Expression

Expressions

Expressions and Predicates can then be used to create the various Enterprise Integration Patterns in the DSL or Xml Configuration like the Recipient List. To support dynamic rules Camel supports pluggable Expression strategies using a variety of different Languages.

API

If you are outside of the DSL and want to create your own expressions you can either implement the Expression interface, reuse one of the other builders or try the ExpressionBuilder class.

Expression

The API for a Camel Expression is defined in the `org.apache.camel.Expression` interface as shown:

```
public interface Expression {

    /**
     * Returns the value of the expression on the given exchange
     *
     * @param exchange the message exchange on which to evaluate the expression
     * @param type the expected type of the evaluation result
     * @return the value of the expression
     */
    <T> T evaluate(Exchange exchange, Class<T> type);
}
```

Predicate

The API for a Camel Predicate is defined in the `org.apache.camel.Predicate` interface as shown:

```
public interface Predicate {

    /**
     * Evaluates the predicate on the message exchange and returns true if this
     * exchange matches the predicate
     *
     * @param exchange the message exchange
     * @return true if the predicate matches
     */
    boolean matches(Exchange exchange);

}
```

Expression Languages

The following languages are supported out of the box

- Bean Language for using Java for expressions
- Constant
- the unified EL from JSP and JSF
- Header
- JSonPath
- JXPath
- Mvel
- OGNL
- Ref Language
- ExchangeProperty / Property
- Scripting Languages such as
    - BeanShell
    - JavaScript
    - Groovy
    - Python
    - PHP

    - Ruby
- Simple
    - File Language
- Spring Expression Language
- SQL
- Tokenizer
- XPath
- XQuery
- VTD-XML

Most of these languages is also supported used as Annotation Based Expression Language.

Using Expressions in your IDE

To use different expression and predicates in your IDE you need to perform a static import of the builder class for the language(s) you wish to use.

Language(s)	Builder class to import
Scripting Languages such as BeanShell, JavaScript, Groovy, PHP, Python and Ruby	org.apache.camel.builder.script.ScriptBuilder
SQL	org.apache.camel.builder.josql.SqlBuilder
XPath	org.apache.camel.builder.xml.XPathBuilder
XQuery	org.apache.camel.builder.saxon.XQueryBuilder

See Also

- Predicate