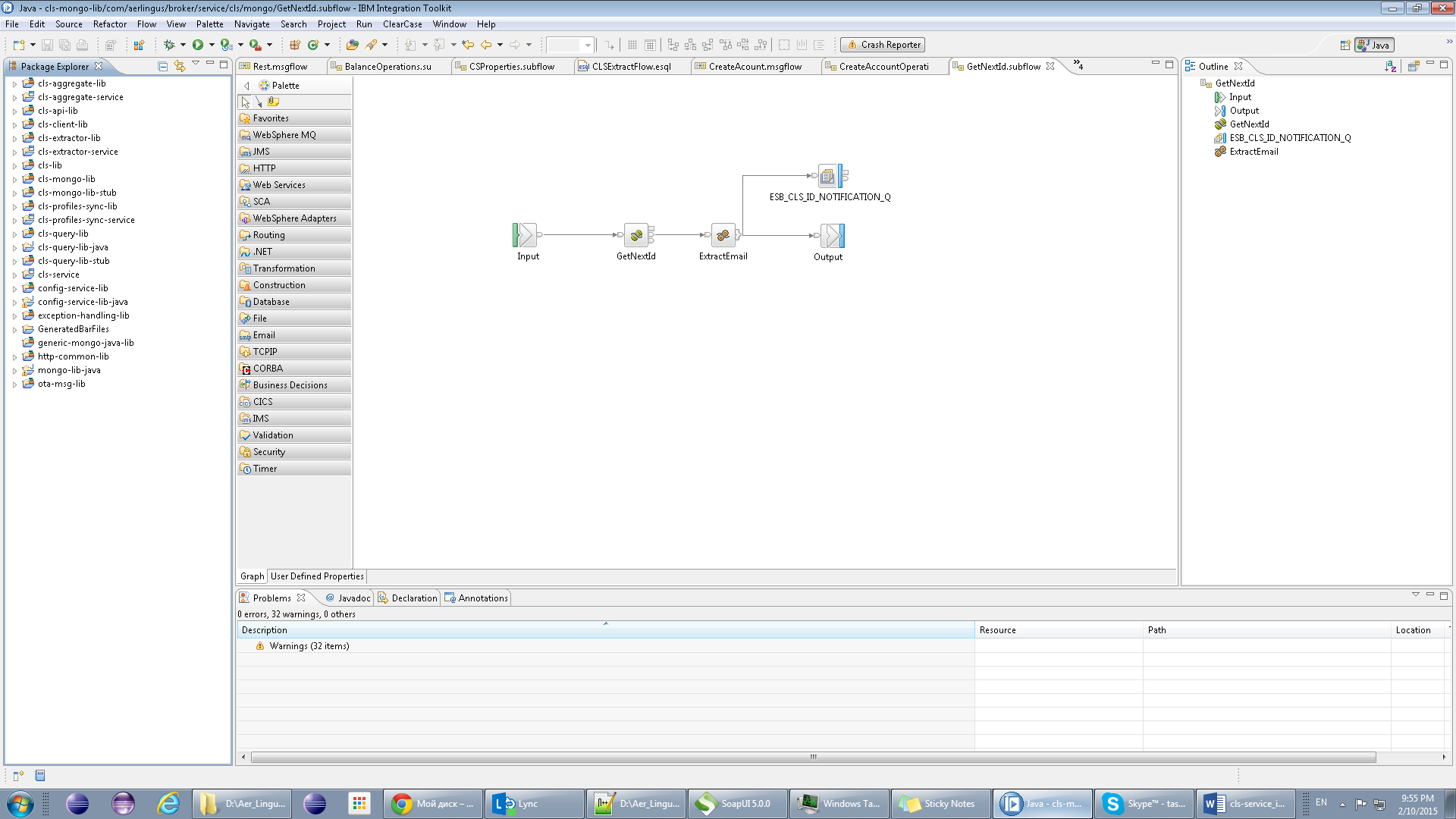
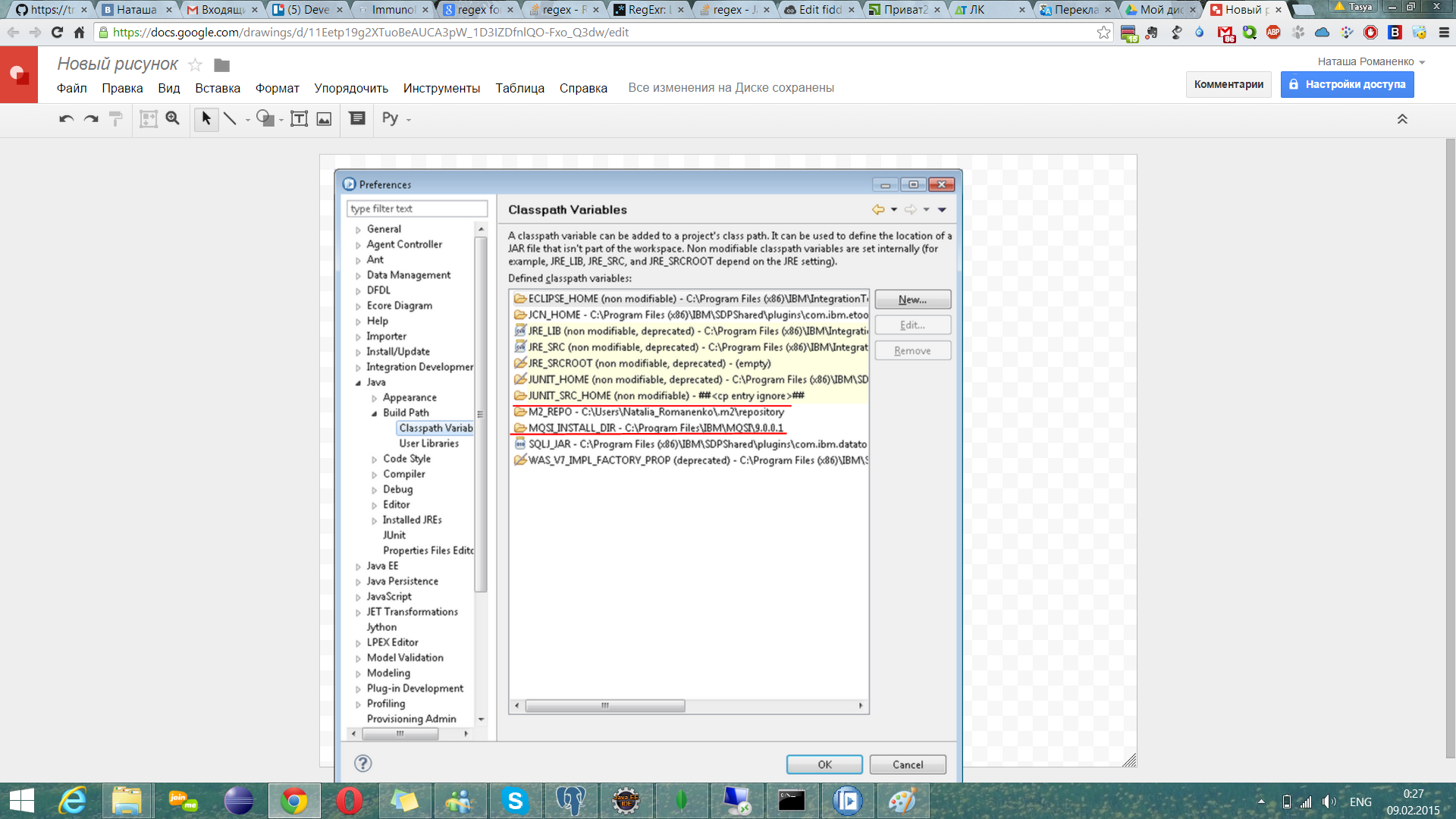
CLS configuration and setup from scratch

