

Übung 1: Einfaches Transformationszenario mit Oxygen

Daten speichern

- Lege einen leeren Ordner an und benenne ihn in „XSLT-Übung1“ um.
- Lade die Datei „Übung1_booklist.xml“ herunter.
- Speichere die Datei in den Ordner „XSLT-Übung1“.

XSLT Datei anlegen

- Öffne den Oxygen XML Editor.
- Öffne das Quelldokument „Übung1_booklist.xml“.
- Öffne eine „Neue Datei“ im Menü Datei/Neue Datei (Strg + N) und wähle die Dateischablone „XSLT Stylesheet“

Aufbau eines XSLT Dokumentes:

XSLT-Dokumente beginnen, wie alle XML-Dokumente, mit der XML-Deklaration `<?xml version="1.0"?>`. Mit dem Attribut `encoding` wird die Zeichenkodierung (z.B.: `encoding="UTF-8"`, eine internationale Kodierung auf Basis der ISO/IEC-10646-Norm mit mindestens 8 Bit Zeichenbreite), die zum Speichern der XML-Datei verwendet wird angegeben.

Darauf folgt das Wurzelement (root-Element) `<xsl:stylesheet>`, das zeigt, dass es sich bei dem XML Dokument um ein XSL Dokument handelt. Ebenfalls angegeben werden muss der xsl Namensraum (Namespace) (`xmlns:xsl`) mit der URL <http://www.w3.org/1999/XSL/Transform> und die Version. Hier die zweite Version des XSLT Standards (`version="2.0"`).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet
3     xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4     xmlns:xs="http://www.w3.org/2001/XMLSchema"
5     exclude-result-prefixes="xs"
6     version="2.0">
7
8     <!-- Dokumentinhalt -->
9
10 </xsl:stylesheet>
11

```

- Speichere die Datei unter „Übung1.xsl“ (vielleicht gleich im Ordner wo auch das XML Quelldokument ist).

Schreibe eine Anweisung

Tippe eine spitze Klammer (`<`). Sofort öffnet sich ein Fenster, in dem alle hier möglichen Elemente aufgelistet sind. Um Attribute hinzuzufügen setze den Cursor nach dem Elementnamen und drücke die Leertaste. Dadurch erscheint wieder ein Fenster, diesmal mit den möglichen Attributnamen.

- Setze den Cursor zwischen das öffnende und schließende Tag des Wurzelements `<xsl:stylesheet> </xsl:stylesheet>`.

- b) Tippe eine öffnende Spitzklammer ein und wähle das Element `<template>`. Füge das Attribut `match` mit dem Wert `/` ein. Alle Elemente in `xsl` tragen den Namensraum `xsl`. Der `xsl`-Namensraum ist notwendig um `xsl`-Elemente von nicht `xsl`-Elementen (z.B. XHTML, TEI) zu unterscheiden.

Mit `<xsl:template>` wird ein Template, eine Schablone, angelegt, womit angegeben wird welche Elemente aus dem Ausgangsdokument transformiert werden sollen. In dem Beispiel unten wird die Wurzel des XML Dokument angesprochen (`/`), aber noch kein Text ausgegeben oder weitere Elemente abgearbeitet.

```
<xsl:template match="/"> </xsl:template>
```

Das Stylesheet sollte nun folgendermaßen aussehen:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   xmlns:xs="http://www.w3.org/2001/XMLSchema" exclude-result-prefixes="xs"
4   version="2.0">
5
6   <xsl:template match="/">
7
8   </xsl:template>
9
10 </xsl:stylesheet>
11
```

