

# Read DOM Session Transformer (2.1 legacy document)

## Table of contents

|                 |   |
|-----------------|---|
| 1 Comments..... | 3 |
|-----------------|---|

## Table of contents

|                                     |   |
|-------------------------------------|---|
| 1 Read DOM Session Transformer..... | 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. Read DOM Session Transformer

With this transformer, a DOM-object that is stored in the session, can be inserted in the SAX stream at a given position.

- Name : readDOMsession
- Class: org.apache.cocoon.transformation.ReadDOMSessionTransformer
- Cacheable: no.

Simply transforms a DOM to SAX-events, which can be used further on in the pipeline. Once you stored the result of a query in the session with the WriteDOMSessionTransformer, you can read it again with the ReadDOMSessionTransformer:

```
<map:transform type="readDOMsession"> <map:parameter name="dom-name"
value="DBresult"/> <map:parameter name="trigger-element" value="users"/>
<map:parameter name="position" value="after"/> </map:transform>
```

In this example, the SAX-events that came from the DOM tree that is stored in the session with name DBresult will be added after the users element. This means as soon that the transformer encounters the end-element 'users', it will start to generate SAX-events from the DOM tree. There are three possible positions, 'before', 'in' and 'after':

1. 'before' means that when the transformer encounters the 'users' element, it will FIRST translate the DOM tree to SAX-events and THEN it will continue to forward the other SAX-events (starting with 'users').
2. 'in' means that the transformer will forward the startElement event for 'users' and that it IMMEDIATELY starts to generate SAX-events from the DOM-tree. After that, it will continue to forward the child elements of users and then all the other elements.
3. 'after' means that the transformer starts to generate SAX-events from the DOM-tree just after it has forwarded the end-element 'users'.

The ReadDOMSessionTransformer is a standalone component, you don't need to use it in combination with the WriteDOMSessionTransformer.

## 1. Comments

add your comments