



Simple API for XML (SAX)

XML-Verarbeitung mit Java

Inhalte dieses Kapitels

Allgemeines

Content Handler

Error Handler

Weitere Komponenten

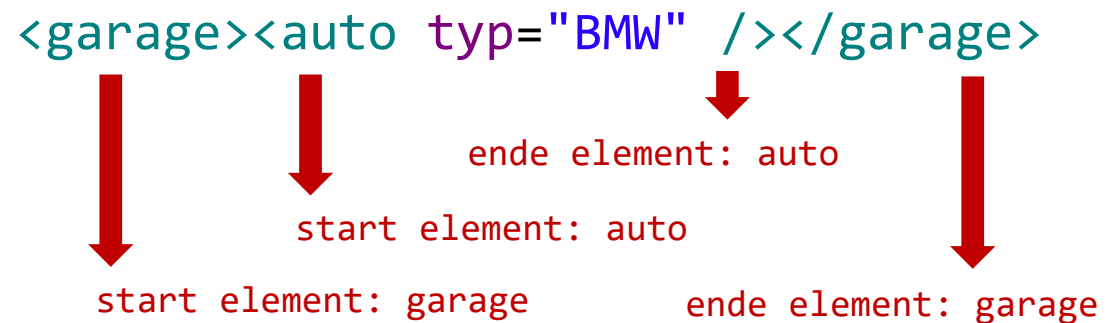
Verwendung der Komponenten

Filter

SAX Features

Allgemeines [1|3]

- Parsen mit ereignisbasierter Verarbeitung
 - Lesen des Dokuments einmalig von Anfang bis Ende
 - Java-Anwendung registriert *Callback Handler* und startet Parser
 - Benachrichtigung der Callback Handler durch SAX Parser für
 - öffnende / schließende Tags
 - Attribute
 - Texte
 - Namespace-Importe
 - ...



- Konzept unabhängig von der Programmiersprache
- Entwicklung vorangetrieben durch Diskussionsgruppe XML-DEV

Allgemeines [2|3]

Bestandteile

XMLReader	SAX-Parser
XMLReaderFactory	Factory zum Erzeugen einer SAX-Parser-Instanz
ContentHandler	Callback Handler zur Ereignis-Behandlung
Locator	Objekt mit Informationen zur Position im Dokument
ErrorHandler	Objekt zur Behandlung von Fehler-Ereignissen
DTDHandler	Callback Handler zur Ereignis-Behandlung (DTD)
EntityResolver	Objekt zum Auflösen von XML-Entities (&...)
Attributes	Liste der XML-Attribute
XMLFilter	Filtern von Elementen

Allgemeines [3|3]

Beispiel

```
// Parser erzeugen
```

```
SAXParserFactory factory = SAXParserFactory.newInstance();
```

```
factory.setNamespaceAware(true);
```

```
SAXParser parser = factory.newSAXParser();
```

```
// ErrorHandler u.a. nur über XMLReader
```

