

How to contribute textual tooling for Apache Camel in several IDEs

Aurélien Pupier

@apupier

Principal Software Engineer in Red Hat Integration Tooling team

ApacheCon @Home 2020

Textual tooling for Camel?

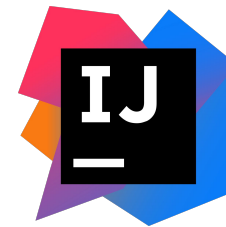
Textual tooling for Camel

- Route DSLs
 - Xml
 - Java
 - Yaml
 - (Groovy, JavaScript, Kotlin)
- Properties
 - Camel main
 - Camel Kafka connector





Several IDEs?

Existing tooling 3 years ago

- Eclipse Desktop Fuse Tooling
 - Textual and graphical tooling
 - Non-reusable code
- IntelliJ
 - Textual tooling
 - Non-reusable code
- What about other IDEs?



Providing tooling for several IDEs...

	Complexity
Eclipse Desktop IDE	
VS Code	
Emacs	
...	

For Camel, **N** implementations
with **high skill** required in **Camel AND IDE extension**

Language Server Protocol

The Promise of the Language Server Protocol

Write Code Editing capabilities once, run in all IDEs!



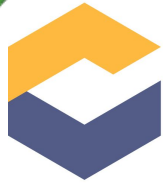
THEIA



Language Server



neovim



emacs



[More](#) (unofficial) coming...

N implementations with **low skill** for IDE

+ 1 implementations with **high skill** required in **language only**

The (real) Promise of the Language Server Protocol

Write limited set of Code Editing capabilities once, run in ~~all~~
most IDEs *with a little setup*

Language Server Protocol Code Editing Capabilities

- Autocomplete
- Hover
- Code Lens
- Goto Definition/References
- Formatting / Coloring
- Renaming
- Code actions
- Highlighting
- Folding
- Diagnostic
- Document symbols

