# Workshop ediarum.WEB Anleitung

Technische Voraussetzungen:

* eXist 3.2 / 4.6 / 5.2

## 1. Beispieldaten

* Die Briefe aus dem Ordner /data nach /db/projects/workshop/data/Briefe kopieren.
* Das Personenregister aus dem Ordner /data nach /db/projects/workshop/data/Briefe kopieren

## 2. Installation

* Über das Dashboard ediarum.web\_VERSION.xar installieren.
* Über das Dashboard workshop.web\_VERSION.xar installieren.

## 3. Grundkonfiguration

### appconf.xml

* In eXide die Datei appconf.xml in /db/apps/workshop.web öffnen.
* Das Datenverzeichnis config/project/collection überprüfen. Es sollte Auf den Ordner /db/projects/workshop/data zeigen.

### Webseite einrichten

* Webseite über das Dashboard workshop.web öffnen.
* Den Header und Footer in /templates/page.html nach Bedarf anpassen.
* Die Startseite unter /views/static-pages/index.html nach Bedarf anpassen.
* Das CSS in /resources/css/main.less nach Bedarf anpassen.
* Ergebnisse auf der Webseite überprüfen:
  + http://localhost:8080/exist/apps/workshop.web/index.html

## 4. Objekte anlegen

* In eXide die Datei appconf.xml in /db/apps/workshop.web öffnen.
* Ein Objekt für die Briefe anlegen, indem folgender Knoten in die appconf.xml eingefügt wird:
* <object xml:id="briefe">  
   <name>Briefe</name>  
   <collection>/Briefe</collection>  
   <item>  
   <namespace id="tei">http://www.tei-c.org/ns/1.0</namespace>  
   <root>tei:TEI</root>  
   <id>@xml:id</id>  
   <label type="xpath">.//tei:titleStmt/tei:title[1]/normalize-space()</label>  
   </item>  
  </object>
* Ein Objekt für das Personenregister anlegen, indem folgender Knoten in die appconf.xml eingefügt wird:
* <object xml:id="personen">  
   <name>Personen</name>  
   <collection>/Register</collection>  
   <item>  
   <namespace id="tei">http://www.tei-c.org/ns/1.0</namespace>  
   <root>tei:person</root>  
   <id>./@xml:id</id>  
   <label type="xpath">.//tei:persName/normalize-space()</label>  
   </item>  
  </object>
* Ergebnisse in der API prüfen:
  + http://localhost:8080/exist/apps/workshop.web/api
  + http://localhost:8080/exist/apps/workshop.web/api/briefe
  + http://localhost:8080/exist/apps/workshop.web/api/personen
  + http://localhost:8080/exist/apps/workshop.web/api?id=all
* Ergebnisse auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/index.html
  + http://localhost:8080/exist/apps/workshop.web/briefe/index.html
  + http://localhost:8080/exist/apps/workshop.web/personen/index.html

## 5. Filter anlegen

* Für die Briefe einen Filter ergänzen. Dazu in der appconf.xml im Object <object xml:id="briefe"> folgenden Code ergänzen:
* <filters>  
   <filter xml:id="correspYear">  
   <name>Jahr</name>  
   <type>union</type>  
   <xpath>.//tei:correspAction//tei:date/@when/substring(.,1,4)</xpath>  
   <label-function type="xquery">function($string) { $string }</label-function>  
   </filter>  
  </filters>
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/briefe/index.html
  + http://localhost:8080/exist/apps/workshop.web/briefe/index.html?cache=no
* Für die Briefe weitere Filter ergänzen. Dazu folgenden Code ergänzen:
* <filter xml:id="correspPlaces">  
   <name>Orte</name>  
   <type>intersect</type>  
   <xpath>.//tei:correspAction//tei:placeName/normalize-space()</xpath>  
   <label-function type="xquery">function($string) { $string }</label-function>  
  </filter>  
  <filter xml:id="correspPersons">  
   <name>Korrespondenzpartner</name>  
   <type>intersect</type>  
   <xpath>.//tei:correspAction//tei:persName/normalize-space()</xpath>  
   <label-function type="xquery">function($string) { $string }</label-function>  
  </filter>
* Ergebnisse in der API prüfen:
  + http://localhost:8080/exist/apps/workshop.web/api
  + http://localhost:8080/exist/apps/workshop.web/api/briefe
* Für die Personen Filter ergänzen. Dazu in der appconf.xml im Object <object xml:id="personen"> folgenden Code ergänzen:
* <filters>  
   <filter xml:id="alphabet">  
   <name>alphabetisch</name>  
   <type>single</type>  
   <root type="label"/>  
   <label-function type="xquery">  
   function($string) {substring(replace(normalize-space($string), '^\(', ''),1,1)}  
   </label-function>  
   </filter>  
   <filter xml:id="birth">  
   <name>Geburtsjahr</name>  
   <type>single</type>  
   <xpath>.//tei:birth/normalize-space()</xpath>  
   <label-function type="xquery">  
   function($string) { $string }  
   </label-function>  
   </filter>  
  </filters>
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/personen/index.html?cache=no
* Den Filter für das Geburtsjahr verbessern, indem die Labelfunktion ersetzt wird mit:
* <label-function type="xquery">  
   function($string) {  
   if (matches($string, "(v\. Chr\.)|(v\. Chr)|(vor Christi)"))  
   then "- v. Chr."  
   else if (matches($string, "(n\. Chr\.)|([^\d]\d\d\d?[^\d])|(^\d\d?\d?$)|([^\d]\d\d\d?$)"))  
   then "1. Jahrtausend"  
   else if (matches($string, "1[0123]\d\d"))  
   then "11.-14. Jh."  
   else if (matches($string, "(1[4567]\d\d)|(1[67]\. Jh\.)"))  
   then "15.-17. Jh."  
   else $string  
   }  
  </label-function>
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/personen/index.html?cache=no