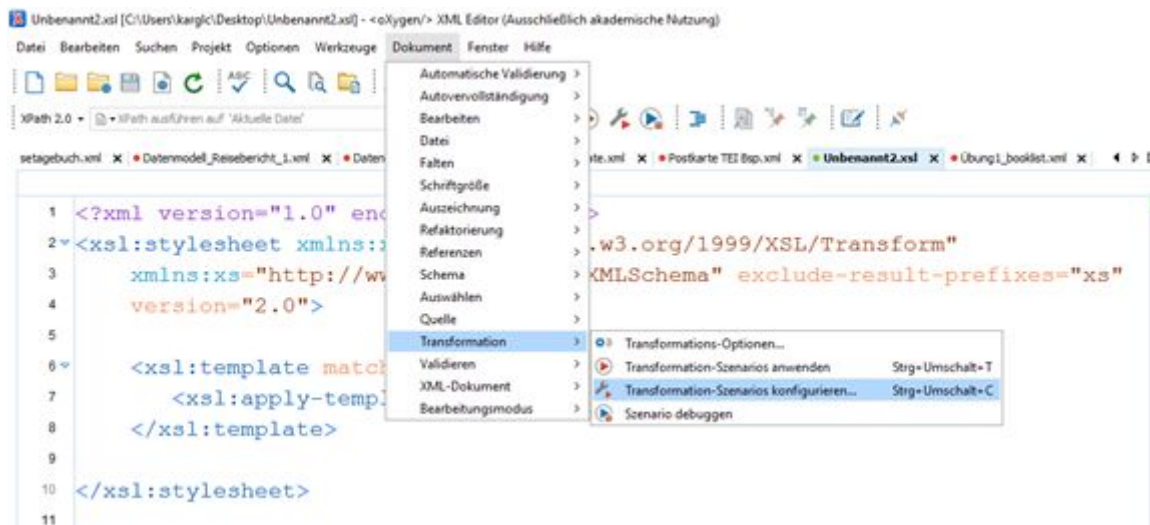
- c) Datei speichern.
- d) Als nächstes wollen wir die Kindelemente des Wurzelements ansprechen. Das erfolgt mit `<xsl:apply-templates/>`. Mit dem Befehl `xsl-apply-templates` werden weitere XSLT Schablonen `<xsl:templates>` aufgerufen. Dadurch können Bezüge und Abfolgen mit dem Attribut `@select` gesteuert werden (z.B. nur bestimmte Elemente ausgewählt werden). Wenn kein `@select` gesetzt ist, spricht das Element `<xsl:apply-templates>` bereits voreingestellt, per *default* alle Kindelemente an.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   xmlns:xs="http://www.w3.org/2001/XMLSchema" exclude-result-prefixes="xs"
4   version="2.0">
5
6   <xsl:template match="/">
7     <xsl:apply-templates/>
8   </xsl:template>
9
10 </xsl:stylesheet>
11
```

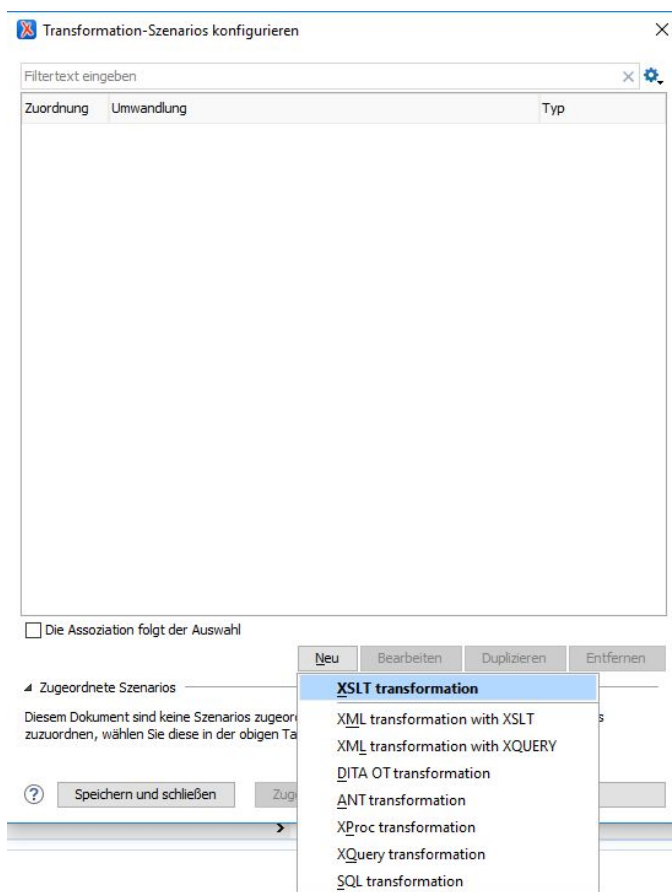
Transformationsszenario durchführen

Um das Ausgangs-XML-Dokument in ein Zieldokument umzuwandeln ist eine Transformation nötig (dabei kann z.B. TEI in XHTML oder ein anderes Format umgewandelt werden). Im oXygen Editor geht das so:

- a) Konfiguriere im Menü unter Dokument/Transformation/Transformation-Szenario konfigurieren (Strg+Umschalt+C) ein Transformationsszenario.




b) Wähle XSLT Transformation

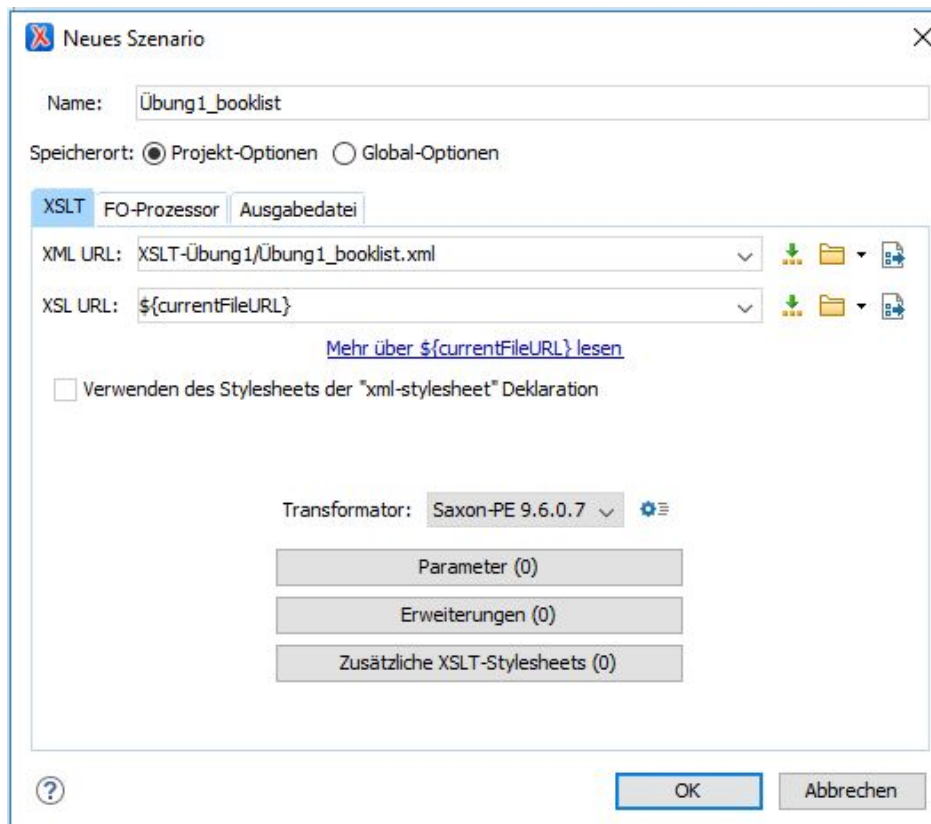


c) Bearbeiten: Szenario bearbeiten

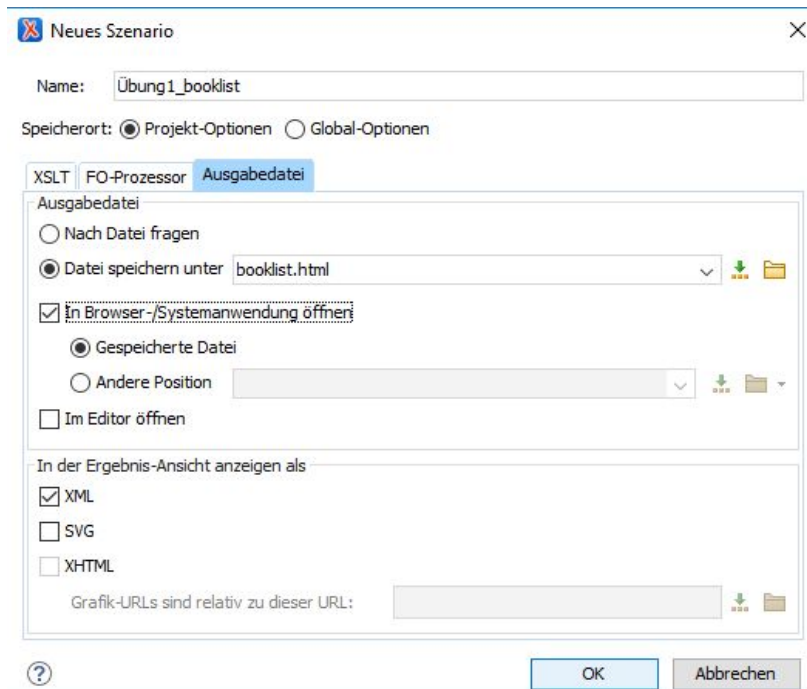
- Reiter XSLT
 - o Name des Transformationszenarios: Übung1_booklist
 - o XML URL: \${currentFileURL} oder file:/C:/.../Übung1_booklist.xml
