



# Instrukcja weryfikacji poprawności instalacji komponentów

Platformy Technologicznej  
[epodreczniki.pl](http://epodreczniki.pl)

Poznań, 14 lutego 2017 r.



## Spis treści

Cassandra.....	3
Moduł wyszukiwania treści .....	4
Portal .....	5
RabbitMQ .....	7
Repozytorium AV .....	7
Repozytorium Treści .....	8
Serwer Kodujący.....	8
Synchronizator KZD .....	9
Transformacje .....	10



## Cassandra

Należy upewnić się czy portal jest w stanie komunikować się z Cassadrą po porcie 9042 oraz 9160. Portal podczas instalacji zdalnie komunikuje się z bazą i tworzy jej schemat.

Należy mieć na uwadze również pozostałe porty 7000 oraz 7001 do komunikacji pomiędzy węzłami oraz 7199 do monitoringu JMX jeśli takowy jest wykorzystywany.

Podczas instalacji wg instrukcji (najnowsza stabilna wersja 2.1) tworzony jest dedykowany użytkownik *epoadmin* dla portalu (można utworzyć również innego i wówczas należy użyć odpowiedniej nazwy użytkownika). Należy przetestować na dowolnym węźle cassyndry czy można się dostać do bazy za pomocą *cqlsh* (argument "-u" oznacza nazwę użytkownika, "-p" hasło użytkownika):

```
cqlsh IP_BIND_ADDRESS -u epoadmin -p epoadmin
```

Jeśli na danym adresie IP działa poprawna baza cassyndry i dany użytkownik istnieje, wówczas otrzymamy widok shell'a cql:

```
[cqlsh 5.0.1 | Cassandra 2.1.15 | CQL spec 3.2.1 | Native protocol v3]
Use HELP for help.
epoadmin@cqlsh>
```

W shellu można wydać komendę listującą keystore'y:

```
desc keyspaces;
```

Przed instalacją portalu widok powinien być taki:

```
epoadmin@cqlsh> desc keyspaces;
system_traces system_auth system
```

Po instalacji portalu na liście dodatkowo powinna pojawić się nazwa skonfigurowanej bazy w ustawieniach portalu - domyślnie: "epo\_production".

Aby sprawdzić schemat bazy portalu (po jego instalacji) należy wykonać komendę:

```
desc keyspace epo_production;
```

Wynikiem powinien być listing schematu bazy danych, który powinien być podobny do zawartości dowolnego pliku cdl zawartego w backupie bazy danych "epo\_production-schema-\*.cdl".

Jeśli tego dodatkowego keystore'a nie ma w ogóle albo schemat jest pusty to oznaczać może problemy w komunikacji między portalem a bazą Cassandy i należy sprawdzić logi portalowe.

Logi usługi bazy Cassandy domyślnie są w lokalizacji */var/log/cassandra*. Można tam sprawdzić czy występują jakiegokolwiek błędy (ERROR). W przypadku dobrze zainstalowanej i skonfigurowanej usługi nie powinno takowych być:

```
cat /var/log/cassandra/cassandra.log | grep "ERROR" | wc -l
cat /var/log/cassandra/system.log | grep "ERROR" | wc -l
```

Wynikiem powyższych komend powinno być "0".



Dodatkowym testem, który jest pomocny w sprawdzeniu stanu całego klastra jest użycie komendy `nodetool`:

```
nodetool status
```

Wynikiem powinna być lista wszystkich skonfigurowanych węzłów za stanem "UN" czyli up and normal.

Można też sprawdzić stan konkretnego keyspace'a:

```
nodetool status epo_production
```

Wynik powinien być podobny - wszędzie status "UN" pokazując też procentowe rozłożenie zawartości bazy pomiędzy węzłami.