## 6. XSLT einrichten

* Detailseite eines Briefe besuchen:
  + http://localhost:8080/exist/apps/workshop.web/briefe/c5z\_vfl\_d3b
* Detailseite einer Person besuchen:
  + http://localhost:8080/exist/apps/workshop.web/personen/Boulliau
* Für die Briefe ein XLST einrichten. Dazu in der appconf.xml im Object <object xml:id="briefe"> folgenden Code ergänzen:
* <views>  
   <view id="default">  
   <xslt>resources/xslt/personen\_details.xsl</xslt>  
   <label>Standard</label>  
   </view>  
  </views>
* Für die Personen ein XLST einrichten. Dazu in der appconf.xml im Object <object xml:id="briefe"> folgenden Code ergänzen:
* <views>  
   <view id="default">  
   <xslt>resources/xslt/personen\_details.xsl</xslt>  
   <label>Standard</label>  
   </view>  
  </views>
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/briefe/c5z\_vfl\_d3b
  + http://localhost:8080/exist/apps/workshop.web/personen/Boulliau

## 7. Eigene Listenansicht anlegen

Wir wollen den alphabetischen Filter für Personen nicht als Listenauswahl sondern kompakter anzeigen.

* Anweisung im Controller controller.xql in Zeile 19 ergänzen:
* edwebcontroller:view-with-feed("/personen/index.html", "data-pages/personen.html", "object-type/personen"),
* Die Datei /views/data-pages/template-list.html kopieren und personen.html nennen.
* Ergebnis auf der Webseite prüfen.
  + http://localhost:8080/exist/apps/workshop.web/personen/index.html?cache=no
* In der Datei personen.html in Zeile 66
* <div data-template="edweb:template-show-filters">
* ersetzen durch
* <div data-template="edweb:load-filter" data-template-filter-name="alphabet">
* In der Datei personen.html in Zeile 71
* <ul class="combobox">
* ersetzen durch
* <ul class="grid">
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/personen/index.html?cache=no

## 8. Eigene Detailansicht anlegen

Wir wollen die Detailseite für Personen individualisieren.

* Anweisung im Controller controller.xql in Zeile 20 ergänzen:
* edwebcontroller:view-with-feed("/personen/", "data-pages/personen-details.html", "object-type/personen/id/"),
* Die Datei /views/data-pages/template-details.html kopieren und personen-details.html nennen.
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/personen/Boulliau

### Die Breadcrumbleiste anpassen

* In der Datei personen-details.html in Zeile 20
* <ol data-template="edweb:add-breadcrumb-items"></ol>
* ersetzen durch
* <ol data-template="edweb:add-breadcrumb-items" data-template-filter="alphabet"></ol>
* Ergebnis auf der Webseite prüfen.
  + http://localhost:8080/exist/apps/workshop.web/personen/index.html?cache=no

### Metadaten per XSLT ergänzen

* In der Datei personen.html in Zeile 36 und 37 folgenden Code ergänzen: XML <div data-template="edweb:template-transform-current" data-template-resource="resources/xslt/personen\_metadata.xsl"></div>
* Ergebnis auf der Webseite prüfen.
  + http://localhost:8080/exist/apps/workshop.web/personen/index.html?cache=no

