
Aktuelle URLs und aktuelle Verweise

Inhaltsverzeichnis

Aktuelle URLs und aktuelle Verweise	1
Datei redirects.xml	2
redirects.rng - Sprechende URLs	3
Datei htaccess.tmp	4
redirects_htaccess.xslt	5
redirects_ht.xslt	8
redirects.css	14
redirects.js	17
redirects	19

Aktuelle URLs und aktuelle Verweise

Ich kann schlecht mit URLs wie `http://kleider.herbaer.de/s2014w16/story.xml#s24_26gsyfh0_0` für meine Website werben. Besser ist eine sprechende URL wie `http://kleider.herbaer.de/aktuell/schwarzer-minirock`. Diese "sprechenden URLs" stelle ich in einer XML-Datei (`supplement/redirects.xml`) zusammen, die wie dieses Beispiel aufgebaut ist:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- file KLEIDER/web/src/aktuell/redirects.xml -->
<!-- Beispiel für eine Quelldatei der "Umleitungen" -->
<redirects xmlns="http://herbaer.de/xmlns/20140430/redirects/">
  <redirect>
    <text>Schwarzer Minirock</text>
    <text>Minirock</text>
    <relurl>s2014w16/story.xml#s24_26gsyfh0_0</relurl>
  </redirect>
  <redirect>
    <relurl>s2014w16/story.xml#s2_26gsyf5c_0</relurl>
    <text>Rotes Tanzkleid</text>
    <text>Tanzkleid</text>
  </redirect>
  <redirect>
    <relurl>s2014w27/story.xml</relurl>
    <text>Kathrin</text>
  </redirect>
</redirects>
<!-- end of file KLEIDER/web/src/aktuell/redirects.xml -->
```

Die Datei `redirects.rng` beschreibt den Aufbau.

Umleitungen

Die umgeleiteten URLs (relativ zur Server-Wurzel) beginnen mit `/aktuell/`. Die Datei `aktuell/.htaccess` im Server-Verzeichnis leitet die URLs um. Sie ist so aufgebaut:

```
# file KLEIDER/web/src/aktuell/htaccess.tmp
# Beispiel einer "Umleitungsdatei"

RewriteEngine On
RewriteBase /
RewriteRule ^schwarzer[-_]?minirock$ s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^minirock$ s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^rotes[-_]?tanzkleid$ s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]
RewriteRule ^tanzkleid$ s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]
RewriteRule ^kathrin$ s2014w27/story.xml [NC,NE,R=301,L]
RewriteRule ^schwarzer[-_]?mini s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^mini s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^rotes[-_]?tanz s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]
RewriteRule ^tanz s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]

# end of file KLEIDER/web/src/aktuell/htaccess.tmp
```

Die Datei `aktuell/.htaccess` wird mit der Transformation `redirects_htaccess.xslt` aus `supplement/redirects.xml` erzeugt.

Übersicht als Index-Datei (XHTML)

Die Datei `aktuell/index.xhtml` im Server-Verzeichnis enthält eine Übersicht über die "sprechenden URL". Sie wird aus der Quelldatei `supplement/redirects.xml` durch die Transformation `redirects_ht.xslt` erzeugt. Das erzeugte XHTML-Dokument nutzt die Dateien `redirects.css`, `redirects.js` sowie `style/common.js`.

Die Dateien `redirects.css` und `redirects.js` werden per `XInclude` in die Transformation eingebunden. Das erfolgt genauso wie zur Darstellung der Bildergeschichten mit denselben Dateien `styleincl_step_1.xslt` und `styleincl_step_2.xslt`. Die Transformation mit den eingebundenen Dateien erzeugt die XHTML-Datei `aktuell/index.xhtml`.

Mit kleinen Anpassungen kann von der Einbindung der Javascript- und CSS-Dateien per `XInclude` abgesehen werden. Die XHTML-Datei enthält dann Verweise auf Dateien, die der Browser einbindet. Man kann auch die Quelldatei `supplement/redirects.xml` mit einem Verweis auf eine XSLT-Transformation (wahlweise mit eingebundenen Javascript- und CSS-Dateien) an den Browser senden und die Transformation dem Browser überlassen. Diese Varianten erfordern aber mehr Server-Anfragen (Request) und sind allenfalls zur Fehlersuche oder Fehlerbehebung nützlich.

Automatische Verarbeitung

Das Skript `redirects` führt die nötigen Schritte nach einer Änderung der Datei `supplement/redirects.xml` aus.

Internationalisierung

Die Internationalisierung und Lokalisierung der sprechenden URL ist schwierig. Einfacher ist die Internationalisierung und Lokalisierung der Texte zu den URL. Sofern die Quelldatei `supplement/redirects.xml` in verschiedenen Sprachen vorliegt, kann auch die Übersichtsdatei `DOCROOT/aktuell/index.xhtml` in verschiedenen Sprachen erzeugt werden. Abhängig von der Sprache der Quelldatei bindet die Transformation `redirects_ht.xhtml` die Lokalisierungsdatei `local.xml.LANG` ein.

Datei `redirects.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Beispiel für eine Quelldatei der "Umleitungen" -->
<redirects xmlns="http://herbaer.de/xmlns/20140430/redirects/">
  <redirect>
    <text>Schwarzer Minirock</text>
    <text>Minirock</text>
    <relurl>s2014w16/story.xml#s24_26gsyfh0_0</relurl>
  </redirect>
  <redirect>
    <relurl>s2014w16/story.xml#s2_26gsyf5c_0</relurl>
    <text>Rotes Tanzkleid</text>
    <text>Tanzkleid</text>
  </redirect>
  <redirect>
    <relurl>s2014w27/story.xml</relurl>
    <text>Kathrin</text>
  </redirect>
</redirects>
```