Current state of Language Server for Apache Camel

Marketplaces clients



Language Server




















APACHE Camel




Examples/readmes



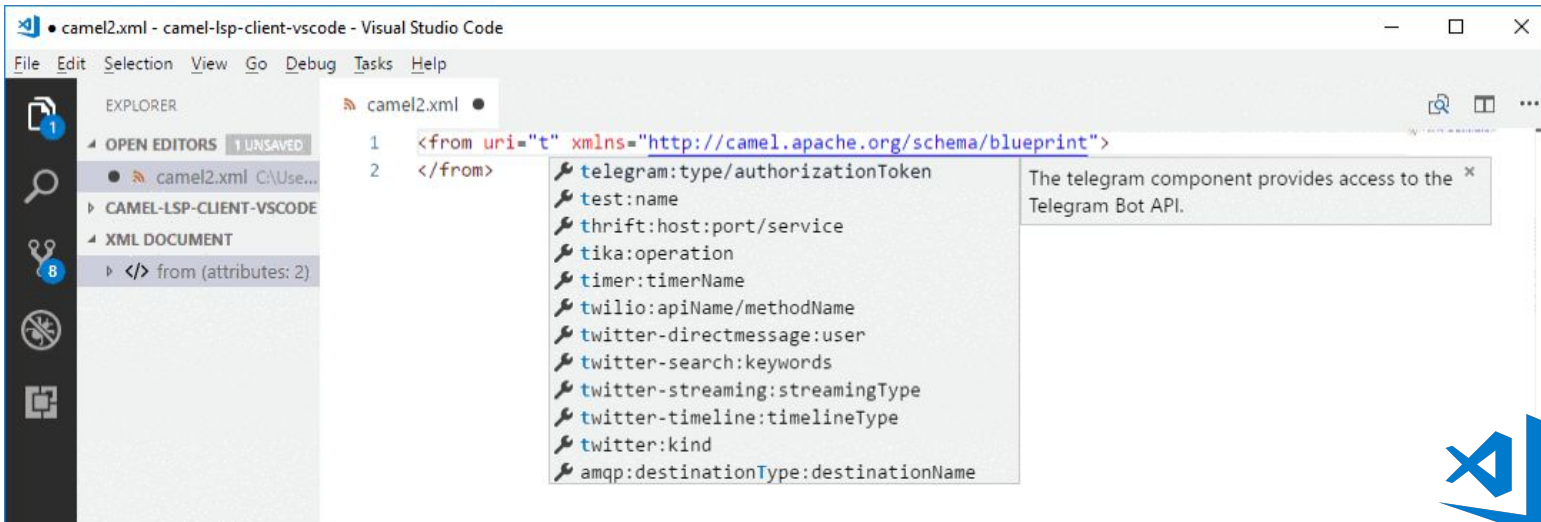
LSP Code Editing Capabilities

- Autocomplete    
- Hover     
- Code Lens
- Goto Definition/References  
- Formatting / Coloring
- Renaming
- Code actions 
- Highlighting
- Folding
- Diagnostic    
- Document symbols 

Some figures

	# of installations
Eclipse Desktop 	4500 (estimated)
VS Code 	8900
Atom 	95
Che / CodeReady Workspaces / Sublime text / Theia / Vim /CodeMirror	No metrics :-)

Demo

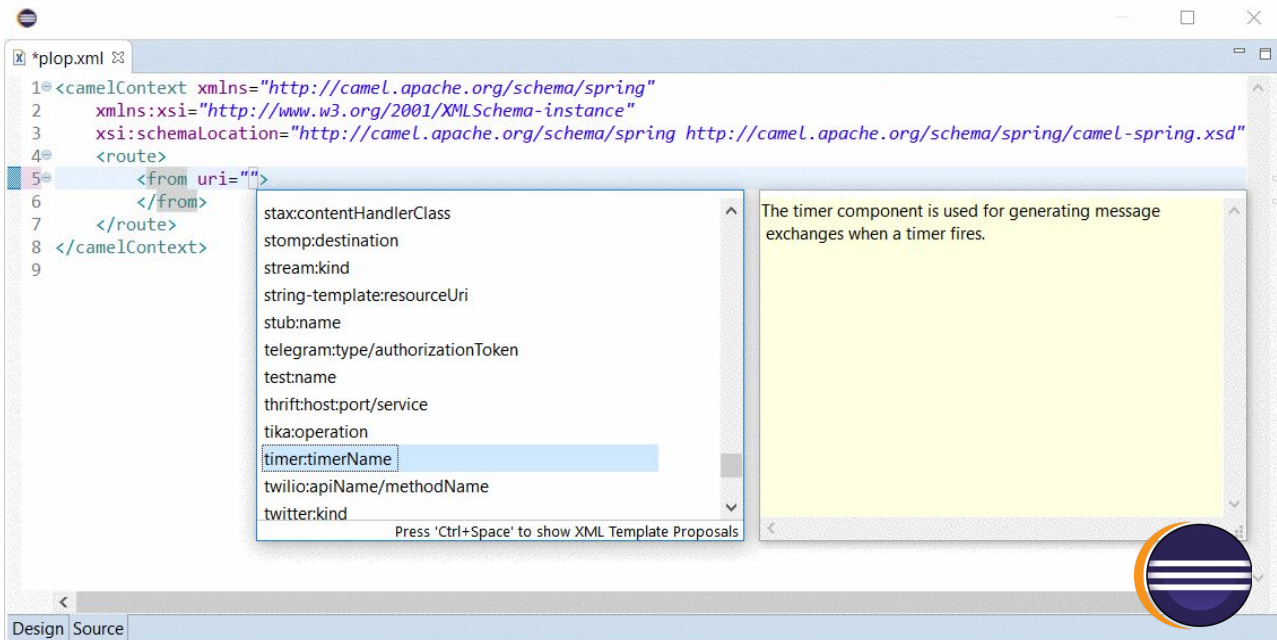


camel2.xml - camel-lsp-client-vscode - Visual Studio Code

```
1 <from uri="t" xmlns="http://camel.apache.org/schema/blueprint">
2 </from>
```