Aby komendy `cqlsh` i `nodetool` zadziałały bez podawania użytkownika i hasła za każdym razem, należy stworzyć plik `cqlshrc` w katalogu `~/cassandra` z zawartością:

```
[authentication]
username=epoadmin
password=epoadmin
```

Powyższy prosty test pozwala sprawdzić czy baza na pewno działa lokalnie. Aby zakończyć działanie `cql` shella należy wykonać komendę: `quit`

Instalacja portalu rzuci błędem gdy jest problem z dostępem do bazy Cassandra, wówczas należy sprawdzić czy połączenie sieciowe działa poprawnie i odpowiednie porty są otwarte.

Wszelkich informacji na temat instalacji i konfiguracji usługi Cassandra można szukać na stronach dokumentacji do wersji 2.1 (wykorzystywanej w platformie):

<http://docs.datastax.com/en/cassandra/2.1/>

## Moduł wyszukiwania treści

W celu weryfikacji poprawności repozytorium wyszukiwania należy upewnić się, że uruchomiony uprzednio moduł SOLR oraz aplikacja indeksująca dane działają. W tym celu należy odwiedzić stronę administracyjną serwera aplikacyjnego (np. w przypadku postępowania zgodnie z instrukcją serwera jetty) i oszacować, czy na liście uruchomionych aplikacji istnieje SOLR jak i SOLRINDEXER:

- `http://<adres url serwera wyszukiwania>:8983/`  
Spodziewane jest by na liście występował `solr` oraz `solrindexer`.

Możliwe jest także skontrolowanie stanu instalacji SOLR poprzez dostęp do jego stron administracyjnych:

- `http://<adres url usługi wyszukiwania>:8983/solr/#/`  
Spodziewane jest istnienie stron standardowych narzędzi administracyjnych SOLR a także brak błędów (prezentowanych w zakładce Logs) oraz istnienie trzech instancji repozytorium (`epub`, `epub_ds` oraz `epub_ac`)



W celu weryfikacji istnienia repozytorium Elastic dla modułu KZD należy skontrolować status usługi elastic poprzez odwiedzenie adresu:

- `http://<adres url usługi wyszukiwania>:9200/_status`

## Portal

O ile nie jest zaznaczone inaczej, wszystkie polecenia należy wywoływać na maszynie wirtualnej portalu. Wszystkie wymienione poniżej polecenia powinny zwracać kod wyjścia równy 0. Wszystkie wymienione poniżej zapytania HTTP powinny zwracać kod 200.

Aby sprawdzić poprawność instalacji serwera cache Redis, należy uruchomić polecenie:

```
redis-benchmark
```

Aby sprawdzić połączenie do serwera cache Redis, należy wywołać polecenie:

```
echo "from surround.django import redis ; redis.get_connection().set('test', 'test')" | ./manage.py shell --plain
```

Aby sprawdzić poprawność instalacji serwera cache Memcached, należy na maszynie wirtualnej portalu uruchomić polecenie:

```
memcached-tool 127.0.0.1:11211 stats
```

Aby sprawdzić połączenie do serwera cache Memcache, należy na maszynie wirtualnej portalu uruchomić polecenie:

```
echo "from django.core.cache import cache ; cache.set('test', 'test') ; cache.get('test')" | ./manage.py shell --plain
```

Aby sprawdzić poprawność instalacji kompilatora lessc, należy na maszynie wirtualnej portalu uruchomić polecenie:

```
lessc --version
```

oraz sprawdzić, czy program nr wersji to 1.7.5

Należy sprawdzić działanie poszczególnych subdomen portalu, poprzez otwarcie w przeglądarce stron:



- `http://www.<domena-instalacji>/begin`
- `http://dev.<domena-instalacji>/`
- `http://api.<domena-instalacji>/`
- `http://user.<domena-instalacji>/auth/login`
- `http://search.<domena-instalacji>/search/classic`

Aby sprawdzić połączenie do klastra Cassandra (po jej poprawnej instalacji), należy wywołać polecenie:

```
echo "from userapi import models ; h = models.HealthCheck.create(myid='test', value='test') ; h.delete()" | ./manage.py shell --plain
```

