



Dokumentacja administracyjno- programistyczna

Platformy Technologicznej
epodreczniki.pl
w wersji 3.0

Poznań, 30 listopada 2015 r.

Spis treści

Instrukcje instalacji - Aplikacje natywne	4
Android.....	4
iOS	4
Windows 8.....	4
Aplikacja offline dla Windows	4
Informacja o aktualnych wersjach aplikacji mobilnych w portalu	5
Instrukcje instalacji - Cassandra	5
Instrukcje instalacji - Dodatek MS Word i konwerter OOXML.....	7
Przygotowanie konwertera OOXML.....	7
Wymagania.....	7
Zależności	8
Konfiguracja i kompilacja.....	8
Przygotowanie Dodatku MS Word	10
Wymagania.....	10
Konfiguracja.....	10
Kompilacja	11
Moduł wyszukiwania treści	11
Wymagania wstępne.....	11
Punkt starowy - instalacja wymagań na przykładzie Centos 6	11
Instalacja SOLR	12
Procedura instalacji:.....	12
Instalacja Elastic	13
Procedura instalacji:.....	13
Instalacja modułu indeksującego dane	14
Kompilacja indeksera SOLR	14
Deployment indeksera	14
Konfiguracja indeksu dla indeksera.....	15
Konfiguracja dostępu do narzędzi wyszukiwania.....	15
Checkpoint.....	16
Konfiguracja platformy.....	17
Wyszukiwanie pełnotekstowe.....	17
Wyszukiwanie oparte na atrybutach.....	17
Indeksowanie danych.....	17



Indeksowanie ręczne.....	17
Indeksowanie automatyczne.....	17
Indeksowanie okresowe.....	18
Portal.....	18
Konfiguracja epodreczniki-edytora.....	18
Katalog epodreczniki-edytora należy umieścić w tym samym katalogu, w którym znajdzie się epodreczniki-portal.....	19
Konfiguracja i uruchomienie epodreczniki-portal.....	19
RabbitMQ.....	21
Instalacja zależności dla RabbitMQ.....	21
Instalacja RabbitMQ.....	21
Konfiguracja RabbitMQ.....	21
Repozytorium AV.....	22
Wymagania.....	22
Instalacja.....	23
Repozytorium Treści.....	24
Przygotowanie środowiska.....	24
Budowanie dLibry.....	25
Budowanie e-podręczniki.....	25
Uruchamianie i zatrzymywanie.....	28
Serwer Kodujący.....	28
Instalacja.....	28
Uruchomienie.....	29
Synchronizator KZD.....	30
Instalacja.....	30
Uruchomienie.....	31
Transformacje.....	32

Instrukcje instalacji - Aplikacje natywne

Android

1. Zainstalować Eclipse
<http://www.eclipse.org>
2. Zainstalować wtyczkę ADT
<http://developer.android.com/sdk/installing/installing-adt.html>
3. Przy pomocy SDK Managera zainstalować Android SDK 5.0.1 (API 21)
4. Zaimportować projekt
5. W klasie `pl.epodreczniki.util.TransferHelper` ustawić poprawną wartość dla stałej `API_BASE`. Serwer o adresie `API_BASE` powinien na zapytanie HTTP GET odpowiadać metadanymi e-podręczników w formacie JSON

iOS

1. Pobierz Xcode z AppStore
2. Kliknij dwukrotnie na plik projektu: `EpodrecznikiApp.xcodeproj`
3. Otwórz plik `EPConstants.h` i podaj URL do api (druga linia, zmienna "API_BASE")
4. W Xcode wybierz "Preferences" potem "Accounts" i dodaj swoje konto developerskie
5. Opcjonalnie zmień "bundle identifier" (domyślna wartość to `pl.edu.ore.EpodrecznikiLite`)
6. Uruchom projekt: Menu "Product" -> "Run"

Windows 8

1. W pliku `epodreczniki\DataModel\CollectionsDataSource.cs` zmienić wartość statycznego pola `Host` w klasie `CollectionsDataSource` na adres, pod którym będą znajdowały się metadane e-podręczników w formacie json
2. Zainstalować Microsoft Visual Studio w wersji 2013, Update 3 lub nowszej
3. Otworzyć w Visual Studio plik: "EP-Win8.1.sln": poprzez dwukrotne kliknięcie na plik, bądź poprzez wybranie w głównym menu Visual Studio, opcji: FILE/Open/Project/Solution
4. Uruchomić kompilację poprzez wybranie, w głównym menu Visual Studio, opcji: BUILD/REBUILD SOLUTION

Aplikacja offline dla Windows

1. Zainstalować Microsoft Visual Studio w wersji 2013, Update 3 lub nowszej
2. Otworzyć w Visual Studio plik: "ePodrecznikiDesktop.sln": poprzez dwukrotne kliknięcie na plik, bądź poprzez wybranie, w głównym menu Visual Studio, opcji: FILE/Open/Project/Solution

3. Uruchomić kompilację poprzez wybranie, w głównym menu Visual Studio, opcji:
BUILD/REBUILD SOLUTION

Informacja o aktualnych wersjach aplikacji mobilnych w portalu

Wydanie nowej wersji którejś z aplikacji mobilnych wymaga dokonania aktualizacji w konfiguracji komponentu portal. Aktualnie wydane numery wersji są przechowywane w pliku epodreczniki-portal/portal/static/other/mobile-versions.json.

Instrukcje instalacji - Cassandra

1. Instalacja najnowszej wersji Javy Oracle 8 (CentOS):

```
cd /opt/  
wget --no-cookies --no-check-certificate --header "Cookie:  
gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie"  
"http://download.oracle.com/otn-pub/java/jdk/8u66-b17/jdk-8u66-linux-x64.tar.gz"  
tar xzf jdk-8u66-linux-x64.tar.gz  
cd /opt/jdk1.8.0_66/  
alternatives --install /usr/bin/java java /opt/jdk1.8.0_66/bin/java 2  
alternatives --config java  
java -version
```

2. Dodajemy repozytorium DataStax Community do '/etc/yum.repos.d/datastax.repo':

Taka ma być zawartość pliku:

```
[datastax]  
name = DataStax Repo for Apache Cassandra  
baseurl = http://rpm.datastax.com/community  
enabled = 1  
gpgcheck = 0
```

3. Instalujemy pakiety Cassandra 2.1.11:

```
sudo yum install dsc21-2.1.11-1 cassandra21-2.1.11-1  
sudo yum install cassandra21-tools-2.1.11-1
```

4. Tworzymy katalogi dla cassandry:

```
w katalogu /opt:  
mkdir /opt/cassandra  
mkdir /opt/cassandra/data  
mkdir /opt/cassandra/commitlog  
mkdir /opt/cassandra/saved_caches
```

5. I zmieniamy właściciela katalogów na cassandra:cassandra:

```
chown -R cassandra:cassandra /opt/cassandra
```

6. Edytujemy plik konfiguracyjny cassandry:

```
vi /etc/cassandra/conf/cassandra.yaml
```

i zmieniamy wpisy podobnie jak niżej (adres IP będzie oczywiście inny w zależności od lokalnej konfiguracji sieci).

Zakładamy, że portal będzie się łączył po lokalnym adresie: 192.168.1.55, porty pozostają standardowe.

```
cluster_name: 'EpoCluster'  
authenticator: PasswordAuthenticator  
authorizer: CassandraAuthorizer  
data_file_directories: /opt/cassandra/data  
commitlog_directory: /opt/cassandra/commitlog  
saved_caches_directory: /opt/cassandra/saved_caches  
seeds: "192.168.1.55"  
listen_address: 192.168.1.55  
rpc_address: 192.168.1.55  
endpoint_snitch: GossipingPropertyFileSnitch
```

Inne ustawienia pozostawiamy bez zmian.