redirects.rng - Sprechende URLs

Namespace	http://herbaer.de/xmlns/20140430/redirects/
Wurzelement	redirects
(foreign_att)	Attribute anderer XML-Namensräume
(anything)	<p><i>Enthalten in:</i> redirects, meta, redirect, text, relurl, kleid</p> <p>Beliebiger Inhalt</p> <p><i>Enthält:</i> (anything) (*)</p> <p><i>Enthalten in:</i> (anything), (foreign_el)</p>
(foreign_el)	<p>Elemente anderer XML-Namensräume</p> <p><i>Enthält:</i> (anything) (*)</p> <p><i>Enthalten in:</i> redirects, redirect</p>
redirects	<p>Wurzelement</p> <p><i>Enthält:</i> (foreign_att), (foreign_el), meta (*), redirect (+)</p> <p><i>Enthalten in:</i> Wurzel</p> <pre><element name="redirects"> <ref name="foreign_att"/> <interleave> <ref name="foreign_el"/> <zeroOrMore> <ref name="el_meta"/> </zeroOrMore> <oneOrMore> <ref name="el_redirect"/> </oneOrMore> </interleave> </element></pre>
meta	<p>Meta-Daten zur der Zusammenstellung der sprechenden URL</p> <p><i>Enthält:</i> Text, (foreign_att), @name</p> <p><i>Enthalten in:</i> redirects</p> <pre><element name="meta"> <ref name="foreign_att"/> <ref name="att_name"/> <text/> </element></pre>
@name	<p>Bestimmt die Bedeutung eines meta-Elements</p> <p>title</p> <p>Die Überschrift über dem Hauptinhalt</p> <p>sideimage</p> <p>URL des (Hintergrund-)Bildes der Seitenleiste</p> <p><i>Erlaubte Werte:</i> "title", "sideimage"</p> <p><i>Enthalten in:</i> meta</p>
redirect	<p>Daten zu einer sprechenden URL</p> <p><i>Enthält:</i> (foreign_att), (foreign_el), text (+), relurl, kleid (*)</p> <p><i>Enthalten in:</i> redirects</p>

	<pre><element name="redirect"> <ref name="foreign_att"/> <interleave> <ref name="foreign_el"/> <oneOrMore> <ref name="el_text"/> </oneOrMore> <ref name="el_relurl"/> <zeroOrMore> <ref name="el_kleid"/> </zeroOrMore> </interleave> </element></pre>
text	<p>Der Text zu einer sprechenden URL. Hieraus wird die URL erzeugt.</p> <p><i>Enthält:</i> Text, (foreign_att)</p> <p><i>Enthalten in:</i> redirect</p> <pre><element name="text"> <ref name="foreign_att"/> <text/> </element></pre>
relurl	<p>Das Ziel der Umleitung der sprechenden URL.</p> <p><i>Enthält:</i> Datentyp data</p> <p><i>Enthalten in:</i> redirect</p> <pre><element name="relurl"> <ref name="foreign_att"/> <data type="data"/> </element></pre>
kleid	<p>Kennung eines Kleidungsstücks</p> <p><i>Enthält:</i> Datentyp word</p> <p><i>Enthalten in:</i> redirect</p> <pre><element name="kleid"> <ref name="foreign_att"/> <data type="word"/> </element></pre>

Datei htaccess.tmp

```
# file KLEIDER/web/src/aktuell/htaccess.tmp
# Beispiel einer "Umleitungsdatei"

RewriteEngine On
RewriteBase /
RewriteRule ^schwarzer[-_]?minirock$ s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^minirock$ s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^rotes[-_]?tanzkleid$ s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]
RewriteRule ^tanzkleid$ s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]
RewriteRule ^kathrin$ s2014w27/story.xml [NC,NE,R=301,L]
RewriteRule ^schwarzer[-_]?mini s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^mini s2014w16/story.xml#s24_26gsyfh0_0 [NC,NE,R=301,L]
RewriteRule ^rotes[-_]?tanz s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]
RewriteRule ^tanz s2014w16/story.xml#s2_26gsyf5c_0 [NC,NE,R=301,L]
```

redirects_htaccess.xslt

[Quelltext]

Namensräume

Präfix	Namensraum
xml	http://www.w3.org/XML/1998/namespace
d	http://herbaer.de/xmlns/20051201/doc
rd	http://herbaer.de/xmlns/20140430/redirects/
xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	text
Encoding	utf-8

Eingebundene Stylesheets

/pool/txt.xslt - Hilfsvorlagen zur Ausgabe und Verarbeitung von Text

Muster-Vorlagen (matching templates)

Muster-Vorlage /rd:redirects

Wurzelement: Kopf der Konfigurationsdatei

Verwendete Modus:

simple
extended

Muster-Vorlage rd:text, simple

Eine Umleitung

Aufgerufene benannte Vorlagen:

txt.replacechars

Muster-Vorlage rd:text, extended

Eine erweiterte Umleitung: "kleid" oder "rock" am Ende des beschreibenden Textes wird zur Bildung der URL ignoriert.

