code contains a working example that generates a random number.
   • Under Data input parameters, enter sample values for testing the data input.
9. If you created a setup page, click the **Add-on Setup Parameters** tab to enter sample values for setup parameters for testing purposes.

10. Click the **Data Input Definition** tab, then click **Test** to run your code and test the data input configuration.

   The **Output** section indicates whether your code succeeded (sample data is displayed) or failed (the error message is displayed).

11. Click **Save** to save the data input, then click **Finish**.

The Add-on Builder displays a summary of the new data input, along with suggested next steps.

**Add authentication to a data input**

When you need to use authentication for a REST input, add a **Global Setup** input field to your data input parameters form:

```python
import json

# To access a username and password in your code, use the following format:

global_account = helper.get_arg('global_account')
username = global_account['username']
password= global_account['password']
```

The **Global account settings** option is automatically selected on your setup page.

To access a username and password in your code, use the following format:

```python
import json

global_account = helper.get_arg('global_account')
username = global_account['username']
password= global_account['password']
```

When users first open your add-on, they set up their accounts on the add-on’s **Configuration** page:
Then, users configure new inputs on the add-on's Inputs page, which displays the data input parameters you configured:

**Pass values from data input parameters**

Add-on Builder generates the Python helper functions for you to access the values from data input parameters as follows, where "server" is the global parameter name, and "local" is the modular input parameter name:

```python
server = helper.get_global_setting("server")
local = helper.get_param("local")
```

For more, see Python helper functions.

**Single-instance and multi-instance modes**

Multi-instance mode runs one instance of a data input per input stanza, resulting in multiple instances (one for each input stanza). This default behavior is useful in multi-thread environments or in situations that require different security contexts or access to different databases.

In a single-thread environment you might want to run an input in single-instance mode. For example, in a WMI environment you would run a single instance of an input to re-use connections.

Multi-instance mode is enabled by default. To enable single-instance mode, do not set a collection interval for the data input, and then uncomment the following
def use_single_instance_mode():
    return True

Learn more

For more information, see the following Splunk Enterprise documentation:

- Source type naming conventions in the *Splunk Add-ons* manual.
- Modular inputs overview in the *Developing Views and Apps for Splunk Web* manual
- How to create modular inputs using Splunk SDK for Python on the Splunk Developer Portal
- Create a setup page for a Splunk app on the Splunk Developer Portal

Create a setup page

If your add-on requires setup parameters, you can create a setup page that is displayed to users when they run your add-on. Use a setup page for your add-on when:

- You need to authenticate users with a username and password, for example to authenticate a third-party API for a data input.
- You need the user to provide proxy information, for example when you need to use a proxy to make outbound API calls from the add-on.

By including a setup page in your add-on, the user provides the information to run your add-on without having to edit a configuration file directly.

An add-on is automatically made visible when it has a setup page.

Setup page workflow

Creating a setup page is part of the workflow for creating a data input and creating an alert action.

Create the setup page

The workflow wizards include an Add-on Setup Parameters tab where you
create or edit the setup page for the current add-on. Select the predefined setup options you want to use. You can also define custom text, password, and checkbox inputs.

Setup parameters can be used in any data input or alert action within the add-on.

Use the setup parameters below when creating a setup page.

<table>
<thead>
<tr>
<th>Setup parameter</th>
<th>Usage</th>
</tr>
</thead>
</table>
| Global account settings | Adds a predefined **Account** tab to the add-on’s **Configuration** page that allows the user to configure multiple accounts, such as one account for production and one for development.  
This **Global account setting** option is selected automatically when you add a **Global Account** field as a data input parameter. |
| Proxy settings        | Adds a predefined **Proxy** tab to the add-on’s **Configuration** page that allows the user to specify a proxy host, port, username, password, and type. |
| Logging settings      | Adds a predefined **Logging** tab to the add-on’s **Configuration** page that lets the user select the logging level to use for the add-on. |
| Text field            | Drag and drop different types of inputs from the Component Library to the setup form. These options will be displayed on a separate **Add-on Settings** tab on the add-on’s **Configuration** page. |
| Password field        | Password fields are automatically obfuscated using the storage password endpoint. |
| Checkbox field        |                                                                    |

**Test the setup page**
The last step of the workflow wizards allow you to test the setup page by entering sample values.

**View the production version of the setup page**

To access the production version of the add-on’s setup page while developing your add-on, open the app in Splunk Web and click **Configuration**.

For detailed procedures, see:

- Configure data collection using a REST API call
- Configure data collection using your Python code
- Create an alert action

**Manage source types**

After you’ve added data inputs in the **Configure Data Collection** section, the modular inputs you created might not collect data quickly enough. To manage source types is useful when:

- Your add-on relies on native core data inputs for data collection (for example, syslog files or the HTTP Event Collector) and you use the Add-on Builder for creating knowledge objects and performing data model mapping.
- You have configured a data collection and you want to upload more sample data to create knowledge objects, such as field extractions, and to perform data model mapping.
• You want to edit the existing source type, configure the timestamp or event line breaking.

You can add new source types in two ways:

• Create a new source type and also upload sample data from one or more files for this source type.
• Import an existing source type from the Splunk platform.

Sample data counts against your license.

**Add new source type**

__To create a source type and add sample data to it__

1. On your add-on homepage, click **Manage Source Types** on the Add-on Builder navigation bar.
2. On the **Manage Source Types** page, click **Add** and then **New Source Type**.
3. Enter a unique source type name.
4. Click **Upload Data**, navigate to and select the sample data file, then click **Open**.

   The preview displays the first 1000 events from the first 2MB of data.

5. Adjust indexing settings as needed:
   ♦ Expand the **Event Breaks** section and select an option that indicates how events for the data in this source type should be separated:
     ◊ **Auto**: Events are auto-detected based on their timestamp location.
     ◊ **Every Line**: Every line is one event.
     ◊ **Regex**: Use a regular expression to define a pattern to split events.
   ♦ Expand the **Timestamp** section and select an option that indicates how to generate timestamps for the data.
Expand the Advanced section to specify additional index-time parameters for parsing data.

6. Click Save.

Sample events are stored in a dedicated "add_on_builder_index" index.

To import an existing source type

1. On your add-on homepage, click Manage Source Types on the Add-on Builder navigation bar.
2. On the Manage Source Types page, click Add and then Import From Splunk.
3. Select a source type from the drop-down list.
4. (Optional). Click Upload Data, navigate to and select the sample data file, then click Open.

The preview displays the first 1000 events from the first 2MB of data.

5. Adjust indexing settings as needed:
   ♦ Expand the Event Breaks section and select an option that indicates how events for the data in this source type should be separated:
     ◊ Auto: Events are auto-detected based on their timestamp location.
     ◊ Every Line: Every line is one event.
     ◊ Regex: Use a regular expression to define a pattern to split events.
   ♦ Expand the Timestamp section and select an option that indicates how to generate timestamps for the data.
   ♦ Expand the Advanced section to specify additional index-time parameters for parsing data.

6. Click Save.

Edit existing source type of this add-on

To edit the existing source type and add sample data to it

1. On your add-on homepage, click Manage Source Type on the Add-on Builder navigation bar.
2. Click Edit on the source type you want to edit.
3. (Optional). Click Upload Data, navigate to and select the sample data file, then click Open.
The preview displays the first 1000 events from the first 2MB of data.