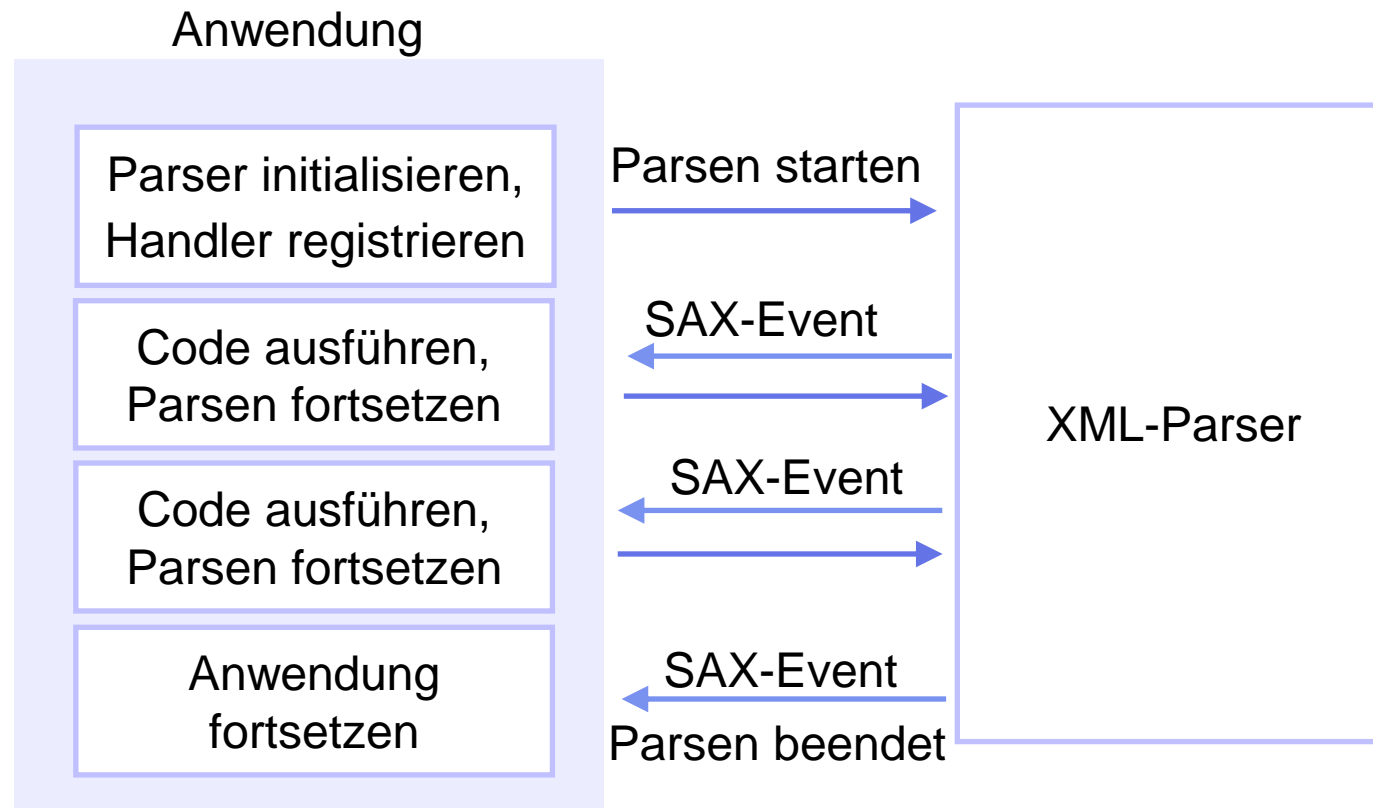
```
parser.getXMLReader().setErrorHandler(new MyErrorHandler());
```

```
// Parsen
```

```
try (InputStream in = ...) {  
    parser.parse(in, new MyContentHandler());  
}
```

Content Handler [1|6]

- Callback-Methoden für verschiedene Ereignisse
- Aufruf der Methoden beim Auftreten eines Ereignisses durch Parser



Content Handler [2|6]

Erstellung – Mindestanforderung

- Interface `org.xml.sax.ContentHandler` implementieren
- Implementierung aller Callback-Methoden (notfalls leer)

Erstellung – Empfehlung

- Unterklasse von `org.xml.sax.ext.DefaultHandler2` erstellen
- Callback-Methoden bei Bedarf überschreiben



Content Handler [3|6]

Beispiel

<garage><auto typ="BMW" /></garage>

```
public class MyContentHandler extends DefaultHandler2 {

    private Auto auto; // zu sammelnde Daten

    public Auto getAuto() { // zum Abfragen nach dem Parsen
        return auto;
    }

    @Override
    public void startElement(String uri, String localName, String qName,
        Attributes attributes) throws SAXException {
        if ("auto".equals(localName)) {
            auto = new Auto();
            auto.setTyp(attributes.getValue("typ"));
        }
    }
}
```


Content Handler [4|6]

