

Δράση 2 - Ηλεκτρονική Πλατφόρμα για την παρακολούθηση δημόσιων συμβάσεων Οδηγός εγκατάστασης

Ιούλιος 2013



ΕΡΓΟ ΣΥΓΧΡΗΜΑΤΟΔΟΤΟΥΜΕΝΟ ΑΠΟ ΤΟ ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ (ΕΚΤ)



!

Περιεχόμενα

Περιεχόμενα.....	2
Έλεγχος εγγράφου.....	3
1 Εισαγωγή.....	4
2 Δράση 2 - frontend.....	5
2.1 Τεχνολογίες που χρειάζονται.....	6
2.2 Εγκατάσταση του κώδικα.....	6
3 Δράση 2 - Middleware.....	9
3.1 Τεχνολογίες που χρησιμοποιούνται.....	10
3.2 Εγκατάσταση του κώδικα.....	10
3.2.1 Ρυθμίσεις κώδικα.....	10
3.2.2 Ρυθμίσεις συστήματος.....	10
3.2.3 Ρυθμίσεις application server (Tomcat 6).....	10
3.2.4 Αλλαγές στην εγκατάσταση για production.....	12

Έλεγχος εγγράφου

Το έγγραφο είναι διαθέσιμο με άδεια [BY-SA](#)

Έκδοση εγγράφου	Ημερομηνία έκδοσης	Παρατηρήσεις
1.0	25/04/2013	Πρώτη έκδοση....
1.1	02/07/2013	Διορθώσεις και προσθήκη τελικού οδηγού εγκατάστασης

1 Εισαγωγή

Το αντικείμενο της συγκεκριμένης δράσης είναι η ανάπτυξη, παράδοση και τεκμηρίωση πρότυπης πλατφόρμας για την καταχώριση και παρακολούθηση συμβάσεων σε κεντρικό επίπεδο. Η πλατφόρμα θα ενσωματώνει βασικές ροές εργασίας για τη διαχείριση των συμβάσεων (αίτημα προμήθειας, αγαθού, έργου, υπηρεσίας, συμβασιοποίηση, πληρωμή) και θα δίνει τη δυνατότητα της προβολής του σχετικού περιεχομένου μέσα από δημόσιες επαφές.

Στο κείμενο που ακολουθεί παρουσιάζονται οι τεχνολογίες που χρησιμοποιούνται, καθώς και αναλυτικά βήματα για την εγκατάσταση της εφαρμογής.

2 Δομή

Όταν κατεβάσετε τον κώδικα αποτελείται απο τα εξής

- Drasi2-middleware: Περιέχει τα βασικά αρχεία που πρέπει να χρησιμοποιήσετε για το middleware
- Drasi2-frontend: Περιέχει τα βασικά αρχεία που πρέπει να χρησιμοποιήσετε για το frontend
- Drasi2 -yii: Χρησιμοποιείται από το frontend
- Drasi2-database: Η βάση που χρησιμοποιεί το middleware. Έχει 2 αρχεία α)μόνο με τη δομή και β) μαζί με τα δοκιμαστικά δεδομένα. Παράλληλα, περιέχει και τη βάση που μπορείτε να χρησιμοποιήσετε για τους φορείς και τα στοιχεία τους. Τα στοιχεία των βάσεων ορίζονται στο κεφάλαιο 2 [middleware]

Στο κείμενο που ακολουθεί περιγράφονται οι διαδικασίες που πρέπει να ακολουθήσετε για να εγκαταστήσετε τη δράση.

3 Δράση 2 - frontend

Το frontend έχει αναπτυχθεί σε yii framework, ένα λογισμικό ανοικτού κώδικα, το οποίο διατίθεται δωρεάν. Το yii framework στηρίζεται σε php και παρέχει ένα γρήγορο και αποδοτικό τρόπο για τη δημιουργία web 2.0 εφαρμογών.

3.1 Τεχνολογίες που χρειάζονται

Οι βασικές τεχνολογίες που χρησιμοποιούνται είναι

- PHP >= 5.3
- Yii framework >=1.1.7
- το captcha χρειάζεται gd και freeType της PHP
- χρειάζεται να είναι ενεργοποιημένο το extension της PHP mbstring. Χρειάζεται για να κάνει σωστά search στα CPV στα ελληνικά.

