Projektseite Selbstzeugnisse

Oktober 2019; ergänzt November 2020 von Maximilian Görmar

# Zusammenfassung

Die Projektseite Selbstzeugnisse ist als Arbeitsumgebung für das Projekt "Selbstzeugnisse der Frühe Neuzeit an der Herzog August Bibliothek" (2015–2017) entwickelt worden. Die Projektseite ist ein in sich geschlossenes System im dem die XML-Quelldateien sowie alle zur Verarbeitung und Anzeige benötigten Skripte usw. vorliegen.

# Baustellen/Korrekturbedarf

## Abgekündigtes Micro-Framework

Da das der Webanwendung zugrundeliegenden Micro-Framwork Silex[[1]](#footnote-1) zum Juni 2018 abgekündigt wurde sollte der Einsatz eines adäquaten Ersatzes in Erwägung gezogen werden. Im IIIF-Pilotprojekt wurden gute Erfahrungen mit Slim[[2]](#footnote-2) gemacht, dass einer ähnlichen Philosophie folgt. Durch den Fokus auf XSLT und einen sehr überschaubaren Einsatz eigenentwickelter PHP-Dienste sollte eine Migration auf ein anderes Mikroframework mit verhältnismäßig wenig Aufwand durchgeführt werden können. [Silex nach wie vor verwendet, MG]

## Lightbox auf kleinen Bildschirmen

Die für die Anzeige der digitalen Faksimile verwendete Lightbox lässt sich auf kleinen Bildschirmen nicht schließen, da der entsprechende Button rechts außerhalb des sichtbaren Bereichs positioniert wird. [bei SZ II durch Bilddatei in iframe ersetzt mit dem Problem, dass die Bilddatei von Gerät zu Gerät (Browser zu Browser?) in anderer Zoomstufe angezeigt wird; ursprünglich DFG-Viewer geplant, aber der hat eine zu schlechte Auflösung; perspektivisch soll auf die zukünftige III-F-Lösung der Hab umgestiegen werden, MG]

# Notwendige Schritte zur Erweiterung der Projektseite

## Festlegung des Publikationsziels

* Übersicht über die kodierten Entitäten und ihrer Gestaltung (Optik & Verhalten)
* Aufgliedern des Quellmaterials in sinnvolle Einheiten ("chunking")
* Festlegen der Zugänge und zugehöriger Navigation
* Übersetzen in die Seitenstruktur (routes)
* Analyse der Auswirkung auf bestehende Seiten und Funktionen

## Umsetzen der Publikation

* Anpassen oder Neuschreiben der XSL-Transformationen [nur 1.0 verarbeitbar, MG]
* Anpassen der Wartungsroutinen
* Ggf. Einrichtung neuer Wartungsroutinen (Indexierug, Aufbereitung, Aggregation usw.)

Beachte: Die Seiten der Edition des Diariums von Herzog August dem Jüngeren sind in der Digitalen Bibliothek[[3]](#footnote-3) direkt verlinkt. Die Links müssen bei Änderung der Seitenstruktur angepasst werden.

# Interna

## Lokale Installation der Projektseite

Die Projektseite ist als Git-Repository auf Github[[4]](#footnote-4) veröffentlicht und muss für eine lokale Installation kopiert werden.

git clone https://github.com/dmj/selbstzeugnisse-hab.git

Alle für den Betrieb der Projektseite notwendigen Abhängigkeiten sind Teil des Repositories. Die Projektseite wird mit einer Webanwendung auf Grundlage des Micro-Frameworks Silex[[5]](#footnote-5) realisiert.

## Verzeichnisstruktur

### Resources

Das Verzeichnis resources enthält alle Quelldateien (Unterverzeichnis daten) und für deren Verarbeitung notwendigen Skripte und Hilfsdateien.

### Webverzeichnis

Das Verzeichnis public enthält den Front Controller (index.php) und die für die HTML-Darstellung benötigten Dateien (CSS, JavaScript usw.).

### Konfiguration der Webanwendung

Die Webanwendung wird im Verzeichnis config konfiguriert. Hier sind Controller (controllers.php), Routes (routes.php) und Anwendungsdienste (services.php) definiert.

### Anpassung der Webanwendung

Das Verzeichnis src enthält die für die Projektseite implementierten Funktionen der Webanwendung.

### Zusatzprogramme

Das Verzeichnis tools enthält die für die Wartungsroutinen erforderliche XProc Software XML Calabash[[6]](#footnote-6).