Aby sprawdzić połączenie do SOLR (po jego poprawnej instalacji), należy wybrać jedną z opublikowanych kolekcji oraz wywołać poniższe polecenie (zakładając, że wybrana kolekcja posiada identyfikator N i wersję V):

```
./manage.py search_index --index N:V:student-canon
```

Aby sprawdzić połączenie do Elasticsearch (po jego poprawnej instalacji), należy wywołać polecenie:

```
./manage.py epo-manage editsearch_index --create
```

Aby sprawdzić połączenie do RabbitMQ (po jego poprawnej instalacji), należy uruchomić polecenie:

```
./manage.py listener --fetch
```

Aby sprawdzić działanie Celery (po zweryfikowaniu połączenia do RabbitMQ), należy wywołać polecenie:

```
echo "import common.tasks ; common.tasks.test.delay()" | ./manage.py shell --plain
```

po czym sprawdzić w logu usługi workera celery (punkt nr 16 instrukcji instalacji), czy jest tam obecny wpis: `IOError: test`

Aby sprawdzić poprawne połączenie do odczytu Repozytorium Treści należy po wybrać kilka przykładowych kolekcji, modułów oraz WOMI znajdujących się w Repozytorium treści, oraz:

- dla każdej kolekcji o identyfikatorze N i numerze wersji V otworzyć w przeglądarce stronę:  
`http://preview.<domena-instalacji>/content/collection/N/V/collection.xml`
- dla każdego modułu o identyfikatorze N i numerze wersji V otworzyć w przeglądarce stronę:  
`http://preview.<domena-instalacji>/content/module/N/V/module.xml`

- dla każdego WOMI o identyfikatorze N otworzyć w przeglądarce następujące strony:
  - <http://preview.<domena-instalacji>/content/womi/N/manifest.json>
  - <http://preview.<domena-instalacji>/content/womi/N/metadata.json>

Aby sprawdzić poprawne działanie zapisu do Repozytorium Treści (po weryfikacji poprawności połączenia do odczytu Repozytorium treści), należy przejść do komponentu Edycji Online portalu, po czym wykonać następujące operacje:

- utworzyć nową kolekcję,
- zabezpieczyć kolekcję (kolekcja może być pusta),
- utworzyć nowy moduł,
- zabezpieczyć moduł (moduł może być pusty).

Aby sprawdzić połączenie do Serwera Kodującego (po weryfikacji poprawności jego instalacji) należy wybrać jedną z kolekcji dostępnych w Repozytorium Treści, i po jej przetworzeniu przez Serwer Kodujący otworzyć następujący zasób w przeglądarce (zakładając, że wybrana kolekcja posiada identyfikator N oraz wersję V): <http://preview.<domena-instalacji>/content/collection/N/V/student-canon/collection.xml>

## RabbitMQ

1. Tak jak w procesie instalacji, należy jeszcze raz otworzyć w przeglądarce stronę: [http://\[ip\\_serwera\]:15672](http://[ip_serwera]:15672) w przeglądarce
2. pierwszym etapem weryfikacji jest sprawdzenie, czy można zalogować się na użytkownika epo z tym hasłem epo
3. jeśli tak, oznacza to, że rabbit został poprawnie skonfigurowany oraz, że została wczytana konfiguracja z pliku rabbit.json
4. dodatkowo można sprawdzić czy utworzono wirtualne hosty:
  1. w zakładce admin
  2. wejść w podstronę virtual hosts
  3. powinny znajdować się tu wirtualne hosty: /, epo oraz portal-production

## Repozytorium AV

1. Po zainstalowaniu oprogramowania RepoAV na każdym węźle należy sprawdzić czy pojawiła się usługa o nazwie "Proca3 - nazwawęzła" i czy ona pracuje
2. Na każdym węźle można wejść przez przeglądarkę na adres:

<http://localhost:8088/LocalNode/config>

widać tam czy wszystkie podsystemy węzła pracują i jakie są ich parametry. Należy sprawdzić czy wszystkie są wystartowane.