Aufgerufene benannte Vorlagen:

txt.replacechars

txt.righthtrim

Modus

Modus simple

Die folgenden Vorlagen implementieren den Modus simple:

Muster-Vorlage rd:text, simple

Der Modus simple wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage /rd:redirects

Modus extended

Die folgenden Vorlagen implementieren den Modus extended:

Muster-Vorlage rd:text, extended

Der Modus extended wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage /rd:redirects

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="#" type="application/xml"?>
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:rd = "http://herbaer.de/xmlns/20140430/redirects/"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  version = "1.0"
>
<!--
erstellt 2014-04-30
2014-07-12 Mehrere Texte zu einer URL
-->
<xsl:include href = "/pool/txt.xslt"/>

<xsl:output method = "text" encoding = "utf-8"/>

<xsl:template match = "/rd:redirects">
  <xsl:text>RewriteEngine On
RewriteBase /
</xsl:text>
  <xsl:apply-templates select = "rd:redirect/rd:text" mode = "simple"/>
  <xsl:apply-templates select = "rd:redirect/rd:text" mode = "extended"/>
</xsl:template>

<xsl:template match = "rd:text" mode = "simple">
  <xsl:variable name = "text">
    <xsl:call-template name = "txt.replacechars">
      <xsl:with-param name = "chars" select = "'äöüÄÖÜß '" />
      <xsl:with-param name = "list" select = "'ae?,oe?,ue?,Äe?,Oe?,Ue?,ss?,-_]?'" />
    </xsl:call-template>
  </xsl:variable>
  <xsl:text>RewriteRule ^</xsl:text>
  <xsl:value-of select =
    'translate($text, 'ABCDEFGHJKLMNPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz')'
  />
  <xsl:text>$ </xsl:text>
  <xsl:apply-templates select = "../rd:relurl"/>
  <xsl:text> [NC,NE,R=301,L]
</xsl:text>
</xsl:template>

<xsl:template match = "rd:text" mode = "extended">
  <xsl:variable name = "text" select =
    'translate(., 'ABCDEFGHJKLMNPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz')'
  />
  <xsl:variable name = "len" select = "string-length($text)"/>
  <xsl:if test = "$len > 5">
    <xsl:variable name = "beg">
      <xsl:choose>
        <xsl:when test = "substring ($text, $len - 3) = 'rock'">
          <xsl:value-of select = "substring ($text, 1, $len - 4)"/>
        </xsl:when>
        <xsl:when test = "substring ($text, $len - 4) = 'kleid'">
          <xsl:value-of select = "substring ($text, 1, $len - 5)"/>
        </xsl:when>
        <xsl:otherwise/>
      </xsl:choose>
    </xsl:variable>
    <xsl:if test = "string-length ($beg) > 1">
      <xsl:text>RewriteRule ^</xsl:text>
      <xsl:call-template name = "txt.replacechars">
        <xsl:with-param name = "txt">
          <xsl:call-template name = "txt.righttrim">
            <xsl:with-param name = "txt" select = "$beg"/>
            <xsl:with-param name = "delim" select = "'-_ '"/>
          </xsl:call-template>
        </xsl:with-param>
        <xsl:with-param name = "chars" select = "'äöüÄÖÜß '" />
        <xsl:with-param name = "list" select = "'ae?,oe?,ue?,ae?,oe?,ue?,ss?,-_]?'" />
      </xsl:call-template>
      <xsl:text> </xsl:text>
      <xsl:apply-templates select = "../rd:relurl"/>
      <xsl:text> [NC,NE,R=301,L]
    </xsl:text>
  </xsl:if>
</xsl:if>
</xsl:template>

