

Adatbázis-alapú entity-resolution szoftvermodul

Telepítési dokumentáció

Tartalomjegyzék

Tartalomjegyzék.....	2
1. Bevezető.....	3
1.1. A dokumentum áttekintése.....	3
1.2. célközönség.....	3
2. Szótár.....	3
3. Architektúrális áttekintő.....	3
3.1. Adatlap.....	3
4. Szoftver telepítése.....	5
4.1. Futató környezet kialakítása.....	5
4.2. Mysql séma telepítése.....	5
4.3. Szoftvermodul beállítása.....	5
4.3.1. Email küldése hibákról:.....	5
4.3.2. Maven profile.....	6
4.3.3. Naplófájl beállítások:.....	8
4.4. Szoftvermodul Buidelése.....	11
4.4.1. Külső jar.....	11
4.4.2. Kód buidelése.....	11
4.5. Szoftvermodul telepítése.....	12
4.5.1. War bemásolása.....	12

1. Bevezető

1.1. A dokumentum áttekintése

Jelen dokumentum célja az Adatbázis-alapú entity-resolution szoftvermodul alkalmazás telepítési leírása.

1.2. célközönség

Azok a fejlesztők, akik az alkalmazás karbantartásában, fejlesztésében részt vettek, részt fognak venni.

2. Szótár

A dokumentumban a következő fogalmak kerülnek használatra; a könnyebb áttekinthetőség és érthetőség érdekében az alábbiakban kerülnek bemutatásra, megfogalmazásra:

Elnevezés	Magyarázat
MAVEN	maven program, ami buildelést végzi
Tomcat	tomcat webserver

3. Architektúrális áttekintő

3.1. Adatlap

Aspektus	Választott megoldás(ok)
Szoftveres környezet	Java SE 8, Spring MVC 1.5.2
Architektúra	Kliens / szerver
Felhasználói felület	nincs
Perzisztencia réteg	A rendszer futásához szükséges entitások, potenciális duplikátumok tárolására

	szolgáltató adatbázis; az implementáció adatbázis-agnosztikus, kizárólagos követelmény, hogy támogassa a megszokott indexelés, keresési, standard ANSI SQL-es szabványokat
Alkalmazáserver	Tomcat 8
Interface-ek, API	Standard java 8 interface-ek, illetve REST (CRUD) API
Nyelvek	Java

4. Szoftver telepítése

A küldött fordított fájl előre beállított beállításokkal rendelkezik, ha a szoftver számára más beállításokra van szükség, akkor be kell állítani a megfelelő beállításokat és buildelni kell a forrás kódot, majd telepíteni azt.

4.1. Futató környezet kialakítása

A program windows és linux alapu gépen is futtatható.

Az ajánlott futtató környezet linux alapu operációs rendszer.

Lépések:

- java8 OpenJDK telepítése
- mysql telepítése
- mysql adatbázis és felhasználók létrehozása és beállítása
- mysql adatbázis séma importálása
- mysql adatbázis inicializálása
- tomcat8 feltelepítése és beállítása
- a document modul program számára hibahatóságát biztosítani kell, pl.: tűzfal beállítással

4.2. Mysql séma telepítése

A telepítő csomag része az adatbázis séma. Az init.sql fájlt kell beimportálni.

4.3. Szoftvermodul beállítása

A szoftver számára szükség van adatbázisra és adatbázis felhasználóra, amivel használja azt, amikor ezek elkészültek, akkor azokat be kell állítani a program forráskódjában és buildelni kell.

A forráskód organization-app mappába a pom.xml fájlban találhatóak a beállítások.

4.3.1. Email küldése hibákról:

Ki/Be kapcsolható hogy hibákról email értesítés jöjjön.