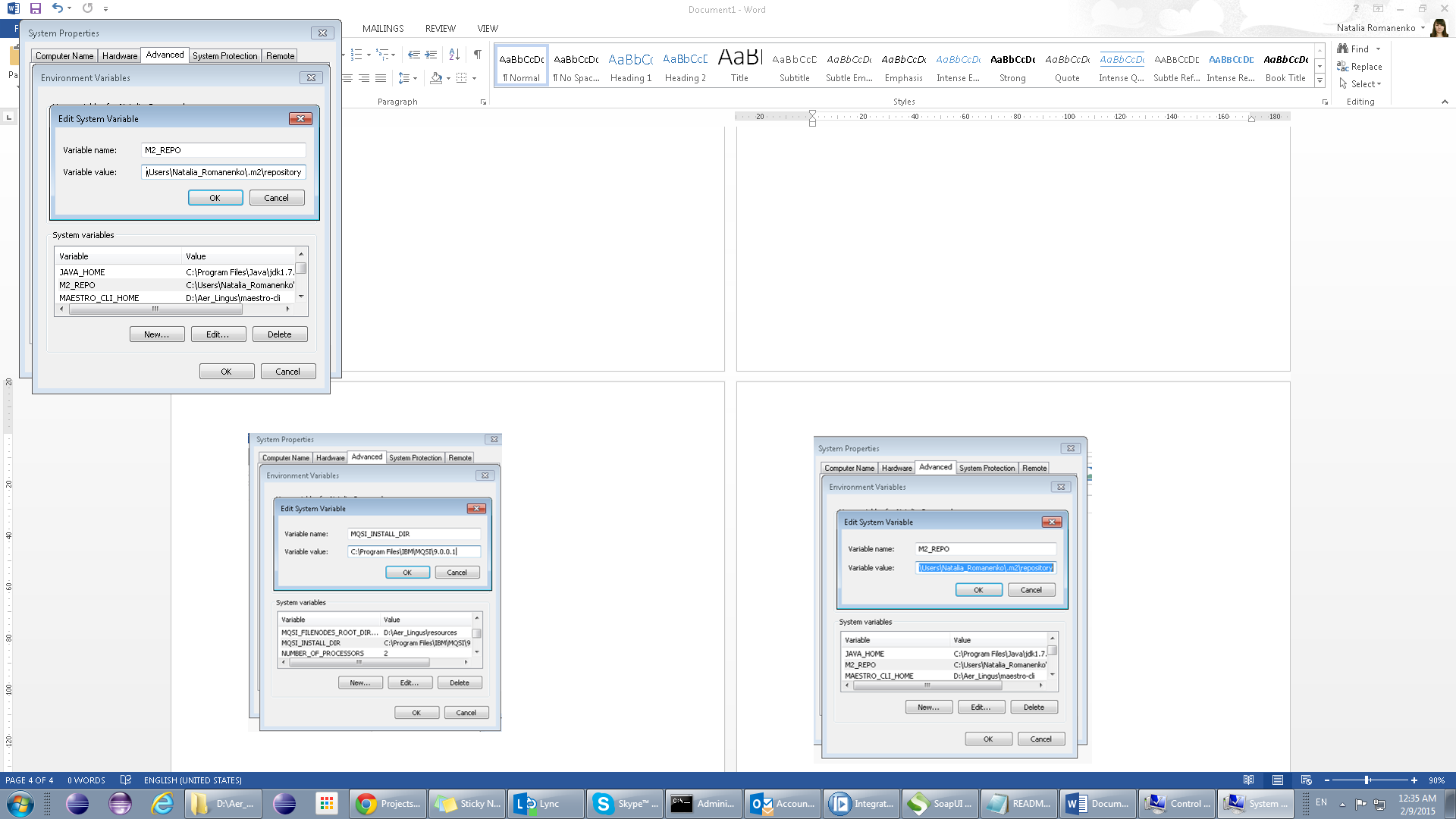
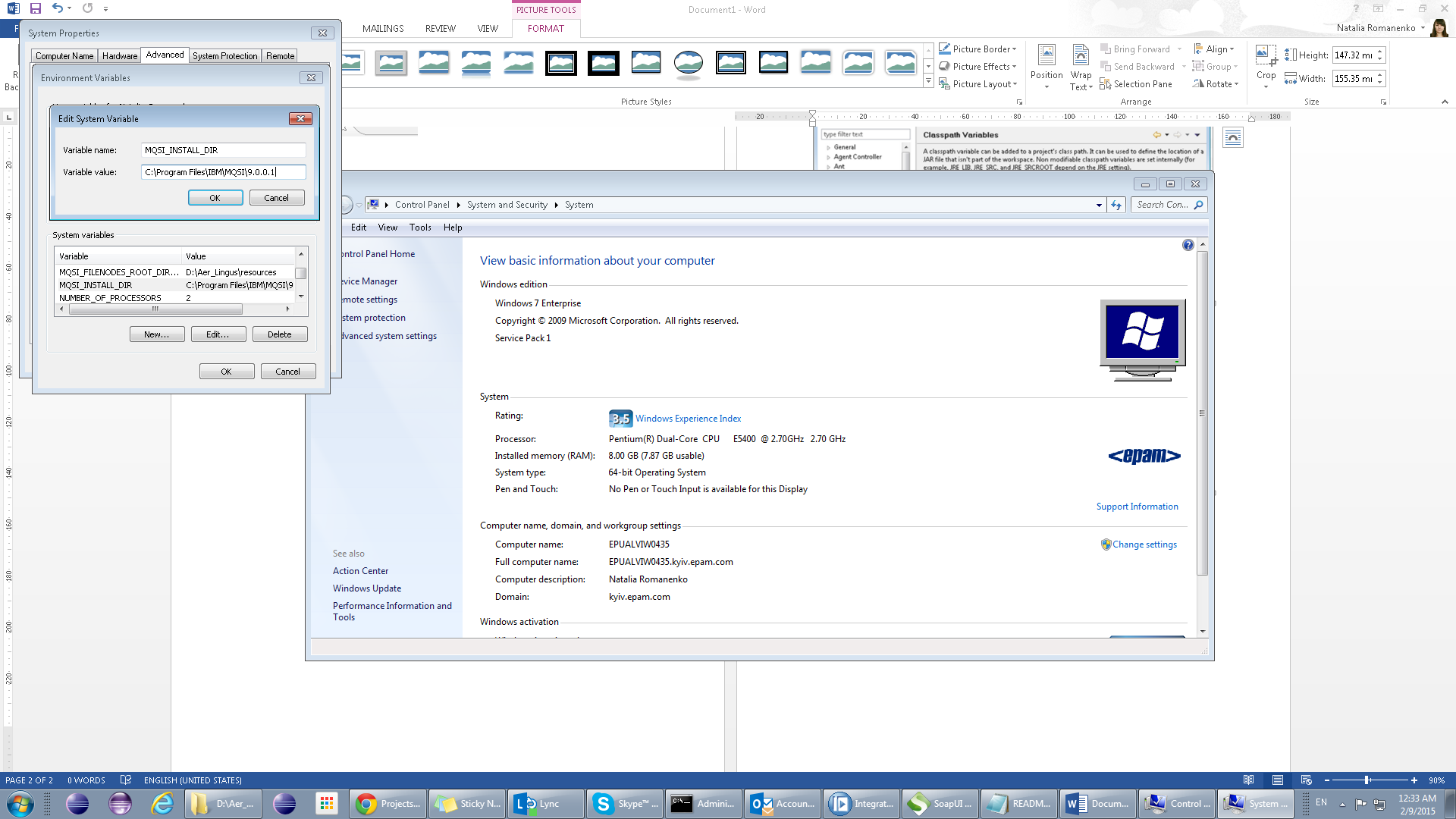
1. Import cls-service, cls-aggregate-service, cls-extractor-service, cls-profiles-sync-service and related libraries:



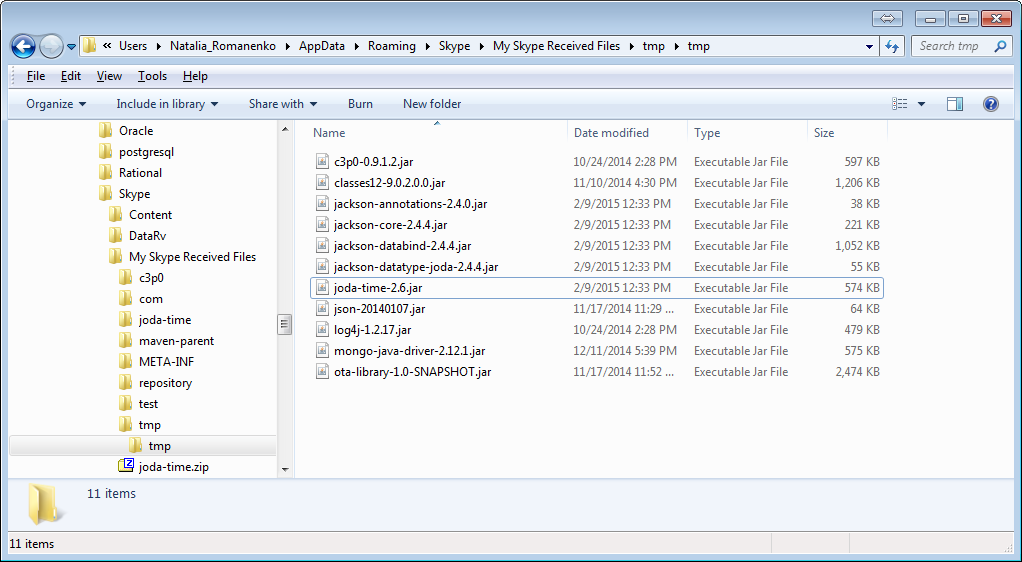
1. Create M2\_REPO & MQSI\_INSTALL\_DIR variables in Eclikkkpse IDE:



1. Create Environment variables M2\_REPO & MQSI\_INSTALL\_DIR:



1. Run mvn dependency:resolve on ‘cls-profiles-sync-service’, ‘cls-extractor-service’, ‘cls-extractor-service’, ‘mongo-lib-java’ folders
2. Put jars into classes folder (C:\Program Files\IBM\MQSI\9.0.0.1\classes) :



**Restart your integration server**

1. Run mvn install commands on classes folder(C:\Program Files\IBM\MQSI\9.0.0.1\classes):

* mvn install:install-file -Dfile=jplugin2.jar -DgroupId=com.ibm -DartifactId=jplugin2 -Dversion=9.0.0.1 -Dpackaging=jar
* mvn install:install-file -Dfile=javacompute.jar -DgroupId=com.ibm -DartifactId=javacompute -Dversion=9.0.0.1 -Dpackaging=jar
* mvn install:install-file -Dfile=classes12-9.0.2.0.0.jar -DgroupId=com.oracle -DartifactId=classes12 -Dversion=9.0.2.0.0 -Dpackaging=jar

After 4-6 steps you must have those jars in maven repository:

\.m2\repository\com\oracle\classes12\9.0.2.0.0\classes12-9.0.2.0.0.jar

\.m2\repository\c3p0\c3p0\0.9.1.2\c3p0-0.9.1.2.jar

\.m2\repository\log4j\log4j\1.2.17\log4j-1.2.17.jar

\.m2\repository\com\aerlingus\ota-library\1.0-SNAPSHOT\ota-library-1.0-SNAPSHOT.jar

\.m2\repository\com\fasterxml\jackson\core\jackson-annotations\2.4.0\jackson-annotations-2.4.0.jar

\.m2\repository\com\fasterxml\jackson\core\jackson-core\2.4.4\jackson-core-2.4.4.jar

\.m2\repository\com\fasterxml\jackson\core\jackson-databind\2.4.4\jackson-databind-2.4.4.jar

\.m2\repository\com\fasterxml\jackson\datatype\jackson-datatype-joda\2.4.4\jackson-datatype-joda-2.4.4.jar

\.m2\repository\joda-time\joda-time\2.6\joda-time-2.6.jar

\.m2\repository\org\json\json\20140107\json-20140107.jar

\.m2\repository\org\mongodb\mongo-java-driver\2.12.1\mongo-java-driver-2.12.1.jar

\.m2\repository\commons-io\commons-io\2.4\commons-io-2.4.jar

\.m2\repository\de\flapdoodle\embed\de.flapdoodle.embed.mongo\1.46.4\de.flapdoodle.embed.mongo-1.46.4.jar

\.m2\repository\de\flapdoodle\embed\de.flapdoodle.embed.process\1.40.1\de.flapdoodle.embed.process-1.40.1.jar

\.m2\repository\net\java\dev\jna\jna-platform\4.0.0\jna-platform-4.0.0.jar

\.m2\repository\net\java\dev\jna\jna\4.0.0\jna-4.0.0.jar

\.m2\repository\org\apache\commons\commons-compress\1.3\commons-compress-1.3.jar

\.m2\repository\org\apache\commons\commons-lang3\3.1\commons-lang3-3.1.jar

\.m2\repository\com\ibm\jplugin2\9.0.0.1\ jplugin2-9.0.0.1.jar

\.m2\repository\com\ibm\javacompute\9.0.0.1\ javacompute-9.0.0.1.jar

1. Install Mongo 2.6 version
2. Create data\db folder
3. Run Mongo (mongod --dbpath data\db) or run like windows service.
4. Add Mongo collections:

* 'cls\_extractor\_job\_status'
* 'seq’
* ‘cls\_new\_customer\_numbers’
* ‘profiles’

1. Add documents in mongo collections:

* Initialize ‘seq’ collection:
  + - {"\_id" : "cls\_extractor\_job\_status","seq" : NumberLong(0)}
* Initialize ‘cls\_new\_customer\_numbers’ collection:
  + - {"batch\_sequence\_no" : 1, "batch\_status" : "O", "first\_cls\_number" : NumberLong(0), "last\_cls\_number" : NumberLong(59999), "next\_cls\_number" : NumberLong(1)}

