



PAX

Power API for XML

von Manfred Duchrow



PAX

PAX

- DOM und SAX
- PAX : Ideen und Konzepte
- Wie PAX funktioniert
- Wie PAX angewandt wird
- WWW-Links

April 2000

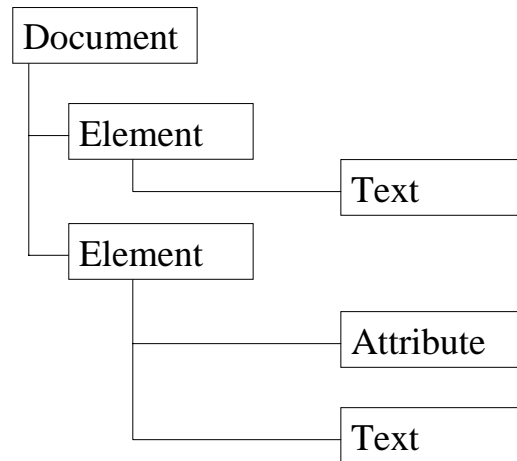
2

Copyright © 2000 by Daedalus Consulting GmbH



DOM (Document Object Model)

PAX



April 2000

3

Copyright © 2000 by Daedalos Consulting GmbH



DOM: Pro und Contra

PAX

Pro

- Sehr leicht zu verwenden
- Sehr gut um XML-Strukturen zu erzeugen

Contra

- Der Parser liest immer das gesamte XML-Dokument ein
- Das Ergebnis des eingelesenen XML-Dokuments ist ein Baum aus generischen Objekten und kein Netz aus Geschäftsobjekten

April 2000

4

Copyright © 2000 by Daedalos Consulting GmbH



SAX (Simple API for XML)

PAX

XML Dokument

```
<?xml version="1.0"?>
...
<Title>
  Design Patterns
</Title>
...
```

SAX-Parser

DocumentHandler

```
startDocument()
startElement()
characters()
endElement()
endDocument()
```

April 2000

5

Copyright © 2000 by Daedalos Consulting GmbH



SAX: Pro und Contra

PAX

Pro

- Sehr einfach
- Arbeitet mit Ereignissen, was die Extraktion von Teilen eines XML-Dokumentes erleichtert

Contra

- Überlässt dem Programmierer die meiste Arbeit (DocumentHandler - Implementierung)
- Die Erzeugung von XML-Dokumenten wird nicht unterstützt.

April 2000

6

Copyright © 2000 by Daedalos Consulting GmbH



PAX

PAX

- Baut auf SAX auf, d.h. es läuft mit jeder SAX-Parser Implementierung
- Unterstützt die direkte Erzeugung von Geschäftsobjekten aus XML-Daten
- Ist eher ein Framework als eine Schnittstelle
- Reduziert den Programmieraufwand
- Führt zu besser wartbarem Code

April 2000

7

Copyright © 2000 by Daedalos Consulting GmbH



Die Grundidee von PAX

PAX

- Jedes XML-Tag soll von einer eigenen Klasse ausgewertet werden
- Unterstützung direkter Erzeugung von Geschäftsobjekten aus XML-Daten
- Berücksichtigung der hierarchischen Strukturen von XML-Dokumenten
- Reduzierung des Programmieraufwandes
- Es sollte immer noch leicht verständlich sein

April 2000

8

Copyright © 2000 by Daedalos Consulting GmbH



PAX - Komponenten

PAX

- Tag interpreter
abstract class BaseXMLTagInterpreter
- Tag interpreter controller
abstract class BaseXMLTagInterpreterController
- Tag interpreter factory
abstract class BaseXMLTagInterpreterFactory

April 2000

9

Copyright © 2000 by Daedalos Consulting GmbH



Randbedingungen von SAX

PAX

- Ein SAX-Parser kennt immer nur eine Instanz eines DocumentHandlers
- Der dem SAX-Parser bekannte DocumentHandler kann während des Parsens eines XML-Dokuments ausgetauscht werden

April 2000

10

Copyright © 2000 by Daedalos Consulting GmbH



PAX - Funktionsweise

PAX

- Zuerst wird ein XMLTagInterpreterController als DocumentHandler dem SAX-Parser zugewiesen
- Dann setzt dieser Controller den jeweils einem beim Parsen auftretenden XML-Tag zugeordneten XMLTagInterpreter als DocumentHandler ein
- Eine XMLTagInterpreterFactory liefert die XMLTagInterpreter Instanzen zu den XML-Tags
- Der Controller verwaltet den Stack der Tag-Interpreter
- Das erzeugte Geschäftsobjekt eines Tag-Interpreters wird an den übergeordneten Tag-Interpreter weitergereicht, um dort in das entstehende Objekt eingefügt zu werden