3. Należy użyć funkcji usługi RepAPI w celu dodania przykładowego materiału video (funkcja AddMaterial), a następnie za pomocą GetMaterialStatus sprawdzić czy materiał się przetworzy.

4. Po zakończeniu przetwarzania można pobrać materiał za pomocą usługi RepositoryAccess/id\_formatu\_materiału

## Repozytorium Treści

1. Zweryfikować, że po uruchomieniu dlibra-server w logu /root/dlibra-server/logs/wrapper.log nie pojawiły się wpisy typu ERROR, na końcu pojawiła się informacja "INFO: Server startup in ... ms".
2. Zweryfikować, że po uruchomieniu apache-tomcat w logu /root/apache-tomcat-6.0.44/logs/catalina.out na końcu pojawiła się informacja "INFO: Server startup in .... ms".  
Błędy przy uruchamianiu, które można zignorować:
  1. SntpManager ERROR:  
Configuration error
  2. PublicationUploadManager WARN:  
Error occured while configuring publication upload form attributes.
  3. ConfigurationContextListener ERROR:  
Error while initializing context:  
java.lang.reflect.InvocationTargetException  
[...]  
Caused by: java.lang.RuntimeException: Couldn't load defaults file
3. Zweryfikować, że wyświetla się strona <https://rt.epo.pl/repo/rt>, a także da się pobrać i uruchomić Aplikację Redaktora: <https://rt.epo.pl/repo/jnlp>

## Serwer Kodujący

Po prawidłowym uruchomieniu SK na standardowym wyjściu można zaobserwować komunikaty:

```
[INFO] Started SelectChannelConnector@0.0.0.0:8080  
[INFO] Started Jetty Server
```

Przy pierwszym uruchomieniu aplikacji kilkanaście linii wcześniej pojawi się wpis:

```
ERROR pl.epodr.sk.womi.IndexDatabase - database file not  
found: ....epodreczniki-public/epodreczniki-sk/sk/indexDb.dat
```

to oczekiwane. Ten komunikat nie pojawi się przy kolejnych uruchomieniach, o ile SK przetworzył już dotąd jakiegokolwiek zlecenia z sukcesem.

Kilka linii wcześniej, po każdym uruchomieniu, powinien pojawić się komunikat, który potwierdza, że SK odnalazł skrypty transformacji (a w szczególności pliki CSS w katalogu transformacji) w oczekiwanym miejscu:

```
INFO nverter.css.AbstractCssManager - PdfCssManager - stylesheets found:  
[standard-2-matematyka, standard-2, standard-2-przyroda, standard-2-uwr]
```

Otwarcie strony [http://<ip\\_sk>:8080/](http://<ip_sk>:8080/) powinno zwrócić przekierowanie na /content/, co powinno skończyć się statusem 404. Tak samo powinno działać otwarcie strony na porcie 80, przy czym port 80 odpowie tylko z zewnątrz tego hosta (prerouting w iptables, skonfigurowany zgodnie z instrukcją instalacji SK, nie działa dla zapytań z localhosta).



Umieszczenie jakiegogo pliku pod dowolną ścieżką w katalogu wskazanym jako outputDir w konfiguracji spowoduje, że plik będzie widoczny po HTTP pod ścieżką /content/.

Wykonanie na hoście z SK polecenia:

```
echo "123456789-1" | POST -Sc text/plain http://localhost:8080/colxml/added
```

(zakładamy, że jest zainstalowany pakiet perl-libwww-perl, żeby komenda "POST" była dostępna) powinno zwrócić status HTTP 204. Na standardowym wyjściu aplikacji powinien pojawić się komunikat:

```
INFO pl.epodr.sk.task.DownloadTask - collection 123456789(1) no longer exist
```

To oznacza, że SK nie odnalazł i nie mógł przetworzyć kolekcji o identyfikatorze "123456789" i wersji 1. Brak takiego komunikatu może oznaczać błąd połączenia z Portalem (lub Portalu z RT) - SK nie mógł połączyć się po HTTP z Portalem w celu pobrania kolekcji o tym identyfikatorze.