1. Create configurable services:

* **name: CLSConnection**

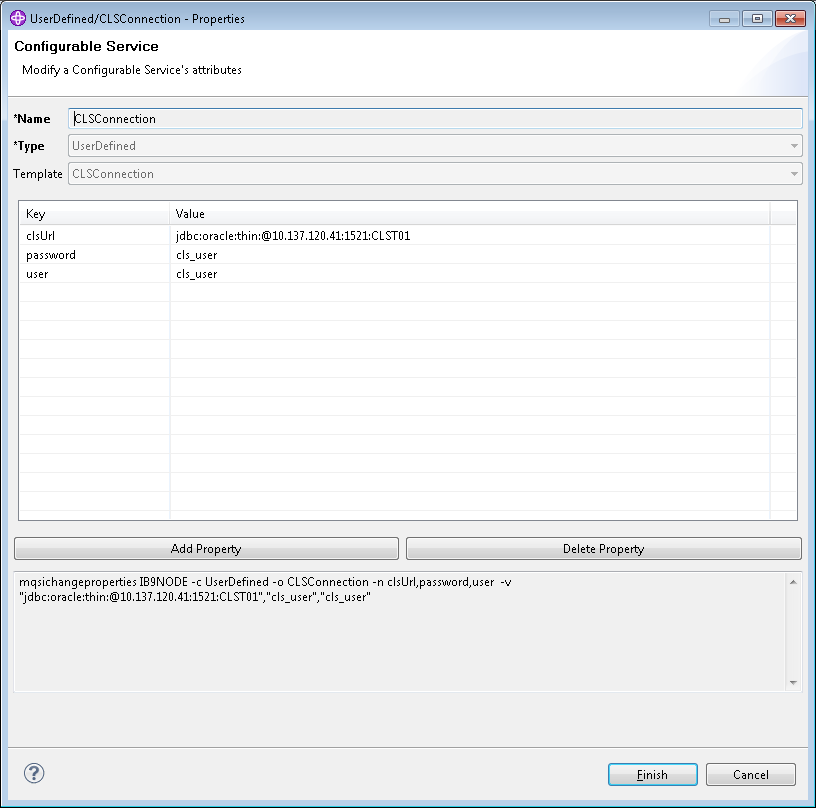
type: UserDefined

key-values for test configuration:

clsUrl: jdbc:oracle:thin:@10.137.120.41:1521:CLST01

user: cls\_user

password: cls\_user



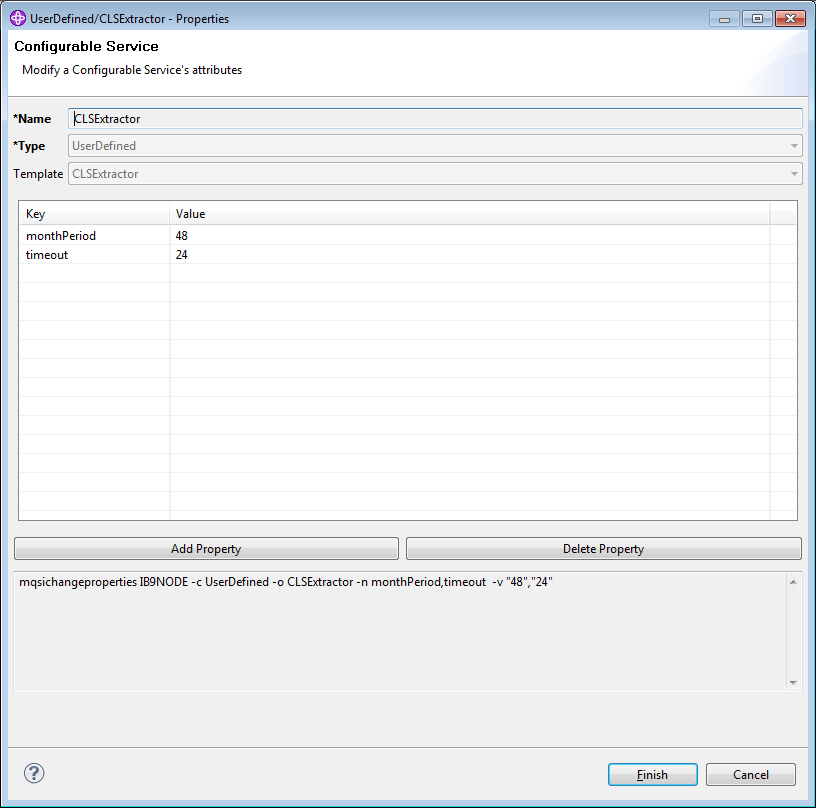
* **name: CLSExtractor**

type: UserDefined

key-values for test configuration:

monthPeriod: 48 (it should be set for period of retrieving customer points age)

timeout: 24 (uses in cls-extractor-service)



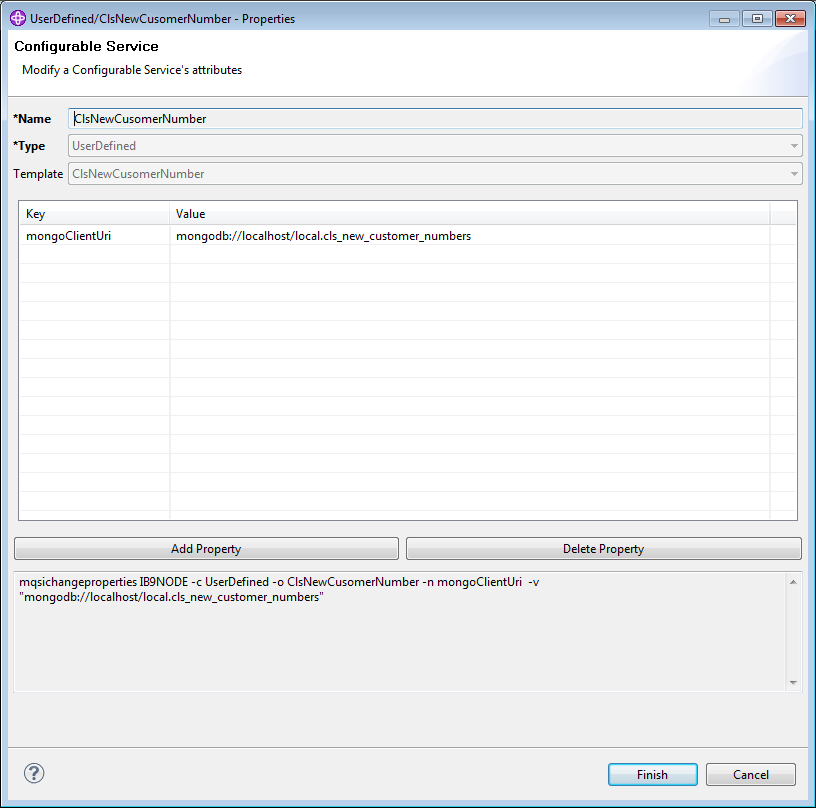
For next configurable services use your mongo [url: mongodb://mongo\_username:mongo\_password@host:port/name\_of\_database](file:///C:\Users\Наталя\Downloads\'mongodb:\mongo_username:mongo_password@localhost:27017\name_of_database):

