

XML, FI und JSON strukturiert in Java verwenden

Wolfgang Nast
MT AG
Ratingen

Schlüsselworte

XML FI FastInfoset JSON Java Java EE

Einleitung

Es wird vorgestellt, wie die verbreiteten, strukturierten Formate XML, FastInfoset(FI) und JSON mit Java genutzt werden können. XML und FI nutzen hierbei die gleiche Basis, da FI die binäre Darstellung und Übertragung von XML ist. Bei XML können auch Schema hinterlegt werden, die die Datenstrukturen beschreiben. Das JSON Format verbreitet sich stark, weil es einfach strukturiert ist und von JavaScript hervorragend unterstützt wird. Damit ist es das bevorzugte Format um Daten zur Weboberfläche zu senden und auch zurück an den Server.

XML Daten klassisch

Der bekannteste Parser ist SAX, auch JAXP genannt. Hier werden die Daten als Events entgegen genommen. Hier lässt sich leicht nach bestimmten Elementen suchen, wobei man nicht benötigte Strukturen ignorieren kann. Hier ist die Aufgabe passende Handler zu bauen. Leider gibt es keinen direkten Weg die Daten zu Schreiben. Mit XSLT ist das möglich, die Datentransformer werden hier nicht beschrieben.

Beim Lesen mit STAX können die Daten gegen ein vorgegebenes Schema validiert werden.

Zum Lesen und Schreiben von XMLDaten eignet sich das Document Object Model (DOM) hier werden die XMLDaten als ein komplettes Dokument geladen. Das Dokument kann dann durchsucht und geändert werden. Zum Schluss kann das Dokument in Form von XMLDaten abgelegt werden. Hier können die Daten mit den typischen Operationen, die in der Webprogrammierung üblich sind, gefunden werden. Dabei sind das Suchen nach einem Key und das Navigieren über die Kind Elemente und das Abfragen von Attributen im Vordergrund. Das Anlegen und Kopieren von Elementen ist möglich, aber meist recht umständlich.

Mit StAX können die Daten ähnlich wie mit SAX gelesen werden. Nur ist es hier möglich sich aktiv zu den nächsten Strukturen sich zu bewegen. Für das Schreiben der Daten wurde eine äquivalente Abbildung gefunden. Diese Struktur macht es angenehm die Daten zu schreiben. Bei StAX gibt es zwei Ansätze die Daten zu behandeln. Mit XmlEvent[Reader/Writer] geht es recht Objektorientiert. Mit XmlStream[Reader/Writer] geht es mehr Zugiffsorientiert. Zugriffsorientiert heißt hier, sind noch Daten da. Von welchem Typ sind die Daten. Nach Typ können die Daten dann abgeholt werden.

XML Daten gemapped

Mit JAXB können die Daten nach Strukturvorgabe gelesen werden. Die Struktur wird meist als XMLSchema vorgegeben. Das Schema kann verwendet werden um Javacode zum Lesen und Schreiben zu generieren. Es ist auch möglich bestehenden Javacode mit Annotationen zu versehen.

Aus den vorgegebenen Javastrukturen entsteht dann ein XML Schema. Dieses lässt sich auch generieren. Der Vorteil von den gemappten Strukturen ist, dass sie sehr leicht ausgelesen und befüllt werden können. JAXB benötigt im Normalfall eine der anderen Technologien um die Daten zu lesen oder zu schreiben. Beim direkten Zugriff von JAXB wird intern eine der anderen Technologien verwendet.

FastInfoset FI

FastInfoset wird recht häufig intern in der Übertragung und Speicherung von XML-Daten verwendet. Überwiegend da wo die Daten nicht menschlich lesbar sein sollen, wie in der DB oder der Übertragung von Binärschnittstellen.

FastInfoset wird nicht direkt vom JDK unterstützt. Hier ist das Java FastInfoset Projekt sehr praktisch. Bitte nicht die im JDK als Unterpackage Intern mit den gleichen Klassen verwenden. Diese werden ab der Java Version 9 nicht mehr ansprechbar sein. In dem Projekt ist die StAX Umsetzung verwendet worden. Für FI ist SAX, DOM und XSLT dort nicht umgesetzt. StAX reicht schon für ein einfaches Lesen und Schreiben der Daten.

JAXB kann hier über die StAX Ein- und Ausgabe direkt verwendet werden. Es sind keinerlei Anpassungen am XML Schema oder Javacode notwendig.

JSON

Hier gibt es eine erste Umsetzung, die mit Java EE 7 verwendet werden kann. Das Projekt kann auch als Jar in Java verwendet werden. Hier sind zwei Anwendungswege möglich. Einmal kann auf die JSON-Daten wie bei StAX zugegriffen werden und alternativ kann das gesamte Dokument ähnlich wie bei DOM in den Speicher geladen werden. Das Dokument lässt sich auch mit Java zusammen bauen und dann direkt speichern.

Zur Zeit gibt es noch ein paar erwähnenswerte Projekte wie JSONB, womit JSON ähnlich wie bei JAXB gemapped wird. Auch wird versucht eine allgemeingültige Umsetzung von JSON für Javaobjekt zu entwickeln. Die Idee ist es den ObjectStream durch einen JSONStream zu ersetzen und so ganze Javaobjekte mit JSON zu übertragen, ohne dass hierfür Codeanpassungen wie Annotationen zu verwenden.

Zusammenfassung

Es gibt viele Möglichkeiten strukturierte Daten in Java zu verarbeiten und zu übertragen. Als binäre Übertragung und Speicherung ist FastInfoset von Vorteil gegenüber den XML-Daten. Bei vorgegebenen Strukturen wie XML Schema ist JAXB vorteilhaft. Damit wird eine gute Abbildung der Strukturen in Javaklassen vorgenommen. Bei sich rasch ändernden Schnittstellen ist JAXB viel zu starr und eine flexible Struktur wie bei JSON ist gefragt. Hier wird JSON in Zukunft eine bedeutende Rolle zu kommen.

Kontaktadresse:

Wolfgang Nast

MT AG

Balcke-Dürr-Allee 9
D-40882 Ratingen

Telefon: +49 (0) 2102 30961-0
Fax: +49 (0) 2102 30961- 101
E-Mail Wolfgang.Nast@mt-ag.com
Internet: www.mt-ag.com