4. Adjust indexing settings as needed:
   - Expand the Event Breaks section and select an option that indicates how events for the data in this source type should be separated:
     - **Auto**: Events are auto-detected based on their timestamp location.
     - **Every Line**: Every line is one event.
     - **Regex**: Use a regular expression to define a pattern to split events.
   - Expand the Timestamp section and select an option that indicates how to generate timestamps for the data.
   - Expand the Advanced section to specify additional index-time parameters for parsing data.

**Learn more**

For more information, see the following Splunk Enterprise documentation:

- For source types, see Configure source types in the *Getting Data In* manual
- For event breaks, see Configure event line breaking in the *Getting Data In* manual
- For advanced settings, see props.conf in the *Admin Manual*

**Extract fields**

In Extract Fields, parse the data in your source types to create field extractions. The source types you created in the Configure Data Collection section or imported from splunk using the Manage source type appear in source type list.

Splunk Add-on Builder provides you three ways to build the field extractions.

- Assisted extraction. Splunk Add-on Builder will detect the format of the data and provides you the recommended regex to parse your data.
- Manual extraction. Configure the field extraction manually in Splunk platform.
- Manual transformation. Configure the field transformation manually in Splunk platform.
If you used assisted extraction before and want to switch to manual extraction or transformation, you need to delete the assisted extraction first or otherwise the manual mode will be disabled.

Assisted extraction

If you choose Assisted extraction, Add-on builder will try to parse your data using the suggested format, the format includes:

- **Unstructured Data.** Typically used for log files.
- **Table.** Data in tabular formats, such as comma-separated values (CSV) and tab-separated values (TSV).
- **Key Value.** Data that contains key-value pairs.
- **JSON.** Data in the JavaScript Object Notation (JSON) format.
- **XML.** Data in the Extensible Markup Language (XML) format.

To parse the data for a sourcetype and extract fields using assisted extractions

1. On your add-on homepage, click Extract Fields on the Add-on Builder navigation bar.
2. On the Extract Fields page, select a source type to parse by clicking Assisted Extraction.
3. From Choose Data Format, select the format of the data.

   A format type is automatically selected if it has been detected. You can change it as needed. If you aren't sure and a format type has not been automatically selected, try "Unstructured Data".

4. Click Submit.

   After parsing the data, the results are displayed on a summary page. Depending on the data format, you can adjust the way fields are extracted.
5. If you are satisfied with the results, click **Save**.

6. If you want to try parsing again using a different format, click **Cancel** to return to the previous page.

Once data for a source type has been parsed, the source type is added to the table on the **Extract Fields** page.

**Unstructured Data**

The Add-on Builder's Field Extractor displays a selection of events in groups, along with the fields that were extracted.

Here are some of the actions you can perform:

- Select one or more groups that best represent the data.
- If you are familiar with creating regular expressions, you can display the regular expression that the Field Extractor used, and modify it to improve the field extraction.
- Click on individual field names to include or exclude the field for extraction.
- Click the Edit icon next to a field name to edit its name.
- Click the Trash icon next to a field name to remove its capture group from the regular expression.

**Table**

The Table format is used with tabular data.
Here are some of the actions you can perform:

- Change how data is parsed by selecting the delimiter character that is used to separate fields. To specify a different character, click Other and enter the character.
- Change the field names, but only after you have selected the correct delimiter. Each time you change delimiters, the number of columns could change and you might lose any changes to field names.

**Key Value**

The Key Value format is used with data containing key-value pairs.

Here are some of the actions you can perform:
• Change how data is parsed. For **Extraction Methods**, select:

  ◊ **Auto** to let the Add-on Builder parse data automatically.
  ◊ **Delimiters** to use delimiters.
  ◊ **Regex** to use regular expressions.

• For **Delimiters**, select the delimiters for the key-value pairs:

  ◊ Specify the pair delimiter character, which is used to separate key-value pairs.
    Using the example `key_a=value_a, key_b=value_b`, the correct character is a comma.
  ◊ Specify the key-value delimiter character, which is used to separate keys and values.
    Using the example `key_a=value_a, key_b=value_b`, the correct character is an equals sign.

• For **Regex**: select the regular expression to use, or create your own.

**JSON**

The JSON format is used with JSON data. There are no additional parsing options.

**XML**

The XML format is used with XML data. There are no additional parsing options.
Manual extraction

If you choose Manual extraction, Add-on builder will direct you to the Field extractions page of Splunk platform.

1. On your add-on homepage, click Extract Fields on the Add-on Builder navigation bar.
2. On the Extract Fields page, select a source type to parse by clicking Manual Extraction.
3. Add-on builder will direct you to the field extractions page of Splunk platform. See more on Use the Field extractions page in the Splunk Enterprise Knowledge Manager Manual.

Manual transformation

If you choose Manual transformation, Add-on builder will direct you to the Field transformation page of Splunk platform.

1. On your add-on homepage, click Extract Fields on the Add-on Builder navigation bar.
2. On the Extract Fields page, select a source type to parse by clicking Manual transformation.
3. Add-on builder will direct you to the field transformation page of Splunk platform. See more on Use the Field transformation page in the Splunk Enterprise Knowledge Manager Manual.
Troubleshooting

What if I need to upload different sample data?

If you decide that you need to upload a different sample data file for a source type, for example you want to clean the data first, go to Manage source types, delete the sample data, then upload additional data files.

A regular expression had too many capture groups, what do I do?

This error is displayed after attempting to parse a file, and the regular expression created by the Field Extractor contains more than 100 capture groups (fields).

This error might indicate a problem with the Event Break setting for the source type:

1. Go to Manage source types.
2. Edit the source type and select a different option for Event Break.
3. Upload the sample events again. Because the Event Break option is applied when indexing the data, changing this value does not affect events that have already been indexed.
4. Parse the data again.

The sample data might contain an event that is too long:

1. Edit the sample data file by splitting the long lines to clean up the data.
2. Go back to Manage source types.
3. Upload the sample events again.
4. Parse the data again.

Why are the field names not detected in my tabular data?

The Add-on Builder uses the first 1000 events for field extraction. If your data contains more than 1000 events, the parser cannot automatically detect the field names.

The parser assumes that all entries except the table header contain a timestamp. If entries in your tabular data do not contain a timestamp, the parser will not correctly detect which entry is the table header.
Learn more

For more information, see the following Splunk Enterprise documentation:

- About fields in the *Knowledge Manager Manual*
- Build field extractions with the field extractor in the *Knowledge Manager Manual*
- Field Extractor: Select Fields step in the *Knowledge Manager Manual*

Map to data model

Version 2.2.0 and later of the Splunk Add-on Builder enables you to map the fields from your data events to the fields in any data model, including CIM data models.

- If you want to map your data to a CIM data model, the Splunk Common Information Model add-on is required to use this feature. Download the Splunk Common Information Model add-on from Splunkbase and see Install the Splunk Common Information Model Add-on for details on how to install this add-on.
- If you want to map to your own data model, the model needs to support the standard defined under the Create a data model section.

Before you apply the data model mapping to your add-on, you must configure one or more source types for your add-on by creating a data input, by adding data from a sample file, or by adding indexed data from Splunk.

Configure the following,

In Map to data model, map the fields from your data to the fields in one of the predefined data models to normalize data at search time.