3.2 Εγκατάσταση του κώδικα

Τα βασικά βήματα που πρέπει να ακολουθήσει κανείς για την εγκατάσταση της εφαρμογής παρουσίασης είναι τα ακόλουθα

1. Το folder με το Yii framework ονομάζεται Yii και βρίσκεται στο ίδιο επίπεδο με την εφαρμογή
2. Στο αρχείο protected/config/main.php ορίζονται οι βασικές παράμετροι της εφαρμογής
 - a. 'name'=>'Πλατφόρμα καταχώρισης Αιτημάτων, Προκηρύξεων, Συμβάσεων & Εντολών Πληρωμής', το όνομα της εφαρμογής
 - b. 'language' => 'el', η γλώσσα που χρησιμοποιείται
3. Στο αρχείο protected/config/params.php αλλάζουν οι παράμετροι για την σύνδεση με την εφαρμογή καταχώρησης των δεδομένων (middleware)
 - a. `$urlMiddle = [PATH_MIDDLEWARE] /` (π . χ `'http://83.212.121.173:8080/agora/';`
 - b. 'agora' -> 'rsslimit' = 20 το όριο των εγγραφών που θα έχει το RSS
 - c. 'parameters'=>array('actionPermitted'=>86400), ο χρόνος που επιτρέπονται οι αλλαγές
 - d. 'files'=>array('maxsize'=> ini_get('post_max_size'), 'types'=>'pdf',) το όριο που

έχει ο apache για τα αρχεία , απλώς το εμφανίζει στον χρήστη για να ξέρει ποιο είναι το όριο των αρχείων pdf.

4. Στο αρχείο protected/models/Units.php πρέπει να ορίσετε το
 - `const _URL_UNITS = '[PATH_MIDDLEWARE]/ taxonomy/units/';`
 - `const _URL_SIGNERS = '[PATH_MIDDLEWARE]/taxonomy/signers/';`
 - `const _URL_ORGANIZATION = '[PATH_MIDDLEWARE]/taxonomy/oda/';`
5. Στο folder documents/ajax υπάρχουν αντίγραφα των στοιχείων των φορεων καθώς και ένα αρχείο για τα crn codes. Όταν πραγματοποιούνται αλλαγές στα αρχεία αυτά πρέπει να αντικαθίστανται. Τα στοιχεία ενημερώνονται από το opendata της Διαύγειας.
6. Τα ακόλουθα αρχεία και folder χρειάζονται δικαιώματα εγγραφής και να ανήκουν στον apache
 - assets
 - protected/runtime
 - protected/data/files
 - protected/data/files/final.xml (χρησιμοποιείται για την προσωρινή εγγραφή του αρχείου)
 - protected/data/auth.php (χρησιμοποιείται για την αυθεντικοποίηση των χρηστών)

3.3 Διαχείριση χρηστών

Για την τοπική διαχείριση χρηστών και δικαιωμάτων χρησιμοποιείται το RBAC μοντέλο.

Πιο συγκεκριμένα δημιουργούνται χρήστες με ρόλους. Κάθε πράξη είναι ένας ρόλος, ο οποίος ανατίθεται σε χρήστη. Η διαδικασία αυτή πραγματοποιείται στο protected/commands/RbacCommand.php. Τα στοιχεία αποθηκεύονται στο αρχείο protected/data/auth.php.

Κάθε φορά που ένας χρήστης κάνει login γράφεται στο αρχείο και του ανατίθενται ρόλοι. Αντίστοιχα με το logout διαγράφεται από το αρχείο.

Για να αρχικοποιηθεί αυτή η διαδικασία και να προστεθεί στο σύστημα πρέπει να εκτελέσετε τα ακόλουθα βήματα. Αρκεί να το κάνετε την πρώτη φορά.

- Σβήνετε το protected/data/auth.php

- στο folder protected τρέχετε ./yiiic rbac commands/RbacCommand.php . Προσοχή η εντολή yiiic πρέπει να είναι εκτελέσιμη. Αρκεί να αλλάξετε τα δικαιώματα.
- Δίνετε κατάλληλα δικαιώματα εγγραφής στο αρχείο protected/data/auth.php.

3.4 Άλλα στοιχεία

3.4.1 Σελίδες και κείμενα

Η εφαρμογή έχει κάποιες στατικές σελίδες και κείμενα - strings. Τα κείμενα τους καθορίζονται μέσα από τα αρχεία που βρίσκονται στο φάκελλο protected/messages/el. Πιο συγκεκριμένα:

- Procurement.php έχει όλα τα κείμενα που χρησιμοποιούνται στα αιτήματα
- Contract.php έχει όλα τα κείμενα που χρησιμοποιούνται στις συμβάσεις
- Rfp.php έχει όλα τα κείμενα που χρησιμοποιούνται στις προκηρύξεις
- Payments.php έχει όλα τα κείμενα που χρησιμοποιούνται στις εντολές πληρωμών
- Form.php γενικά κείμενα που χρησιμοποιούνται στις φόρμες
- Yii.php γενικά κείμενα που χρησιμοποιούνται σε όλη την εφαρμογή.