* **name: ClsNewCusomerNumber**

type: UserDefined

key-values for test configuration:

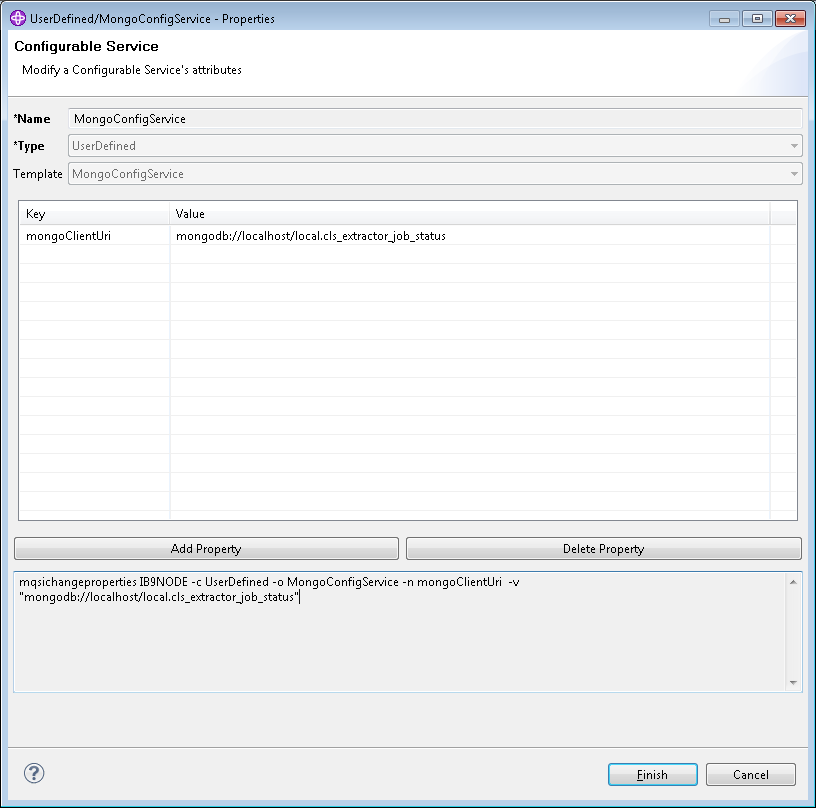
mongoClientUri: mongodb://localhost/local. cls\_new\_customer\_numbers(for generate customer id)



* **name: MongoConfigService**

type: UserDefined

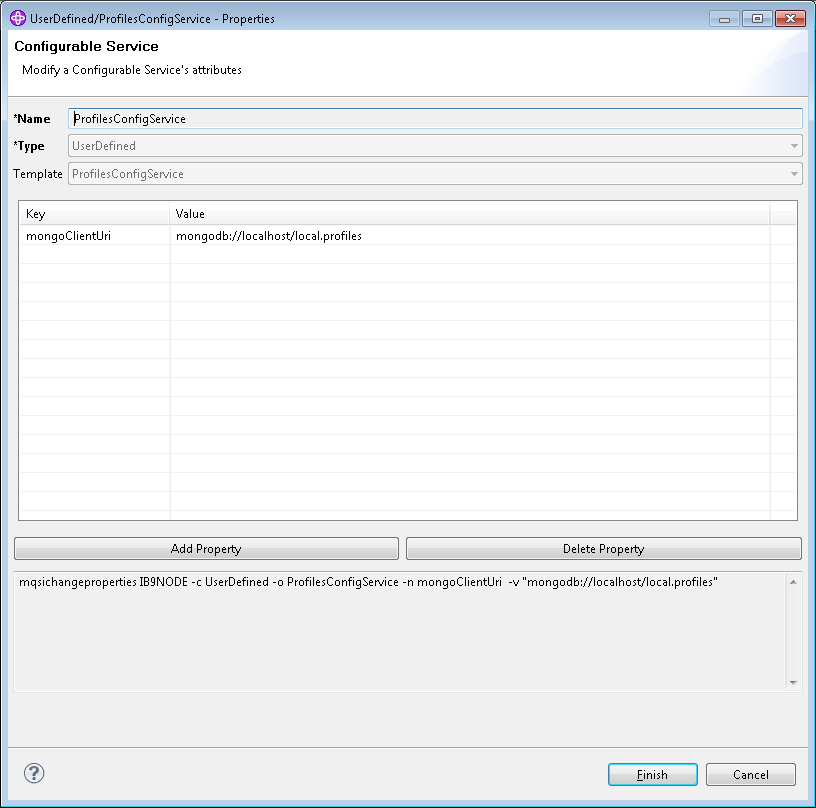
mongoClientUri - path to job status collection(mongodb://localhost/local.cls\_extractor\_job\_status)