1. On your add-on homepage, click **Map to data model** on the Add-on Builder navigation bar.
2. On the **Data Model Mapping** page, click **New Data Model Mapping**.
3. On the **Data Model Mapping >> Define Event Type** page, define an event type to generate events from which to extract fields:
   - Enter a name for the event type.
   - Select a source type from which to generate events.
   - Enter a search to select events. By default, the search selects all events for the source type you selected, but you can apply
additional search criteria as needed.
♦ Click Save.

5. On the Data Model Mapping >> Select Data Models page, select the data model to use for mapping:
   ♦ From the center panel, select one or more data models to use. Then you can also select individual datasets within a data model. Fields from your event type are displayed for reference, and fields from the selected data models are also displayed.
   ♦ When you have finished selecting data models, click Done.
6. On the Data Model Mapping >> Data Model Mapping Details page, click New Knowledge Object and select the type of mapping to create:
   ♦ Select FIELDALIAS to map a field from the data model to a field from your event type.
   ♦ Select EVAL to map a field from the data model to an expression based on a field from your event type.
7. Define a field alias or expression in the new row that was added to the Data Model Mapping List:
   ♦ If you are defining a field alias, click one field name from the Data Model Fields list and one from the Event Type Fields list, and then click OK at the end of the new row in the Data Model Mapping List.
   ♦ If you are defining an expression, click one field name from the Data Model Fields list and one or more fields from the Event Type Fields list. Edit the expression in the Event Type Field or Expression column, then click OK at the end of the new row in the
Data Model Mapping List.
8. Repeat steps 6-7 as needed.
9. Click **Done** when you have finished data model mapping.

The Data Model Mapping page displays an entry for the mapping you just completed.

**Learn more**

For more information, see the following Splunk Enterprise documentation:

- About event types in the *Knowledge Manager Manual*
- About tags and aliases in the *Knowledge Manager Manual*
- `eval` in the *Search Reference* manual.
- Use the CIM to normalize data at search time in the *Common Information Model Add-on Manual*

**Create alert actions**

In *Create Alert Actions*, create and configure alert actions, which appear in Splunk Enterprise for users of your add-on. Alert actions might define third-party integrations, add custom functionality, or support adaptive response actions for Splunk Enterprise Security.

Once you have added alert actions to your add-on, you can manage them from the Alert Action page under *Create Alert Actions*. Your alert actions also appear on the *Settings > Alert Actions* page in Splunk Web.

**Create an alert action**

1. On your add-on homepage, click the **Create Alert Actions** icon.
2. On the *Alert Actions* page, click **New Alert Action**.
   The **Create Alert Action** wizard starts.
3. On the *Alert Action Properties* page, enter the properties for this alert action:
   - Enter a name, label, and description for this alert action. The label is the friendly name that appears in Splunk Web.
   - Optionally, click **Upload my icon** if you want to add an icon, such as a logo, to this alert action.
If you want to create an adaptive response action, see Create an adaptive response action for Enterprise Security, below.

4. Click Next.
5. On the Alert Action Inputs tab, specify one or more input fields for this alert action.

For each input, drag a field from the Component Library to the center panel, then specify its properties in the Property Editor.
If certain permission is required to trigger the alerts, you can drag Text field and Password field on the Component Library, specify the properties and then add the account in the Add-on Setup Parameters page as the example below.

6. Optionally, click the Add-on Setup Parameters tab to define any parameters that are required for users to configure your add-on.
   ♦ Select predefined options to prompt for account, proxy, or logging settings.
   ♦ Add custom inputs by dragging fields from the Component Library to the center panel panel. Select an input to specify its properties in the Property Editor.

For more about setup pages, see Create a setup page.

7. Click Next.
8. On the Alert Action Parameters tab, in the Code Editor panel edit the Python code to create your alert action.
   ♦ Find the comment # TODO: Implement your alert action logic here, which indicates where to add your code.
   ♦ Use the Python helper functions, which are dynamically generated as commented code, as a guide to working with the input parameters you defined in the previous step.
If you want to collect data using SSL certificate, use `send_http_request` function as follows.

```python
import solnlib.splunkenv

cert_file_path =
    solnlib.splunkenv.make_splunkhome_path(['<TA_folder>', 'default', 'example.pem'])
response =
    helper.send_http_request('https://www.example.com/api',
        'GET', verify=cert_file_path)
```

- `helper.settings` is a dict that contains information including SPL, Splunk environment configurations and alert configurations. If you want to use the information in `helper.settings`, use the following sample code as an example.

  ```python
  search_name = helper.settings.get('search_name')
  sid = helper.settings.get('sid')
  ```

9. On the **Alert Action Parameters** tab, enter sample values for testing this alert action.
10. If you created a setup page, on the **Add-on Setup Parameters** tab enter sample values to test the setup page.
11. Click **Test** to test your code and the alert action configuration. The **Output** section indicates whether the command succeeded or failed.
12. Click **Save** to save this alert action, then click **Finish**.

### Create an adaptive response action for Enterprise Security

An adaptive response action is a type of alert action that is specifically created for Splunk Enterprise Security. An adaptive response action can be triggered from correlation searches or on an ad hoc basis when examining a notable event on the Incident Review dashboard.

You can use the Splunk Add-on Builder to create an alert action that supports adaptive response. Adaptive response actions must conform to the Common
Action Model, which is included with the Splunk Common Information Model add-on.

**Note** The Splunk Common Information Model add-on version 4.6.0 or later is required by:

- Add-on developers to create adaptive response actions
- End users of your add-on to run adaptive response action

Download the Splunk Common Information Model add-on from Splunkbase.

To create an adaptive response

1. On your add-on homepage, click the Create Alert Actions icon.

The Create Alert Action wizard starts.

3. On the Alert Action Properties page, enter the properties for this alert action:
   - Enter a name, label, and description for this alert action. The label is the friendly name that appears in Splunk Web.
   - Optionally, click Upload my icon if you want to add an icon, such as a logo, to this alert action.

4. Select **Support as an adaptive response action in Splunk Enterprise Security** and fill out the fields as follows:
   - For **Category**, enter the categories the action belongs to, such as "Information Gathering".
   - For **Task**, enter the functions performed by the action, such as "scan".
   - For **Subject**, enter the objects that the action's tasks can be performed on, such as "endpoint.file".
   - For **Vendor**, enter the technology vendor that the action supports.
- For **Product**, enter the products that the action supports.
- For **Version**, enter the versions of the product that the action supports.

5. Select **Support as an ad hoc action** if the action supports ad hoc invocation from the Actions menu on the Incident Review dashboard in Splunk Enterprise Security. For help determining whether your action supports ad hoc invocation, see Determine whether your action supports ad hoc invocation on the Splunk Developer Portal.

Then, fill in the related fields as follows:
- Optionally, for **Custom drilldown**, enter a URL to a custom drilldown or view for the link that appears in the detailed view of a notable even on the Incident Review dashboard in Splunk Enterprise Security. If you don't specify a URL, the default URL runs a search for the result events created by this response action.

To specify a target in an app outside Enterprise Security, use the format

```
../<app_context>/<viewname>?<additional drilldown parameters>
```

For example:

```
../my_app/my_view?form.sid=$orig_sid$&form.rid=$orig_rid$
```

To redirect to a custom view within Enterprise Security, use the format

```
/<viewname>?<additional drilldown parameters>
```

For example:

```
/my_view?form.sid=$orig_sid$&form.rid=$orig_rid$
```

- For **Sourcetype**, enter the source type to which to assign the events produced as a result of this response action.