3.4.2 Theme

Το yii έχει ένα πολύ απλό theme το οποίο αποτελείται από τα εξής σημεία:

- Main.php ο βασικός κορμός που περιέχει όλα τα στοιχεία
- mainPerOrg.php ο βασικός κορμός που χρησιμοποιείται στους φορείς
- Column2.php η δεξιά μπάρα
- column2PerOrg.php η δεξιά μπάρα που χρησιμοποιείται στους φορείς.

Όλα αυτά τα στοιχεία βρίσκονται στο /protected/views/layouts/. Όλες οι σελίδες της εφαρμογής βρίσκονται στο protected/views/site.

4 Δράση 2 - Middleware

Το middleware υλοποιείται με Spring στη λογική του contract-first service. Τα XSD έχουν οριστεί στο `src/main/resources/schemas`. Πιο συγκεκριμένα, περιλαμβάνουν JAXB annotations ώστε ο jxc να παράγει classes με διαφορετικά ονόματα από αυτά που χρησιμοποιούνται εσωτερικά στο domain (προσθέτοντας την κατάληξη OXM, πχ ContractOXM)

Η επικοινωνία με το middleware γίνεται μέσω REST calls, με χρήση του public API (Δείτε: RESTful API). Ο χειρισμός του REST γίνεται από τους `ContractController`, `ProcurementRequestController`, `PaymentController` η βασική λειτουργικότητα των οποίων, κληρονομείται από τον `DecisionGenericController` με χρήση Spring MVC. Τα objects γίνονται αυτόματα unmarshal από XML μέσω JAXB, αλλά ΔΕΝ χρησιμοποιούνται εσωτερικά. Το business logic χρησιμοποιεί τα classes του `gr.opengov.agora.model`. Οι converters (`PublicOrganizationDecisionOXMConverter`, `ContractOXMConverter`, `ProcurementRequestOXMConverter`, `PaymentOXMConverter`, `CommonsOXMConverter`) μετατρέπουν τα OXM objects σε domain objects και αντίστροφα.

Για την αποθήκευση χρησιμοποιείται DAO pattern. Τα υλοποιημένα DAO χρησιμοποιούν Hibernate. Τα επισυναπτόμενα αρχεία (πχ PDF) αποστέλονται με Base64 encoding και αποθηκεύονται ξεχωριστά. Η αποθήκευση των δεδομένων γίνεται στο filesystem και για την αποθήκευση:

- Υπολογίζεται το SHA-256 hash του UID του εγγράφου που αφορά το document
- Οι 10 πρώτοι χαρακτήρες του hash χωρίζονται σε 5 τμήματα, μήκους 2 χαρακτήρων το καθένα (πχ aa-bb-cc-dd-ee)
- Δημιουργείται μια δομή directories με τα ονόματα των 5 τμημάτων (πχ aa/bb/cc/dd/ee)
- Το αρχείο αποθηκεύεται στο τελικό directory με όνομα hash.original
- Στο ίδιο directory αποθηκεύεται και μια μορφή του αρχείου με embedded το UID του εγγράφου, με όνομα hash.stamped

Τα μεταδεδομένα των αρχείων αποθηκεύονται στη σχεσιακή βάση, στον πίνακα DOCUMENT. Στην ίδια εγγραφή αποθηκεύονται πληροφορίες και για την original και για την stamped μορφή του αρχείου.

Όλες οι λειτουργίες απαιτούν πιστοποίηση. Η πιστοποίηση γίνεται μέσω Basic Authentication και τη χειρίζεται το Spring Security. Χρησιμοποιούνται τα στοιχεία πρόσβασης των μελών των ΟΔΕ της Διαύγειας μέσω ενός custom authentication provider (gr.opengov.agora.security.OdeAuthenticationProvider).

Το backend αποθηκεύει τα αιτήματα, τις προκηρύξεις, τις συμβάσεις, τις εντολές πληρωμών και ορισμένα βασικά μεταδεδομένα. Δεν συντηρεί ταξινομίες φορέων, μονάδων και τελικών υπογράφωντων. Αυτές αντλούνται δυναμικά από τη Διαύγεια.