</xsl:stylesheet>

```

redirects_ht.xslt

[Quelltext]

Namensräume

Die Namensraum-Präfixe, die aus dem erzeugten Dokument ausgeschlossen sind, sind durch einen Stern (*) in der ersten Spalte gekennzeichnet.

	Präfix	Namensraum
	xml	http://www.w3.org/XML/1998/namespace
	(default)	http://www.w3.org/1999/xhtml
*	rd	http://herbaer.de/xmlns/20140430/redirects/
	l	http://herbaer.de/xmlns/20141210/localization
*	d	http://herbaer.de/xmlns/20051201/doc
*	xsl	http://www.w3.org/1999/XSL/Transform

Ausgabe (output)

Method	xml
Encoding	utf-8
Indent	no

Parameter

Parameter p_title

Der Titel des erzeugten XHTML-Dokuments

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_head

Die zweite Überschrift

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage rd:redirects

Parameter p_sideimage

Das Bild an der Seite

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_styleprefix

Präfix des Pfades von und CSS-Dateien

Select: '/style/'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Muster-Vorlage /

Parameter p_localbase

Basis des Pfades der Lokalisierungsdatei ohne das Suffix der Sprachkennung

Select: 'local.xml'

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Parameter p_local

Parameter p_lang

Kennung der Sprache

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Parameter p_local

Parameter p_local

Pfad der Lokalisierungsdatei

Select: concat(\$p_localbase, '.', \$p_lang)

Verwendete globale Parameter oder Variable:

Parameter p_localbase

Parameter p_lang

Parameter p_local

Der Parameter wird in den folgenden Toplevel-Elementen benutzt:

Parameter p_local

Variable g_l

Globale Variable

Variable g_l

Wurzelement der Lokalisierungsdatei

Select: document(\$p_local)/l:localization

Verwendete globale Parameter oder Variable:

Parameter p_local

Muster-Vorlagen (matching templates)

Muster-Vorlage /

Der HTML-Rahmen

Verwendete globale Parameter oder Variable:

Parameter p_title
Parameter p_sideimage
Parameter p_styleprefix

Muster-Vorlage rd:redirects

Zusammenstellung der sprechenden URLs als Tabelle

Verwendete globale Parameter oder Variable:

Parameter p_head

Muster-Vorlage rd:redirect

Eine Tabellenzeile zu einem oder mehreren sprechenden Verweisen

Verwendete Modus:

image

Muster-Vorlage rd:text

Ein sprechender Verweis

Muster-Vorlage rd:relurl, image

Bild zu einem sprechenden Verweis

Modus

Modus image

Die folgenden Vorlagen implementieren den Modus image:

Muster-Vorlage rd:relurl, image

Der Modus image wird in den folgenden Stylesheet-Elementen benutzt:

Muster-Vorlage rd:redirect

Quelltext

[Beschreibung]

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="/pool/xslt_ht.xslt" type="application/xml"?>
<!--
  XHTML-Darstellung der "sprechenden URLs"
  2014 Herbert Schiemann <h.schiemann@herbaer.de>
  Borkener Str. 167, D-46284 Dorsten, Germany
  Diese Datei wird unter den Bedingungen der GPL Version 3 oder
  einer neueren Version weitergegeben.
  Jede Gewährleistung ist ausgeschlossen.
  erstellt 2014-05-01
  2014-07-12 Mehrere Texte zu einer URL
  2015-04-23 Anpassung an verschiedene Lokalisierungen
-->
<xsl:stylesheet
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
  xmlns:d = "http://herbaer.de/xmlns/20051201/doc"
  xmlns:l = "http://herbaer.de/xmlns/20141210/localization"
  xmlns:rd = "http://herbaer.de/xmlns/20140430/redirects/"
  xmlns = "http://www.w3.org/1999/xhtml"
  exclude-result-prefixes = "xsl d rd"
  version = "1.0"
>

<xsl:param name = "p_title">
  <l:ph id = "kleider_von_gestern"/>
</xsl:param>

<xsl:param name = "p_head">
  <xsl:choose>
    <xsl:when test = "/rd:redirects/rd:meta[@name = 'title']">
      <xsl:value-of select = "/rd:redirects/rd:meta[@name = 'title']"/>
    </xsl:when>
    <xsl:otherwise>
      <l:ph id="empfehlungen"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:param>

<xsl:param name = "p_sideimage">
  <xsl:choose>
    <xsl:when test = "/rd:redirects/rd:meta[@name = 'sideimage']">
      <xsl:value-of select = "/rd:redirects/rd:meta[@name = 'sideimage']"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>/s2012w10/smallimg/1kjsyitt.jpg</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</xsl:param>

<xsl:param name = "p_styleprefix" select = "/style"/>

<xsl:param name = "p_localbase" select = "'local.xml'"/>

<xsl:param name = "p_lang">
  <xsl:choose>
    <xsl:when test = "not (/rd:redirects/@xml:lang)">
      <xsl:text>de</xsl:text>
    </xsl:when>
    <xsl:when test = "contains (/rd:redirects/@xml:lang, '-')">
      <xsl:value-of select = "substring-before (/rd:redirects/@xml:lang, '-')"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>de</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</xsl:param>

<xsl:param name = "p_local" select = "concat($p_localbase, '.', $p_lang)"/>

<xsl:variable name = "g_l" select = "document($p_local)/l:localization"/>

<xsl:output method = "xml" encoding = "utf-8" indent = "no"/>

```



```
<xsl:template match = "rd:relurl" mode = "image">
  <xsl:choose>
    <xsl:when test = "contains (., '/story.xml#')">

      <xsl:variable name = "story" select = "substring-before (., '/story.xml#')"/>
      <xsl:variable name = "f" select = "substring-after (., '/story.xml#')"/>
      <xsl:if test = "contains ($f, '_')">
        <xsl:variable name = "f2" select = "substring-after ($f, '_')"/>
        <xsl:if test = "contains ($f2, '_')">
          <img
            alt = "Vorschau"
            src = "/{$story}/thumbs/{substring-before ($f2, '_')}.jpg"
          />
        </xsl:if>
      </xsl:if>
    </xsl:when>
    <xsl:when test = "contains (., '/story.xml')">
      <img
        alt = "Vorschau"
        src = "{substring-before (., '/story.xml')}/thumbnail.jpg"
      />
    </xsl:when>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

redirects.css

[Quelltext]

Fenstergröße und Layout

Diese Datei legt Mindestmaße des `body`-Elements fest. Mittels Javascript wird das `body`-Element vergrößert, dass es das ganze Fenster ausfüllt.

Das Fenster ist in vier absolut positionierte "Kacheln" aufgeteilt: Links oben das Logo (ID `div_logo`), rechts oben das Titelfeld (ID `div_title`), links unten die Seitenleiste (ID `div_side`), rechts unten der Hauptinhalt (ID `div_main`).

Die anfänglichen Maße des Logo sind festgelegt, die anfänglichen Maße des Titelfeldes ergeben sich dagegen aus dem Inhalt. Javascript setzt die Höhen des Logo und des Titelfeldes beide auf das Maximum der anfänglichen Höhen und passt die Positionen der Seitenleiste und des Hauptinhalts an.

Seitenleiste

Auf der Seitenleiste soll lediglich ein Hintergrundbild zentriert angezeigt werden. Dabei sollen Abstände vom Bild zu den Kanten der Seitenleiste eingehalten werden. Ich nutze dazu ein verschachteltes `div`-Element (ID `div_side-img`). Dieses nimmt horizontal zentriert 95% der verfügbaren Breite ein.

Quelltext

[Beschreibung]