telegram:type/authorizationToken
test:name
thrift:host:port/service
tika:operation
timer:timerName
twilio:apiName/methodName
twitter-directmessage:user
twitter-search:keywords
twitter-streaming:streamingType
twitter-timeline:timelineType
twitter:kind
amqp:destinationType:destinationName

The telegram component provides access to the Telegram Bot API.

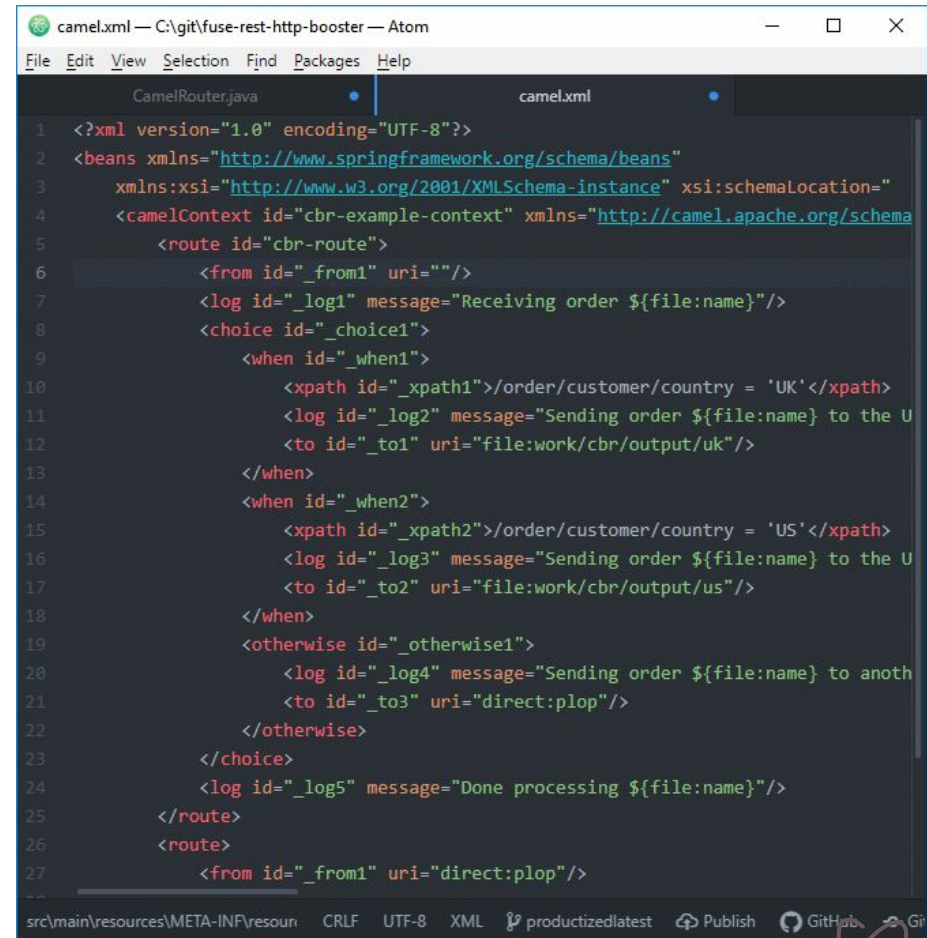


*plop.xml

```
1 <camelContext xmlns="http://camel.apache.org/schema/spring"
2 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xsi:schemaLocation="http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd"
4 <route>
5 <from uri="">
6 </from>
7 </route>
8 </camelContext>
```

stax:contentHandlerClass
stomp:destination
stream:kind
string-template:resourceUri
stub:name
telegram:type/authorizationToken
test:name
thrift:host:port/service
tika:operation
timer:timerName
twilio:apiName/methodName
twitter:kind