7. Uruchamiamy usługę i sprawdzamy czy działa poprawnie:

```
service cassandra start
```

```
service cassandra status  
cassandra (pid 29915) is running...
```

Logi usługi: /var/log/cassandra - tutaj można sprawdzić czy nie ma błędów po uruchomieniu

8. Użytkownicy

Po instalacji jest dostępny administrator: cassandra:cassandra

Tworzymy użytkowników dla portalu epo:

Logujemy się na konsole cqlsh:

```
cqlsh 192.168.1.55 -u cassandra -p cassandra
```

Można zmienić hasło dla administratora:



```
alter user cassandra with password 'newpassword';
```

Tworzymy superusera dla portalu epo:

```
create user epoadmin with password 'epoadmin' superuser;
```

oraz zwykłego użytkownika:

```
create user epo with password 'epo';
```

Użytkownik epoadmin jest superuser'em i zakładamy, że może wszystko robić, natomiast dla użytkownika epo nadajemy uprawnienia dla poszczególnych tabel.

Będzie to zrobione w skrypcie ze schematem danych - portal się tym już zajmuje.

Gdyby był problem z uprawnieniami można ręcznie dodać je dla poszczególnych keyspace'ow lub poszczególnych tabel:

```
GRANT select ON KEYSPACE nazwa-bazy-epo TO epo;
```

```
GRANT modify ON KEYSPACE nazwa-bazy-epo TO epo;
```

9. Otwieramy porty na firewall:

7000 Cassandra inter-node cluster communication.

7001 Cassandra SSL inter-node cluster communication.

7199 Cassandra JMX monitoring port.

9042 Cassandra client port.

9160 Cassandra client port (Thrift).

W CentOS w pliku `/etc/sysconfig/iptables` dodajemy (np. pod linią ssh z portem 22):

```
-A INPUT -p tcp --dport 9160 -j ACCEPT
```

```
-A INPUT -p tcp --dport 9042 -j ACCEPT
```

```
-A INPUT -p tcp --dport 7199 -j ACCEPT
```

```
-A INPUT -p tcp --dport 7000 -j ACCEPT
```

```
-A INPUT -p tcp --dport 7001 -j ACCEPT
```

```
service iptables restart
```

i sprawdzamy:

```
iptables -L -n
```

Instrukcje instalacji - Dodatek MS Word i konwerter OOXML

Przygotowanie konwertera OOXML

Wymagania

- System operacyjny Windows w wersji 7 lub nowszy,

- Java w wersji 7 lub nowszej,
- serwer WWW lub przestrzeń dostępna przez HTTP,
- ActiveState PerlApp 5.3.0,
- ActiveState Perl 5.8.9.

Zależności

- 7-ZIP (9.20 wersja linii komend),
- JEuclid 3.1.9,
- Apache FOP 1.1 bin,
- Saxon 9.1.0.8j,
- Docbook XSLT 1.77.1

Konfiguracja i kompilacja

1. Ustalić:

- adres URL repozytorium modułów (adres wyspecyfikowanego w wymaganiach serwera WWW lub przestrzeni dostępnej przez HTTP),
- nazwę domenową Otwartego Repozytorium Treści,
- hasło zdalne - wykorzystywane do deszyfrowania pakietów pobieranych przez konwerter z repozytorium,
- hasło lokalne - wykorzystywana do zabezpieczania lokalnych plików konfiguracyjnych,
- hasło zip - wykorzystywane dla modułów składających się z wielu plików.

2. Wygenerować metryki fontów (w osobnym katalogu, zgodnie z dokumentacją FOP):

- wypakować z archiwum Apache FOP 1.1 bin następujące pliki jar:
 - avalon-framework-4.2.0.jar;
 - commons-io-1.3.1.jar;
 - commons-logging-1.0.4.jar;
 - fop.jar;
 - xmlgraphics-commons-1.5.jar,
- uruchomić (w tym samym katalogu) następujące instrukcje:
- `java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\arial.ttf arial.ttf.xml`

- `java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\ARIALBD.ttf ARIALBD.ttf.xml`
- `java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\ARIALBI.ttf ARIALBI.ttf.xml`
- `java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\ARIALI.ttf ARIALI.ttf.xml`
- `java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\times.ttf times.ttf.xml`
- `java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\timesbd.ttf timesbd.ttf.xml`
- `java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\timesbi.ttf timesbi.ttf.xml`

`java -cp avalon-framework-4.2.0.jar;commons-io-1.3.1.jar;commons-logging-1.0.4.jar;fop.jar;xmlgraphics-commons-1.5.jar org.apache.fop.fonts.apps.TTFReader c:\WINDOWS\Fonts\timesi.ttf timesi.ttf.xml`

- umieścić wygenerowane metryki w katalogu repository/modules/FOP_UPDATE/fop-fonts.

3. Dokonać konfiguracji:

- executable/konwerter_OOXML.pl - ustawić adres URL repozytoriów modułów, hasło zdalne, hasło lokalne,
- repository/build_repo.cfg - ustawić adres URL repozytoriów modułów, hasło zdalne, hasło zip,
- repository/modules/MAIN_MODULE.pl - ustawić hasło zip oraz fonty,
- modules/XSLT_PACKAGE/config.xml - ustawić nazwę domenową Otwartego Repozytorium Treści.

4. Skompilować konwerter:

5.

- wypakować zawartość pakietu 7-ZIP do katalogu executable,
- uruchomić:
- `cd executable`

COMPILE.bat

6. Przygotować moduły dla repozytorium:
 - umieścić w katalogu repository/dist następujące pakiety (w formie archiwów zip):
 - JEuclid 3.1.9,
 - Apache FOP 1.1 bin,
 - Saxon 9.1.0.8j,
 - Docbook XSLT 1.77.1,
 - uruchomić:
 - cd repository

build_repo.pl

7. Stworzyć repozytorium modułów:
 - udostępnić zawartość katalogu repository/dist pod adresem URL repozytorium modułów.

Przygotowanie Dodatku MS Word

Wymagania

- System operacyjny Windows w wersji 7 lub nowszy,
- Microsoft Visual Studio 2013 (pełna wersja) lub nowszy (ze wszystkimi dostępnymi składnikami, w tym Microsoft Visual C# 2013),
- Office Developer Tools for Visual Studio 2013 lub nowszy
- Microsoft .NET Framework w wersji 4.5.

Minimalne wymagania sprzętowe to procesor Intel Core i7-2760QM 2,4 GHz, 8 GB pamięci operacyjnej, 10 GB miejsca na dysku.

Konfiguracja

1. Przekopiować eprt-tools (katalog lib oraz pliki config.properties) do katalogu Apps/DirRetrieverAndPubUploader
2. Otworzyć program Visual Studio 2013
3. W ramach Visual Studio otworzyć dodatek do Worda (Menu Plik -> Otwórz -> Projekt/Rozwiązanie) wskazać plik EP_WordPlugin.sln
4. Ustawić wartości zmiennych we właściwościach projektu: Menu Projekt -> Właściwości EP_WordPlugin -> Zakładka "Ustawienia":
 - URL_KONWERTER - URL pod którym dostępny jest główny plik wykonywalny konwertera OOXML,

- URL_KONWERTER_BACKUP - URL pod którym dostępna jest zapasowa kopia głównego pliku wykonywalnego konwertera OOXML,
- URL_OTWARTE_REPOZYTORIUM_TRESCI - adres bazowy (domena wraz z protokołem) Otwartego Repozytorium Treści.

Kompilacja

1. Otworzyć program Visual Studio 2013
2. W ramach Visual Studio otworzyć dodatek do Worda (Menu Plik -> Otwórz -> Projekt/Rozwiązanie) wskazać plik EP_WordPlugin.sln
3. Skompilować projekt: Menu Kompiluj -> Kompiluj rozwiązanie
4. Opublikować: Menu Projekt -> Właściwości EP_WordPlugin -> Zakładka publikuj -> Klawisz "Publish Now"

Moduł wyszukiwania treści

Wymagania wstępne

- Java jdk 1.8.0_40
- Maven 3.3.3
<https://maven.apache.org/download.cgi>
<http://www.eu.apache.org/dist/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.zip>
<http://www.eu.apache.org/dist/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.tar.gz>

Punkt starowy - instalacja wymagań na przykładzie Centos 6

Poniższe kroki są opcjonalne w przypadku spełnienia wymagań dotyczących zainstalowania na maszynie środowiska Oracle Java i Apache Maven.