April 2000

11

Copyright © 2000 by Daedalus Consulting GmbH



Schritt für Schritt

PAX

- Es ist sinnvoll eine Klasse anzulegen, die alle Konstanten (Tag-Namen, Attribut-Namen etc.) enthält
- Für jedes vorkommende Tag eine Unterklasse von *BaseXMLTagInterpreter* anlegen
- Eine Unterklasse von *BaseXMLTagInterpreterFactory* erstellen, die für jedes XML-Tag den entsprechenden Tag-Interpreter liefert
- Eine Unterklasse von *BaseXMLTagInterpreterController* anlegen

April 2000

12

Copyright © 2000 by Daedalus Consulting GmbH



Funktionsweise - *XMLTagInterpreter*

PAX

Jede Unterklasse von *BaseXMLTagInterpreter* muss die folgenden Methoden implementieren:

- **public String getTagName()**
Liefert den Namen des Tags für das der Interpreter zuständig ist
- **public void start(String elementName, AttributeList attributes)**
Wird aufgerufen, wenn das zugeh. Tag gefunden wurde und der Interpreter die Auswertung der eingelesenen Daten beginnen soll
- **public void restart(String subTagName, Object subResult)**
Wird aufgerufen, wenn ein Interpreter wieder die Kontrolle übernimmt nachdem ein untergeordnetes Tag ausgewertet wurde
- **public Object getResult()**
Liefert das aus den XML-Daten erzeugte Objekt

April 2000

13

Copyright © 2000 by Daedalus Consulting GmbH



Funktionsweise - *XMLTagInterpreterFactory*

PAX

Eine Unterklasse *BaseXMLTagInterpreterFactory* sollte alle verfügbaren Tag-Interpreter-Klassen registrieren:

```
public class MyFactory extends
    BaseXMLTagInterpreterFactory {
public MyFactory() {
    super() ;
    registerTagInterpreter( "Booklist",
        "com.libri.xml.BooklistInterpreter");
    registerTagInterpreter( "Book",
        "com.libri.xml.BookInterpreter");
    registerTagInterpreter( "Title",
        "com.libri.xml.TitleInterpreter");
    }
}
```

April 2000

14

Copyright © 2000 by Daedalus Consulting GmbH



Funktionsweise - *XMLTagInterpreterController*

PAX

Eine Unterklasse von *BaseXMLTagInterpreterController* muss die folgenden Methoden implementieren:

- **protected void handleFinalResult(Object result)**
Diese methode wird aufgerufen, nachdem das letzte Ende-Tag eines XML-Dokumentes interpretiert wurde. Sie erhält das Ergebnis-Objekt des letzten Tag-Interpreters.
- **protected XMLTagInterpreterFactory
createTagInterpreterFactory()**
Wird vom Constructor aufgerufen, um die Variable *interpreterFactory* zu initialisieren.

April 2000

15

Copyright © 2000 by Daedalus Consulting GmbH



Ausführung

PAX

Beispiel:

```
...
parser = new SAXParser() ;
controller = new MyPaxController( parser ) ;
parser.parse( filename ) ;
result = controller.getResult() ;
...
```

April 2000

16

Copyright © 2000 by Daedalus Consulting GmbH



Links

PAX

- Daedalos Download Seite für PAX
<http://www.daedalos.de/DE/DOWNLOAD/pax/>
- W3C - DOM Specification
<http://www.w3.org/TR/REC-DOM-Level-1/>
<http://www.w3.org/TR/DOM-Level-2/>
- SAX Specification
<http://www.megginson.com/SAX/sax.html>
<http://www.megginson.com/SAX/SAX2/>

April 2000

17

Copyright © 2000 by Daedalos Consulting GmbH



Kontakt

PAX

Daedalos Consulting GmbH
Ruhrtal 5
D-58456 Witten
Germany

Tel.: +49 (0)2302 979-0
Fax: +49 (0)2302 979-199
E-Mail: info@daedalos.com

Web: <http://www.daedalos.com>

April 2000

18

Copyright © 2000 by Daedalos Consulting GmbH