6. Click **Next**.
7. On the **Alert Action Inputs** tab, specify one or more input fields that are required for configuring this alert action.

   For each input, drag a field from the Component Library to the center panel, then specify its properties in the Property Editor.

8. Optionally, click the **Add-on Setup Parameters** tab to define any parameters that are required for users to configure your add-on.
   - Select predefined options to prompt for account, proxy, or logging settings.
Add custom inputs by dragging fields from the Component Library to the center panel panel. Select an input to specify its properties in the Property Editor.

For more about setup pages, see Create a setup page.

9. Click Next.
10. On the Alert Action Parameters tab, in the Code Editor panel edit the Python code to create your alert action.
   ♦ Find the comment # TODO: Implement your alert action logic here, which indicates where to add your code.
   ♦ Use the Python helper functions, which are dynamically generated as commented code, as a guide to working with the input parameters you defined in the previous step.
   ♦ For example Python code showing how to create an adaptive response, see Walkthrough: Create an ES adaptive response action on the Splunk Developer Portal.
11. On the Alert Action Parameters tab, enter sample values for testing this alert action.
12. If you created a setup page, on the Add-on Setup Parameters tab enter sample values to test the setup page.
13. Click Test to test your code and the alert action configuration. The Output section indicates whether the command succeeded or failed.
14. Click Save to save the alert action, then click Finish.

You should also test and validate your response action in Enterprise Security. For more, see Validate your response action in Enterprise Security on the Splunk Developer Portal.

Pass values from setup parameters

When referring to setup parameters, you must include a namespace string before the parameter name to get the value from the parameter. The following shows an example, where param_name is the parameter name:

${__settings__.additional_parameters.param_name}

For example, let’s say you have a text input on your setup page that prompts the user for an API token, with the internal name "api_token". To pass the value of the API token to the REST call, use the following format:

api-key=${__settings__.additional_parameters.api_token}
We recommend you do not set account, proxy, or logging fields directly. You can access the global account values as follows:

${global_account.username}
{{global_account.username}}

${global_account.password}
{{global_account.password}}

**Read user credentials from multiple accounts**

Alert actions do not support the Global Account input field. When you have alert actions that require different user credentials for different accounts, such as one account for production and one for development, you can set up your alert actions to read specific user credentials from the setup page.

The following workflow shows one way to read multiple credentials of a setup page.

1. Create an alert action.
2. When creating a setup page on the Add-on Setup Page Parameters tab in the Create Alert Action wizard, select Add Global account settings to prompt users to add credentials for one or more accounts.
3. When defining the input variables for the alert action, add a Text field that prompts for the username of an account. The value of this text field acts as a dictionary key to look up the corresponding credentials from the setup page.
4. When defining the code for the alert action, use the following code to retrieve a username and password from the setup page.

   Use the same internal name (replace "internal_name" below) that you used for the text field from the previous step:

   ```python
   account = 
   helper.get_user_credential(helper.get_arg("internal_name"))
   ```

   The data returned by the `helper.get_user_credential` function is a JSON dictionary, so retrieve the username and password from the `account` dictionary as follows:

   ```python
   username = account["username"]
   password = account["password"]
   ```

5. Save your alert action, then restart Splunk Enterprise.

For more, see the Python helper functions.
Test your alert action

To test your alert action, you can use the `sendalert` command as follows using a hard-coded value:

1. Build your alert action and save it.
2. Restart Splunk Enterprise.
3. Use `sendalert` command as follows

   ```
   index=_internal | head 1| eval fieldname="xyz" | sendalert myalertname param.abc="myvalue"
   ```

To pass search result values dynamically to different alert action parameters, use the `$result.fieldname$` format as follows:

   ```
   index=_internal | head 1| eval fieldname="xyz" | sendalert myalertname param.abc="$result.fieldname$"
   ```

For a list of possible alert action tokens, see Pass search result values to alert action tokens in the Developing Views and Apps for Splunk Web manual.

End-to-end walkthroughs

For end-to-end walkthroughs, see the Splunk Developer Portal.

Learn more

For more information, see the following documentation:

- For creating adaptive response actions, see the Adaptive Response Framework on the Splunk Developer Portal
- For creating adaptive response actions, see Use the common action model to build custom alert actions in the Common Information Model Add-on Manual
- For installing the common action model, see Install the Splunk Common Information Model Add-on in the Common Information Model Add-on Manual
Validate and package

In Validate & Package:

- Validate your add-on for best practices and Splunk App Certification, and test your data inputs, field extractions, and data model mappings. The Splunk Add-on Builder shows you any errors or warnings, along with recommendations about how to address them.
- Download your add-on package as an SPL file. The SPL format is identical to the tar archive format, but uses the .spl extension rather than .tar.gz.

You can also validate and package apps and add-ons that were not created using the Add-on Builder.

About Splunk App Certification

Note: The Splunk app certification program has been retired. This page has been deprecated. For more information, see the App certification retirement FAQs.

Splunk App Certification is a process that analyzes your add-on and certifies that it meets certain criteria and verifies that your add-on conforms to best practices. Criteria and best practices may change, so for the latest information see About app certification on the Splunk Developer Portal.

The Splunk Add-on Builder can validate your add-on to determine whether it is ready to be submitted to Splunk App Certification. When you select App Precertification from the types of validation to perform, the Add-on Builder logs into the App Certification server using your Splunk.com credentials, runs the app precertification validation, then displays a "ready" or "not ready" message.

Before you run validation, be sure to provide your Splunk.com credentials.

To set Splunk.com credentials

1. From the Splunk Web home page or from the Apps menu, click Splunk Add-on Builder.
2. On the Splunk Add-on Builder home page, click Configuration at the top left of the page.
On the Configuration page, enter your Splunk.com username and password.

4. If you use a proxy server, select **Enable proxy** and fill out the proxy settings.

5. To verify your login credentials and proxy settings, click **Test**.

6. Click **Save**.

## Validate the add-on

You can select from the following types of validation to perform on your add-on:

- Best Practices
- Data Model Mapping
- Field Extraction
- App Precertification

### To validate your add-on

1. On your add-on homepage, click **Validate & Package** on the Add-on Builder navigation bar.
2. Click **Validate** to run the validation and display the results.

### Overall Health Report

This report displays a health validation score based on the errors, warnings, and failed rules. If you are validating for app precertification, this section also lets you know whether your app is ready to be submitted to Splunk App Certification.

This health score is valid in a local environment. The score could be different for the same add-on if the
There were add-ons were validated in other environments or at different times due to differences in global knowledge management and differences in indexed data. Use the health score as a subjective indicator about the overall quality of your add-on.

<table>
<thead>
<tr>
<th>Error</th>
<th>The total number of errors that were found.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warning</td>
<td>The total number of warnings that were found.</td>
</tr>
<tr>
<td>Pass</td>
<td>The number of rules that your add-on passed.</td>
</tr>
<tr>
<td>Validation result distribution</td>
<td>A chart displaying the breakdown of the validation results.</td>
</tr>
</tbody>
</table>

### Validation Rule Details

A list of all the validation rules, with details about the validation category, a description of the rule, and a suggested solution to fix the problem for rules that your add-on did not pass.