4.1 Τεχνολογίες που χρησιμοποιούνται

Οι βασικές τεχνολογίες που χρησιμοποιούνται είναι

- tomcat6,
- libmysql-java (φέρνει apache dbcp),
- sun-java-jdk6

4.2 Εγκατάσταση του κώδικα

Στο κεφάλαιο αυτό παρουσιάζονται τα βήματα που πρέπει να ακολουθηθούν για να εγκατασταθεί ο κώδικας του middleware.

4.2.1 Ρυθμίσεις κώδικα

- Λήψη του κώδικα και εγκατάσταση του στο μηχάνημα
- Αντιγραφή του build.properties.default σε build.properties και προσαρμογή τοπικών ρυθμίσεων

4.2.2 Ρυθμίσεις συστήματος

- Εγκαθιστούμε τα el_GR.UTF-8 locales
- Δημιουργούμε μια MySQL database (πχ agora) με character set utf8. Η πρόσβαση στη βάση θα ρυθμιστεί μέσω JNDI datasource στον Tomcat

4.2.3 Ρυθμίσεις application server (Tomcat 6)

- Εγκατάσταση Apache DBCP

Διαγράφουμε τις default βιβλιοθήκες για connection pooling:
\$CATALINA_HOME/lib/tomcat-dbc.jar

Αντιγράφουμε το Apache DBCP (commons-dbc-1.4.jar και commons-pool-1.5.5.jar) στο \$CATALINA_HOME/lib

- Δημιουργία JNDI datasources: jdbc/apofaseisReader στη read-only βάση της Διαύγειας και jdbc/agora στη βάση της εφαρμογής. Για να μπορέσει να έχει πρόσβαση στα στοιχεία της Διαύγειας είναι απαραίτητη η σύνδεση με την βάση της διαύγειας. Μπορείτε να χρησιμοποιήσετε την τοπική ή να ζητήσετε να συνδεθείτε με την κεντρική βάση της Διαύγειας. Για το περιβάλλον οι ενδεικτικές ρυθμίσεις στο \$CATALINA_HOME/conf/context.xml είναι:

```
<Resource name="jdbc/apofaseisReader" auth="Container"
type="javax.sql.DataSource"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="100" maxIdle="30" maxWait="10000" username="USERNAME"
password="PASSWORD" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://xxx.xxx.xxx.xxx/apofaseis?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding
=UTF-8" removeAbandoned="true" logAbandoned="true"
testOnBorrow="true" validationQuery="SELECT 1"/>

<Resource name="jdbc/apofaseisWriter" auth="Container"
type="javax.sql.DataSource"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="100" maxIdle="30" maxWait="10000" username="USERNAME"
password="PASSWORD" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://xxx.xxx.xxx.xxx/apofaseis?
zeroDateTimeBehavior=convertToNull&useUnicode=true&characterEncoding
=UTF-8" removeAbandoned="true" logAbandoned="true"
testOnBorrow="true" validationQuery="SELECT 1"/>

<Resource name="jdbc/agora" auth="Container"
type="javax.sql.DataSource"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="100" maxIdle="30" maxWait="10000" username="USERNAME"
password="PASSWORD" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/agora?
useUnicode=true&characterEncoding=UTF-8" removeAbandoned="true"
logAbandoned="true" testOnBorrow="true" validationQuery="SELECT 1"/>
```

- Αντιγράφουμε τον Mysql Connector (mysql-connector-java-5.1.14-bin.jar) στο \$CATALINA_HOME/lib

4.2.4 Αλλαγές στην εγκατάσταση για production

- Εγκατάσταση tomcat6, libmysql-java (φέρνει apache dbcp), sun-java-jdk6 ή βάζουμε στο μηχανήμα το tomcat6 puppet module
- Τα resources τα ορίζουμε στο /etc/tomcat6/context.xml
- Στο /etc/tomcat6/catalina.properties αλλάζουμε την γραμμή common.loader=.. ώστε να περιέχει στο τέλος το path του mysql connector:

```
Από common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/lib,${catalina.home}/lib/*.jar,/var/lib/tomcat6/common/classes,/var/lib/tomcat6/common/*.jar
```

```
Σε common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/lib,${catalina.home}/lib/*.jar,/var/lib/tomcat6/common/classes,/var/lib/tomcat6/common/*.jar,/usr/share/java/mysql-connector-java-5.1.10.jar
```