### Paketverwaltung

Die Abhängigkeiten der Webanwendung werden mit Composer[[7]](#footnote-7) verwaltet. Sind werden in composer.json und composer.lock definiert und sind im Verzeichnis vendor installiert.

## METS-Datei

Herz der Projektseite bildet die METS-Datei resources/mets.xml in der Inhalt und Struktur des Portals beschrieben werden. In ihrem Aufbau folgt die Datei dem "Dispatcher-Prinzip".

Das heißt:

1. Alle möglichen Unterseiten der Projektseite sind durch eine mets:div in der mets:structMap repräsentiert. Der Wert des @ID-Attributs entspricht der Route in der der Pfadtrenner / durch einen Punkt . ersetzt ist.
2. Das jeweilige XML-Quelldokument ist in der mets:fileSec aufgeführt und mit der Beschreibung der Unterseite über einen mets:fptr verknüpft.
3. Das zur Darstellung verwendete XSLT ist in einem mets:behavior beschrieben, das über das @STRUCTID-Attribut mit der oder den Unterseiten verbunden ist.

Die Struktur der METS-Datei wird durch die Klasse HAB\Selbstzeugnisse\Configuration interpretiert. Sie implementiert eine Funktion zum Lokalisieren des XML-Quelldokuments (resolveSourceDocumentUri) und eine Funktion zum ermitteln der notwendigen XSL-Transformation (resolveBehavior). Der einfache HAB\Selbstzeugnisse\StaticController lädt das Quelldokument, ermittelt die XSL-Transformation, führt selbige aus und sendet das Ergebnis an den Browser.

## Definition der Routes

Die Beziehung zwischen der aufgerufenen Unterseite (Route) und der korrespondierenden Struktur in der METS-Datei wird in einer separaten Konfigurationsdatei config/routes.php gepflegt. Ursprünglich war geplant, eine gesonderte Routing-Komponente des zugrundeliegenden Frameworks zu implementieren, die direkt auf die Informationen der METS-Datei zugreift. Aus Mangel an Ressourcen wurde diese "doppelte Buchführung" beibehalten.

# Anlegen einer neuen Seite

Das Anlegen einer neuen Seite erfolgt in fünf Schritten: [NB: statisches html kann scheinbar nicht verarbeitet werden, MG]

1. Das XML-Quelldokument muss in das Verzeichnis resources an eine angemessene Stelle kopiert werden. Im Allgemeinen folgt die Struktur dieses Verzeichnisses der Hierarchie der Unterseiten der Projektseite.
2. Das XML-Quelldokument muss in der mets:fileSec der METS-Datei verzeichnet werden.
3. Die Unterseite muss in der mets:structMap beschrieben und mit der Datei in der mets:fileSec über ein mets:fptr verbunden werden.
4. Das XSLT [NB: Server kann nur xsl 1.0, MG] für die Umwandlung muss in der mets:behaviorSec in einem mets:behavior beschrieben und mit der Unterseite über das @STRUCTID-Attribut verbunden werden.
5. In der Konfigurationsdatei config/routes.php ist eine Route anzulegen.

# Wartungsroutinen

[Wartungsroutinen werden nicht mehr über den Makefile und die dort definierten Targets ausgeführt, sondern über die im Ordner bin abgelegten .bat-Dateien, MG]

## Lokaler Entwicklungsserver

Das Makefile definiert das Target server, mit dessen Hilfe ein lokaler Entwicklungsserver gestartet werden kann.

make server

Die lokale Instanz ist dann unter der Adresse http://127.0.0.1:9999 im Browser abrufbar. [bisher nur unter Ubuntu realisiert; Befehl entspricht nun Datei start\_server.bat, MG]

## Indexieren der Projektseiten

Das Makefile Target solr indexiert die Repertoriumseinträge und den Inhalt der Projektseiten in den respektiven Solr Index 'selbst' (Repertorium) und 'selbst-ft' (Inhalt der Projektseiten). Die Indexierung erfolgt in beiden Fällen über eine XProc-Pipeline, die die betreffenden Dokumente selektiert und mittels http an Solr überträgt.

make solr

Die Volltextindexierung der Projektseiten basiert auf der METS-Datei, in der alle möglichen Projektunterseiten definiert sind.