Sometimes the suggested remedy is to directly edit a configuration file (.conf). The configuration files for your app are located in `$SPLUNK_HOME/etc/apps/TA_your_addon_name`, and you can edit them in a text editor. After you save changes to the file, refresh Splunk Web by going to `http://<localhost:port>/en-US/debug/refresh`.

For more information about the best practices for add-ons, see:

- About Splunk add-ons in the *Splunk Add-ons* manual.
- Add-on Best Practice Check Tool

#### Troubleshoot validation for app precertification

If the validation process hangs during app precertification, the App Certification server might not be responding.

To troubleshoot the source of the problem, run the following search command:

```
index=_internal source=*validation.log
```

Then, if you see results such as the entry below, the problem is with the App Certification server:

```
```
Click **Stop** at any time during the validation process if you think the process is not responding.

You can also switch to other tasks in Add-on Builder while validation is running. The validation process will continue to run in the background until the process has completed or you restart Splunk Enterprise.

**Download the add-on package**

Creating the installation package for an add-on includes packaging the props, transforms, tags, event types, lookups, modular inputs, setup screens, modular alerts, and prebuilt panels into a single new app and exporting it to a downloadable file.

If you have added or modified any files directly in the file system rather than using Add-on Builder, be sure you follow the recommendations in Modify files directly before you package your add-on.

splunkd must have OS-level access to all files that you manually copy to your add-on directory. splunkd permission is defined by the user running Splunk Enterprise. Insufficient access permission is indicated by a grayed-out Download Package button.

**To download your add-on package file**

1. On your add-on homepage, click **Validate & Package** on the Add-on Builder navigation bar.
2. Click **Download Package** to download the installation package for the add-on as an SPL file.

**Credit third-party libraries**

If your add-on includes modular inputs or custom alert actions, Add-on Builder packages third-party libraries in the add-on. Credit these libraries in your documentation.

- Backbone.validation
- Bootstrap
- Decorator
- functools32
- httplib2
• Jinja2
• jqTree
• jQuery
• jQuery-resize
• jQuery UI
• jsl
• JSONPath RW
• jsonschema
• Lodash
• LowPro for jQuery
• Mako
• markupsafe
• Moment.js
• munch
• PLY
• PySocks
• Requests
• sax-js
• Schematics
• Select2
• simpleyaml
• six
• SortedContainers
• splunk-sdk-python
• Underscore.js

If your add-on includes neither modular inputs nor custom alert actions, Add-on Builder does not include any third-party libraries in the add-on.
Use the Add-ons

Use the add-on

After you download the add-on package you created using Splunk Add-on Builder, you can use this add-on to get related data in.

1. From the Splunk Web home page, click the name of the add-on in the left navigation panel.
2. Click **Configuration**.

The following example demonstrates how to set up and configure a data input in an add-on created by Splunk Add-on Builder. The example add-on requires an account and password for the data source, and uses checkpointing

Set up the add-on

**Account settings**

1. Create a account with a unique name in the **Account** field.
2. Enter the **Username** and **Password** of the account which you want to get data from.

**Proxy settings**
1. If you are using a proxy, check **Enable** and fill in the required fields. If you do not use a proxy, skip this section.
2. Select the type of proxy to use in the **Proxy Type** field.
3. Provide the proxy server address in the **Host** field.
4. Provide the proxy server port in the **Port** field. For example: 8081.
5. Provide a proxy username if you have one in the **Username** field.
6. If you provided a proxy username, type the proxy password in the **Password** field.
7. Check the **Remote DNS resolution** box if you want to perform DNS resolution through your proxy.

**Logging Settings**

1. If you want to change the **Log level**, select a new level from the drop down menu.

**Configure inputs**
After you set up the add-on, you can configure the inputs to collect data via Splunk Web. Click **Create New Input** in the Inputs tab and configure the following.

1. Enter a **Name**, **Interval** and **Index**.
   - **Name**: The name of the input.
   - **Index**: The index in which you want the Splunk platform to store indexed data.
   - **Interval**: The number of seconds to wait before the Splunk platform runs the command again.

2. Choose the **Checkpoint type** and enter the checkpoint value. If you need more details on setting modular input checkpoints, see Data checkpoints in *Developing Views and Apps for Splunk Web*.
   - **Checkpoint type**: There are two checkpoint type, **Auto** and **File**.
     - ♦ If you choose **Auto**, the checkpoint file will be saved in the **KVstore**.
     - ♦ If you choose **File**, the checkpoint file will be saved under folder $SPLUNK_HOME$/var/lib/splunk/modinputs.

3. Restart the Splunk platform after configuring the inputs.
Advanced

Create a new data model

From version 2.2.0, Splunk add-on builder supports the user to map the data event to the data model you create. Users can design and maintain data models and use them in Splunk Add-on builder. Splunk recommends you to use Splunk web first and then modify the data model JSON file to follow the standard of Add-on builder.

To create data models which can be used by Splunk add-on builder, you need to understand

- What is data models and how to create a data model in Splunk platform.
- The format and semantics of their indexed data and are familiar with the Splunk search language. In building a typical data model, knowledge managers use knowledge object types such as lookups, transactions, search-time field extractions, and calculated fields.
- The data model standard of Splunk add-on builder. See syntax of data model for details.

Make sure you have sufficient access permissions to any files you place in your add-on directory.

Syntax of data model

After building data model using Splunk web, the generated JSON file cannot be used by Splunk Add-on builder directly, add the following fields to the existing JSON file.

**Required field**

objects.comment.tags

**Syntax:** $.objects[*].comment.tags  
**Description:** It defines the tags of object in the data model. Eventtype which has the same tag(s) will be mapped to this data model.
**Optional fields**

objects.comment.description

**Syntax:** `$.objects[*].comment.description`

**Description:** The description of the data model.

object.fields.comment.description

**Syntax:** `$.object[*].fields[*].comment.description:<string>`

**Description:** The description of the data model field.

object.fields.comment.expected_values

**Syntax:** `$.object[*].fields[*].comment.expected_values:<string>`

**Description:** It defines the expected value(s) of the data model field. Splunk add-on builder verifies the expect value(s) when user validates the add-on.

**Example: Create a data model named test**

1. Create the data model using Splunk Web and name it as `test`.
2. Open the `test.json` file under `$SPLUNK_HOME/etc/apps/<your_addon_folder>/default/data/models/test.json` and add the field required by Splunk Add-on builder as follows.

3. Save the file and then restart Splunk.
4. The data model you create will be listed on the **Select Data Models** page. Follow the instruction on how to map to data model.
Modify conf files directly

For certain situations, you might want to modify your add-on outside of the Add-on Builder by modifying files directly. Follow these guidelines for best results:

- Make sure you have sufficient access permissions to any files you place in your add-on directory.
- When modifying configuration files, for example to add stanzas, modify the version in the $SPLUNK_HOME/etc/apps/your_addon/default directory.
- If you need to add a configuration file, add the file to the $SPLUNK_HOME/etc/apps/your_addon/default directory.

Advanced modifications might affect the add-on's eligibility for Splunk App Certification.

Learn more

For more information, see the following Splunk Enterprise documentation:

- About configuration files
- Configuration file precedence

Python helper functions

The Add-on Builder provides Python code templates for creating modular data inputs and alert actions, with helper functions for working with parameters.