A pom.xml-be a lenti beállítások kezelik a hiba email küldést.

```
<log.email.global.level>OFF</log.email.global.level>
<log.email.global.smtpHost>example.hu</log.email.global.smtpHost>
<log.email.global.smtpPort>25</log.email.global.smtpPort>
<log.email.global.smtpProtocol>smtp</log.email.global.smtpProtocol>
<log.email.global.username></log.email.global.username>
<log.email.global.password></log.email.global.password>
<log.email.global.to>uzemeltetes@example.hu</log.email.global.to>
<log.email.global.from>uzemeltetes@example.hu</log.email.global.from>
```

level = ON/OFF értékű lehet és ezzel lehet be/ki kapcsolni funkciót

smtpHost = itt lehet beállítani mi legyen az SMTP domain

smtpPort = SMTP portja

smtpProtocol = SMTP protokollja

username = SMTP fiók felhasználó neve

passord = SMTP fiók jelszava

to = erre a címre mennek ki levelek

from = erről a címről jönnek a levelek

4.3.2. Maven profile

Ez tartalmazza a beállításokat, amiket használ program és „\${változo neve}” lehet hivatkozni más változókra.

profile.name = profile neve

id = profile idje

database.username = mysql szerver felhasználó neve, amit használ

database.password = mysql szerver adatbázis felhasználó jelszava, amit használ

database.port = mysql szerver portja

database.dbname = mysql szerver adatbázis, amit használ szoftver

database.location = mysql szerver elérése

database.show_sql = false/true értéke lehet ezzel lehet állítani mutassa-e logba a mysql queryket

database.auto = amikor indul a program, akkor mit csináljon adatbázissal: create/update

database.init = false, amikor indul program akkor inicializálja-e adatbázist.

tomcat.host = a szoftver futtató tomcat hostja

tomcat.port = a szoftvert futtató tomcat portja

document.host= a külső document modul hostja ahol fut

document.port = a külső document modul port ahol fut

document.apiKey = a külső document modul API key-e amivel tud autentikálni a document modulba, ez van minden document kérés esetén headerbe elküldve.

resolve.uuidEnabled = az UUID feloldó modul be /ki kapcsolása, true/false értéke lehet.

resolve.euTaxNumberEnabled = az Eu adószám feloldó modul be/ki kapcsolása, true/false értéke lehet.

resolve.threads = mennyi szálon végezze a feloldást, pl.: egyszerre hány példányba kérdezi le a cég információkat.

resolve.uuidScheduleRate = UID ütemező esetén mennyi legyen újra ütemezési idő

resolve. euTaxNumberScheduleRate = UID ütemező esetén mennyi legyen újra ütemezési idő

organization.apiKey = API key amivel lehet hívni a szoftver REST API interface-t.

war.moduleName = tomcat WAR modul neve, ez bele kerül a war mint nev

war.path = a tomcat számára ezen a néven lesz deployolva.

application.logdir = ebbe a könyvtára fogja rakni a naplófájlokat, amit készít.

cors.allowed.origins = ezekről a domainnek esetén engedélyezett a CORS kérés

```
<profile>
<id>prod</id>
<properties>
<profile.name>prod</profile.name>
<database.username>organization</database.username>
```

```

        <database.password>...</database.password>
    <database.location>organization_db</database.location>
        <database.port>3306</database.port>
    <database.dbname>organization</database.dbname>
        <database.show_sql>false</database.show_sql>
    <database.format_sql>true</database.format_sql>
        <database.auto>update</database.auto>
        <database.init>false</database.init>
        <tomcat.host>...</tomcat.host>
        <tomcat.username>...</tomcat.username>
        <tomcat.password>...</tomcat.password>
    <document.host>organization.hu</document.host>
        <document.port>...</document.port>
        <document.context></document.context>
        <document.apiKey>.....</document.apiKey>
    <resolve.uuidEnabled>true</resolve.uuidEnabled>
    <resolve.euTaxNumberEnabled>true</resolve.euTaxNumberEnabled>
        <resolve.threads>2</resolve.threads>
    <resolve.uuidScheduleRate>3000</resolve.uuidScheduleRate>
    <resolve.euTaxNumberScheduleRate>3000</resolve.euTaxNumberScheduleRate>

    <organization.apiKey>...</organization.apiKey>

    <war.moduleName>Dev</war.moduleName>
    <war.path>${project.artifactId}</war.path>

    <application.logdir>/home/data/logs/${project.artifactId}</application.logdir>

        <log.email.level>All</log.email.level>
    <log.email.smtpHost>${log.email.global.smtpHost}</log.email.smtpHost>
        <log.email.smtpPort>${log.email.global.smtpPort}</log.email.smtpPort>
    <log.email.smtpProtocol>${log.email.global.smtpProtocol}</log.email.smtpProtocol>
    <log.email.username>${log.email.global.username}</log.email.username>
    <log.email.password>${log.email.global.password}</log.email.password>
        <log.email.to>${log.email.global.to}</log.email.to>
        <log.email.from>${log.email.global.from}</log.email.from>

    <cors.allowed.origins>http://localhost</cors.allowed.origins>
    </properties>
</profile>

