# Selectors (2.1 legacy document)

## Table of contents

# Table of contents

**Warning:**
This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

## 1. Goal

This document lists all of the available selectors of Apache Cocoon and describes their purpose. You may also wish to read Using and Implementing Matchers and Selectors.

## 2. Overview

Selectors in Apache Cocoon have a role similar to matchers with additional flexibility. If you haven't learned about matchers yet, read about them here before continuing. Selectors are designed to evaluate a generally simple boolean expression regarding some part of the environment (request URI, headers, or cookies, for example). The result of this evaluation determines which pipeline fragments should be combined within a given pipeline. Unlike matchers, selectors can be active decision-driving components. For example, a matcher makes only simple "yes/no" decisions. If a match is successful, a pipeline is executed. If not, it is ignored. Selectors go further by allowing more complex, multiple-choice use cases. In short, consider matchers to be simple "if" statements. By extension, consider selectors to have all the power of an "if-else if-else" or "switch-case" constructs. The selector syntax should be familiar to anyone who uses XSLT's <xsl:choose> statement.

As an example, consider the typical scenario in which a page should be rendered differently based on the client browser. Given the large number and diversity of available browsers, it would be awkward and counterintuitive to address this need with a set of matchers. The BrowserSelector tests a given parameter against the user-agent request header. Using this single selector, we can deploy a consistent and readable setup.

```
<map:match pattern="docs/*.html"> <map:generate src="xdocs/{1}.xml"/> <map:select type="browser"> <map:when test="netscape"> <map:transform src="stylesheets/netscape.xsl" /> </map:when> <map:when test="explorer"> <map:transform src="stylesheets/ie.xsl" /> </map:when> <map:when test="lynx"> <map:transform src="stylesheets/text-based.xsl" /> </map:when> <map:otherwise> <map:transform src="stylesheets/html.xsl" /> </map:otherwise> </map:select> <map:serialize/> </map:match>
```

## 3. The Selectors in Cocoon

Available Selectors in Cocoon include the following:

- **BrowserSelector**: matches the value of the "test" parameter against the HTTP User-Agent header, allowing it to recognize the browser issuing the request;
- **CodeSelector**: matches a snippet of Java code given as the "test" parameter against the environment;
- **HostSelector**: matches the "test" parameter value against the Host request header
- ParameterSelector: matches the string specified in the "test" parameter against a specified Cocoon internal (e.g. sitemap) parameter;
- **HeaderSelector**: same as the Parameter selector, but matches against the request headers;
- RegexpHeaderSelector: same as the Header selector, but uses a regular expression for matching;
- **RequestSelector**: again, same as the Parameter selector, but matches against the Request parameters;
- **SessionSelector**: finally, this selector is used as the Parameter selector to match against an

arbitrary session attribute;

## 1. Comments

add your comments