You can only access input arguments using helper functions.
## Functions to get values from input parameters

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>get_app_name()</code></td>
<td>Returns the name of the add-on.</td>
</tr>
<tr>
<td><code>get_input_type()</code></td>
<td>Returns the value of <strong>Input name</strong> on the <strong>Data Input Properties</strong> tab of the <strong>Inputs &amp; Parameters</strong> step.</td>
</tr>
</tbody>
</table>
| `get_input_stanza(input_stanza_name=None)`   | If `input_stanza_name` is provided, returns a `dict` containing all stanza parameters for the specified stanza name, or "None" if the stanza name does not exist.  
If `input_stanza_name` is not provided, returns all stanzas in a `dict`, with the stanza name as the key and detailed stanza parameters as the value.  
**Note** When `input_stanza_name` is not provided, one key-value pair is returned for multi-instance mode, and multiple key-value pairs are returned for single-instance mode. |
| `get_input_stanza_names()`                   | Returns all stanza names for this input. A single string value is returned for multi-instance mode. A list of stanza names is returned for single-instance mode.                                                   |
| `get_arg(arg_name, input_stanza_name=None)`  | Returns an argument value based on `arg_name` and `input_stanza_name`. Arguments are defined on the **Data Input Parameters** tab of the Data Transformation step.                                        |
the Inputs & Parameters step. **arg_name** refers to the input field’s Internal name.

For multi-instance mode, **input_stanza_name** is not necessary, and the returned arg value is "None".

For single-instance mode, if **input_stanza_name** is "None", a dict with the stanza name as the key and the arg value as the value is returned. If **input_stanza_name** is not "None", a single value of "None" is returned.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>get_sourcetype(<strong>input_stanza_name=None</strong>)</td>
<td>Returns the source type.</td>
</tr>
<tr>
<td>get_output_index(<strong>input_stanza_name=None</strong>)</td>
<td>Returns the output index.</td>
</tr>
</tbody>
</table>

### Functions to get values from setup parameters

Helper functions to get credential accounts are used in rare cases. If you need an account in your modular input code, add a **Global Account** field to your setup parameters, then use `helper.get_arg("internal_name")`, where **internal_name** is the internal name of the **Global Account** field, to retrieve the value.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| get_proxy()                             | If **Proxy settings** is selected, returns a dict: {
|                                        | "proxy_url": ..., "proxy_port": ..., "proxy_username": ..., "proxy_password": ..., "proxy_type": ..., "proxy_rdns": ...
|                                        | Otherwise, an empty dict is returned.                                       |
| get_user_credential_by_username(username) | If **username** exists, returns the dict: {
|                                        | "name": account_id, "username": username, "password": ...}                 |
get_user_credential_by_id(account_id)  
If account_id exists, returns the dict:  
"name": account_id,  
"username": username,  
"password": password  
Otherwise, returns "None".

global_setting(var_name)  
Returns a global setting value for var_name. This method is used to get custom settings.

get_log_level()  
Returns the log level that was selected. If not set, returns "INFO".

set_log_level(level)  
Sets the log level to level.

log(msg)  
Logs a message using the configured logging level.

log_debug(msg)  
Logs a message using the logging.DEBUG level.

log_info(msg)  
Logs a message using the logging.INFO level.

log_warning(msg)  
Logs a message using the logging.WARNING level.

log_error(msg)  
Logs a message using the logging.ERROR level.

log_critical(msg)  
Logs a message using the logging.CRITICAL level.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>save_check_point(key, state)</td>
<td>Saves the checkpoint {key: state} to the Splunk KV store.</td>
</tr>
<tr>
<td>delete_check_point(key)</td>
<td>Deletes the checkpoint corresponding to key.</td>
</tr>
</tbody>
</table>
get_check_point(key)

Get the value of the checkpoint corresponding to `key`.

batch_save_check_point(states)

Saves a batch of `key: state` pairs, where `states` is a dict, the `key` is the checkpoint state key, and `state` is the value.

### Functions for alert actions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| addevent(event_message, sourcetype) | Adds a custom event to the Adaptive Response Framework.  
  • `event_message`: The event message.  
  • `sourcetype`: The source type of this event. |
| writeevents(index, host, source) |Flushes the custom Adaptive Response Framework events. Call this function only once in your adaptive response.  
  • `index`: The index that stores these events.  
  • `host`: The host fields of these events.  
  • `source`: The source fields of these events. |
| get_events()                 | Gets an event list that contains all of the search results when this alert is triggered. |
| get_param(param_name)        | Gets the parameter value when the alert is triggered for `param_name`, which is the internal name of the alert parameter. |

### Other functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| send_http_request(url, method, parameters=None, payload=None, headers=None, cookies=None, verify=True, cert=None, timeout=None, use_proxy=True) | Sends an HTTP request and returns the response.  
  • `url`: URL for the new Request object.  
  • `method`: Method for the new Request object (GET | POST |
PUT | DELETE.

- **parameters**: Optional. Dictionary or bytes to be sent in the query string for the request.
- **payload**: Optional. Dictionary, bytes, or file-like object to send in the body of the request.
- **headers**: Optional. Dictionary of HTTP headers to send with the request.
- **cookies**: Optional. Dict or CookieJar object to send with the request.
- **verify**: Optional. Indicates whether the SSL certificate will be verified. A CA_BUNDLE path can also be provided.
- **cert**: Optional. If a string, contains the path to the SSL client cert (.pem) file. If Tuple, a ('cert', 'key') pair.
- **timeout** (float or tuple): Optional. Specifies how long to wait for the server to send data, as a float or a (connect timeout, read timeout) tuple.
- **use_proxy**: Optional. Indicates whether to use a proxy. If True, the proxy in the Add-on Builder Configuration settings is used.

### new_event

**new_event**(data, time=None, host=None, index=None, source=None, sourcetype=None, done=True, unbroken=True)

Creates a new event. This function is used to index data in Splunk Enterprise.

The example in the code template shows `ew.write_event(event)`. Refer to `splunklib.modularinput.Event` in splunklib.
Release Notes

What's new

Here's what's new in each version of the Splunk Add-on Builder.

Version 2.2.0

What's new in this release:

- The Add-on Builder allows you to map your data events to any data model which follows the Splunk Add-on Builder standard. For details, see Map to data model.
- The Add-on Builder allows you to manage source types and extract fields of the other apps and add-ons which were not created using Add-on Builder.
- The Add-on Builder provides you three ways to do the field extractions: assisted extractions, manual extractions, and manual transformation. For details, see Extract fields.
- You can now import and export project on Splunk Web or by using the migration scripts. For details, see Import and export an add-on project.
- Improvements have been made to Manage source type page. For details, see Manage source types.
- The documentation now provides guidance for what third-party libraries to credit in the add-ons you create with Add-on Builder. See Credit third-party libraries.

Version 2.1.2

What's new in this release:

- Bug fixes. For details, see Fixed issues.

Version 2.1.1

What's new in this release:

- Bug fixes. For details, see Fixed issues.
Version 2.1.0

What's new in this release:

- The Add-on Builder has a new and enhanced setup library consistent with modern Splunk-built add-ons.
- The Add-on Builder no longer requires you to restart Splunk Enterprise when building new data inputs, creating a new add-on, or any other step.
- The Add-on Builder home page has been updated and improved.
- You can now import and export add-on projects, allowing you to work on an add-on on different computers and share projects with others. For details, see Import and export add-on projects.
- You can now use the Add-on Builder to add CIM mapping to apps and add-ons that were not created using Add-on Builder. For details, see Work with other apps and add-ons.
- When you configure data inputs using a REST API, you can now add checkpointing for incremental data collection and set JSON event breaking. For details, see Add a data input using a REST API.
- Improvements have been made to the CIM mapping UI. For details, see Map to CIM.
- Changes have been made to the way global accounts are implemented in the setup page for an add-on. For details, see Create a setup page.

