

Using Input Transforms

Date published: 2019-12-17

Date modified: 2021-03-25



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Creating Input Transforms.....	4
---------------------------------------	----------

Creating Input Transforms

Input Transforms are a powerful way to clean, mutate, and arrange data that is poorly organized, has changing format, has data that is junk or otherwise hard to use. You create a javascript function to transform the data after it has been consumed from the source and before queries happen in SQL Stream Builder.

About this task

You define your schema on the output of the transform. You can use the input transforms on the following sources:

- Not in your control, for example, data feed from a third party provider
- The format is hard to change, for example, a legacy feed, other teams of feeds within your organization
- Inconsistent message to message
- Have no uniform keys, or data without keys (like nested arrays), but are still valid JSON
- The schema you want does not match the incoming topic exactly



In case your input data is JSON, and it is hard to express with a schema, you can use Input Transformations.

- Input Transformations are defined for a virtual table source.
- One transformation per source.
- Takes record as a JSON formatted string input variable. The input is always named record.
- Emits the output of the last line to the calling JVM. It could be any variable name. In the following examples, out and emit is used as a JSON formatted string.

A basic input transformation looks like this:

```

var out = JSON.parse(record);    // record is input, parse JSON formatted s
tring to object                  tring to object
                                // do whatever you want here..
JSON.stringify(out);             // emit JSON formatted string of o
bject
  
```

Procedure

1. Select Console from the left hand menu.
2. Select the Virtual Tables tab.
3. Select the edit button (pencil icon) for the Virtual Table Name you would like to add a transform to.
4. Select the Display/Edit Transformations link.
5. Add your transform code.
6. Make sure the output of your transform matches the Manual Schema definition.