```
/*
  Index-Datei der "sprechenden Verweise"
  2014-05-02 Herbert Schiemann <h.schiemann@herbaer.de>
*/

body {
  margin: 0;
  padding: 0;
  min-width: 300px;
  min-height: 200px;
  overflow: auto;
}

/* Logo links in der Kopfleiste */
div#div_logo {
  position: absolute;
  left: 0px;
  top: 0px;
  width: 230px;
  height: 68px; /* Mindesthöhe der Titelleiste*/
  background-image: url(logo.png);
  background-position: center center;
  background-repeat: no-repeat;
  color: transparent;
  overflow: hidden;
}

/* rechtes Feld in der Kopfleiste */
#div_title {
  position: absolute;
  left: 230px;
  top: 0px;
  overflow: hidden;
}

div#div_links div { padding: 0.2em 0; }
div#div_links p {
  display: inline;
  padding: 0 1.0em 0 0.5em;
  margin: 0;
}

h1 {
  margin: 0;
  padding-bottom: 0.2em;
  padding-left: 5px;
  white-space: nowrap;
  font-family: sans-serif;
}
h2 {
  margin-left: 10px;
  font-family: sans-serif;
}

/* Feld auf der linken Seite */
div#div_side {
  position: absolute;
  left: 0;
  top: 68px;
  width: 230px;
  bottom: 0;
  overflow: hidden;
}

div#div_sideimg {
  width: 95%;
  height: 100%;
  margin: auto;
  overflow: hidden;
  background-position: center center;
  background-repeat: no-repeat;
}
```

```
/* Hauptinhalt */
div#div_main {
  position: absolute;
  top: 68px;
  left: 230px;
  right: 0;
  bottom: 0;
  background-image: url(/style/maintext.png);
  background-position: right center;
  background-repeat: no-repeat;
  overflow: auto;
}

table      { margin-left: 10px; }
td + td    { padding-left: 1em; }
```


redirects.js

[Quelltext]

Übersicht

Die Datei `redirects.js` enthält Code für die Übersicht über die "sprechenden URL".

Sie setzt voraus, dass die Javascript-Datei `common.js` geladen ist.

Diese Datei steuert die Anpassung der Darstellung an die Fenstergröße.

```
var sz_logo; // Größe des Logo
var e_body; // das body-Element
var s_title; // "Stile" der Elemente
var s_side;
var s_main;

onResize (event)
initialize ()
```

Funktionen und Daten

`sz_logo`

Die Größe des Logo (ID `div_logo`). Die Maße werden beim ersten Aufruf der Funktion `onResize` ermittelt. Wenn die anfängliche Höhe des Titelfeldes (ID `div_title`) größer ist als die Höhe des Logo, wird die Höhe bis zur Höhe des Titelfeldes vergrößert. Die Maße des Logo bleiben danach unverändert.

`e_body`

Das body-Element. Die Funktion `onResize` ermittelt seine Größe.

`s_title`

Der "Stil" des Titelfeldes (ID `div_title`). Die Funktion `onResize` passt die Größe an die neue Größe des Fensters an.

`s_side`

Der "Stil" des Seitenfeldes (ID `div_side`). Die Funktion `onResize` passt die Größe an die neue Größe des Fensters an.

`s_main`

Der "Stil" des Hauptinhalts (ID `div_main`). Die Funktion `onResize` passt die Größe an die neue Größe des Fensters an.

`onResize (event)`

Diese Funktion passt die Darstellung an die (neue) Fenstergröße an. Die Breiten des Titelfeldes (ID `div_title`) und des Hauptinhalts (ID `div_main`) reichen bis zur rechten Kante des Fensters, die Höhen des Seitenfeldes (ID `div_side`) und des Hauptinhalts reichen bis zum unteren Kante des Fensters.

`initialize ()`

Diese Funktion wird aufgerufen, sobald das XHTML-Dokument geladen ist. Sie initialisiert die Variablen, ruft erstmals `onResize` auf und ordnet `onResize` als Ereignisbehandlung für eine Änderung der Fenstergröße zu.

Quelltext

[Beschreibung]

```
// -*- coding: utf-8 -*-
// Anzeige der "sprechenden URLs"
// 2014-05-02 Herbert Schiemann <h.schiemann@herbaer.de>

var sz_logo; // Größe des Logo
var e_body; // das body-Element
var s_title; // "Stile" der Elemente mit den jeweiligen IDs
var s_side;
var s_main;

// Fenstergröße geändert
function onResize (event) {
  var _sz = getWindowSize ();
  if (_sz == null) return;
  if (sz_logo == null) {
    sz_logo = get_size (document.getElementById ("div_logo"));
    var _h = get_height (document.getElementById ("div_title"));
    if (_h > sz_logo.height) {
      sz_logo.height = _h;
      set_position (s_logo, 0, 0, sz_logo.width, sz_logo.height);
    }
  }
  var _sz2 = get_size (e_body);
  if (_sz2.height > _sz.height) _sz.height = _sz2.height;
  if (_sz2.width > _sz.width) _sz.width = _sz2.width;
  set_position (s_title, sz_logo.width, 0, _sz.width - sz_logo.width, _sz.height);
  set_position (
    s_side,
    0, sz_logo.height, sz_logo.width,
    _sz.height - sz_logo.height
  );
  set_position (
    s_main,
    sz_logo.width, sz_logo.height,
    _sz.width - sz_logo.width, _sz.height - sz_logo.height
  );
} // onResize

function initialize () {
  e_body = document.getElementsByTagName ("body")[0]; // das body-Element
  s_logo = document.getElementById ("div_logo").style;
  s_title = document.getElementById ("div_title").style;
  s_side = document.getElementById ("div_side").style;
  s_main = document.getElementById ("div_main").style;
  onResize ();
  window .addEventListener ("resize", onResize, false);
} // initialize
```

redirects

[Quelltext]

Übersicht