Beispiel: Hierarchie

```
<garage><auto typ="BMW" /></garage>
```

```
private boolean garage; // Hierarchie beachten!
```

```
@Override
```

```
public void startElement(String uri, String localName, String qName, Attributes attributes)
```

```
    throws SAXException {
```

```
    switch (localName) {
```

```
        case "garage": garage = true;          break; // ab jetzt innerhalb von <garage>!
```

```
        case "auto"   : if(garage) { /*...*/ } break;
```

```
    }
```

```
}
```

```
@Override
```

```
public void endElement(String uri, String localName, String qName) throws SAXException {
```

```
    if ("garage".equals(localName)) {garage = false; } // ab jetzt außerhalb von <garage>!
```

```
}
```

Content Handler [5|6]

Beispiel: Namespace

```
<garage xmlns="http://www.ars.de/garage">  
  <auto typ="BMW" />  
</garage>
```

```
private static final String NS = "http://www.ars.de/garage";  
  
@Override  
public void startElement(String uri, String localName, String qName, Attributes attributes)  
    throws SAXException {  
    if (NS.equals(uri) && "auto".equals(localName)) {  
        auto = new Auto();  
        auto.setTyp(attributes.getValue(NS, "typ"));  
    }  
}
```

Content Handler [6|6]

Beispiel: Locator

```
private Locator locator;

@Override
public void setDocumentLocator(Locator locator) {
    this.locator = locator;
}

@Override
public void startElement(String uri, String localName, String qName, Attributes attributes)
    throws SAXException {
    int column = locator.getColumnNumber();
    int row = locator.getLineNumber();
    // ...
}
```

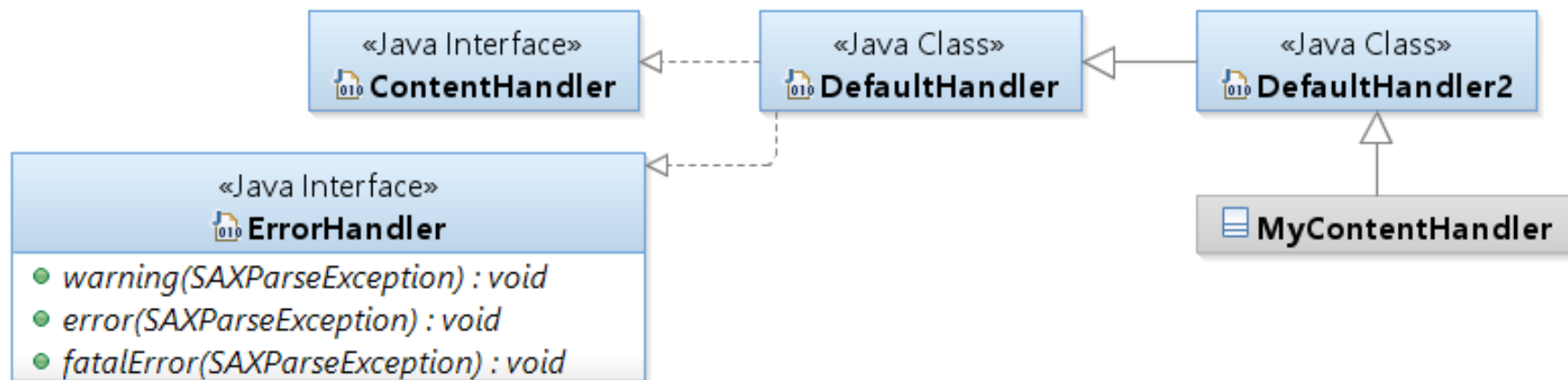
Error Handler [1|4]

Erstellung – Mindestanforderung

- Interface `org.xml.sax.ErrorHandler` implementieren
- Implementierung aller Callback-Methoden (notfalls leer)

1 Implementierungsklasse für Content- und ErrorHandler

- Klasse `org.xml.sax.helper.DefaultHandler` implementiert auch ErrorHandler-Interface
- Callback-Methoden bei Bedarf überschreiben



Error Handler [2|4]

Beispiel: Separate Implementierungsklasse

```
public class MyErrorHandler implements ErrorHandler {  
  
    @Override  
    public void warning(SAXParseException exception) throws SAXException { /*...*/ }  
  
    @Override  
    public void error(SAXParseException exception) throws SAXException { /*...*/ }  
  
    @Override  
    public void fatalError(SAXParseException exception) throws SAXException { /*...*/ }  
  
}
```

