

Welcome to xml-ezparse

1. Simple and Clean XML Parsing

xml-ezparse is a simple and hence easy to use API built upon the SAX API. In a nutshell, listeners are registered for XML elements. Each listener handles only the element it is registered for thereby eliminating unsightly if statements.

Reasons why you may consider using this:

1. Simple. Start using it within 5 minutes.
2. Uses SAX so it has a smaller memory footprint.
3. No reflection.

2. How it works.

Exact Pattern Matching

It works like event processing in AWT/Swing. A listener is defined for the pattern `"/states/state"`. The listener's `startElement()` method will be invoked when the element is first encountered. When the closing tag is found the listener's `endElement()` method is called. Another listener may handle the pattern `"/states/state/capital"`. Since each Listener manages only one pattern you don't need messy if statements. Please also see the code example. Note that there is no trailing slash character in the X-Path like pattern.

Regular Expression Support

`src/test/SimpleParserTestRegEx.java` tests the Regular Expression capabilities. For example you could add an `ElementListener` for the `"/states/state/capital"` element using the following path `"/state/state/c*"`.

The parser will try to find an exact match on the path to pick the `ElementListener` to call. If there are no exact matches, regular expressions are used to see if any listeners have been registered using regular expressions for the path string. Finally, if no matches are found the default listener is used which does nothing. A log message is generated when no match is found.

3. Other choices

There are other XML parsing utilities such as Apache's Digester, which is what inspired me

to write xml-ezparse. The major drawback (IMHO) with Digester is that it uses reflection which adds complexity. I also thought the API could be simpler, so voila.