```
redirects --help | --version
```

```
redirects [ --verbose ... | --no_verbose ] [ --overwrite | --no_overwrite ]  
[ --srcdir SRCDIR ] [ --stsrc STSRC ] [ --rdsrc RDSRC ] [ --aktdir AKTDIR ]  
[ --htaccfile HTACCFILE ] [ --ixfile IXFILE ] [ --tempdir TMPDIR ]  
--htaccess --xhtml  
--no_htaccess --no_xhtml
```

Optionen

Alle Optionen haben Vorgabewerte. Der Befehl **redirects --help** zeigt die Vorgabewerte an.

--help

Gibt eine kurze Hilfe und die Vorgabe-Werte zu allen Optionen aus.

--version

Gibt kurze Hinweise zum Programm und die Version aus.

--verbose

Erstellte Dateien und Meldungen über den Programmablauf werden nach STDOUT ausgegeben.

--no_verbose

Diese Option hebt die Wirkung der Option **--verbose** auf.

--overwrite

Existierende Dateien werden überschrieben.

--no_overwrite

Existierende Dateien werden weder gelöscht noch ersetzt.

--srcdir SRCDIR

Das Quelltext-Verzeichnis (XSLT-, Javascript- und CSS-Dateien).

--stsrc STSRC

Die Javascript- und CSS-Dateien werden auf dieselbe Weise in die XSLT-Datei eingebunden wie zur Darstellung einer Bildergeschichte. Es werden dieselben Dateien `styleincl_step_1.xslt` `styleincl_step_2.xslt`, `clean_css.pl` und `clean_js.pl` benutzt. Diese Dateien liegen im Verzeichnis `STSRC`.

--rdsrc RDSRC

Die XML-Datei mit den Texten und den "festen" URL zu den sprechenden URL. Die Datei `redirects.rng` beschreibt den Aufbau. Aus dieser Datei werden die Server-Konfigurationsdatei (`HTACCFILE`) und die XHTML-Darstellung der Zusammenstellung (`IXFILE`) erzeugt.

--aktdir *AKTDIR*

Das Server-Unterverzeichnis für die "sprechenden URL". Dieses Verzeichnis enthält die Dateien *HTACCFILE* und *IXFILE* und bestimmt das Präfix der sprechenden URL.

--htacfile *HTACCFILE*

Der Pfad der Server-Konfigurationsdatei (htaccess-Datei) zu den Umleitungen der "sprechenden ULR".

--ixfile *IXFILE*

Der Pfad der XHTML-Datei, die die sprechenden URL zusammenstellt.

--tempdir *TEMPDIR*

Verzeichnis für Zwischendateien ("verdichtete" Javascript- und CSS-Dateien, XSLT-Dateien mit eingebundenen Javascript- und CSS-Dateien).

--htaccess

Erzeugt aus der Datei *RDSRC* die Server-Konfigurationsdatei *HTACCFILE*.

--xhtml

Erzeugt aus der Datei *RDSRC* die Zusammenstellung *IXFILE* als XHTML.

Aus den Dateien *redirects.js* und *redirects.css* werden Kommentare und unnötige Leerzeichen entfernt. Die so "verdichteten" Dateien werden in die Datei *redirects_ht.xslt* eingebunden. Die so entstandene Transformation erzeugt aus *RDSRC* die Datei *IXFILE*. Zusätzlich wird ein gzip-komprimierte Version mit der zusätzlichen Dateinamensendung *.gz* erstellt.

--no_*

Wenn keine der "Aktionsoptionen" `--htaccess` oder `--xhtml` genutzt wird, können auch die Optionen mit dem Präfix "no_" (`--no_htaccess` und `--no_xhtml`) genutzt werden. Diese schließen die zugehörige Aktion aus. Alle nicht ausgeschlossenen Aktionen werden ausgeführt.

Aufgerufene Programme

xsltproc

Wendet XSLT-Transformationen an.

gzip

Komprimiert Dateien.

Quelltext

[Beschreibung]

```
#!/bin/bash
# -*- coding:utf-8 -*-
# 2014-04-30 Herbert Schiemann <h.schiemann@herbaer.de>
# aktuelle URL-Umleitungen pflegen

# Zähler, Variable, Aktionen
declare_vars ()
{
    # Ein Leerzeichen als Wert bedeutet, dass Positionsargumente verarbeitet werden
    _argv="";

    # Suchpfad für rc-Dateien, : - getrennte Liste von Verzeichnispfaden
    # Falls leer, wird die Option --rc nicht speziell behandelt
    # g_configpath=$HOME/etc/redirects.:. ;

    # Zähler
    g_counters=" \
        verbose \
        overwrite ";

    # Variable
    g_variables=" \
        srcdir \
        stsrc \
        rdsrc \
        aktdir \
        htaccfile \
        ixfile \
        tempdir ";

