



Dokumentacja  
konfiguracyjna  
platformy

Jarosław Legierski  
Henryk Rosa  
Tomasz Janisiewicz

Dokumentacja  
utrzymaniowa  
platformy

Flickr.com, jgonzac, CC BY

# Spis treści

1	Cel projektu	2
2	Opis architektura platformy	2
3	Konfiguracja platformy produkcyjnej	3
4	Konfiguracja platformy testowej	4
5	Dokumentacja utrzymaniowa platformy	6
	5.1 Data Server	6
	5.2 Function Server	9
6	Podsumowanie	10
7	Słowniczek – lista skrótów	11
8	Bibliografia	13

# 1 Cel projektu

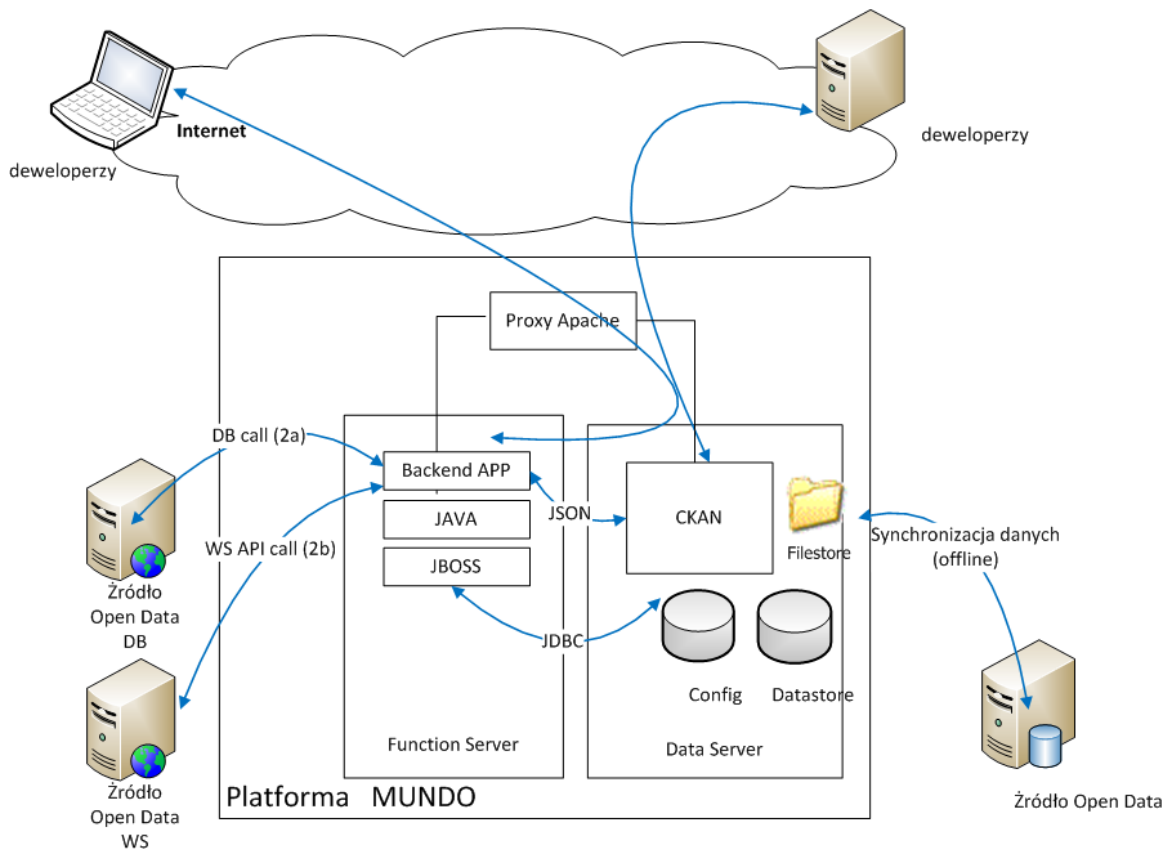
Celem projektu MUNDO [1] jest diagnoza możliwości ekspozycji oraz przygotowanie danych, które mogą zostać udostępnione przez Miasto Warszawa w modelu Open Data, a przede wszystkim budowa warstwy ekspozycji dla tych danych w formie platformy eksponującej API. Projekt MUNDO łączy w sobie opracowanie technologii do udostępnienia otwartych danych z metodologią społeczną dotyczącą włączania różnych grup społecznych w działania na rzecz poprawy życia w mieście. W ramach prac konsorcjum prowadzone były działania, które przygotowały techniczne zaplecze dla części społecznej projektu. Natomiast w części pilotażowej konsorcjum przeprowadziło konkurs na aplikacje internetowe dla Warszawy, by w ten sposób przekonać społeczeństwo o wadze otwartych danych (szczególnie publicznych) dla rozwoju miasta i ułatwienia funkcjonowania jego mieszkańców.