The timer component is used for generating message exchanges when a timer fires.



camel.xml - C:\git\fuse-rest-http-booster - Atom

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
4 http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
5 <camelContext id="cbr-example-context" xmlns="http://camel.apache.org/schema
6 http://camel.apache.org/schema/blueprint">
7 <route id="cbr-route">
8 <from id="_from1" uri=""/>
9 <log id="_log1" message="Receiving order ${file:name}"/>
10 <choice id="_choice1">
11 <when id="_when1">
12 <xpath id="_xpath1">/order/customer/country = 'UK'</xpath>
13 <log id="_log2" message="Sending order ${file:name} to the U
14 <to id="_to1" uri="file:work/cbr/output/uk"/>
15 </when>
16 <when id="_when2">
17 <xpath id="_xpath2">/order/customer/country = 'US'</xpath>
18 <log id="_log3" message="Sending order ${file:name} to the U
19 <to id="_to2" uri="file:work/cbr/output/us"/>
20 </when>
21 <otherwise id="_otherwise1">
22 <log id="_log4" message="Sending order ${file:name} to anoth
23 <to id="_to3" uri="direct:plop"/>
24 </otherwise>
25 </choice>
26 <log id="_log5" message="Done processing ${file:name}"/>
27 </route>
28 </camelContext>
29 </beans>
```

src\main\resources\META-INF\resour... CRLF UTF-8 XML productizedlatest Publish GitHub

How to contribute Code Editing capabilities

Development Environment setup

- Demo with Eclipse Desktop
 - Nice support of Language Server writing
 - Powerful Java support
- VS Code good too
- Any others IDE for Java. It will require manual configuration to test in an editor.

Environment setup - Eclipse Desktop example

<https://github.com/camel-tooling/camel-language-server/blob/master/Contributing.md>

How to contribute a new IDE client

You are on your own!

- Specific to each IDE
- One-time effort
- If I knew it,
 - it would have been done
 - there would be no more need to do it

Not completely on your own :-)

- Standalone Jar to launch
- Connection through stdio or websocket
- File associations

















Contribution ideas to Camel LSP

Ways of contributing

- Implement Code Editing capabilities
- Create the client for your preferred IDE
- Provide new feature ideas!
- Help review Pull requests
- ...






LSP Code Editing Capabilities

- Autocomplete    
- Hover     
- Code Lens
- Goto Definition/References  
- Formatting / Coloring
- Renaming
- Code actions
- Highlighting
- Folding
- Diagnostic    
- Document symbols 

Language Server feature ideas

- Renaming of direct/seda ids
- Folding of *choice* for Java
- Support of placeholder
- Autocomplete proposing some patterns
- Autocomplete in *connected mode*
- Support incremental mode
- Support Templated routes
- Provide icons for Document symbols
- Provide link to online documentation on hover
- Native version of the Language Server (Quarkus?)
- ...

Maintenance  Red Hat
CodeReady
Workspaces



[camel-tooling/camel-language-server](https://github.com/camel-tooling/camel-language-server)



APACHE Camel



Maven™



Travis CI

new features / maintenance

Create readmes
Provide specific client in Marketplaces



OPENSIFT

Your beloved IDE

Improve readmes
Provide specific client in Marketplaces



Going further...

- Cross-Language Server for a tighter integration:
 - Jdt LS (Java)
 - Lemminx (xml)
- Other similar protocols:
 - Debug Adapter Protocol
 - Graphical Language Server Protocol

Make it an Apache Camel subproject?

- Spread the idea
- Migration
- Maintenance and governance

Go enhance the Language Server for Camel!

Aurélien Pupier

Principal Software Engineer

@apupier

ApacheCon @Home 2020