

# Stream Generator (2.1 legacy document)

## Table of contents

|                 |   |
|-----------------|---|
| 1 Comments..... | 4 |
|-----------------|---|

## Table of contents

|                           |   |
|---------------------------|---|
| 1 Stream Generator.....   | 3 |
| 1.1 PostInputStream.....  | 3 |
| 1.2 See it in Action..... | 3 |

**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. Stream Generator

The StreamGenerator is a class that reads XML from an HttpRequest InputStream and generates SAX Events. StreamGenerator expects XML data coming as HTTP request message.

- Name : stream
- Class: org.apache.cocoon.generation.StreamGenerator
- Cacheable: no.

For POST requests with mimetype of application/x-www-form-urlencoded, the xml data expects to be associated with the name specified in the sitemap parameter.

For HTTP requests with mimetypes: text/plain, text/xml, application/xml the xml data is in the body of the HTTP request and its length is specified by the value returned by getLength() method.

### 1.1. PostInputStream

The StreamGenerator uses helper class org.apache.cocoon.util.PostInputStream for InputStream reading operations. At the time that Parser reads the data out of InputStream - Parser has no knowledge about the length of data to be read. The only way to signal to the Parser that all data was read from the InputStream is to control reading operation - PostInputStream- and to return to the requestor -1 when the number of bytes read is equal to the getLength() value.

### 1.2. See it in Action

The Generator is a generic object, i.e. it can process any stream out of the HTTP message. There are two ways to see StreamGenerator in action:

- To invoke URL <http://localhost:8080/cocoon/Order>
- To use telnet program to generate POST request

The first option is not a "pure" stream invocation, but it is quick way to observe desired effects. The result of this invocation is a form containing the XML document embedded in the textarea of the form. Submission of this form will invoke StreamGenerator. The textarea name/value pair is specified as a parameter in the sitemap definition for the StreamGenerator. The expected result is the submitted xml document send back to the browser.

The second or "pure" option of testing StreamGenerator "in action," requires the use of Telnet program or any other process able to generate correct HTTP message. The procedure is:

- To invoke telnet, connect to localhost 8080 and to use content of [telnet.txt](#) file as a post message.
- Here, the Copy-Paste method should be used.
- Remember to hit the enter button twice enter after the contents of the post are set in telnet.

It is important because Content-len is calculated assuming two "enter" in the end of http message. Once again, the performed task results in the mirror of the original document being sent back to the requestor.

The "pure" stream generation can be observed using the telnet utility where you can invoke a message targeting my processing. Any other method is good (URL object connection) as long the message is well formed.

```
<map:generate type="stream"/>
```

If you want to process XML streams sent by clients that don't set the Content-Type HTTP header just use the defaultContentType parameter.

```
<map:generate type="stream"> <map:parameter name="defaultContentType"
value="text/xml"/> </map:generate>
```

## 1. Comments

add your comments