

# **Spring Integration Splunk Adapter**

1.1.0.RELEASE

Jarred Li, Mark Pollack, Damien Dallimore

Copyright ©

© SpringSource Inc., 2012

# Table of Contents

I. What's new? .....	1
1. What's new? .....	2
II. Integration Adapters .....	3
2. Splunk Adapter .....	4
2.1. Outbound Channel Adapter .....	4
2.2. Inbound Channel Adapter .....	5
III. Appendices .....	7
A. Additional Resources .....	8
A.1. Spring Integration Home .....	8
A.2. Splunk Home .....	8
B. Change History .....	9

---

# Part I. What's new?

If you are interested in the changes and features, that were introduced in earlier versions, please take a look at chapter: Appendix B, *Change History*

## 1. What's new?

The Spring Integration adapter for Splunk includes two adapters:

- Inbound Channel Adapter to search data from Splunk.
- Outbound Channel Adapter to push event data into Splunk.

---

## Part II. Integration Adapters

Spring Integration adapter for Splunk includes inbound channel adapter to read data from Splunk and outbound channel adapter to write data into Splunk.

## 2. Splunk Adapter

The Spring Integration Splunk Adapter provides outbound and inbound channel adapters:

- [Outbound Channel adapter](#)
- [Inbound Channel Adapter](#)

To use Spring Integration adapter for Splunk, you have to import the XML namespace. For example, you can have following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:int="http://www.springframework.org/schema/integration"
  xmlns:int-splunk="http://www.springframework.org/schema/integration/splunk"
  xsi:schemaLocation="http://www.springframework.org/schema/integration/splunk
    http://www.springframework.org/schema/integration/splunk/spring-integration-splunk.xsd
    http://www.springframework.org/schema/integration
    http://www.springframework.org/schema/integration/spring-integration.xsd
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">

</beans>
```

Meanwhile, you have to define your Splunk server information. For example you can define server as following:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:int-splunk="http://www.springframework.org/schema/integration/splunk"
  xsi:schemaLocation="http://www.springframework.org/schema/integration/splunk
    http://www.springframework.org/schema/integration/splunk/spring-integration-splunk.xsd
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">
  ...
  <int-splunk:server id="splunkServer" host="somehost" port="8089"
    userName="user" password="password" owner="admin"/>
  ...
</beans>
```

### 2.1 Outbound Channel Adapter

Outbound channel adapter is used to put data into Splunk from channels in Spring Integration. There are 3 kinds of method to put data:

- Submit (HTTP REST)
- Stream
- Tcp

The main difference between using the REST inputs vs plain TCP/UDP inputs is really in the Splunk event handling pipeline.

With REST, you have to declare your event meta data (index, source, source type...) in the HTTP request at the source. You can't really transform the log event anymore after you have created and

sent it to Splunk. Typically though, for people using REST, this is fine because they are well formatting their log events before sending them anyway ie: no further processing/transforming and manipulation is required. You can however still do dynamic search time transforms/filtering on the data when later searching over it in Splunk.

To use outbound channel adapter with submit, you can define the adapter as following:

```
<int-splunk:outbound-channel-adapter id="splunkOutboundChannelAdapter"
  auto-startup="true" order="1"
  channel="outputToSplunk"
  splunk-server-ref="splunkServer"
  pool-server-connection="true" sourceType="spring-integration" source="example"
  ingest="submit">
</int-splunk:outbound-channel-adapter>
```

With TCP inputs, you can manipulate and transform the event data in Splunk before it gets indexed (using entries in props.conf/transforms.conf). The event meta data (index, source, source type...) gets declared on the Splunk side when you establish the TCP/UDP input and can also be dynamically created, so essentially you have a lot more control over the indexing of the event data. This is generally more important when you don't control the format of the data at the source and it needs manipulating/filtering ie: network devices syslogging etc...