## BEACON

Das Makefile Target beacon erstellt eine BEACON-Datei auf Basis einer Übersicht über die in der Edition verlinkten Personen.

make beacon

Die BEACON-Datei beruht auf dem XRD-Dokument[[8]](#footnote-8) resources/daten/edition/diarium/linkmap.xml, in dem jeder Tagebucheintrag mit den in ihn erwähnten Personen verknüpft ist. Das XRD-Dokument wird durch das Target linkmap erzeugt, das eine Abhängigkeit des Targets beacon ist. [NB: Beacon-Datei muss überarbeitet werden; ist z.B. von Hartmut Beyers Beaconfinder (http://beaconfinder.mww-forschung.de/) nicht nutzbar, MG; 27.06.2022: neue Beacon-Datei erstellt, die auf das Personenregister verlinkt, MG]

## Aktualisieren der Normdaten

Das Makefile Target normdaten lädt die Normdatensätze des Personen- und Ortsregisters in das Projektverzeichnis.

make normdaten

## Sitemap erstellen

Das Makefile Target sitemap erstellt eine Google Sitemap aus der zentralen METS-Datei.

make sitemap

## OAI-PMH Static Repository für Repertorium

Das Makefile Target repository erstellt ein OAI-PMH Static Repository für die Selbstzeugnisse des Repertoriums.

make repository

# Geobrowserdateien

Zur SZII-Edition liegen Datensätze zum Ortsregister sowie den Itineraren im DARIAH-Geobrowser vor. [Stand: 31.05.2022]

## Gesamtdateien

* Itinerar: <https://geobrowser.de.dariah.eu/?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-60B1-195E-0614-0>
* Ortsregister: <https://geobrowser.de.dariah.eu/?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-47B5-8B5E-5D70-0>

## Cod. Guelf. 28 Blank

* Itinerar: <https://geobrowser.de.dariah.eu/index.html?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-C308-CBCF-8C56-0>
* Ortsregister: <https://geobrowser.de.dariah.eu/index.html?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-960B-78B5-D859-0>

## Cod. Guelf. 211.1 Extrav.

* Itinerar: <https://geobrowser.de.dariah.eu/?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-2D6B-A11D-78A1-0>
* Ortsregister: <https://geobrowser.de.dariah.eu/?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-1594-5C1F-AC61-0>

## Cod. Guelf. 286 Blank.

* Itinerar: <https://geobrowser.de.dariah.eu/index.html?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-C396-D651-1ACD-0>
* Ortsregister: <https://geobrowser.de.dariah.eu/index.html?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-0218-0228-50FF-0>

## Cod. Guelf. 286a Blank.

* Itinerar: <https://geobrowser.de.dariah.eu/index.html?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-C9E8-87D2-095C-0>
* Ortsregister: <https://geobrowser.de.dariah.eu/index.html?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-14BB-52C2-B076-0>

## Synoptische Darstellung der Einzelsets:

* Itinerar: <https://geobrowser.de.dariah.eu/index.html?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-C308-CBCF-8C56-0&csv2=https://cdstar.de.dariah.eu/dariah/EAEA0-2D6B-A11D-78A1-0&csv3=https://cdstar.de.dariah.eu/dariah/EAEA0-C396-D651-1ACD-0&csv4=https://cdstar.de.dariah.eu/dariah/EAEA0-C9E8-87D2-095C-0>
* Ortsregister: <https://geobrowser.de.dariah.eu/?csv1=https://cdstar.de.dariah.eu/dariah/EAEA0-1594-5C1F-AC61-0&csv2=https://cdstar.de.dariah.eu/dariah/EAEA0-14BB-52C2-B076-0&csv3=https://cdstar.de.dariah.eu/dariah/EAEA0-0218-0228-50FF-0&csv4=https://cdstar.de.dariah.eu/dariah/EAEA0-960B-78B5-D859-0&currentStatus=mapChanged=Historical+map+of+1715>

1. https://silex.sensiolabs.org [↑](#footnote-ref-1)
2. https://www.slimframework.com [↑](#footnote-ref-2)
3. https://doi.org/10.15499/edoc/ed000225 [↑](#footnote-ref-3)
4. https://github.com [↑](#footnote-ref-4)
5. https://silex.sensiolabs.org [↑](#footnote-ref-5)
6. http://xmlcalabash.com [↑](#footnote-ref-6)
7. https://getcomposer.org [↑](#footnote-ref-7)
8. http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.html [↑](#footnote-ref-8)