```

4.3.3. Naplófájl beállítások:

A organization-app/src/main/resources mappába a log4j.xml tartalmazza.

```

<Configuration>

    <Properties>
        <Property name="logDir">${application.logdir}</Property>
        <Property name="datePattern">%d{yyyy-MM-dd}</Property>
        <Property name="filteredPackages">

```



```

org.springframework,sun.reflect,org.apache.javax.servlet,java.lang.reflect,com.sun.proxy,java.util.concurrent,java.lang

</Property>
<Property name="logEntryPattern">%d{HH:mm:ss} %p %marker %c{3.} %m%n</Property>
<!-- NOTE: a log-ba bekerül az artifactId is; ez csak olyan esetben szükséges, amikor egy fájlba
több bejegyzés alkalmazás is naplózik -->
<Property name="logEntryPattern_minified">%d{HH:mm:ss} %p %c{1.} ${artifactId}
%xEEx{filters(${filteredPackages})} %m%n
</Property>
<Property name="level">${log.email.level}</Property>
<Property name="logEntryPattern_minified">%d{HH:mm:ss} %p %c{1.} ${artifactId}
%xEEx{filters(${filteredPackages})} %m%n
</Property>
<Property name="smtpHost">${log.email.smtpHost}</Property>
<Property name="smtpPort">${log.email.smtpPort}</Property>
<Property name="smtpProtocol">${log.email.smtpProtocol}</Property>
<Property name="smtpUsername">${log.email.username}</Property>
<Property name="smtpPassword">${log.email.password}</Property>
<Property name="to">${log.email.to}</Property>
<Property name="from">${log.email.from}</Property>
</Properties>
<Appenders>
<Console name="Console" target="SYSTEM_OUT">
<PatternLayout pattern="${logEntryPattern_minified}"/>
<ThresholdFilter level="ERROR"/>
</Console>
<RollingFile name="FullFile" fileName="${logDir}/full.log"
filePattern="${logDir}/full-${datePattern}.log.gz">
<PatternLayout pattern="${logEntryPattern}"/>
<Policies>
<TimeBasedTriggeringPolicy/>
</Policies>
<!-- only keep last 30 day -->
<DefaultRolloverStrategy max="30"/>
</RollingFile>

<RollingFile name="WarnFile" fileName="${logDir}/warn.log"
filePattern="${logDir}/warn-${datePattern}.log.gz">
<ThresholdFilter level="WARN" onMatch="ACCEPT" onMismatch="DENY"/>
<PatternLayout pattern="${logEntryPattern_minified}"/>
<Policies>
<TimeBasedTriggeringPolicy/>
</Policies>
<!-- only keep last 30 day -->
<DefaultRolloverStrategy max="30"/>
</RollingFile>

<SMTP name="Email"
subject="KHfnew - ${artifactId} - ${profile.name} - ERROR Log"
to="${to}"
from="${from}"