Instalacja wget:

```
sudo yum install wget
```

Instalacja Oracle Java SDK

```
wget --no-cookies --no-check-certificate --header "Cookie:
gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie"
"http://download.oracle.com/otn-pub/java/jdk/8u60-b27/jdk-8u60-linux-x64.rpm"
```

```
sudo yum localinstall jdk-8u60-linux-x64.rpm
```

```
rm ~/jdk-8u60-linux-x64.rpm
```

Instalacja Apache Maven

```
wget http://www.eu.apache.org/dist/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.tar.gz
```

```
tar -xvf apache-maven-3.3.3-bin.tar.gz
```

```
sudo mkdir -p /opt/maven/3.3.3
sudo mv apache-maven-3.3.3/* /opt/maven/3.3.3/
sudo ln -s /opt/maven/3.3.3/ /opt/maven/current
vi ~/.bash_profile
##### MAVEN 3.3.3 #####
export MAVEN_HOME=/opt/maven/current
export PATH=$PATH:$MAVEN_HOME/bin
##### MAVEN 3.3.3 #####
source ~/.bash_profile
mvn -version
```

Instalacja SOLR

SOLR dostępny jest bezpośrednio ze strony produktu (w zależności od wersji systemu):

- <https://archive.apache.org/dist/lucene/solr/4.10.4/solr-4.10.4.tgz>
- <https://archive.apache.org/dist/lucene/solr/4.10.4/solr-4.10.4.zip>

Dodatkowe biblioteki analizy leksykalnej:

- <http://central.maven.org/maven2/org/apache/solr/solr-analysis-extras/4.10.4/solr-analysis-extras-4.10.4.jar>
- <http://central.maven.org/maven2/org/apache/lucene/lucene-analyzers-stempel/4.10.4/lucene-analyzers-stempel-4.10.4.jar>

Powyższe biblioteki należy umieścić w katalogu aplikacji webowej SOLR po jego zainstalowaniu (<SOLR HOME>/example/solr-webapp/WEB-INF/lib/)

Procedura instalacji:

1. Pobranie paczki produktu
2. cd ~

```
wget https://archive.apache.org/dist/lucene/solr/4.10.4/solr-4.10.4.tgz
```

3. Rozpakowanie paczki
4. cd ~

```
tar xvf solr-4.10.4.tgz
```

5. Przygotowanie repozytoriów wyszukiwania pełnotekstowego
6. cd ~
7. cp -r solr-4.10.4/example/solr/collection1 solr-4.10.4/example/solr/epub
8. echo "name=epub" > solr-4.10.4/example/solr/epub/core.properties

9. `cp -r solr-4.10.4/example/solr/collection1 solr-4.10.4/example/solr/epub_ac`
10. `echo "name=epub_ac" > solr-4.10.4/example/solr/epub_ac/core.properties`
11. `cp -r solr-4.10.4/example/solr/collection1 solr-4.10.4/example/solr/epub_ds`
`echo "name=epub_ds" > solr-4.10.4/example/solr/epub_ds/core.properties`
12. Przygotowanie skryptów uruchomieniowych SOLR
Skrypty uruchomieniowe umieszczone w katalogu instalacyjnym pozwolą na szybkie uruchomienie SOLR w późniejszym czasie.
Przykładowy plik uruchomieniowy `runSolr.bat` (dla systemów Windows)

runSolr.bat

```
@echo off
cd solr-4.10.4\example\
set JVM_ARGS="-Xmx1024m -XX:MaxPermSize=256m"
java -jar start.jar -OPTIONS=jsp -Xmx1024m -XX:MaxPermSize=256m
```

Przykładowy plik uruchomieniowy `runSolr.sh`

runSolr.sh

```
#!/bin/bash
cd solr-4.10.4/example
java -jar start.jar -OPTIONS=jsp -Xmx1024m -XX:MaxPermSize=256m
```

Instalacja Elastic

Elastic pobrany powinien być bezpośrednio z witryny produktu:

<https://www.elastic.co/downloads/past-releases/elasticsearch-1-7-3>

Pobierane odpowiedniej paczki dla systemu:

- <https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.3.tar.gz>
- <https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.3.zip>

Procedura instalacji:

1. Pobranie paczki produktu
2. `cd ~`

```
wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-1.7.3.tar.gz
```

3. Rozpakowanie paczki
4. `cd ~`

```
tar xvf elasticsearch-1.7.3.tar.gz
```

Instalacja modułu indeksującego dane

Kompilacja indeksera SOLR

1. Pobranie kodów źródłowych indeksera i kompilacja
2. `cd ~`
3. `cd epodreczniki-wyszukiwanie`
4. `cd PublicationParser`
5. `mvn compile`
6. `mvn install`
7. `cd ../SOLRIndexerWebapp/`
8. `mvn compile`

`mvn war:war`

Kody źródłowe indeksera oraz pliki konfiguracyjne dla SOLR znajdują się w podkatalogu `epodreczniki-wyszukiwanie` w głównym pakiecie oprogramowania.

Deployment indeksera

1. Skopiowanie pliku z aplikacją do serwera aplikacyjnego uruchamiającego SOLR
2. `cd ~`

`cp epojavaindexer/SOLRIndexerWebapp/target/SOLRIndexerWebapp-1.0.*-localhost.war solr-4.10.4/example/webapps/`

3. Przygotowanie pliku kontekstu dla aplikacji indeksera
4. `cd ~`
5. `vim solr-4.10.4/example/contexts/solrindexer-jetty-context.xml`

Plik konfiguracji kontekstu indeksera powinien mieć w takim przypadku postać:

solrindexer-jetty-context.xml

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure.dtd">
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="contextPath"><SystemProperty name="hostContext" default="/solrindexer"/></Set>
  <Set name="war"><SystemProperty name="jetty.home"/>/webapps/SOLRIndexerWebapp-1.0.21-localhost.war</Set>
  <Set name="defaultsDescriptor"><SystemProperty name="jetty.home"/>/etc/webdefault.xml</Set>
```

```
<Set name="tempDirectory"><Property name="jetty.home" default="."/>/solrindexer-  
webapp</Set>
```

```
</Configure>
```

Konfiguracja indeksu dla indeksera

Konfiguracja indeksu polega na skopiowaniu przygotowanych dla SOLR 4 plików konfiguracyjnych, które zawierają definicję pól wykorzystywanych przez aplikację indeksera

1. Repozytorium główne epub
2. cd ~

```
cp epojavaindexer/PublicationParser/src/main/resources/conf/test/epub/conf/*.xml solr-  
4.10.4/example/solr/epub/conf/
```

3. Repozytorium epub_ac
4. cd ~

```
cp epojavaindexer/PublicationParser/src/main/resources/conf/test/epub_ac/conf/*.xml solr-  
4.10.4/example/solr/epub_ac/conf/
```

5. Repozytorium epub_ds
6. cd ~

```
cp epojavaindexer/PublicationParser/src/main/resources/conf/test/epub_ds/conf/*.xml solr-  
4.10.4/example/solr/epub_ds/conf/
```

7. Narzędzia analizy leksykalnej:
8. cd ~
9. cd solr-4.10.4/example/solr-webapp/webapp/WEB-INF/lib
10. wget http://central.maven.org/maven2/org/apache/solr/solr-analysis-extras/4.10.4/solr-analysis-extras-4.10.4.jar

```
wget http://central.maven.org/maven2/org/apache/lucene/lucene-analyzers-  
stempel/4.10.4/lucene-analyzers-stempel-4.10.4.jar
```

Konfiguracja dostępu do narzędzi wyszukiwania

Należy zapewnić dostęp do uruchomionych modułów SOLR i Elastic z poziomu maszyn serwujących portal.

1. Konfiguracja iptables - konieczne jest dodanie pozwolenia na dostęp do portów 8983 (SOLR) i 9200, 9300 (Elastic)

/etc/sysconfig/iptables

```
-A INPUT -p tcp -m tcp --dport 8983 -j ACCEPT
```

```
-A INPUT -p tcp -m tcp --dport 9200 -j ACCEPT
```



```
-A INPUT -p tcp -m tcp --dport 9300 -j ACCEPT
```

Należy upewnić się, że powyższe ustawienia znajdują się w odpowiednim miejscu pliku `/etc/sysconfig/iptables` (koniecznie przed wpisami z klauzulą REJECT).

Checkpoint

W celu sprawdzenia przeprowadzonej instalacji należy uruchomić oba moduły (Elastic oraz SOLR) i skontrolować możliwość załadowania ich strony statusowych (najlepiej korzystając z przeglądarki z urządzenia w obrębie sieci testowej).

1. SOLR
Uruchomienie:

2. `cd ~`