Upgrade add-ons from earlier versions

When you open an add-on project that was created in an earlier version of Add-on Builder, the add-on is automatically upgraded. Note the following changes:

- Any UI or dashboards that were created outside of Add-on Builder will no longer be available.
- If you made any changes outside of Add-on Builder to configuration files, review these files in case your changes were not migrated.
- Setup pages have changed in version 2.1.0 and are upgraded to the new style. Due to changes between these styles, review your setup pages and verify the text appears correctly. Ensure that display labels for input fields are unique.
- If you created data inputs with Python code in Add-on Builder 2.0.0, you must upgrade your code for version 2.1.0.

Replace: def collect_events(helper, inputs, ew):
With: def collect_events(helper, ew):

85
Version 2.0.0

What's new in this release:

- The validation process has been enhanced to include App Certification readiness. This validation process can also be performed on apps and add-ons that were created outside of Add-on Builder. For details, see About Splunk App Certification.
- You can now create alert actions for your add-ons using the Add-on Builder. For details, see Create alert actions.
- You can now create adaptive response actions for use with Splunk Enterprise Security. For details, see Create an adaptive response action for Enterprise Security.
- The Add-on Builder is easier to use with a new workflow and enhanced UI.
- Bug fixes. For details, see Fixed issues.

Version 1.1.0

What's new in this release:

- Additional data formats (JSON, XML, and tabular) are now supported, improving field extraction.
- Adding sample data is easier, with the option to upload sample data files and indexed data from Splunk Enterprise.
- Support for field extraction and CIM mapping has been added for indexed data in Splunk Enterprise.
- The framework for modular inputs has changed. See below for details about migrating your code.
- Bug fixes. For details, see Fixed issues.

Migrating modular input code from an earlier release

The framework for modular inputs has changed. If you are upgrading the Add-on Builder app from a previous release, update your Python code for any modular inputs you created as described below.

Create events

Use the `self.new_event` method to create events for modular inputs rather than creating an event directly using `smi.Event`. The arguments are the same.

For example, in the auto-generated code for modular inputs, you would update the sample code as follows.
1. Find this line:

```python
event = smi.Event(source=self.input_name, index=self.output_index,
sourcetype=self.output_sourcetype, data=data)
```

2. Replace it with this line:

```python
event = self.new_event(source=self.get_input_name(),
index=self.get_output_index(), sourcetype=self.get_sourcetype(),
data=data)
```

**Read configurations from setup pages**

The following example code, also included in the auto-generated code for modular inputs, shows how to read configurations from setup pages in the new framework for version 1.1.0:

```python
arg_value = self.get_arg('arg_name')
proxy_settings = self.get_proxy()
account = self.get_user_credential("admin")
userdefined = self.get_global_setting("userdefined")
```

**Version 1.0.1**

What's new in this release:

- Bug fixes. For details, see Fixed issues.

**Version 1.0.0**

This is the first release of the Splunk Add-on Builder.

**Known issues for Splunk Add-on Builder**

This version of Splunk Add-on Builder has the following known issues and workarounds.

<table>
<thead>
<tr>
<th>Date filed</th>
<th>Issue number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-09-25</td>
<td>TAB-2762</td>
<td>Non-standard characters in other add-on's clear passwords can cause Add-on Builder setup to hang. Workaround: Disable the offending TA.</td>
</tr>
</tbody>
</table>
Incorrect license agreement packaged in Add-on Builder

Workaround:
See the license linked from the Splunkbase page for the correct licensing information.

Renaming add-on and then deleting it causes an internal error.

Some characters in sourcetype are not supported by Add-on Builder

Alert action not working on Windows if TA path is too long

Sourcetype set for data input log file doesn't take effect until restart

Error when user installs both Splunk DB Connect and Add-on Builder.

---

**Fixed issues for Splunk Add-on Builder**

<table>
<thead>
<tr>
<th>Date resolved</th>
<th>Issue number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-06-20</td>
<td>TAB-2043</td>
<td>On Windows, could not rename or delete add-on with modular input running.</td>
</tr>
<tr>
<td>2017-05-23</td>
<td>TAB-2247</td>
<td>Remove setup configurations when all data inputs and alert actions are deleted</td>
</tr>
<tr>
<td>2017-05-17</td>
<td>TAB-1263</td>
<td>Best practice check for duplicated stanza header or field will pass even the duplication exists.</td>
</tr>
<tr>
<td>2017-04-25</td>
<td>TAB-1978</td>
<td>Renaming a visible add-on will create an icon in Splunk Web home page before restart.</td>
</tr>
<tr>
<td>2017-04-13</td>
<td>TAB-2045</td>
<td>Error occurred in importing add-on</td>
</tr>
<tr>
<td>2017-04-11</td>
<td>TAB-1934</td>
<td>REST data input highlights the wrong lines.</td>
</tr>
<tr>
<td>2017-03-13</td>
<td>TAB-2124</td>
<td>Modular input checkpoint highlighting does not work when response JSON is an array.</td>
</tr>
<tr>
<td>2017-03-08</td>
<td>TAB-2275</td>
<td>JSON path highlight rendering is slow for some expression.</td>
</tr>
</tbody>
</table>
Third-party software credits for Splunk Add-on Builder

Some of the components included in Splunk Add-on Builder are licensed under free or open source licenses. We wish to thank the contributors to these projects.

**Backbone.validation**

Version 0.9.1

Copyright © 2014 Thomas Pedersen, http://thedersen.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**Bootstrap**

Version 2.3.1

The MIT License (MIT)

Copyright (c) 2011-2014 Twitter, Inc

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Decorator

https://github.com/micheles/decorator

Version 4.0.11

Copyright (c) 2005-2017, Michele Simionato All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in bytecode form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DefiantJS

https://github.com/hbi99/defiant.js

Version 1.4.0

Copyright (c) 2013 Hakan Bilgin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

functools32

https://pypi.python.org/pypi/functools32/3.2.3-2

Version 3.2.3-2

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using this software ("Python") in source or binary form and its associated
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006 Python Software Foundation; All Rights Reserved" are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0
1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.
IMPORTANT: PLEASE READ THE FOLLOWING AGREEMENT CAREFULLY.

BY CLICKING ON "ACCEPT" WHERE INDICATED BELOW, OR BY COPYING, INSTALLING OR OTHERWISE USING PYTHON 1.6, beta 1 SOFTWARE, YOU ARE DEEMED TO HAVE AGREED TO THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT.

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6, beta 1 software in source or binary form and its associated documentation, as released at the www.python.org Internet site on August 4, 2000 ("Python 1.6b1").

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6b1 alone or in any derivative version, provided, however, that CNRI's License Agreement is retained in Python 1.6b1, alone or in any derivative version prepared by Licensee.

Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6, beta 1, is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1011. This Agreement may also be obtained from a proxy server on the Internet using the URL: http://hdl.handle.net/1895.22/1011".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6b1 or any part thereof, and wants to make the derivative work available to the public as provided herein, then Licensee hereby agrees to indicate in any such work the nature of the modifications made to Python 1.6b1.

4. CNRI is making Python 1.6b1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6b1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING PYTHON 1.6b1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by and interpreted in all respects by the law of the State of Virginia, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6b1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

httplib2

https://github.com/httplib2

Version 0.9.2

Copyright (c) 2006 by Joe Gregorio

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ImmutableJS

https://github.com/facebook/immutable-js

Version 3.8.1

Copyright (c) 2014-2015, Facebook, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
• Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

• Neither the name Facebook nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JavaScript Cookie

https://github.com/js-cookie/js-cookie

Version 2.1.2

Copyright 2014 Klaus Hartl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Jinja2

http://jinja.pocoo.org/

Version 2.8

Copyright (c) 2009 by the Jinja Team, see AUTHORS for more details.

Some rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The names of the contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
**jqTree**

Version 0.19.0

https://github.com/mbraak/jqTree

Copyright 2011 Marco Braak

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

**jQuery**

Version 2.1.0

Copyright 2014 jQuery Foundation and other contributors

http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
 OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jQuery-resize

https://github.com/cowboy/jquery-resize

Version 1.1

Copyright (c) 2010 "Cowboy" Ben Alman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jQuery UI

https://github.com/jquery/jquery-ui

Version 1.10.4

Copyright 2014 jQuery Foundation and other contributors, http://jqueryui.com/

This software consists of voluntary contributions made by many individuals (AUTHORS.txt, http://jqueryui.com/about) For exact contribution history, see the revision history and logs, available at http://jquery-ui.googlecode.com/svn/
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in the
Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jsl

https://github.com/aromanovich/jsl

Version 0.2.4

Copyright (c) 2016 by Anton Romanovich.

Some rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

- The names of the contributors may not be used to endorse or promote
  products derived from this software without specific prior written
  permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**JSONPath RW**

https://github.com/kennknowles/python-jsonpath-rw

Version 1.4.0

Copyright 2013- Kenneth Knowles

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

**jsonschema**

https://github.com/Julian/jsonschema

Version 2.5.1

Copyright (c) 2013 Julian Berman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the
Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**Lodash**

https://github.com/lodash/lodash

Version 4.17.3

Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

**LowPro for jQuery**

https://github.com/danwrong/low-pro-for-jquery

Copyright (c) 2009 Dan Webb

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**Mako**

http://docs.makotemplates.org/

Version 1.0.4

Copyright (C) 2006-2016 the Mako authors and contributors <see AUTHORS file>. Mako is a trademark of Michael Bayer.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**markupsafe**

https://github.com/pallets/markupsafe

Version 1.0

Copyright (c) 2010 by Armin Ronacher and contributors. See AUTHORS for more details.

Some rights reserved.
Redistribution and use in source and binary forms of the software as well as
documentation, with or without modification, are permitted provided that the
following conditions are met:

- Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

- The names of the contributors may not be used to endorse or promote
  products derived from this software without specific prior written
  permission.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED BY THE
COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE AND DOCUMENTATION, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

Moment.js

https://github.com/moment/moment

Version 2.8.3

Copyright (c) 2011-2015 Tim Wood, Iskren Chernev, Moment.js contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in the
Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

munch

http://github.com/Infinidat/munch

Version 2.0.4

Copyright (c) 2010 David Schoonover

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
PLY

http://www.dabeaz.com/ply/

Version 3.9

Copyright (C) 2001-2016 David M. Beazley (Dabeaz LLC) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the David Beazley or Dabeaz LLC may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PySocks

https://github.com/Anorov/PySocks

Version 1.6.5

Copyright 2006 Dan-Haim. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of Dan Haim nor the names of his contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY DAN HAIM "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DAN HAIM OR HIS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

React

https://github.com/facebook/react

Version 15.5.4

Copyright (c) 2013-present, Facebook, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name Facebook nor the names of its contributors may be used to endorse or promote products derived from this software without specific
prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

React Redux

https://github.com/reactjs/react-redux

Version 4.4.8

Copyright (c) 2015 Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Redux

https://github.com/reactjs/redux

Version 3.6.0

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

redux-actions

https://github.com/acdlite/redux-actions

Version 1.2.0

Copyright (c) 2016 Andrew Clark

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

redux-immutable

https://github.com/gajus/redux-immutable

Version 3.0.9

Copyright (c) 2016, Gajus Kuizinas (http://gajus.com/) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Gajus Kuizinas (http://gajus.com/) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANUARY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**redux-observable**

https://github.com/redux-observable/redux-observable

Version 0.12.2

Copyright (c) 2016 Ben Lesh, Jay Phelps, and redux-observable contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**Regulex**

https://github.com/JexCheng/regulex

Version 0.0.2

Copyright (c) 2014 Jex

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Requests

https://github.com/kennethreitz/requests

Version 2.12.4

Copyright 2016 Kenneth Reitz

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

RxJS

https://github.com/ReactiveX/rxjs

Version 5.0.2

Copyright 2015-2016 Netflix, Inc., Microsoft Corp. and contributors

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the
License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

sax-js

https://github.com/isaacs/sax-js

Version 1.1.0

Copyright (c) Isaac Z. Schlueter ("Author") All rights reserved.

The BSD License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
The file "examples/strict.dtd" is licensed by the W3C and used according to the terms of the W3C SOFTWARE NOTICE AND LICENSE. See LICENSE-W3C.html for details.

"String.fromCharCode" used under the terms of the MIT license. Its license follows:

Copyright Mathias Bynens <http://mathiasbynens.be/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Schematics

https://pypi.python.org/pypi/schematics/2.0.0a1

Version 2.0.0a1

Copyright (c) 2013?2016 Schematics Authors; see AUTHORS for details All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of Schematics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Select2

http://select2.github.io/select2/

Versions 3.4.6

Copyright 2014 Igor Vaynberg

This software is licensed under the Apache License, Version 2.0 (the "Apache License") or the GNU General Public License version 2 (the "GPL License"). You may choose either license to govern your use of this software only upon the condition that you accept all of the terms of either the Apache License or the GPL License.

You may obtain a copy of the Apache License and the GPL License in the LICENSE file, or at:

http://www.apache.org/licenses/LICENSE-2.0
http://www.gnu.org/licenses/gpl-2.0.html
Unless required by applicable law or agreed to in writing, software distributed under the Apache License or the GPL License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Apache License and the GPL License for the specific language governing permissions and limitations under the Apache License and the GPL License.

**simpleyaml**

https://pypi.python.org/pypi/simpleyaml

Version 1.0

Copyright (c) 2012 Kirill Simonov and Reuven V. Gonzales

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**six**

https://github.com/benjaminp/six

Version 1.10.0

Copyright (c) 2010-2015 Benjamin Peterson
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in the
Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Sortable

https://github.com/RubaXa/Sortable

Version 1.5.1

Copyright 2013-2016 Lebedev Konstantin ibnRubaXa@gmail.com
http://rubaxa.github.io/Sortable/

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in the
Software without restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
Underscore.js

Version 1.6.0

Copyright (c) 2009-2014 Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

uuid

https://github.com/kelektiv/node-uuid

Version 3.0.1

Copyright (c) 2010-2016 Robert Kieffer and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.