    # Aktionen
    g_actions=" \
        htaccess \
        xhtml ";
} # declare_vars

# setzt Vorgabe-Werte
set_defaults ()
{
    local w=${realpath $0};
    w=${w%/src/aktuell/redirects};
    [[ -n "$verbose" ]] || verbose=1 ;
    [[ -n "$rdsrc" ]] || rdsrc="$w/supplement/redirects.xml" ;
    [[ -n "$aktdir" ]] || aktdir="$w/docroot/aktuell" ;
    [[ -n "$htaccfile" ]] || htaccfile="$aktdir/.htaccess" ;
    [[ -n "$ixfile" ]] || ixfile="$aktdir/index.xhtml." ;
    [[ -n "$srcdir" ]] || srcdir="$w/src/aktuell" ;
    [[ -n "$stsrc" ]] || stsrc="$w/src/style" ;
    [[ -n "$tempdir" ]] || tempdir="$w/web/temp" ;
} # set_defaults

# Zeigt eine kurze Hilfe an
show_help ()
{
    local cmd=${0#*/} ;
    set_defaults ;
    cat << .HELP ;
$cmd --version
$cmd --help
$cmd ([Aktion] | [Option])*

Aktionen
--htaccess      Konfigurationsdatei für URL-Umleitungen erzeugen
--xhtml        XHTML-Datei zu Umleitungen erzeugen

Einstellungen
--[no_]verbose      Erhöht den Umfang der Ausgabe des Scripts ($verbose)
--[no_]overwrite    Existierende Dateien überschreiben ($overwrite)
--rdsrc RDSRC       XML-Quelldatei der umzuleitenden URLs
                   ($rdsrc)
--aktdir AKTDIR     Serververzeichnis für umzuleitenden URLs
                   ($aktdir)
--htaccfile HTACCFIPE Pfad der Konfigurationsdatei
                   ($htaccfile)
--ixfile IXFILE     Pfad der Indexdatei
                   ($ixfile)
--srcdir SRCDIR     Verzeichnis der Skripte
                   ($srcdir)
--stsrc STSRC       weiteres Skriptverzeichnis zum Stil
                   ($stsrc)
--tempdir TEMPDIR   Verzeichnis für temporäre Dateien
                   ($tempdir)
.HELP
} # show_help
```

```

# Zeigt die Version an
show_version ()
{
    cat << .VERSION ;
}
$0
aktuelle URL-Umleitungen pflegen
2014-04-20, Herbert Schiemann, h.schiemann@herbaer.de
GPL Version 2 oder neuer
.VERSION
} # show_version

# Variable und Zähler initialisieren
init_vars () {
    local v;
    declare_vars ;
    for v in $g_counters $g_variables $g_actions; do
        eval "$v=" ;
    done;
} # init_vars

# Argumente verarbeiten
read_args ()
{
    local wd ;
    local lastwd ;
    local var ;
    local ok ;

    has_actions=0 ;
    for wd in "$@"; do
        if [[ "$lastwd" = "--" ]]; then
            _argv="$argv $wd";
        elif [[ -n "$lastwd" ]]; then
            if [[ "$wd" =~ ^[\ a-zA-Z0-9./_#-]+$ ]]; then
                if [[ "$lastwd" == "rc" && -n "$g_configpath" ]]; then
                    if ! read_configuration $wd; then
                        (( verbose )) && echo "Kann Konfiguration $wd nicht lesen" ;
                        exit 10 ;
                    fi ;
                else
                    ok=0 ;
                    for var in $g_variables; do
                        if [[ "$var" == "$lastwd" ]]; then
                            (( ++ok )) ;
                            eval "$var=\"\$wd\"";
                            break ;
                        fi ;
                    done ;
                    if (( ! ok )); then
                        (( verbose )) && echo "Unbekannte Option --$lastwd $wd" ;
                        exit 11 ;
                    fi ;
                fi ;
            else
                (( verbose )) && echo "Ungültiger Optionswert --$lastwd $wd" ;
                exit 12;
            fi;
            lastwd= ;
        else
            case "$wd" in
                --version )
                    show_version ;
                    exit 0 ;
                    ;;
                --help )
                    show_version ;
                    show_help ;
                    exit 0 ;
                    ;;
                -- )
                    if [[ -n "$argv" ]]; then
                        lastwd--;
                        continue;
                    else
                        (( verbose )) && echo "Ungültige Option $wd" ;
                        exit 13 ;
                    fi ;
                    ;;
                -* )
                    if [[ "$wd" =~ ^--[a-z][a-z0-9_]*$ ]]; then
                        lastwd=${wd#--} ;
                        ok=0 ;
                        for var in $g_counters ; do
                            if [[ "$lastwd" == $var ]] ; then
                                eval "(( ++$lastwd ))" ;
                            elif [[ "$lastwd" == "no_$var" ]]; then
                                eval "${lastwd#no_}=0" ;
                            else
                                continue;
                            fi;
                            (( ++ok )) ;
                            break ;
                        done;
                        if (( !ok )); then

```

```

        for var in $g_actions; do
            if [[ "$lastwd" == "$svar" ]]; then
                eval "(( ++$svar ))" ;
                (( ++ok ));
                has_actions=1;
                break;
            elif [[ "$lastwd" == "no_$svar" ]]; then
                eval "(( ++no_$svar ))" ;
                (( ++ok ));
                break;
            fi;
        done;
    fi;
    (( ok )) && lastwd=;
else
    (( verbose )) && echo "Ungültige Option $wd" ;
    exit 14 ;
fi ;
;;
* )
if [[ -n $argv ]]; then
    _argv="$argv $wd";
else
    (( verbose )) && echo "Ungültige Option $wd" ;
    exit 15 ;
fi;
;;
esac ;
fi ;
done ;
if [[ -n $lastwd && "$lastwd" != "--" ]]; then
    (( verbose )) && echo "Unverarbeitete Option --$lastwd";
    exit 16 ;
fi ;
[[ "$argv" =~ ^[[:space:]]+$ ]] && _argv="" ;
} # read_args