Wykonanie powyższego polecenia z podaniem istniejącej w RT kolekcji (id i wersja, połączone minusem) powinno skutkować rozpoczęciem przetwarzania, co można śledzić na podstawie informacji pojawiających się w logu. W międzyczasie w katalogu wskazanym w konfiguracji jako outputDir, w podkatalogu <id\_kolekcji>/<wersja\_kolekcji>/ zaczną stopniowo pojawiać się pliki wytworzone dla tej kolekcji przez SK: w pierwszej kolejności plik metadanych metadata.xml, a dalej m.in. formaty emisyjne HTML dla portalu, a później (zazwyczaj trwa to dłużej) plik (pliki) PDF i ZIP (format emisyjny dla aplikacji mobilnych). To dowiedzie również prawidłowości działania transformacji, które uruchamia SK w celu wytworzenia formatów emisyjnych.

Po zakończeniu kolejnych etapów przetwarzania kolekcji SK rozgłasza komunikat o tym fakcie za pośrednictwem Rabbita, co można zaobserwować na stronach administracyjnych Rabbita. Jeśli jest problem z połączeniem z Rabbitem to w logu SK będą wyświetlane wyjątki na ten temat, a wyjątki pojawią się już po starcie aplikacji i będą logowane co kilka sekund, ponieważ SK nie może się połączyć w celu pobrania zleceń przetwarzania. Brak tego rodzaju wyjątków na standardowym wyjściu dowodzi prawidłowej komunikacji SK z Rabbitem.

## Synchronizator KZD

Uruchomienie "run.sh --create-new" spowoduje wyjątek java.net.ConnectException: Connection refused (Connection refused), jeśli nie można połączyć się z serwerem FTP wskazanym w konfiguracji. Przy prawidłowym połączeniu pojawi się w logu komunikat "retrieving resource descriptors...", a następnie wydrukują się kilka linii informacji, że nie znaleziono nowych zasobów na FTP i program zakończy się. Jeśli nowe zasoby na FTP zostałyby znalezione to rozpocznie się przetwarzanie.

Uruchomienie "run.sh --listen" spowoduje wyjątek IOException wylatujący z pakietów com.rabbitmq.client.\*, jeśli nie da się podłączyć do Rabbita, w celu pobrania zleceń z edytora KZD z Portalu. Przy prawidłowym połączeniu aplikacja zakończy się bez żadnego komunikatu (status wyjścia 0), jeśli nie ma żadnych zleceń do przetworzenia, a w razie odnalezienia zleceń w Rabbicie rozpocznie się przetwarzanie.



Zlecenia z edytora KZD w Portalu można wygenerować wykonując na hoście portalu, w wenvie:

```
echo "from common import messaging ;  
messaging.publish_json('resource.modified', 'kzd-editor', custom_id='test',  
womi_id='123456789', womi_version=1)" | ./manage.py shell --plain
```

Podanie namiarów na nieistniejący zasób (jak w powyższej komendzie) spowoduje, że `run.sh --listen` zakończy się wyjątkiem `FileNotFoundException` przy pobieraniu zasobu z portalu przez HTTP, ale dowodzi to poprawności komunikacji Synchronizatora z Rabbitem (otrzymanie zlecenia) i bezpośrednio z Portalem (połączenie HTTP w celu pobrania metadanych zasobu).

## Transformacje

- aby zweryfikować poprawność instalacji PrinceXML należy wywołać `"prince --version"` - powinna wyświetlić się informacja o wersji i licencji
- poprawność instalacji samych skryptów Transformacji weryfikuje się poprzez weryfikację działania Serwera Kodującego - prawidłowo przetworzą się zlecenia przetworzenia kolekcji