

# Automating PDI ETL Testing using Behaviour Driven Development Framework

Ali Akkas  
January 2020

# Questions?

- How do you verify that an ETL process work as expected?
- Are you able to automate integration or end-to-end testing?
- How do you integrate Behaviour Driven Development testing using Pentaho Data Integration?

# Example of an ETL testing process

- Understand user stories and acceptance criteria
- Test preparation
- Design test cases
- Validate data sources
- Execute ETL scripts
- Validate & verify result
- Write test report

# Behaviour Driven Development

- BDD is a way for software teams to work that closes the gap between business people and technical people by:
- Encouraging collaboration across roles to build shared understanding of the the problem to be solved
- Working in rapid, small iterations to increase feedback and the flow of value
- Producing system documentation that is automatically checked against the system's behaviour

# Let's start with a simple user story

Feature: convert a date to ISO8601 date format

Given the ETL script will receive a date field as input, convert it to the ISO8601 standard format and output it as xml tag.

When mapping the following inputs

#	input	expected_result
1	2019-12-01	2019-12-01T00:00:00.000+0000
2	Mon, 12 Aug 2019 08:47:24 +0200	2019-08-12T06:47:24.000+0000
3	2020-01-18T00:00:00.000+0000	2020-01-18T00:00:00.000+0000
4	2019-12-10T12h34:45:678+0400	2019-12-10T08:34:45.678+0000
5	<null>	INVALID DATE

Then the following outputs will be generated

#	input	expected_result	enriched_xml
1	2019-12-01	2019-12-01T00:00:00.000...	<dateiniso8601 calculationFailed="true"/>
2	Mon, 12 Aug 2019 08:47:24...	2019-08-12T06:47:24.000...	<dateiniso8601 calculationFailed="true"/>
3	2019-12-10T12h34:45:678...	2019-12-10T08:34:45.67...	<dateiniso8601 calculationFailed="true"/>
4	2020-01-18T00:00:00.000+...	2020-01-18T00:00:00.000...	<row><dateiniso8601>2020-01-18T00:00:00.000+0000</dateiniso8601></row>
5	<null>	INVALID DATE	<row><dateiniso8601></dateiniso8601></row>

# Create a feature file using Gherkin language

```
Feature: convert a date to ISO8601 date format
will receive a date field as input, convert it to the ISO8601 standard format and output it as xml tag.
```

```
Background: Perform basic authentication and set JSESSIONID
```

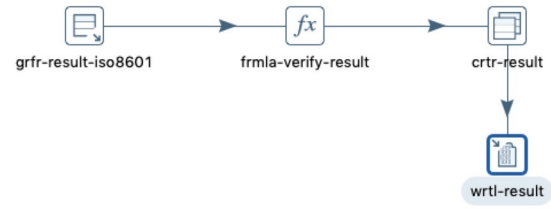
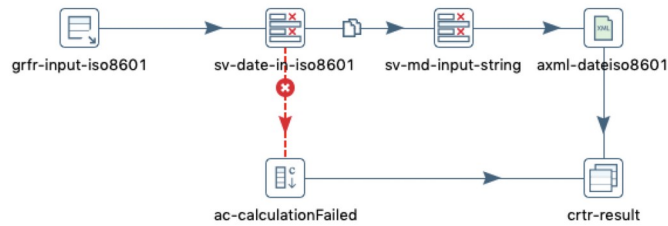
```
* url 'http://127.0.0.1:9999'
* def setup = callonce read('pdi-cookies.feature')
* cookies { 'JSESSIONID': #(setup.jsessionid) }
```

```
@GetenrichedData
```

```
Scenario: Get enriched data
```

```
Given path 'pdi/execute'
And params { P_TRANSFORMER: 'dateiso8601', filename: '|/Users/aliakkas/apps/spring/etl/jb-acceptancetest.kjb' }
When method get
Then status 200
And match $.pdi-result.result.nr_errors == '0'
* def result = $.pdi-result.result.result-rows.row-data
* print result
* def log = $.pdi-result.result.log_text
* print log
* def data = $result[*].value-data[*]
* print data
```

# Execute Feature file using Rest Endpoint



# Test result

Cucumber Report

Features

Tags

Steps

Trends

Failures

Tag	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
@GetenrichedData	22	0	0	0	0	22	1	0	1	0.291	Passed



Feature: [/Users/aliakkas/apps/spring/run-acceptancetest.feature](#)

Tags: [@GetenrichedData](#)

Scenario [Get enriched data](#) >

0.291

Steps >

Given path 'pdi/execute'	0.001
And params { P_TRANSFORMER: 'dateiso8601', filename: '/Users/aliakkas/apps/spring/etl/jb-acceptancetest.kjb' }	0.000
When method get	0.278
Then status 200	0.000
And match \$.pdi-result.result.nr_errors == '0'	0.001
* def result = \$.pdi-result.result.result-rows.row-data	0.000
* print result	0.000

[Doc string](#)

* def log = \$.pdi-result.result.log_text	0.000
* print log	0.000

[Doc string](#)

```
01:54:45.266 [print] 2020/01/21 01:54:45 - tr-verify-result - Using run configuration [Pentaho local]
2020/01/21 01:54:45 - tr-verify-result - Using legacy execution engine
2020/01/21 01:54:45 - tr-verify-dateiso8601 - Dispatching started for transformation [tr-verify-dateiso8601]
2020/01/21 01:54:45 - grfr-result-iso8601.0 - Finished processing (I=0, O=0, R=5, W=5, U=0, E=0)
2020/01/21 01:54:45 - wrtl-result.0 -
2020/01/21 01:54:45 - wrtl-result.0 - -----> Linenr 1-----
2020/01/21 01:54:45 - wrtl-result.0 - input = 2019-12-01
2020/01/21 01:54:45 - wrtl-result.0 - expected_result = 2019-12-01T00:00:00.000+0000
2020/01/21 01:54:45 - wrtl-result.0 - enriched_xml = <dateiniso8601 calculationFailed="true"/>
2020/01/21 01:54:45 - wrtl-result.0 - passed = N
```



# Software used

- Embedded PDI – for executing Kettle jobs or transformations
- Karate (Cucumber) – executing feature files and producing test reports - <https://github.com/intuit/karate>
- Spring Boot – for RESTful web services