 - o XSL URL: \${currentFileURL} oder file:/C:/.../Übung1_booklist.xsl
- '\${currentFileURL}' kann unter XML URL und XSL URL ausgewählt werden. Das ist davon abhängig, welche der Dateien aktuell im Editor ausgewählt ist. Ist die XSL-Datei ausgewählt

kann unter XSL URL '`${currentFileURL}`' eingegeben bzw. ausgewählt werden. Mit Hilfe des Ordnersymbols rechts kann die Datei auch ausgewählt werden und man muss den Pfad nicht eintippen.

Wählt man bei der XSL URL die aktuell ausgewählte Datei ist das ein Vorteil, wenn man verschiedene Stylesheets mit dem Quelldokument testen will, da sich das Quelldokument nicht verändert. Über das Symbol  in der Werkzeugleiste kann man somit per Knopfdruck eine neue Transformation starten, ohne sie erneut konfigurieren zu müssen.



- Reiter Ausgabedatei:
 - o Datei speichern unter: booklist.html (Damit die Datei automatisch im Browser geöffnet werden kann, verwenden wir die .html Dateiendung. Obwohl wir in diesem Beispiel eigentlich kein valides HTML produzieren.)
 - o In Browser- /Systemanwendung öffnen anhacken



- „OK“ klicken
- „Zugeordnete anwenden“ klicken

Nun wird die Transformation im Browser angezeigt. Auch wenn unsere Übungsdatei noch keine HTML tags enthält wird der Text im Browser dargestellt.

Da wir eigentlich nur Text ausgeben, könnte die Datei auch eine einfache TXT-Datei sein. In dem Falle würde oXygen einen Texteditor als Systemanwendung aufrufen. Versuch es doch gleich mal (siehe nächste Abbildung).