Innowacyjność projektu MUNDO polega na jego złożoności oraz na łączeniu ram technicznych z ramami metodologii partycypacji społecznej w życiu miasta. Wartością projektu jest również jego replikowalność w innych miastach Polski, gdyż zarówno oprogramowanie platformy do ekspozycji otwartych danych, jak i metodologia prowadzenia działań społecznych zostaną udostępnione do wykorzystania dla zainteresowanych jednostek z użyciem wolnych licencji (GPL, LGPL, FDL, CC).

## 2 Opis architektura platformy

W skład platformy wchodzi następujące komponenty:

- Proxy Server (opcjonalny) – warstwa bezpieczeństwa odpowiedzialna min za rozdział ruchu
- Data Server – oparty na systemie CKAN serwer będący: katalogiem danych, serwerem www i repozytorium danych plikowych i tabelarycznych
- Function Server – oparty na Java serwer, będący middleware dla wywołań API do źródeł danych opartych o Web Services, bazy danych itp. System ten ma zapewnić dostęp do danych dynamicznych, tj. danych, których źródłem są usługi sieciowe oraz bazy danych. Ten podsystem jest również odpowiedzialny za limitowanie wywołań, buforowanie danych, konwersje wywołań oraz konwersję formatu danych w celu uzyskania możliwie spójnego formatu API do wszystkich danych i funkcji UM Warszawa.



Rysunek 1 Elementy platformy

### 3 Konfiguracja platformy produkcyjnej

W poniższym rozdziale opis konfiguracji produkcyjnej platformy MUNDO zainstalowanej w UM Warszawa. Na poniższym rysunku przedstawiono elementy platformy natomiast w tabeli ich opis.

Tabela 1 konfiguracja sprzętowa - platforma produkcyjna

VM	OS	Procesor	RAM	HD	Uwagi
Frontend (PHP)	Linux 64 bit server (CentOS 6.6)	bd	bd	bd	Element nie należy do zakresu projektu MUNDO
Data Server (CKAN)	Linux 64 bit server (CentOS 6.6)	2 Cores	8GB	80 GB	VM
Function Server (Java)	Linux 64 bit server (CentOS 6.6)	2 Cores	8GB	80 GB	VM

## Dostęp do systemu

interface	IP	Port	Sposób dostępu	opis
ssh	10.10.10.10	22/TCP	Klient ssh	Platforma produkcyjna - zarządzanie Data Server
www (CKAN)	10.10.10.10	80/TCP	http://10.10.10.10	Platforma produkcyjna - www
Baza danych	10.10.10.10	5432/TCP	Klient PostgreSQL	Platforma produkcyjna - baza danych
CKAN Datapusher	10.10.10.10	8800/TCP	http://10.10.10.10:8800 /	Platforma produkcyjna - rozszerzenie CKAN Datapusher
Apache Tomcat	10.10.10.10	8080/TCP	http://10.10.10.10:8080 /	Platforma produkcyjna - Apache Tomcat
Apache SOLR	10.10.10.10	8080/TCP	http://10.10.10.10:8080 /solr	Platforma produkcyjna - Apache SOLR
ssh	10.10.10.11	22/TCP	Klient ssh	Platforma produkcyjna - zarządzanie Function Server
JBossAS	10.10.10.11	9990/TCP	http://10.10.10.11:9990	Platforma produkcyjna - JBOSSAS 7 konsola
Aplikacja MUNDO Backend	10.10.10.11	8080/TCP	http://10.10.10.11:8080 /cbr/mundo-java-backend/	Aplikacja MUNDO Backend

## 4 Konfiguracja platformy testowej

Analogicznie poniżej zamieszczono konfigurację platformy testowej używanej do prac developerskich i testowania nowych wersji oprogramowania.

