

# Projektliste nach parametrisierbarem Stichwort filtern (XSLTrans, Dynamic Documents) (ALT)

Vorausgesetzte Kenntnisse: HTML, TAL, Python

Im folgenden wird ein Workaround vorgestellt, mit dem es möglich ist, mit Hilfe eines XSLTrans Dokuments, sowie eines Dynamic Documents Projektlisten auszugeben, die mittels eines Parameters (keyword) nach dem Stichwort in der Forschungsdatenbank gefiltert werden können.

Zuerst muss man wie üblich ein XSLTrans Dokument anlegen mit den üblichen Parametern, um eine Projektliste auszugeben. Weitere Informationen dazu findet man z.B. bei der [Beschreibung zur Erstellung einer Projektliste](#). Dieses XSLTrans Dokument nennen wir in unserem Beispiel `xsltrans_master_forschungsthemen` und merken uns diesen Kurznamen, da er weiterverwendet wird.

Die Änderungen an diesem XSLTrans müssen wiederum für Anonyme Besucher erlaubt werden. Siehe dazu [Beschreibung zur Erstellung einer Projektliste](#).

Anschließend erstellt man ein Dynamic Document, in dem dieses XSLTrans referenziert wird. Der Quellcode für dieses Dynamic Document sieht dann wie folgt aus:

```
<div tal:define="kostenstelle python:str('1102*')">
  <div tal:define="projectid
python:str(context.REQUEST.get('projectId',''));">
    <div tal:define="keyword
python:str(context.REQUEST.get('keyword',''));">
      <div tal:define='additionalparams python:(str("LfdNr=") +
projectid,str("xql=Stichwort~\~\"") + keyword + str("*\~\""))[len(projectid) ==
0]'\>
        <!-- Comment in to show debug output <div
tal:content="python:str(additionalparams)">XQL goes here</div-->
        <div tal:define='Parameters python:("Jahr=alle", "Ausgabeart=xml",
"Sprache=D", "dokumentart=Projekt", "Kostenstelle=" + kostenstelle, "Lfdnr="
+ projectid, "language=D", "image_float=right", additionalparams); foo
python:context.xsltrans_master_forschungsthemen.setParam(Parameters)'\>
          <div tal:content="structure
python:context.xsltrans_master_forschungsthemen.get_result()" />
        </div>
        <!-- reset der Parameter von xsltrans -->
        <div tal:define='Parameters python:("Jahr=alle", "Ausgabeart=xml",
"Sprache=D", "dokumentart=Projekt", "Kostenstelle=" + kostenstelle,
"language=D", "image_float=right"); foo
python:context.xsltrans_master_forschungsthemen.setParam(Parameters)'\ />
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
</div>
```

Im folgenden möchte ich die einzelnen Bestandteile kurz erläutern, da einige Tricks notwendig sind. Zuerst muss man sich überlegen, dass man der Forschungsdatenbank entweder den XQL Parameter `xql=Stichwort~~,<MeinStichwort>` übergeben will, oder den Parameter `LfdNr=42`. Beide Parameter zu kombinieren funktioniert nicht, da die Forschungsdatenbank dann anscheinend nur den XQL Parameter auswertet.

Damit das funktioniert, muss man eine if-then-else Bedingung aufbauen. Das einfachste wäre nun mittels eines ternären Operators zu überprüfen, ob eine `projectId` übergeben wurde, und falls ja, das Projekt mit dieser laufenden Nummer abzufragen, anderenfalls Projekte mit dem angegebenen Stichwort abzufragen.

Das sähe dann in etwa so aus:

```
parameter = (<Bedingung>) ? "xql=Stichwort~~\" + keyword + "\" : "LfdNr=" + projectId
```

Nun gibt es in Python auch den ternären Operator, d.h. in Python sähe das so aus:

```
"xql=Stichwort~~\" + keyword + "\" if <Bedingung> else "LfdNr=" + projectId
```

Das funktioniert in Plone/TAL auf dem CMS der Universität aber leider nicht (oder ich habe die Syntax nicht korrekt hinbekommen), weshalb man einen Trick mit einem Array anwenden muss.

```
("a", "b")[<bedingung>]
```

Hier wird entweder „a“ zurückgegeben, oder „b“ in Abhängigkeit davon, ob `<bedingung>` zu true oder false, d.h. 1 oder 0 evaluiert. Das funktioniert an dieser Stelle nur, weil boolean eine Subklasse von int ist.

Jetzt braucht man noch eine geeignete Bedingung, die allerdings schnell gefunden ist. Wir überprüfen einfach die Länge des Parameters `projectId`, nachdem wir diesen nach String konvertiert haben:

```
<div tal:define="projectId"
python:str(context.REQUEST.get('projectId', ''));">
... len(projectid) == 0 ...
</div>
```

Kurz rekapituliert, je nachdem wie diese Bedingung evaluiert, wollen wir entweder nach Stichwort filtern:

```
str("xql=Stichwort~~\" + keyword + str("*\"")
```

oder eben ein spezifisches Projekt ausgeben:

```
str("LfdNr=") + projectId
```

Weiterer Trick: Wir ergänzen hier an dieser Stelle bei der Stichwortabfrage noch einen \* nach dem

keyword, falls kein Stichwort übergeben wurde, wird dann die Liste aller Projekte ausgegeben, anstatt keines. Damit das funktioniert, verwenden wir an dieser Stelle im HTML single Quotes, anstatt der Double Quotes, damit das Escapen einfacher wird, da wir für die Forschungsdatenbank `xql=Stichwort~~, <Stichwort>` und nicht `xql=Stichwort~~<Stichwort>` übergeben müssen.

Das kombinieren wir jetzt mit unserer Bedingung und unserem Array-Trick:

```
(str("LfdNr=") + projectid, str("xql=Stichwort~~\"") + keyword + str("*\""))[len(projectid) == 0]
```

und verpacken das ganze in eine TAL-Anweisung:

```
<div tal:define='additionalparams python:(str("LfdNr=") + projectid, str("xql=Stichwort~~\"") + keyword + str("*\""))[len(projectid) == 0] '>
```

Damit kann dann in den Parametern der zusätzliche Parameter `additionalparams` eingebaut werden, der entweder das Stichwort abfragt, oder eben ein bestimmtes Projekt.

Jetzt kann dann das Dynamic Document (nennen wir es `forschungsthema-details-nach-stichwort`) mit folgenden Parameterkombinationen aufrufen: `forschungsthema-details-nach-stichwort?keyword=Brennstoffzellen` oder auch `forschungsthema-details-nach-stichwort?projectId=8076`. Somit kann man auf einer HTML-Seite dann die erstgenannten Links einbindungen und Ergebnislisten, gefiltert nach Stichwort ausgeben.

— *Christoph Hermann* 2012/09/20 16:20

[techniker](#), [plone](#), [tf-infoportal](#), [xsltrans](#), [parameter](#)

From: <https://wiki.uni-freiburg.de/tf-infoportal/> - **Entwicklungs-Wiki für das Infoportal der Technischen Fakultät**

Permanent link: <https://wiki.uni-freiburg.de/tf-infoportal/doku.php?id=tf-infoportal:dokumentation:plone:plone-erweiterte-projektliste-nach-stichwort> 

Last update: **2013/11/04 14:47**