```

```

smtpHost="${smtpHost}" smtpPort="${smtpPort}" smtpProtocol="${smtpProtocol}"
    bufferSize="50"
smtpUsername="${smtpUsername}" smtpPassword="${smtpPassword}"
    smtpDebug="false">
    </SMTP>
</Appenders>
<Loggers>
    <Root level="info">
        <AppenderRef ref="Console"/>
        <AppenderRef ref="FullFile"/>
        <AppenderRef ref="WarnFile"/>
        <AppenderRef ref="Email" level="${level}"/>
    </Root>
    <!--HtmlUnitDriver-->
    <Logger name="com.gargoylesoftware" level="fatal"/>
    <!-- Log all SQL DML statements as they are executed -->
    <Logger name="org.hibernate.SQL" level="info"/>
    <!-- Log all JDBC parameters -->
    <Logger name="org.hibernate.type" level="warn"/>
    <!-- Log all SQL DDL statements as they are executed -->
    <Logger name="org.hibernate.tool.hbm2ddl" level="debug"/>
    <!-- Log the state of all entities (max 20 entities) associated with the session at flush time -->
    <Logger name="org.hibernate.pretty" level="warn"/>
    <!-- Log all second-level cache activity -->
    <Logger name="org.hibernate.cache" level="warn"/>
    <!-- Log transaction related activity -->
    <Logger name="org.hibernate.transaction" level="debug"/>
    <!-- Log all JDBC resource acquisition -->
    <Logger name="org.hibernate.jdbc" level="warn"/>
    <!-- Log HQL and SQL ASTs during query parsing -->
    <Logger name="org.hibernate.hql.ast.AST" level="debug"/>
    <!-- Log all JAAS authorization requests -->
    <Logger name="org.hibernate.secure" level="warn"/>
    <!-- Log everything (a lot of information, but very useful for troubleshooting) -->
    <Logger name="org.hibernate" level="warn"/>
</Loggers>
</Configuration>

```

4.4. Szoftvermodul Buidelése

4.4.1. Külső jar

A buidelés feltétele hogy segéd Jar-ok fel vannak telepítve és maven számára elérhetőek.

<https://maven.apache.org/guides/mini/guide-3rd-party-jars-local.html> ezen a linken levan írva hogyan lehet telepíteni külső jarokat. (mvn install:install-file -Dfile=<path-to-file> -DpomFile=<path-to-pomfile>)

A telepítő csomag lib mappájában található a libek, amiket be kell rakni.

Ha ezeken kell változtatni, a fejlesztők számára elérhetővé tudja tenni cég a forráskódot ezek számára.

A lenti kettő jart kell installálni a maven számára.

```
<organization.commons.version>1.0</organization.commons.version>
<dependency>
    <groupId>hu.organization</groupId>
    <artifactId>organization-common-aspects</artifactId>
    <version>${organization.commons.version}</version>
</dependency>

<dependency>
    <groupId>hu.organization</groupId>
    <artifactId>organization-common-mail</artifactId>
    <version>${organization.commons.version}</version>
</dependency>
```

4.4.2. Kód buidelése

A buidelés feltétele hogy segéd libek maven számára elérhetőek, és app megfelelően be van állítva.

lépései

- be kell lépni organization mappába
- futtatni kell maven install -P [profile név] parancsot

A program kód alapba tartalmazza a „prod” nevű profile-t.

A maven buildeli a organization-data és organization-app modulokat.

A tomcat számára a war fájl a organization-app/target mappába lesz található.

4.5. Szoftvermodul telepítése

Előfeltétele hogy futatási környezet be van állítva, document modul elérhető, mysql adatbázis elérhető és fut.

A megfelelő beállításokat tartalmazó fordított WAR fájl megvan.

A war fájlt deployolni kell tomcat-be, amit meg lehet többféle képpen csinálni.

4.5.1. War bemásolása

A war fájlt be kell másolni a tomcat webapps mappába és restartolni kell tomcat-et, ekkor automatikusan installálja programot és elérhető lesz az app: a `http://[tomcathost]:[port]/[war fájlneve]` url-en.