To use outbound channel adapter with tcp, you can define the adapter as following:

```
<int-splunk:outbound-channel-adapter
  id="splunkOutboundChannelAdapter" auto-startup="true" order="1"
  channel="outputToSplunk" splunk-server-ref="splunkServer"
  ingest="tcp" tcpPort="9999">
</int-splunk:outbound-channel-adapter>
```

To use outbound channel adapter with stream, you can define the adapter as following:

```
<int-splunk:outbound-channel-adapter
  id="splunkOutboundChannelAdapter" auto-startup="true" order="1"
  channel="outputToSplunk" splunk-server-ref="splunkServer"
  ingest="stream">
</int-splunk:outbound-channel-adapter>
```

## 2.2 Inbound Channel Adapter

Inbound channel adapter is used to get data out of Splunk and put into Spring Integration's channel. There are 5 ways to get data out of Splunk:

- Blocking
- Non blocking
- Saved search
- Realtime
- Export

For more information on the difference, please refer [Splunk SDK](#)

To use blocking inbound channel adapter, you can define the adapter as following:

```
<int-splunk:inbound-channel-adapter id="splunkInboundChannelAdapter"
  auto-startup="true" search="search spring:example"
  splunk-server-ref="splunkServer"
  channel="inputFromSplunk" mode="blocking" initEarliestTime="-1d">
  <int:poller fixed-rate="5" time-unit="SECONDS"/>
</int-splunk:inbound-channel-adapter>
```

To use non blocking inbound channel adapter, you can define the adapter as following:

```
<int-splunk:inbound-channel-adapter id="splunkInboundChannelAdapter"
  auto-startup="true" search="search spring:example"
  splunk-server-ref="splunkServer"
  channel="inputFromSplunk" mode="normal" initEarliestTime="-1d">
  <int:poller fixed-rate="5" time-unit="SECONDS"/>
</int-splunk:inbound-channel-adapter>
```

To use saved search inbound channel adapter, you can define the adapter as following:

```
<int-splunk:inbound-channel-adapter id="splunkInboundChannelAdapter"
  auto-startup="true" savedSearch="test" splunk-server-ref="splunkServer"
  splunk-server-ref="splunkServer"
  channel="inputFromSplunk" mode="saved" initEarliestTime="-1d">
  <int:poller fixed-rate="5" time-unit="SECONDS"/>
</int-splunk:inbound-channel-adapter>
```

To use realtime search inbound channel adapter, you can define the adapter as following:

```
<int-splunk:inbound-channel-adapter id="splunkInboundChannelAdapter"
  auto-startup="true" search="search spring:example"
  splunk-server-ref="splunkServer"
  channel="inputFromSplunk" mode="realtime" initEarliestTime="-1d">
  <int:poller fixed-rate="5" time-unit="SECONDS"/>
</int-splunk:inbound-channel-adapter>
```

To use export inbound channel adapter, you can define the adapter as following:

```
<int-splunk:inbound-channel-adapter id="splunkInboundChannelAdapter"
  auto-startup="true" search="search spring:example"
  splunk-server-ref="splunkServer"
  channel="inputFromSplunk" mode="export" initEarliestTime="-1d">
  <int:poller fixed-rate="5" time-unit="SECONDS"/>
</int-splunk:inbound-channel-adapter>
```

As Splunk support range search, you can specify the search range by using "latestTime", "earliestTime", "initEarliestTime".

"initEarliestTime" is the value for "earliestTime" when the application first start. If you specify "earliestTime" and "latestTime", the poller will only search data in that range. Otherwise, the range will move forward as time goes. That means, the "latestTime" is equal to the time where the polling trigger, the "earliestTime" is equal to the time where the last polling is run.

You can get more information on the range search from [Splunk](#).



---

## Part III. Appendices

---

# Appendix A. Additional Resources

## A.1 Spring Integration Home

The definitive source of information about Spring Integration is the [Spring Integration Home](http://www.springsource.org) at <http://www.springsource.org>. That site serves as a hub of information and is the best place to find up-to-date announcements about the project as well as links to articles, blogs, and new sample applications.

## A.2 Splunk Home

You can get more information on Splunk from [Splunk Home](#).

Splunk SDK API is in [Splunk Dev](#).

# Appendix B. Change History

*Table B.1.*

Release	Date	Changes
0.5.0	2012.9.28	Initial release