```
./runSolr.sh
```

Status:

-

- Strona administracyjna
SOLR: <http://192.168.1.54:8983/solr/#/> (<http://150.254.208.58:8983/solr/#/>)
- Strona serwera
jetty: <http://192.168.1.54:8983/> bądź <http://150.254.208.58:8983/> (powinna prezentować dwie zainstalowane aplikacje - solr i solrindexer)

Jeżeli uruchomiona została platforma, możliwe jest testowe przeindeksowanie istniejącej na platformie kolekcji:

```
curl -X
```

```
PUT http://192.168.1.54:8983/solrindexer/index?colURL=http://content.epo.pl/content/collection/<id>/<version>/<variant>/collection.xml
```

I ewentualnie możliwe jest posprzątanie po indeksacji testowej (powyższej):

```
curl -X DELETE http://192.168.1.54:8983/solrindexer/clear/<id>/<version>/<variant>
```

Uwaga: Wyszukiwanie przyjmuje variant kolekcji do usunięcia w postaci "student" bądź "teacher" (nie "student-canon", "teacher-canon"), zgodnie z tym, jak kolekcje są indeksowane.

1. Elastic
Uruchomienie:

2. `cd ~`

```
./elasticsearch-1.7.3/bin/elastic
```

Status (alternatywnie):

- http://192.168.1.54:9200/_status
- http://150.254.208.58:9200/_status

Konfiguracja platformy

Konfiguracja platformy polega na ustawieniu adresów uruchomionych w powyższych krokach serwerów SOLR i Elastic na maszynie obsługującej portal.

Wyszukiwanie pełnotekstowe

Konfiguracja znajduje się w pliku `portal/portal/settings/solr/simple.py`. Należy tu ustawić adres serwera Elastic:

```
SOLR_HOST = '192.168.1.54'
```

```
SOLR_PORT = '8983'
```

Wyszukiwanie oparte na atrybutach

Konfiguracja znajduje się w pliku `portal/portal/settings/editsearch/elasticsearch/simple.py`. Należy tu ustawić adres serwera Elastic:

```
EPO_EDITSEARCH_HAYSTACK = {  
    'IP': '192.168.1.54',  
    'PORT': '9200',  
}
```

Powyższe ustawienia spowodują, że platforma zostanie zintegrowana z modułem wyszukiwania, publikacja treści uruchomi mechanizmy automatycznej indeksacji zawartości.

Indeksowanie danych

Indeksowanie odbywa się mechanizmami automatycznymi platformy uruchamianymi po publikacji treści. Możliwe jest ręczne zainicjowanie procesu indeksowania z poziomu platformy.

Indeksowanie ręczne

Indeksowanie dla wyszukiwania pełnotekstowego (przeindeksowanie wszystkiego):

```
epo_manage search_index --all
```

Indeksowanie dla wyszukiwania atrybutowego w bibliotece

```
epo_manage editsearch_index --kzd
```

Indeksowanie automatyczne

Indeksowanie automatyczne uruchamiane jest po publikacji treści i określone jest poprzez ustawienia w plikach `signals.py` odpowiednich modułów:

wyszukiwanie pełnotekstowe: `portal/search/signals.py`

wyszukiwanie w bibliotece: `portal/editsearch/signals.py`

Indeksowanie okresowe

Indeksowanie okresowe zdefiniowane jest w konfiguracji zdarzeń okresowych w `portal/portal/settings/deductions/decude_crontab.py`

Portal

Konfiguracja epodreczniki-edytora

Edytor modułu działa na serwerze HTTP. Domyślnie jest to serwer portalu z Edycją Online (EO), który serwuje pliki w formacie epXML do otwarcia w edytorze.

Możliwe jest uruchomienie edytora na dowolnym serwerze HTTP w celach testowych. W tym celu należy skopiować katalog główny edytora `/export` na wybrany serwer HTTP. W tym przypadku nie mamy możliwości bezpośredniego serwowania plików do edytora, dlatego należy wpisać ręcznie adres pliku epXML do edycji w pliku startowym `start.html`. Należy ustawić rolę użytkownika w pliku `export/epeditor/js/epo-api.js` - `userRoles_EPO_do_not_change = [AUTHOR_ROLE]`.

W środowisku produkcyjnym wymiana informacji (w szczególności epXMLi modułów) między ramkami edytora modułu a stroną portalu jest zaimplementowana z użyciem **HTML5-Messaging**. Ramka edytora po otrzymaniu wiadomości ze źródłem modułu w postaci string, uruchamia funkcję `XOPUS Editor.getActiveCanvas.loadDocument()`, która powoduje wywołanie funkcji `loadCustomXML` (ustawionej przy pomocy narzędzia `XOPUS IO.setLoadXMLFunction(loadCustomXML)`), w której następuje załadowanie nadesłanego epXML do edytora.

Ze względu na pochodzenie ramek z różnych domen oraz bezpieczeństwo systemu, konieczne jest ustawienie odpowiednich adresów przy wysyłaniu wiadomości, wartości domyślne na stronie portalu ustawiane są przez serwer portalu, a w kodzie edytora za pomocą zmiennej **ALLOWED_IFRAME_URLS**.

Licencja Xopus uzyskana od firmy SDL (<http://xopus.com/>):

- umieścić w katalogu `export/license/` w pliku o nazwie 'domena'.txt
- w przypadku braku licencji w edytorze będzie wyświetlać się informacja o wersji testowej

W przypadku zmiany domen należy zmienić następujące URL (w katalogu `export/epeditor/js`):

- w pliku `womi-reference.js`:
 - `var WOMIURI = "http://www.epodreczniki.pl";`
- w pliku `main.js`:
 - `var ALLOWED_IFRAME_URLS = ["http://www.epodreczniki.pl"];`
- w pliku `lister-dialog.js`:
 - `var searchURI = "https://www.epodreczniki.pl/edit/tiles/searchforxopus";`
 - `var foldersURI = "https://www.epodreczniki.pl/edit/tiles/foldersforxopus";`
 - `var WOMIURI = "http://www.epodreczniki.pl";`
 - `var WOMI_IMAGE_URL = "http://epodreczniki.pcss.pl/repo/womi/"` - adres do repozytorium treści
- w pliku `bookmark-dialog.js`:
 - `var collectionListURI = "//www.epodreczniki.pl/edit/common/api/interlinks";`

- przykład zwracanego pliku: interlinks.xml
- var collectionURI =
"//preview.epodreczniki.pl/content/collection/{0}/{1}/collection.xml";
- var referablesURI =
"//www.epodreczniki.pl/edit/common/api/object/collection/{0}/{1}/referables/xml";
 - przykład zwracanego pliku: export/epeditor/references.xml
- w pliku glossary-reference-dialog.js:
 - var referablesURI = "<https://www.epodreczniki.pl/edit/common/api/object/collection/>" +
getCollectionId() + "/referables/xml";
- w pliku load-womi-metadata.js:
 - var previewAddr = "<https://preview.epodreczniki.pl/content/womi/>";
- w pliku stylesheet.xsl ((w katalogu export/epeditor):
 - <xsl:variable name="womi-image-url"
select="<https://epodreczniki.pcss.pl/repo/womi/>"> - adres do repozytorium treści

Katalog epodreczniki-edytor należy umieścić w tym samym katalogu, w którym znajdzie się epodreczniki-portal.

Konfiguracja i uruchomienie epodreczniki-portal

1. Instalacja narzędzi systemowych, Pythona 2.7.5, narzędzia pip, zależności dla Pythona oraz JDK według instrukcji instalacji komponentu Transformacji
2. Instalacja serwera cache Redis (<http://redis.io/>) i uruchomienie na standardowym porcie
3. Instalacja wirtualnego środowiska venv w katalogu epodreczniki-portal
4. Instalacja zależności wyspecyfikowanych w plikach reqs/req-common.txt reqs/problematic.txt reqs/preview.txt
5. Instalacja kompilatora lessc (<http://lesscss.org/>) w wersji 1.7.5, tak aby był dostępny w wirtualnym środowisku
6. Konfiguracja ustawień portalu:
 1. w pliku settings/instances/simple.py
 1. TOP_DOMAIN - domena portalu,
 2. EPO_KZD_MEN_FTP_HOST, EPO_KZD_MEN_FTP_USER, EPO_KZD_MEN_FTP_PASSWORD - dane dostępowe do serwera FTP MEN przeznaczanego do importu zasobów KZD,
 3. EPO_KZD_SYNC_SECRET - losowy token uwierzytelniający Synchronizatora KZD przy komunikacji z portalem; musi być identyczny z ustawieniem portal.secret w konfiguracji Synchronizatora,
 4. EPO_REPO_ADVANCED_USER, EPO_REPO_ADVANCED_PASSWORD - użytkownik i hasło dla narzędzia eprt-tools
 5. ustawić odpowiednie parametry konfiguracyjne dla konta pocztowego SMTP: SMTP_CONF

2. w plikach settings/*/simple.py, settings/deductions/deduce_messaging.py oraz settings/deductions/deduce_celery.py adresy IP oraz porty poszczególnych skonfigurowanych usług
3. w plikach: portal/repo/templates/repo/advanced/collection.properties, portal/repo/templates/repo/advanced/module.properties, portal/repo/templates/repo/advanced/womi/root_publication.properties ustawienie publication.destination.directoryId wskazujące na odpowiedni katalog w repozytorium treści
4. w pliku: portal/editsearch/utils.py, w linii 399, identyfikator katalogu dla KZD
7. Stworzenie początkowej bazy danych SQL poleceniem: ./manage.py restoredb
8. Dodanie superużytkownika zgodnie ze standardem Django
9. Stworzenie schematu bazy danych kontekstu użytkownika poleceniem: EPO_CASSANDRA_ADMIN_CONFIG=1 ./manage.py epo_sync_cassandra
10. Podpięcie zewnętrznych bibliotek dostarczonych przez partnerów w takich ścieżkach, aby były dostępne pliki: portal/3rdparty/ge/v1_1/runner.html oraz portal/3rdparty/cp/v1_0/index.html w subdomenie static
11. Udostępnienie narzędzia eprt-tools w katalogu epodreczniki-portal/./epodreczniki-eprt-tools/deploy/uploader/
12. Instalacja serwera nginx i jego skonfigurowanie tak, aby serwer Django był jedynym serwerem backendowym, z włączonym przekazywaniem nagłówka Host
13. Skonfigurowanie w /etc/hosts wszystkich subdomen platformy wskazujących na adres prywatny tego serwera: www.epo.pl api.epo.pl dev.epo.pl static.epo.pl content.epo.pl preview.epo.pl edit.epo.pl repo.epo.pl user.epo.pl search.epo.pl
14. Uruchomienie serwera Django: ./manage.py runserver 0.0.0.0:8000
15. Uruchomienie obsługi wiadomości przychodzących do portalu: ./manage.py listener --run
16. Uruchomienie roboczego procesu dla zadań asynchronicznych: ./manage.py celery worker

W przypadku zmiany domen należy zmienić następujące URL:

- w pliku portal/reader/static/reader/js/modules/core/womi/WOMIAudioContainer.js podmienić domenę dla AV:

url: "<http://av.epodreczniki.pl/RepositoryAccess/>"

- w katalogu portal/portal/settings/av należy ustawić w plikach konfiguracyjnych odpowiedni adres serwera AV (zmienna AV_SERVER_ADDRESS)



RabbitMQ

Instalacja zależności dla RabbitMQ

Instalacja języka programowania *erlang*. W tym celu należy najpierw dodać odpowiednie repozytorium do systemu:

```
sudo tee /etc/yum.repos.d/erlang.repo <<EOF
[erlang-solutions]
name=Centos $releasever - $basearch - Erlang Solutions
baseurl=http://packages.erlang-solutions.com/rpm/centos/$releasever/$basearch
gpgcheck=0
gpgkey=http://packages.erlang-solutions.com/debian/erlang_solutions.asc
enabled=1
EOF
```

a następnie zainstalowanie tego języka:

```
sudo yum install erlang
```

Instalacja RabbitMQ

RabbitMQ był użytkowany w wersji 3.2.4-1. W celu zapewnienia pełnej kompatybilności zalecamy instalację tej samej wersji serwera. Można tego dokonać poprzez wykonanie komendy:

```
rpm -i http://www.rabbitmq.com/releases/rabbitmq-server/v3.2.4/rabbitmq-server-3.2.4-1.noarch.rpm
```

Konfiguracja RabbitMQ

Utworzenie i nadanie praw katalogowi do zapisu:

```
sudo mkdir /opt/rabbitmq
sudo chown rabbitmq:rabbitmq /opt/rabbitmq/
Następnie zawartość dostarczonego katalogu config umieszczamy w katalogu
konfiguracyjnym RabbitMQ na serwerze:
sudo cp config/* /etc/rabbitmq/
Kiedy wszystko jest już skonfigurowane włączyć serwer i ustawić
automatyczne włączanie serwisu po restarcie komputera:
sudo service rabbitmq-server start
sudo chkconfig rabbitmq-server on
W zależności od konfiguracji konieczne może być otwarcie portów
przychodzących do usługi:
iptables -I INPUT -p tcp -m multiport --dports 5672,15672 -j ACCEPT
```

Utworzenie przestrzeni oraz zestawienie połączeń dla poprawnego działania systemu

W tym celu należy pobrać plik z konfiguracją: *rabbit.json*. Należy się zalogować w panelu administratora serwera rabbitmq. W przeglądarce internetowej wpisać adres: `http://[ip serwera]:15672` i zalogować się przy pomocy użytkownika i hasła: `guest:guest`. Następnie w sekcji import, podać lokalizację pobranego pliku konfiguracyjnego i wcisnąć przycisk: "Upload broker definitions".

Brawo serwer rabbitmq został poprawnie skonfigurowany, a na konto administratora możesz zalogować się przy pomocy użytkownika i hasła: `epo:epo`.



Repozytorium AV

Wymagania

System operacyjny

Węzły repozytorium zostały stworzone i przetestowane do pracy pod kontrolą systemu Windows 2012R2 w wersji angielskiej.

Zainstalowane komponenty systemu operacyjnego

Węzeł repozytorium wymaga instalacji następujących komponentów:

- .NET Framework 4.5.1 lub nowszy
- Serwer Web (IIS) z włączonym ASP.NET 4.5
- MS SQL Express 2012 lub nowszy

<https://www.microsoft.com/en-us/download/confirmation.aspx?id=29062>

lub w katalogu setup/install/SQLEXPR_x64_ENU.exe

UWAGA: SQL Server należy zainstalować jako domyślna instancję o nazwie:

- o MSSQLSERVER (trzeba zaznaczyć "Default instance" w zakładce "Instance Configuration")
- o W zakładce "Database Engine Configuration" należy wybrać Mixed Mode i wpisać wybrane hasło dla użytkownika sa
- o Pozostałe parametry domyślne
- Microsoft® SQL Server® Data-Tier Application Framework

<https://www.microsoft.com/en-us/download/details.aspx?id=36842>

lub w katalogu setup/install/DACFramework.msi

Po instalacji SQL Servera należy uruchomić narzędzie SQL Server Configuration Manager i w zakładce SQL Server Network Configuration/Protocols

for MSSQLSERVER należy włączyć protokoły: Named Pipes i TCP/IP



Instalacja

1. przygotować katalog instalacyjny przez skompilowanie kodów źródłowych poprzez wykonanie skryptu `repoav_prepareinstall.ps1`.

Aby skompilować kody źródłowe należy użyć maszyny z systemem operacyjnym Windows na którym zainstalowane jest oprogramowanie MS Visual Studio 2013 lub nowsze.

Jeżeli zainstalowano SQL Server 2014 to w pliku `repoav_setup.ps1` należy zmodyfikować linię:

```
$global:SqlPackage = ${env:ProgramFiles(x86)} + '\Microsoft SQL  
Server\110\DAC\bin\SqlPackage.exe'
```

na