Error Handler [3|4]

Beispiel: Gemeinsame Implementierungsklasse

```
public class MyErrorHandler extends DefaultHandler2 {  
  
    @Override  
    public void startElement(String uri, String localName, String qName, Attributes attributes)  
        throws SAXException { /*...*/ }  
  
    @Override  
    public void fatalError(SAXParseException exception) throws SAXException { /*...*/ }  
  
}
```

Error Handler [4|4]

Arten von Fehlern

- Einteilung nach W3C XML 1.0 Recommendation (<http://www.w3.org/TR/REC-xml/#sec-terminology>)
- `public void warning(SAXParseException exception)`
 - Warnung vom XML-Parser/Validierer
 - Beispiel: Schema nicht gefunden
- `public void error(SAXParseException exception)`
 - Fehler bei Validierung nach Schema
 - Kleinere Syntax-Fehler
- `public void fatalError(SAXParseException exception)`
 - Fehler, die zum Abbruch des Parsens führen
- Callback-Methoden können `SAXException` werfen, um das Parsen selbst zu beenden

Weitere Komponenten

Entity Resolver

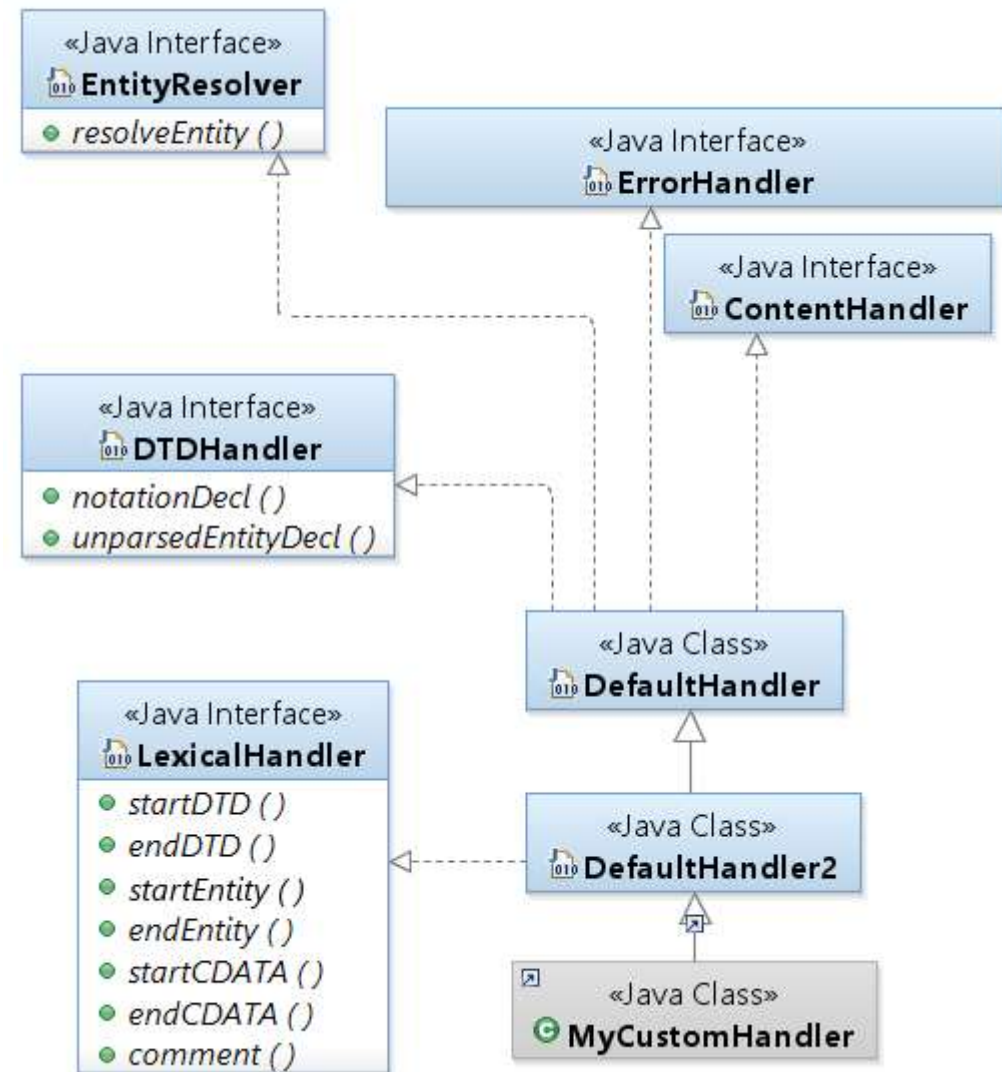
- Auflösen von Entity-Referenzen
- Interface `org.xml.sax.EntityResolver`