**Tabela 2 konfiguracja sprzętowa - platforma produkcyjna**

VM	OS	Procesor	RAM	HD	Uwagi
Frontend (PHP)	Linux 64 bit server (CentOS 6.6)	bd	bd	bd	Element nie należy do zakresu projektu MUNDO
Data Server (CKAN)	Linux 64 bit server (CentOS 6.6)	2 Cores	8GB	80 GB	VM
Function Server (Java)	Linux 64 bit server (CentOS 6.6)	2 Cores	8GB	80 GB	VM

**Dostęp do systemu**

interface	IP	Port	Sposób dostępu	opis
ssh	10.10.20.10	22/TCP	Klient ssh	Platforma testowa - zarządzanie Data Server
www (CKAN)	10.10.20.10	80/TCP	http://10.10.20.10	Platforma testowa - www
Baza danych	10.10.20.10	5432/TCP	Klient PostgreSQL	Platforma testowa - baza danych
CKAN Datapusher	10.10.20.10	8800/TCP	http://10.10.20.10:8800 /	Platforma testowa - rozszerzenie CKAN Datapusher
Apache Tomcat	10.10.20.10	8080/TCP	http://10.10.20.10:8080 /	Platforma testowa - Apache Tomcat
Apache SOLR	10.10.20.10	8080/TCP	http://10.10.20.10:8080 /solr	Platforma testowa - Apache SOLR
ssh	10.10.20.11	22/TCP	Klient ssh	Platforma testowa - zarządzanie Function Server
JBossAS	10.10.20.11	9990/TCP	http://10.10.20.11:9990	Platforma testowa - JBOSSAS 7 konsola
Aplikacja MUNDO	10.10.20.11	8080/TCP	http://10.10.20.11:8080 /cbr/mundo-java-	Aplikacja MUNDO

Backend		P	backend/	Backend
---------	--	---	----------	---------

## 5 Dokumentacja utrzymaniowa platformy

W poniższym rozdziale zamieszczono procedury utrzymaniowe elementów platformy MUNDO dotyczące Data Server i Function Server zarówno platformy testowej jak i produkcyjnej.

### 5.1 Data Server

#### A) Baza danych

Uruchomienie PostgreSQL

```
service postgresql start
```

Restart usługi postgresql.

```
service postgresql restart
```

Wyłączenie PostgreSQL

```
service postgresql stop
```

Sprawdzenie czy na porcie 5432 działa postgresql

```
netstat -an | grep 5432
```

#### B) Server Apache Tomcat

Uruchomienie serwera Tomcat:

```
$CATALINA_HOME/bin/startup.sh
```

Wyłączenie serwera tomcat

```
$CATALINA_HOME/bin/shutdown.sh
```

Sprawdzenie czy na porcie 8080 działa serwer tomcat

```
netstat -an | grep 8080
```

Weryfikacja Tomcat i SOLR

Uruchomić przeglądarkę i wpisać [http://adres\\_ip\\_serwera:8080](http://adres_ip_serwera:8080) – powinniśmy otrzymać dostęp do strony www Tomcat

Uruchomić przeglądarkę i wpisać [http://adres\\_ip\\_serwera:8080/solr](http://adres_ip_serwera:8080/solr) – powinniśmy otrzymać dostęp do strony www SOLR

### C) Server Apache

Uruchomienie usługi httpd

**service httpd start**

Wyłączenie usługi httpd

**service httpd stop**

Restart usługi httpd

**service httpd restart**

Przeładowanie usługi httpd

**service httpd reload**

Weryfikacja CKAN

Uruchomić przeglądarkę i wpisać nazwę FQDN lub adres ip serwera – powinniśmy otrzymać dostęp do strony www CKAN Po wejściu na główną stronę CKAN serwer nie powinien zwracać jakichkolwiek błędów.