```
$global:SqlPackage = ${env:ProgramFiles(x86)} + '\Microsoft SQL  
Server\120\DAC\bin\SqlPackage.exe'
```

2. Na każdym węźle wchodzącym w skład repozytorium AV należy:

1. skopiować katalog instalacyjny `setup` i wykonać skrypt `repoav_setup.ps1` z odpowiednimi parametrami. Skrypt należy uruchomić z prawami administratora.

Skrypty znajdują się w folderze `setup`.

Parametry wywołania:

- id - numeryczny identyfikator węzła, parametr wymagany
- name - nazwa węzła, parametr wymagany
- IntIP - wewnętrzny adres IP, parametr wymagany
- ExtIP - zewnętrzny adres IP, jeżeli nie podano - taki sam jak adres wewnętrzny
- folder - katalog w którym zainstalowany zostanie węzeł repozytorium, domyślnie: `C:\RepoAV`
- type - typ węzła (Snode, Recoder, Manager, SNodeRecoder), domyślnie: Snode
- repodb - czy instalować centralną bazę RepoDB, domyślnie: 0
- mfdb - czy instalować bazę węzła - MaterialFormatDB, domyślnie: 1
- MFDBName - nazwa bazy danych węzła domyślnie: MaterialFormatDB



- RDBName - nazwa centralnej bazy, domyślnie: RepDB
- RDBServer - nazwa servera centralnej bazy, domyślnie: (local)
- RDBUser - nazwa użytkownika do centralnej bazy, wymagane jeżeli RDBServer różne od (local)
- RDBPass - hasło centralnej bazy, wymagane jeżeli RDBServer różne od (local)
- IntNetwork - adres sieci wewnętrznej domyślnie: 10.0.0.0/8
- otherNodes - lista innych węzłów w systemie postaci: id1:IP1;id2:IP2
- resize - rozmiar lokalnego repozytorium w Gb, domyślnie: 30
- RAName - nazwa usługi udostępniającego pliki, domyślnie: RepositoryAccess
- RepApiName - nazwa usługi API dla repozytorium, domyślnie: RepAPI
- vdir - nazwa virtual directory udostępniającego pliki, domyślnie: Repository
- repfolder - katalog z dla materiałów, domyślnie: c:\Storage
- port - port TCP dla usługi proca, domyślnie: 8088
- sapass - hasło administratora w przypadku gdy RepoDB jest na innej maszynie

Przykład

```
.\repoav_setup.ps1 -id 1 -name Repo1 -IntIP 192.168.1.57 -ExtIP 150.254.208.61 -replib 1 -mfdb 1 -  
IntNetwork 192.168.0.0/16 -resize 40 -type SNodeRecorder -sapass haslo
```

Icon

UWAGA: należy podać odpowiednie hasło dla użytkownika sa.

Repozytorium Treści

Przygotowanie środowiska

- Instalacja javy
 1. yum install java-1.7.0-openjdk-devel
- Instalacja Maven-1.1
 1. Pobrać i rozpakować <http://archive.apache.org/dist/maven/binaries/maven-1.1.tar.gz> do katalogu /root
 2. export PATH=/root/maven-1.1/bin:\$PATH
 3. export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk.x86_64
- Instalacja Tomcata

1. Pobrać i rozpakować <http://ftp.ps.pl/pub/apache/tomcat/tomcat-6/v6.0.44/bin/apache-tomcat-6.0.44.tar.gz> do katalogu /root
 2. W pliku /root/apache-tomcat-6.0.44/conf/server/xml przestawić connector http z portu 8080 na 80 dodać connector https na porcie 443
 3. Wygenerować samopodpisany certyfikat: `keytool -genkey -alias tomcat -keyalg RSA` (hasło changeit, jako first and last name trzeba wpisać domenę, pod którą będzie dostępny serwis - rt.epo.pl)
 4. Dodać plik /root/apache-tomcat-6.0.44/conf/Catalina/localhost/repo.xml:

```
<Context docBase="/root/dlibra-webapp">
</Context>
```
- Instalacja Inkscape i ImageMagick
 1. `yum install wget gcc ImageMagick-devel inkscape libpng-devel`
 2. `wget ftp://sunsite.icm.edu.pl/packages/ImageMagick/ImageMagick-6.8.6-9.tar.bz2`
 3. `tar -xjvf ImageMagick-6.8.6-9.tar.bz2`
 4. `cd ImageMagick-6.8.6-9`
 5. `./configure --host=x86_64-redhat-linux-gnu --build=x86_64-redhat-linux-gnu --program-prefix= --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin --sysconfdir=/etc --datadir=/usr/share --includedir=/usr/include --libdir=/usr/lib64 --libexecdir=/usr/libexec --localstatedir=/var --with-quantum-depth=8`
 6. `make`
 7. `make install`
 - Przygotowanie bazy danych
 1. `yum install postgresql-server`
 2. `su postgres -c 'initdb'`
 3. `/etc/init.d/postgresql start`
 4. `su postgres -c 'psql'`
 1. `create database dlibra;`
 2. `create user dlibra password 'dlibra';`
 3. `grant all on database dlibra to dlibra;`
 5. W /var/lib/pgsql/data/pg_hba.conf zmienić wszędzie ident na md5

Budowanie dLibry

1. Pobranie/wypakowanie źródeł (docelowo z <https://git.man.poznan.pl/stash/scm/dlr/dlibra-repository.git>) do /root/git/dlibra-repository
2. W pliku /root/git/dlibra-repository/server/project.properties ustawić contentHome (na duży dysk: /opt/content), oraz serverHostname (na domenę rt.epo.pl)
3. W pliku /root/git/dlibra-repository/webapp/project.properties ustawić webappWorkingDir (na duży dysk: /opt/webapp-work) oraz serverHostname i webappDomain (na domenę rt.epo.pl)
4. z katalogu /root/git/dlibra-repository wywołać `maven dlibra:distall -Dmaven.test.skip=true`
5. `mv server/base/target/dist /root/dlibra-server`
6. `mv webapp/base/target/dist /root/dlibra-webapp`

Budowanie e-podręczniki

1. Pobranie/wypakowanie źródeł do /root/git/epodreczniki-rt
2. Z katalogu /root/git/epodreczniki-rt wywołać `maven installall`

3. kopiowanie i podpisywanie jarów Aplikacji Redaktora:

```
#!/bin/bash
cp /root/git/dlibra-repository/app/base/target/jars-for-webapp/* /root/dlibra-webapp/jnlp
cp /root/git/epodreczniki-rt/ep-rt-app-plugins/target/ep-rt-app-plugins-3.0.0.jar /root/dlibra-
webapp/jnlp
cp /root/git/epodreczniki-rt/ep-rt-common/target/ep-rt-common-3.0.0.jar /root/dlibra-webapp/jnlp
cp /root/git/epodreczniki-rt/ep-rt-xml/target/ep-rt-app-extension-mf-xml-3.0.0.jar /root/dlibra-
webapp/jnlp
rm /root/dlibra-webapp/jnlp/dlibra-app-extension-ds-url-1.0.0.jar

echo 'Permissions: all-permissions' >/root/jnlp-manifest.mf
echo 'Application-Name: E-podreczniki: Aplikacja Redaktora' >>/root/jnlp-manifest.mf
echo 'Codebase: *' >>/root/jnlp-manifest.mf

keytool -genkeypair -alias epodreczniki -keystore /root/jnlp-keystore.jks -storepass "epodreczniki"
keytool -exportcert -keystore /root/jnlp-keystore.jks -storepass "epodreczniki" -alias epodreczniki
-file /root/cert.cer

for jarname in /root/dlibra-webapp/jnlp/*.jar; do
    jar -ufm $jarname /root/jnlp-manifest.mf
    jarsigner -keystore /root/jnlp-keystore.jks -tsa http://timestamp.digicert.com -storepass
"epodreczniki" $jarname epodreczniki
done
```

Plik /root/cert.cer należy zaimportować do certyfikatów Java na komputerach, gdzie będzie uruchamiana Aplikacja Redaktora (w Panelu Sterowania -> Java -> Security -> Manage Certificates, Certificate type: Signer CA -> Import).

4. Skopiować zawartość katalogu /root/git/ep-rt-server/target/conf/files do /root/dlibra-server. Wewnątrz skopiowanego katalogu, w pliku service.properties należy ustawić ścieżkę dostępu do Repozytorium AV (rt.content.url), URI dostępowe do kolejki komunikatów Rabbit (message.queue.uri) oraz domenę, pod którą będzie dostępna strona www Aplikacji Czytelnika (rt.content.url).
5. Dopisać serwis ep w pliku /root/dlibra-server/conf/server.xml, analogicznie do istniejących wpisów, z atrybutem serviceFactoryClass=pl.psnc.ep.rt.server.EPServiceFactory
6. Zmienić ustawienia wyszukiwania - katalog /root/dlibra-server/conf/se:
 1. W pliku service.properties ustawić wartość use.search.config.file na true
 2. W pliku searchConfig.properties zmienić treść na:

```
attributeName.1=Tytul
attributeName.2=Autor
attributeName.3=SlowaKluczowe
attributeName.4=IdentyfikatorWlasny
```

7. Skopiować potrzebne biblioteki do katalogu /root/dlibra-server/lib/java. Znajdują się one w lokalnym repozytorium Maven (katalog /root/.maven/repository):
 1. org.apache.httpcomponents/jars/httpclient-4.2.1.jar
 2. org.apache.httpcomponents/jars/httpcore-4.2.1.jar
 3. com.rabbitmq/jars/amqp-client-3.3.4.jar
 4. org.json/jars/json-20131018.jar
 5. com.mchange/jars/mchange-commons-java-0.2.10.jarDodatkowo zbudowane pliki:
 6. ep-rt-server/target/ep-rt-server-3.0.0.jar
 7. ep-rt-common/target/ep-rt-common-3.0.0.jar
8. Wyedytować plik /root/dlibra-server/conf/wrapper/java.conf - dopisać ścieżki do plików skopiowanych w kroku powyżej, analogicznie jak istniejące już tam wpisy (z kolejnymi numerami).
9. Skopiować katalog /root/git/ep-rt-conversion/target/ep-rt-conversion-3.0.0 do katalogu /root/dlibra-server/extensions.

10. Dostosowanie plików strony www:

1. Skopiować całą zawartość katalogu /root/git/ep-rt-webapp do katalogu /root/dlibra-webapp, zastępując istniejące pliki.
2. Aby działały linki do samouczków na stronie głównej, należy utworzyć katalog /root/dlibra-webapp/samouczki i skopiować do niego zasoby z katalogu z dokumentacją:

```
cp -rf /root/docs/edytor_tresci_online/cz1_uklad_edytora/html/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/edytor_tresci_online/cz1_uklad_edytora/exe/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/edytor_tresci_online/cz2_edycja_tresci/html/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/edytor_tresci_online/cz2_edycja_tresci/exe/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/edytor_tresci_online/cz3_recenzje/html/* /root/dlibra-webapp/samouczki/  
cp -rf /root/docs/edytor_tresci_online/cz3_recenzje/exe/* /root/dlibra-webapp/samouczki/  
  
cp -rf /root/docs/plugin_ms_word/1_instalacja_plugin/html/* /root/dlibra-webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/1_instalacja_plugin/exe/* /root/dlibra-webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/2_tworzenie_modulu/html/* /root/dlibra-webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/2_tworzenie_modulu/exe/* /root/dlibra-webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/3_tworzenie_struktury_epodrecznika/html/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/3_tworzenie_struktury_epodrecznika/exe/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/4_wykorzystanie_stylow_znak_blok_womi/html/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/4_wykorzystanie_stylow_znak_blok_womi/exe/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/6_zadania_z_losowaniem_odpowiedzi/html/* /root/dlibra-  
webapp/samouczki/  
cp -rf /root/docs/plugin_ms_word/6_zadania_z_losowaniem_odpowiedzi/exe/* /root/dlibra-  
webapp/samouczki/
```

3. Wprowadzić poprawny adres do instalatora dodatku MS Word, w pliku /root/dlibra-webapp/WEB-INF/components/templates/welcome.vm - tam, gdzie występuje fraza TUTAJ_WSTAW_ADRES_DO_INSTALATORA_DODATKU_MS_WORD
 4. W razie potrzeby zmiany adresów najlepiej wyszukać we wszystkich plikach w /root/dlibra-webapp frazy ".epo.pl" i odpowiednio zmienić.
11. Do katalogu /root/dlibra-webapp/WEB-INF/lib skopiować potrzebne katalogi z repozytorium maven (katalog /root/.maven/repository):
1. org.apache.commons/jars/commons-compress-1.9.jar
 2. org.json/jars/json-20131018.jar
- Dodatkowo zbudowane pliki:
3. ep-rt-common/target/ep-rt-common-3.0.0.jar
 4. ep-rt-web-components/target/ep-rt-web-components-3.0.0.jar
12. W pliku /root/git/ep-rt-tools/target/ep-rt-tools-3.0.0.zip znajduje się zestaw narzędzi do interakcji z Repozytorium Treści z linii komend, który należy przekazać pozostałym komponentom, które z niego korzystają, oraz zainteresowanym użytkownikom. Jeśli docelowy adres Repozytorium Treści jest inny niż rt.epo.pl, należy go wcześniej zmienić w pliku config.properties w tym archiwum zip.
13. Inicjalizacja bazy danych: `psql -U dlibra -d dlibra -f git/epodreczniki-rt/ep-rt-server/db-init.sql`
14. Nadanie uprawnień do wykonywania plików:
1. `chmod +x /root/dlibra-server/dlibra-server.sh`
 2. `chmod +x /root/dlibra-server/bin/wrapper-linux-x86-64`

Uruchamianie i zatrzymywanie

1. Serwer: w katalogu /root/dlibra-server należy wywołać:
 1. `./dlibra-server.sh start` - aby uruchomić
 2. `./dlibra-server.sh stop` - aby zatrzymać
2. Aplikacja Czytelnika: w katalogu /root/apache-tomcat-6.0.44/bin należy wywołać:
 1. `./startup.sh` - aby uruchomić
 2. `./startup.sh` - aby zatrzymać. Uwaga: czasami aplikacja nie zamyka się poprawnie. Należy upewnić się przy pomocy polecenia `ps -ef | grep tomcat`, że proces java został zakończony, albo zakończyć go poleceniem `kill -9 <PID>`.
3. Pobranie Aplikacji Redaktora:
 1. Komputer klienta musi być tak skonfigurowany, aby domena `rt.epo.pl` wskazywała na Repozytorium Treści.
 2. Pobrać i uruchomić plik <https://rt.epo.pl/repo/jnlp/> (wymagana Java, wersja 7 lub 8).
 3. Przy logowaniu trzeba w profilu zaznaczyć "Używaj konfiguracji NAT" i wpisać domyślny serwer `rt.epo.pl`
 4. dane logowania: autor/autor ma prawo do katalogu Testy zewnętrzne
 5. użytkownik PCSS/PCSS ma prawa do wszystkiego

Serwer Kodujący

Instalacja

Do działania Serwera Kodującego (SK) wymagany jest komponent Transformacji ze wszystkimi swoimi zależnościami.

Ponadto:

1. Rozpakować komponent SK do katalogu `~/epodreczniki-sk`
2. Zainstalować Apache Maven 3.x:
3. `cd /opt`
4. `sudo wget http://www.eu.apache.org/dist/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz`
5. `sudo tar -zxf apache-maven-3.3.9-bin.tar.gz`
6. `sudo mv apache-maven-3.3.9 apache-maven`
- 7.
8. `sudo tee /etc/profile.d/maven.sh <<EOF`
9. `#!/bin/bash`
10. `MAVEN_HOME=/opt/apache-maven`

11. `PATH=\$MAVEN_HOME/bin:\$PATH`

12. `export PATH MAVEN_HOME`

13. `export CLASSPATH=.`

14. `EOF`

15. `sudo chmod +x /etc/profile.d/maven.sh`

16. `source /etc/profile.d/maven.sh`

17.

`mvn --version`

18. Skompilować i zainstalować moduł `stat-alt-validator` w lokalnym repozytorium Mavena:

19. `cd ~/epodreczniki-sk/stat-alt-validator`

`mvn install`

20. Wprowadzić niezbędną konfigurację SK do poniższych plików:

1. `~/epodreczniki-sk/sk/src/main/resources/profiles/production/conf.properties` - konfiguracja ogólna; należy utworzyć katalogi robocze wskazane w konfiguracji;

2. `~/epodreczniki-sk/sk/src/main/resources/profiles/production/urls.properties` - konfiguracja związana z plikami źródłowymi podręczników; należy poprawić adres domenowy portalu (pozostawiając subdomenę "preview.*"), adres domenowy Repozytorium Treści oraz identyfikatory roboczych WOMI.

21. Skompilować aplikację SK:

22. `cd ~/epodreczniki-sk/sk`

`mvn compile`

23. Aplikacja będzie uruchomiona na porcie 8080, ale na zewnątrz będzie dostępna na porcie 80. Ruch z portu 80 na 8080 zostanie przekierowany za pomocą iptables. Należy dodać odpowiednią regułę:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```

Uruchomienie

Uruchamiając SK, należy podać port, na którym ma zostać uruchomiony serwer HTTP:



```
cd ~/epodreczniki-sk/sk
```

```
mvn jetty:run -Djetty.port=8080 -DSK_MODE=production
```

Logi działania aplikacji są zapisywane w katalogu ~/epodreczniki-sk/log. Do odrębnego pliku logu warto przekierować standardowe wyjścia aplikacji.

Zatrzymanie aplikacji: Ctrl+C.

Synchronizator KZD

Instalacja

1. Zainstalować narzędzia systemowe oraz JDK 1.7 jak tutaj.
2. Zainstalować Apache Maven 3.x jak tutaj.
3. Rozpakować komponent Synchronizatora do roboczego katalogu, np. ~/kzd-sync-install/.
4. Plik ~/kzd-sync-install/xml/epo-kzd-schema.xsd należy umieścić na dowolnym serwerze HTTP.
5. W pliku ~/kzd-sync-install/xml-validator/src/main/resources/pl/epo/kzd/xml/conf.properties należy ustawić:
 1. schemaUrl - adres URL pliku epo-kzd-schema.xsd
 2. categoryListUrl - adres URL do API portalu zwracającego listę kategorii KZD
6. W pliku ~/kzd-sync-install/sync/src/main/resources/application.properties należy ustawić:
 1. ftp.host, ftp.username, ftp.password - dane połączenia do FTP
 2. ftp.xml_directory - ścieżka do katalogu na FTP, w którym znajdują się pliki XML z opisem zasobów
 3. amqp.connection_url - connection URL do połączenia z serwerem RabbitMQ
 4. amqp.queue - nazwa kolejki w RabbitMQ, którą "nasłuchuje" Synchronizator
 5. portal.domain - adres domenowy portalu
 6. portal.protocol - protokół "http" lub "https" portalu

7. portal.secret - token uwierzytelniający Synchronizatora przy komunikacji z portalem; musi być identyczne z ustawieniem EPO_KZD_SYNC_SECRET w konfiguracji portalu

8. rt.domain - adres domenowy Repozytorium Treści (RT)

9. rt.directory_id - identyfikator katalogu w RT, w którym będą umieszczane zasoby KZD

10. rt.username, rt.password - dane użytkownika RT z prawami zapisu do w/w katalogu

7. Skompilować aplikację Synchronizatora:

8. cd ~/kzd-sync-install/xml-validator

9. mvn install

10. cd ../sync

mvn package

11. Umieścić skompilowaną aplikację oraz skrypt uruchomieniowy w nowym katalogu:

12. mkdir ~/kzd-sync

13. cd ~/kzd-sync

14. cp ~/kzd-sync-install/sync/target/kzd-sync-jar-with-dependencies.jar kzd-sync.jar

15. cp ~/kzd-sync-install/sync/misc/run.sh .

chmod +x run.sh

16. W katalogu z aplikacją umieścić pakiet eprt-tools w katalogu o nazwie "eprt-tools" i nadać skryptom prawo wykonywania:

17. cd ~/kzd-sync

18. wget http://example.com/eprt-tools.zip

19. unzip eprt-tools.zip -d eprt-tools

chmod +x eprt-tools/*.sh

Uruchomienie

Synchronizator jest okresowo uruchamiany przez crona w dwóch trybach:

- poszukiwanie nowych zasobów na FTP - np. co dobę

- obsługa edycji wykonanych przez edytor KZD w portalu - np. co 5 minut

W tym celu należy dodać odpowiednią konfigurację do /etc/hosts, np.:

```
10***epo /home/epo/kzd-sync/run.sh --create-new
```

```
*/5****epo /home/epo/kzd-sync/run.sh --listen
```

Transformacje

1. Zainstalować potrzebne narzędzia systemowe:

```
2. sudo yum groupinstall -y development
   sudo yum install -y wget zlib-devel openssl-devel bzip2-devel xz-libs
   readline-devel sqlite-devel
```

3. Zainstalować Pythona 2.7.5:

```
4. wget http://www.python.org/ftp/python/2.7.5/Python-2.7.5.tar.xz
5. xz -d Python-2.7.5.tar.xz
6. tar -xf Python-2.7.5.tar
7. cd Python-2.7.5
8. ./configure --prefix=/usr/local
9. make
10. sudo make install
11. sudo ln -s /usr/local/bin/python2.7 /usr/bin/python2.7
12.
13. python -V
    sudo python2.7 -V
```

14. Zainstalować narzędzie pip

```
15. curl https://bootstrap.pypa.io/get-pip.py | sudo python2.7 -
16. sudo ln -s /usr/local/bin/pip2.7 /usr/bin/pip
17. sudo pip install setuptools
18.
19. pip -V
    sudo pip -V
```

20. Zainstalować zależności dla Pythona:

```
21. sudo yum install -y libxslt-devel libxml2-devel librsvg2 stix-fonts python-
    devel libjpeg-devel libffi-devel libpng-devel
    sudo pip install lxml reportlab cairosvg Pillow argparse qrcode
```

22. Zainstalować Java Development Kit (JDK) 1.7.x:

```
23. sudo yum install java-1.7.0-openjdk-devel
24.
    java -version
```

25. Zainstalować PrinceXML 8.1 rev 5

```
26. wget http://www.princexml.com/download/prince-8.1-5.centos60.x86_64.rpm
    sudo rpm -Uvh prince-8.1-5.centos60.x86_64.rpm
```

W pliku /usr/lib/prince/license/license.dat należy umieścić plik licencji. Domyślnie znajduje się tam licencja demo.

27. Rozpakować komponent Transformacji do katalogu ~/epodreczniki-transformacje.

28. Rozpakować bibliotekę Euler MathSpeech (<http://sklep.altix.pl/pl/euler-mathspeech>) do katalogu `~/epodreczniki-transformacje/epomath/` tak, żeby bezpośrednio w tym katalogu znajdował się główny plik `__init__.pyc` tej biblioteki.
29. Zainstalować w systemie niezbędne czcionki - `msttcorefonts` oraz `latinmodern`:
30. `sudo yum install -y cabextract xorg-x11-font-utils fontconfig`
31. `sudo rpm -i https://downloads.sourceforge.net/project/mscorefonts2/rpms/msttcore-fonts-installer-2.6-1.noarch.rpm`
- 32.
33. `sudo mkdir /usr/share/fonts/latinmodern-math`
34. `sudo cp epodreczniki-transformacje/SVGMath/fonts/latinmodern-math.ttf /usr/share/fonts/latinmodern-math/`
`sudo fc-cache`
35. W pliku `~/epodreczniki-transformacje/xsl/common.xslt` należy poprawić adresy URL prowadzące do portalu - należy skorygować adres domenowy portalu:
 - o Wartość elementu "param" o nazwie "www_link" zawiera główny link do portalu epodreczniki.pl. Wartość linku jest wykorzystywana do tworzenia odwołań do portalu epodreczniki.pl z formatów statycznych (pdf, aplikacja mobilna), np. w celu obejrzenia filmu.
 - o Zmienna "licenseMap" zawiera zbiór licencji [nazwa:url]. Licencje są używane w elementach kolekcji np. womi.