DTD-Handler

- Ereignisse beim DTD Parsen
- Interface `org.xml.sax.DTDHandler`

Lexical Handler

- erst seit SAX 2.0
- Kommentare
- DTD-Deklarationen und Entities
- CDATA
- Interface `org.xml.sax.ext.LexicalHandler`



Verwendung der API [1|2]

Beispiel: Separate Implementierungsklassen

```
DefaultHandler contentHandler = new MyCustomHandler();  
SAXParserFactory factory = SAXParserFactory.newInstance();  
SAXParser parser = factory.newSAXParser();  
parser.getXMLReader().setErrorHandler(...);  
parser.getXMLReader().setEntityResolver(...);  
parser.getXMLReader().setDTDHandler(...);  
parser.getXMLReader().setProperty(  
    "http://xml.org/sax/properties/lexical-handler", ...);  
parser.parse(in, handler);
```

Verwendung der API [2|2]

Beispiel: eine Implementierungsklasse

```
DefaultHandler handler = new MyCustomHandler();
SAXParserFactory factory = SAXParserFactory.newInstance();
SAXParser parser = factory.newSAXParser();
// Redundant
parser.getXMLReader().setErrorHandler(handler);
parser.getXMLReader().setEntityResolver(handler);
parser.getXMLReader().setDTDHandler(handler);
parser.getXMLReader().setProperty(
    "http://xml.org/sax/properties/lexical-handler", ...);
parser.parse(in, handler);
```

Filter [1|3]

Konzept

- Implementierung von *Cross-Cutting Concerns*
- Verkettung mehrerer Filter möglich

Erstellung

- Implementierung des Interface `org.xml.sax.XMLFilter`
 - Wrapper für `XMLReader`
 - Oberklasse `org.xml.sax.helpers.XMLFilterImpl`
 - Überschreiben der Methoden, wenn notwendig

Filter [2|3]

Beispiel: Implementierung

```
public class MyCustomFilter extends XMLFilterImpl {  
    @Override  
    public void startElement(String uri, String localName, String qName,  
        Attributes attributes) throws SAXException {  
        AttributesImpl attr = new AttributesImpl(attributes);  
        if (localName.equals("car") && null == attr.getValue("typ")) {  
            attr.addAttribute(uri, "typ", "", "CDATA", "BWM");  
        }  
        // Delegation an Original-XML-Reader  
        super.startElement(uri, localName, qName, attr);  
    }  
}
```

Filter [3|3]

Beispiel: Verwendung

```
XMLReader reader = XMLReaderFactory.createXMLReader();
XMLFilter filter = new MyCustomFilter();
filter.setParent(reader);
// Eigene Verarbeitung konfigurieren
filter.setContentHandler(new MyCustomHandler());
filter.setErrorHandler(new MyErrorHandler());
// Parsen
try (InputStream in = ...) {
    filter.parse(new InputSource(in));
}
```

SAX Features

- Konfiguration der Einstellungen des Parsers
- Unterstützung nicht in allen Parsern
- Präfix <http://xml.org/sax/features/>
- Aktivieren der Einstellungen über Methode `XMLReader#setFeature()`

Beispiel: Validierung aktivieren

```
XMLReader parser = XMLReaderFactory.createXMLReader();  
parser.setFeature("http://xml.org/sax/features/validation", true);
```