**http:// adres\_ip\_serwera**

Weryfikacja działania rozszerzenia CKAN Data Pusher

Należy w przeglądarce połączyć się z hostem na porcie 8800

**http:// adres\_ip\_serwera:8800/**

W wyniku powinniśmy otrzymać następujący JSON

```
{
  "help": "\n    Get help at:\n    http://ckan-service-provider.readthedocs.org/."
}
```

### D) Backup /Restore

**Backup systemu plików**



W celu wykonania backup konfiguracji i danych Data Server (CKAN) należy wykonać backup następujących katalogów:

Katalog	Zawartość	Uwagi
/var/lib/pgsql/	PostgreSQL	
/etc/ckan	CKAN – konfiguracja	
/usr/lib/ckan/	CKAN	
/var/lib/ckan	CKAN filestore	Filestore – pliki
/opt/Tomcat	Tomcat	
/etc/profile.d/	Tomcat	Plik tomcat
/etc/rc.d/init.d/	Tomcat	Plik tomcat
/usr/share/solr/	SOLR	
/var/lib/solr/	SOLR	
/etc/solr/	SOLR – konfiguracja	
/etc/httpd/	Apache – konfiguracja	
/var/log/	Logi	Opcja

### Backup bazy danych

W celu wykonania kopii bazy danych należy wykonać:

```
./usr/lib/ckan/default/bin/activate
```

```
cd /usr/lib/ckan/default/src/ckan
```

```
paster db dump --config=/etc/ckan/default/production.ini production_database.pg_dump
```

### Restore bazy danych

W celu przywrócenia bazy danych z kopii należy wykonać:

```
./usr/lib/ckan/default/bin/activate
```

```
cd /usr/lib/ckan/default/src/ckan
```

```
paster db clean -c /etc/ckan/default/production.ini
```

```
paster db load -c /etc/ckan/default/production.ini my_database_dump.sql
```

Uwaga w przypadku nowej instancji CKAN przed db load należy wykonać :

```
paster db init -c /etc/ckan/default/production.ini
```

## 5.2 Function Server

### E) Jboss

Uruchomienie serwera JBOSS

```
service jbossas7 start
```

Wyłączenie serwera JBOSS

```
service jbossas7 stop
```

Restart serwera JBOSS

```
service jbossas7 restart
```

Przeładowanie serwera JBOSS

```
service jbossas7 reload
```

Sprawdzanie statusu serwera JBOSS

```
service jbossas7 status
```

Weryfikacja działania JBOSS

Uruchomić przeglądarkę i wpisać nazwę FQDN lub adres ip serwera i port 8080 – powinniśmy otrzymać dostęp do strony www.

Po wejściu na stronę [http://adres\\_ip\\_serwera:8080/cbr/mundo-java-backend/](http://adres_ip_serwera:8080/cbr/mundo-java-backend/) powinna się wyświetlić strona logowania:

Konsola administracyjna umożliwiająca zaawansowaną diagnostykę i konfigurację Jboss AS7 jest dostępna pod adresem [http://adres\\_ip\\_serwera:9990](http://adres_ip_serwera:9990)

## Backup/Restore

### F) Backup /Restore

#### Backup systemu plików

W celu wykonania backup konfiguracji i danych Function Server (CKAN) należy wykonać backup następujących katalogów:

Katalog	Zawartość	Uwagi
/usr/local/share/jboss	JBOSS	
/etc/init.d/	JBOSS	Plik jbossas7

#### Backup bazy danych

W celu wykonania kopii bazy danych należy wykonać poniższy skrypt:

```
export PGHOST=localhost
export PGDATABASE=java_backend
export PGUSER=postgres
export PGPASSWORD=tu wpisać hasło
pg_dump -a --inserts -f dataonly.sql
pg_dump --inserts -f all.sql
```

#### Restore bazy danych

W celu przywrócenia bazy danych z kopii należy wykonać:

```
psql java_backend < all.sql
```

lub w przypadku przywracanie jedynie danych:

```
psql java_backend < dataonly.sql
```

## 6 Podsumowanie

Niniejszy dokument zawiera dokumentację utrzymaniową platformy MUNDO. Należy podkreślić, iż zawartość rozdziałów 3 i 4 zawiera informacje z przykładowymi adresami IP nie zawiera również informacji o wewnętrznej strukturze sieci UM Warszawa w której zainstalowane są obie instancje platformy.