* **name: ProfilesConfigService**

type: UserDefined

mongoClientUri - path to profile collection(mongodb://localhost/local.profiles)



1. Required queues:

ESB\_AUDIT\_Q

ESB\_CLS\_ACCOUNT\_DB\_INSERT\_Q

ESB\_CLS\_ACCOUNT\_IN\_Q - CLS

ESB\_CLS\_ACCOUNT\_OUT\_Q

ESB\_CLS\_BALANCE\_IN\_Q - CLS

ESB\_CLS\_BALANCE\_OUT\_Q

ESB\_CLS\_ID\_NOTIFICATION\_Q

ESB\_CLS\_REDEMPTION\_IN\_Q

ESB\_CLS\_REDEMPTION\_OUT\_Q

ESB\_CLS\_PROFILES\_OUT\_Q

ESB\_CLS\_PROFILES\_IN\_Q

ESB\_CLS\_LOOP\_Q

1. In case of any errors clean you workspace
2. Build test and dev bars on cls-service(use command: mvn clean install)
3. Deploy test bar on broker; import test project(D:\Aer\_Lingus\branch\_repo\_cls\aerl-mobl-esb\cls-service\test\cls-service-soapui-project.xml) to soapUI and run tests or run with maven command:

com.smartbear.soapui:soapui-maven-plugin:5.0.0:test -Dhost=localhost:7080

(-Dhost=localhost:7080 – is a parameter for overriding hostname and port properties in cls-service-soapui-project.xml project)

1. Deploy dev bar. To try account creation, getting balance and redemption follow the instructions below.
2. To create loyaltyAccount:

- Path: 'POST /loyalties/createAccount'

- Request: {'http://www.opentravel.org/OTA/2003/05'}: OTA\_LoyaltyAccountCreateRQ

- Response: {'http://www.opentravel.org/OTA/2003/05'}:OTA\_LoyaltyAccountRS

Request:

<OTA\_LoyaltyAccountCreateRQ Version="2013.1" xmlns="http://www.opentravel.org/OTA/2003/05">

<AccountInfo>

<MemberInfo BirthDate="1988-11-27">

<PersonName>

<GivenName>John</GivenName>

<MiddleName>T</MiddleName>

<Surname>Smith</Surname>

</PersonName>

<Telephone PhoneNumber="11-111-1111" PhoneLocationType="HP" AreaCityCode="2222"/>

<Telephone PhoneNumber="22-222-2222" PhoneLocationType="BP" AreaCityCode="2222"/>

<Address Type="B">

<AddressLine>1234 Main Street</AddressLine>

<AddressLine>Suite 22</AddressLine>

<CityName>Dublin</CityName>

<PostalCode>23255</PostalCode>

<CountryName Code="IE">Ireland</CountryName>

</Address>

<Email EmailType="EH">testemail@aerlingus.com</Email>

<Email EmailType="EB">Btestemail@aerlingus.com</Email>

<TPA\_Extensions>

<PersonGender>M</PersonGender>

</TPA\_Extensions>

</MemberInfo>

</AccountInfo>

</OTA\_LoyaltyAccountCreateRQ>

Response:

<NS1:OTA\_LoyaltyAccountRS Version="2013.1" xmlns:NS1="http://www.opentravel.org/OTA/2003/05">

<NS1:Success/>

<NS1:UniqueID ID="7770000002" Type="21"/>

<NS1:AccountInfo>

<NS1:MemberInfo BirthDate="1988-11-27">

<NS1:PersonName>

<NS1:GivenName>John</NS1:GivenName>

<NS1:MiddleName>T</NS1:MiddleName>

<NS1:Surname>Smith</NS1:Surname>

</NS1:PersonName>

<NS1:Telephone PhoneNumber="11-111-1111" PhoneLocationType="HP" AreaCityCode="2222"/>

<NS1:Telephone PhoneNumber="22-222-2222" PhoneLocationType="BP" AreaCityCode="2222"/>

<NS1:Address Type="B">

<NS1:AddressLine>1234 Main Street</NS1:AddressLine>

<NS1:AddressLine>Suite 22</NS1:AddressLine>

<NS1:CityName>Dublin</NS1:CityName>

<NS1:PostalCode>23255</NS1:PostalCode>

<NS1:CountryName Code="IE">Ireland</NS1:CountryName>

</NS1:Address>

<NS1:Email EmailType="EH">testemail@aerlingus.com</NS1:Email>

<NS1:Email EmailType="EB">Btestemail@aerlingus.com</NS1:Email>

<NS1:TPA\_Extensions>

<NS1:PersonGender>M</NS1:PersonGender>

</NS1:TPA\_Extensions>

</NS1:MemberInfo>

</NS1:AccountInfo>

</NS1:OTA\_LoyaltyAccountRS>

<NS1:UniqueID ID="7770000002" Type="21"/> Id is generated by /cls-mongo-lib/com/aerlingus/broker/service/cls/mongo/GetNextId.subflow in range of first\_cls\_number & last\_cls\_number of ‘mongodb://localhost/local.cls\_new\_customer\_numbers’ collection

To check if it is working you can select data on oracle db:

select \* from tcuemal where FK\_TCUID\_NUMBER = 7770000002;

select \* from tcualia where FK1\_CUSTOMER\_NBR = 7770000002;

select \* from tcuphon where FK2\_CUSTOMER\_NBR = 7770000002;

select \* from tcuaddr where FK3\_CUSTOMER\_NBR = 7770000002;

select \* from tcu where ID\_NUMBER = 7770000002;