# Aktionen ausführen
run_actions ()
{
    local act ;
    for act in $g_actions; do
        eval "(( ! has_actions && ! no_$act || $act )) && process_$act";
    done;
} # run_actions

# show_variables VARNAME1 VARNAME2
# Werte der Variablen anzeigen
show_variables ()
{
    local v ;
    for v in $g_counters $g_variables $g_actions $!; do
        eval "echo \"\$v = \${$v}\"" ;
    done;
} # show_variables

# Können die Eingabedateien gelesen werden?
# check_infiles first/path/to/file path/to/second_file ;
check_infiles ()
{
    local f ;
    for f in "$@"; do
        if [[ ! -f "$f" ]]; then
            (( verbose )) && echo "\"$f\" ist keine gewöhnliche Datei";
            return 1;
        fi;
        if [[ ! -s "$f" ]]; then
            (( verbose )) && echo "\"$f\" ist leer";
            return 1;
        fi;
        if [[ ! -r "$f" ]]; then
            (( verbose )) && echo "Kann Datei \"$f\" nicht lesen";
            return 1;
        fi;
    done;
    return 0;
} # check_infiles

```

```

# Können die Ausgabedateien erstellt werden?
# erstellt fehlende Verzeichnisse und löscht existierende Dateien
# nach Maßgabe der Variablen overwrite
# check_outfiles first/path/to/file path/to/second_file ;
check_outfiles ()
{
    local fp;
    local dir;
    local verb;
    (( verbose )) && verb=--verbose ;
    for fp in "$@"; do
        if [[ ! -e $fp ]]; then
            dir=${fp%/*};
            if [[ -n $dir && ! -e $dir ]]; then
                mkdir -p $verb $dir ;
                if [[ ! -d $dir ]]; then
                    (( verbose )) && echo "$dir ist kein Verzeichnis";
                    return 1;
                fi;
            fi;
            elif [[ -d $fp ]]; then
                (( verbose )) && echo "$fp ist ein Verzeichnis";
                return 1;
            elif (( overwrite )); then
                (( verbose )) && echo "lösche $fp";
                rm $fp;
            else
                (( verbose )) && echo "$fp existiert";
                return 1;
            fi;
            (( verbose )) && echo "$fp";
        done;
    return 0;
} # check_outfiles

# gzip-komprimierte Datei(en) hinzufügen
add_gzip ()
{
    local f;
    local g;
    for f in "$@"; do
        [[ -f $f ]] || continue;
        g=${f%.}.gz ;
        [[ -f $g ]] && rm $g;
        [[ -e $g ]] && continue;
        (( verbose )) && echo "erstelle $g";
        gzip --best --stdout $f > $g ;
    done ;
} # add_gzip

# Hilfsfunktion: temporäre Javascript und CSS-Dateien erzeugen
# proc_tempfiles subdir
proc_tempfiles ()
{
    local sd=$1 ;
    for s in $srcdir/*.js ; do
        [[ -f $s ]] || continue;
        o=${d}/${s#$srcdir/} ;
        check_outfiles $o \
            && $stsrc/clean_js.pl --in $s --out $o ;
    done;
    for s in $srcdir/*.css ; do
        [[ -f $s ]] || continue;
        o=${d}/${s#$srcdir/} ;
        check_outfiles $o \
            && $stsrc/clean_css.pl --in $s --imageprefix '${imageprefix}' --out $o ;
    done;
} # proc_tempfiles

# Konfigurationsdatei für URL-Umleitungen erzeugen
process_htaccess ()
{
    (( verbose )) && echo "process_htaccess" ;
    local t=$srcdir/redirects_htaccess.xslt ;
    check_infiles $t $rdsrc || return 1;
    check_outfiles $htaccfile || return 1;
    xsltproc -o $htaccfile $t $rdsrc ;
} # process_htaccess

```



```
# XHTML-Datei zu Umleitungen erzeugen
process_xhtml ()
{
  (( verbose )) && echo "process_xhtml" ;
  local t=$srcdir/redirects_ht.xslt ;
  check_infiles $t $rsrc || return 1;
  check_outfiles $ixfile || return 1;
  local td="$tempdir/(date +%Y%m%d%H%M%S%N)" ;
  local t2=$td/redirects_ht.xslt ;
  proc_tempfiles $td ;
  xsltproc \
    --stringparam p_tmpprefix $td/ \
    --stringparam p_shortids none \
    $stsrc/styleincl_step_1.xslt $t \
    | xsltproc -o $t2 --xinclude $stsrc/styleincl_step_2.xslt - ;
  xsltproc \
    --stringparam p_localbase $srcdir/local.xml \
    -o $ixfile $t2 $rsrc ;
  add_gzip $ixfile ;
  rm --recursive $td;
} # process_xhtml

# Sicherheit
export PATH=/bin:/usr/bin:$HOME/bin ;
IFS=$' \t\n' ;
set -o noclobber ; # existierende Dateien werden nicht überschrieben
shopt -s extglob nullglob ;

init_vars ;
read_args "$@" ;
set_defaults ;

(( verbose > 1 )) && show_variables " HOME " ;

run_actions ;
exit 0;
```