# 7 Słowniczek – lista skrótów

Glossary		
Nb.	Abbreviation	Explanation
1	API	Application Programming Interface
2	GPL	GNU Public License – Licencja wolnego i otwartego oprogramowania
3	LGPL	Lesser GNU Public License – słabsza powszechna licencja publiczna GNU, powszechna licencja publiczna dla bibliotek
4	FDL	Free Documentation License – Licencja Wolnej Dokumentacji
5	CC	Creative Commons
6	CKAN	Comprehensive Knowledge Archive Network – sieć archiwizacyjna dla kompleksowej wiedzy – system www będący repozytorium danych i eksponujący dane otwarte stworzony przez Open Knowledge Foundation
7	WS	Web Service – usługa sieciowa oparta o SOA/SOAP lub ROA/REST
8	SOA	architektura zorientowana na usługi (ang. Service Oriented Architecture)
9	SOAP	SOAP (ang.) Simple Object Access Protocol – protokół wywoływania zdalnego dostępu do obiektów oparty o XML
10	ROA	Architektura oparta o zasoby (ang. Resource-oriented architecture)
11	REST	Representational State Transfer – wzorzec architektury oprogramowania
12	MD	metadane (dane o danych),
13	WMS	Web Map Service – stworzony przez Open Geospatial Consortium (OGC) standard udostępniania map w postaci rastrowej za pomocą interfejsu HTTP
14	WFS	Web Feature Service stworzony przez Open Geospatial Consortium (OGC) standard udostępniania map w postaci wektorowej za pomocą interfejsu http
15	DS	DS – data set zbiór danych
16	FS	FS – function set – zbiór funkcji (np. realizowanych przez web service)
17	APIKey	Kod alfanumeryczny przekazanego do serwera API przez programy komputerowe wywołujące API
18	RDF	Ang. Resource Description Framework – język/metoda pozwalająca na opisywanie zasobów sieci Web, ze składnią opartą na XML, opracowana przez W3C
19	Dublin Core	Dublin Core Metadata Element Set, DC, DCE(S) – ogólny standard metadanych. Przyjęty jako standard ISO 15836-2003.
20	DCAT	W3C Data Catalog Vocabulary – standard schematu katalogowego dla danych
21	CSV	Comma separated values – plik z wartościami oddzielanymi przecinkami
22	CMS	system zarządzania treścią (content management system)
23	SPARQL	SPARQL Protocol And RDF Query Language – język zapytań i protokół dla plików RDF.
24	OWL	Web Ontology Language – język ze składnią opartą na XML, a semantyką

		opartą na logice opisowej
25	HTTP	Hypertext Transfer Protocol
26	SSL	Secure Socket Layer
27	BI UM Warszawa	Biuro Informatyki UM Warszawa

## 8 Bibliografia

- [1] Instalacja CKAN2.0 na CentOS 6.3 <https://github.com/ckan/ckan/wiki/How-to-install-CKAN-2.0-on-CentOS-6.3-%28new%29>
- [2] Instalacja Tomcat 6.0 na CentOS 7.0 <http://www.howtoforge.com/how-to-install-tomcat-on-centos-7>
- [3] Instalacja Tomcat 6 z JDK 7 na CentOS RHEL Fedora Debian Ubuntu <http://linuxdrops.com/install-tomcat-6-with-jdk-7-on-centos-rhel-fedora-debian-ubuntu/>
- [4] Instalacja Tomcat 6 na CentOS lub RHEL [http://www.davidghedini.com/pg/entry/install\\_tomcat\\_6\\_on\\_centos](http://www.davidghedini.com/pg/entry/install_tomcat_6_on_centos)
- [5] Instalacja CKAN Filestore <http://docs.ckan.org/en/latest/maintaining/filestore.html>
- [6] Instalacja CKAN DataStore <http://docs.ckan.org/en/latest/maintaining/datastore.html>
- [7] Instalacja CKAN DataPusher <http://docs.ckan.org/projects/datapusher/en/latest/>
- [8] CKAN filestore preview problem <https://github.com/ckan/ckan/pull/1160>
- [9] Wniosek do 1 konkursu Innowacje Społeczne NCBIR projektu MUNDO, Warszawa, 2013
- [10] Portal CKAN <http://ckan.org/> [30.10.2014]
- [11] Portal The Open Knowledge Foundation <https://okfn.org/> [30.10.2014]
- [12] Portal <https://www.digitalocean.com/community/tutorials/how-to-backup-postgresql-databases-on-an-ubuntu-vps> [30.07.2015]
- [13] PostgreSQL 9.1.18 Documentation Backup and Restore <http://www.postgresql.org/docs/9.1/static/backup.html> [30.07.2015]