### Beziehungen hinzufügen

* Für die Absender von Briefen eine Beziehung ergänzen. Dazu folgenden Knoten in die appconf.xml eingefügen:
* <relation xml:id="absender" subject="personen" object="briefe">  
   <name>Absender</name>  
   <collection>/Briefe</collection>  
   <item>  
   <namespace id="tei">http://www.tei-c.org/ns/1.0</namespace>  
   <root>tei:correspAction[@type='sent']/tei:persName</root>  
   <label type="xquery">function ($node as node()) { "Absender" }</label>  
   </item>  
   <subject-condition>  
   function($this as map(\*), $subject as map(\*)) {  
   $this?xml/@key = $subject?id  
   }  
   </subject-condition>  
   <object-condition>  
   function($this as map(\*), $object as map(\*)) {  
   $this?absolute-resource-id = $object?absolute-resource-id  
   }  
   </object-condition>  
  </relation>
* Ergebnisse in der API prüfen:
  + http://localhost:8080/exist/apps/workshop.web/api/absender
* Die Beziehungen bei den Personen anzeigen. Dazu in der Datei personen.html in Zeile 49 folgenden Code ergänzen:
* <div id="section/relations-msdesc">  
   <h3>  
   Absender folgender Briefe:  
   </h3>  
   <div data-template="edweb:load-relations-for-subject"   
   data-template-relation="absender">  
   <div data-template="edweb:template-switch">  
   <switch><span data-template="edweb:insert-count"   
   data-template-from="relations"></span></switch>  
   <p case="0">Keine Briefe vorhanden.</p>  
   <span case="default">  
   <table class="table">  
   <thead>  
   <tr>  
   <th scope="col">Titel</th>  
   <th scope="col">Kontextrolle</th>  
   </tr>  
   </thead>  
   <tbody>  
   <div data-template="templates:each"   
   data-template-from="relations"   
   data-template-to="item">  
   <tr>  
   <td><a data-template="edweb:template-detail-link"   
   data-template-from="item">  
   <span data-template="edweb:insert-string"   
   data-template-path="item?label"></span></a></td>  
   <td><span data-template="edweb:insert-string"   
   data-template-path="item?predicate"></span></td>  
   </tr>  
   </div>  
   </tbody>  
   </table>  
   </span>  
   </div>  
   </div>  
  </div>
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/personen/Boulliau

## 9. Eigene Routinen schreiben

Wir wollen die Detailseite für Briefe individualisieren, indem eine zweispaltige Ansicht und ein seitenweises Blättern eingerichtet werden.

* Anweisung im Controller controller.xql in Zeile 20 ergänzen:
* edwebcontroller:view-with-feed("/briefe/", "data-pages/briefe-details.html", "object-type/briefe/id/"),
* Die Datei /views/data-pages/template-details.html kopieren und briefe-details.html nennen.
* Ergebnis auf der Webseite prüfen:
  + http://localhost:8080/exist/apps/workshop.web/briefe/c5z\_vfl\_d3b

### Appconf anpassen

* Ein Seitenreferenzierung für die Briefe anlegen, indem in der appconf.xml im Object <object xml:id="briefe"> folgender Code ergänzt wird:
* <parts separator="." prefix="-">  
   <part xml:id="page" starts-with="p">  
   <root>tei:milestone[@type='page']</root>  
   <id>./@n</id>  
   </part>  
  </parts>
* Ergebnisse in der API prüfen:
  + http://localhost:8080/exist/apps/workshop.web/api/briefe/c5z\_vfl\_d3b?part=page

### Eigenen Code schreiben

* Die Datei /modules/app.xql durch die gleichname aus dem Zip-Ordner examples kopieren.

### Eigenen Code einbinden

* In der Datei briefe-details.html in Zeile 45 bis 52
* <div class="row">  
   <div class="col-md-1"></div>  
   <div class="col-md-7 book-like" id="object-content">  
   <div data-template="edweb:template-transform-current"></div>  
   </div>  
   <div class="col-md-1"></div>  
   <div class="col-md-3">  
   </div>  
  </div>
* ersetzen durch
* <div class="row" data-template="app:add-page-id">  
   <div data-template="app:load-parts" data-template-part="page">  
   <div data-template="app:load-page">  
   <div class="col-6 box whitebox page-height">  
   <div class="nav-scroller py-1 mb-2">  
   <nav class="nav d-flex justify-content-center">  
   <div data-template="app:select-page"></div>  
   </nav>  
   </div>  
   <div data-template="app:transform"   
   data-template-from="part"   
   data-template-resource="resources/xslt/briefe\_details.xsl"></div>  
   </div>  
   <div class="col-1"></div>  
   <div class="col-5 box whitebox page-height">  
   <div data-template="edweb:template-transform-current"   
   data-template-resource="resources/xslt/briefe\_abstract.xsl"></div>  
   </div>  
   </div>  
   </div>  
  </div>
* Ergebnis auf der Webseite prüfen.
  + http://localhost:8080/exist/apps/workshop.web/briefe/c5z\_vfl\_d3b