You can run min request to create loyalty profile:

<OTA\_LoyaltyAccountCreateRQ Version="2013.1" xmlns="http://www.opentravel.org/OTA/2003/05">

<AccountInfo>

<MemberInfo >

<PersonName>

<GivenName>John</GivenName>

<Surname>Smith</Surname>

</PersonName>

<Email EmailType="EB">testemail@aerlingus.com</Email>

<TPA\_Extensions>

<PersonGender>M</PersonGender>

</TPA\_Extensions>

</MemberInfo>

</AccountInfo>

</OTA\_LoyaltyAccountCreateRQ>

Then there will be created rows only in three tables:

select \* from tcuemal where FK\_TCUID\_NUMBER = 7770000005;

select \* from tcualia where FK1\_CUSTOMER\_NBR = 7770000005;

select \* from tcu where ID\_NUMBER = 7770000005;

(7770000005 is generated!!! You have to check what id has just created profile)

<NS1:OTA\_LoyaltyAccountRS Version="2013.1" xmlns:NS1="http://www.opentravel.org/OTA/2003/05">

<NS1:Success/>

<NS1:UniqueID ID="7770000005" Type="21"/>

<NS1:AccountInfo>

<NS1:MemberInfo>

<NS1:PersonName>

<NS1:GivenName>John</NS1:GivenName>

<NS1:Surname>Smith</NS1:Surname>

</NS1:PersonName>

<NS1:Email EmailType="EB">testemail@aerlingus.com</NS1:Email>

<NS1:TPA\_Extensions>

<NS1:PersonGender>M</NS1:PersonGender>

</NS1:TPA\_Extensions>

</NS1:MemberInfo>

</NS1:AccountInfo>

</NS1:OTA\_LoyaltyAccountRS>

1. To get points data:

* Path: 'GET /loyalties/{id}' id - customer ID in Golden Circle account

Request: {'http://www.opentravel.org/OTA/2003/05'}: OTA\_ReadRQ

Response: {'http://www.opentravel.org/OTA/2003/05'}:OTA\_LoyaltyAccountRS

Request:

<http://localhost:7080/loyalties/1801921>

where 1801921 – uniqueId of user account

(in CLSExtractor configurable service in ‘monthPeriod’ property you set period for getting points data)

Response:

<NS1:OTA\_LoyaltyAccountRS Version="2013.1" xmlns:NS1="http://www.opentravel.org/OTA/2003/05">

<NS1:Success/>

<NS1:UniqueID ID="1801921" Type="21"/>

<NS1:AccountInfo PointBalance="4000">

<NS1:CustLoyalty ExpireDate="3999-03-31" SignupDate="2006-03-22" LoyalLevel="G"/>

</NS1:AccountInfo>

</NS1:OTA\_LoyaltyAccountRS>

1. To proccess redemption:

Path: 'POST /loyalties/redemption'

- Request: {'http://www.opentravel.org/OTA/2003/05'}: OTA\_LoyaltyCertificateRedemptionRQ

- Response: {'http://www.opentravel.org/OTA/2003/05'}: OTA\_LoyaltyAccountRS

Request:

<OTA\_LoyaltyCertificateRedemptionRQ Version="2013.1" xmlns="http://www.opentravel.org/OTA/2003/05">

<RedemptionInfo>

<RedemptionDetail MemberNumber="1801921" ProgramName="FF"/>

<Customer>

<TPA\_Extensions>

<Redemption PointsToRedeem="1000"/>

</TPA\_Extensions>

</Customer>

</RedemptionInfo>

</OTA\_LoyaltyCertificateRedemptionRQ>

To get a response, you must have profile at mongoDB at ‘profiles’ collection with MembershipID = 1801921 and programID =’ FF’

Response:

<NS1:OTA\_LoyaltyAccountRS xmlns:NS1="http://www.opentravel.org/OTA/2003/05">

<NS1:Success>Success</NS1:Success>

<NS1:UniqueID ID="1801921"/>

<NS1:AccountInfo PointBalance="3000">

<NS1:CustLoyalty ExpireDate="3999-03-31" LoyalLevel="G"/>

</NS1:AccountInfo>

</NS1:OTA\_LoyaltyAccountRS>

To check if point balance was decreased, you can run sql query on oracle DB:

select \* from TASAGEG where FK1\_CUSTOMER\_NBR = 1801921;

To delete all redemptions made by above-mentioned request, you can run sql query on oracle DB:

DELETE FROM TASAGEG WHERE POINTS\_AMOUNT=-1000 AND FK1\_CUSTOMER\_NBR=1801921;commit;

1. To run job extractor you have to deploy ‘cls-extractor-service’, ‘cls-profiles-sync-service’, ‘cls-aggregate-service’ and it will run automatically with time period declared in ‘CLSExtractor’ configurable service in ‘timeout’ property.

* This job will extract all changed profiles from oracle DB updated after last success sync.
* If you want to test extractor job in large amount of profiles you have to change

SET Environment.time= CURRENT\_TIMESTAMP;

in

/cls-extractor-lib/com/aerlingus/broker/service/cls/extractor/CLSExtractFlow.esql -> CLSExtractFlowAddJobRequest

To some other date, for example:

SET Environment.time= ' 2005-09-10';

To check extractor job status run ‘db.cls\_extractor\_job\_status.find()’ command on ‘cls\_extractor\_job\_status’ collection

To check profiles in mongoDB run ‘db.profiles.find()’ command on ‘profiles’ collection

1. To run tests on ‘mongo-lib-java’ project:

Run “mvn clean package” command on ‘mongo-lib-java’ project.