



## คู่มือการใช้งานสำหรับนักพัฒนาระบบ XML Generator (Java)

โครงการ จ้างที่ปรึกษาเพื่อบริหารโครงการปรับเปลี่ยนบริการภาครัฐที่เกี่ยวกับการ  
ออกใบอนุญาต หรือหลักฐานสำคัญ ให้เป็นดิจิทัล ด้วยมาตรฐานที่จำเป็น

สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์

23 เมษายน 2564

Strategy Consulting & Digital Transformation

FRONTIS 

บริษัท ฟรอนทิส จำกัด

## สารบัญ

การกำหนดค่าสำหรับ Library .....	3
1. Environment และ Software ที่เกี่ยวข้อง.....	3
2. การกำหนด Dependencies .....	3
ข้อมูลรายละเอียด Library.....	8
1. Class and method .....	8
การใช้งานและการ Deploy library .....	15
1. การเตรียม Project .....	15
2. การเรียกใช้งานสำหรับการทดสอบ (Debug).....	16
3. การ Deploy library (Executable jar) .....	18
การเรียกใช้งานผ่าน Command-line interface .....	20
1. รายละเอียด Argument.....	20
2. ตัวอย่างการเรียกใช้งาน .....	21

## การกำหนดค่าสำหรับ Library

### 1. Environment และ Software ที่เกี่ยวข้อง

Library นี้พัฒนาด้วยภาษา Java ซึ่งมี environment และ software ที่จำเป็นในการใช้พัฒนา ดังนี้

1. Java JDK 8
2. Eclipse (Editor สำหรับใช้การพัฒนา)
3. Microsoft SQL Server (สามารถเปลี่ยนเป็น Database อื่นได้ โดยจะต้องเปลี่ยนการประกาศ Dependency ในหัวข้อถัดไปด้วย)

### 2. การกำหนด Dependencies

Library มีการใช้งาน maven library อื่น ๆ เพิ่มเติมประกอบในการพัฒนา เพื่อให้ Library สามารถทำงานได้อย่างสมบูรณ์ จึงจำเป็นต้องติดตั้ง Dependency ที่เกี่ยวข้องทั้งหมด (ประกาศในไฟล์ pom.xml) ดังนี้

#	groupId	artifactId	version
1	com.googlecode.json-simple	json-simple	1.1.1
2	com.fasterxml.jackson.core	jackson-databind	2.12.0
3	org.json	json	20201115
4	com.microsoft.sqlserver	mssql-jdbc	8.4.1.jre8
5	joda-time	joda-time	2.10.10
6	ca.uhn.hapi.fhir	hapi-fhir-structures-r5	5.3.0
7	ca.uhn.hapi.fhir	hapi-fhir-base	5.3.0
8	ca.uhn.hapi.fhir	hapi-fhir-validation	5.3.0
9	com.squareup.okhttp3	okhttp	4.9.0
10	org.slf4j	slf4j-simple	1.7.28

#### 2.1 วิธีการติดตั้ง Dependency

2.1.1 ค้นหา Maven library ได้ที่ <https://mvnrepository.com/>

The screenshot shows the Maven Repository search results for the term 'fhir'. The search found 299 results. The results are sorted by relevance. The top three results are:

- HAPI FHIR Core Library**  
ca.uhn.hapi.fhir » hapi-fhir-base  
HAPI FHIR Core Library  
Last Release on Apr 14, 2021
- HAPI FHIR Structures DSTU3**  
ca.uhn.hapi.fhir » hapi-fhir-structures-dstu3  
HAPI FHIR Structures DSTU3  
Last Release on Apr 14, 2021
- HAPI FHIR Structures FHIR R4**  
ca.uhn.hapi.fhir » hapi-fhir-structures-r4  
HAPI FHIR Structures FHIR R4  
Last Release on Apr 14, 2021

On the left side, there is a sidebar with 'Repository' and 'Group' sections. The 'Repository' section lists various repositories like Central, Sonatype, Mulesoft, etc. The 'Group' section lists various groups like ca.uhn, com.ibm, org.openehealth, etc.

2.1.2 เพื่อพบ library ที่ต้องการแล้ว ให้เลือกเวอร์ชันที่ต้องการ

The screenshot shows the Maven Repository page for the artifact 'HAPI FHIR Structures FHIR R4'. The page displays the following information:

- Indexed Artifacts (20.3M)**: A line graph showing the number of projects (in millions) over time, from 2008 to 2018.
- Popular Categories**: A list of categories including Aspect Oriented, Actor Frameworks, Application Metrics, Build Tools, Bytecode Libraries, and Command Line Parsers.
- Home » ca.uhn.hapi.fhir » hapi-fhir-structures-r4**: The breadcrumb navigation path.
- HAPI FHIR Structures FHIR R4**: The artifact name.
- License**: Apache 2.0.
- Used By**: 45 artifacts.
- Central (23) Trifork (8)**: The distribution management system (DMS) counts.
- Version List**: A table showing the available versions for the artifact.
 

Version	Count
5.3.x	23
5.2.x	8

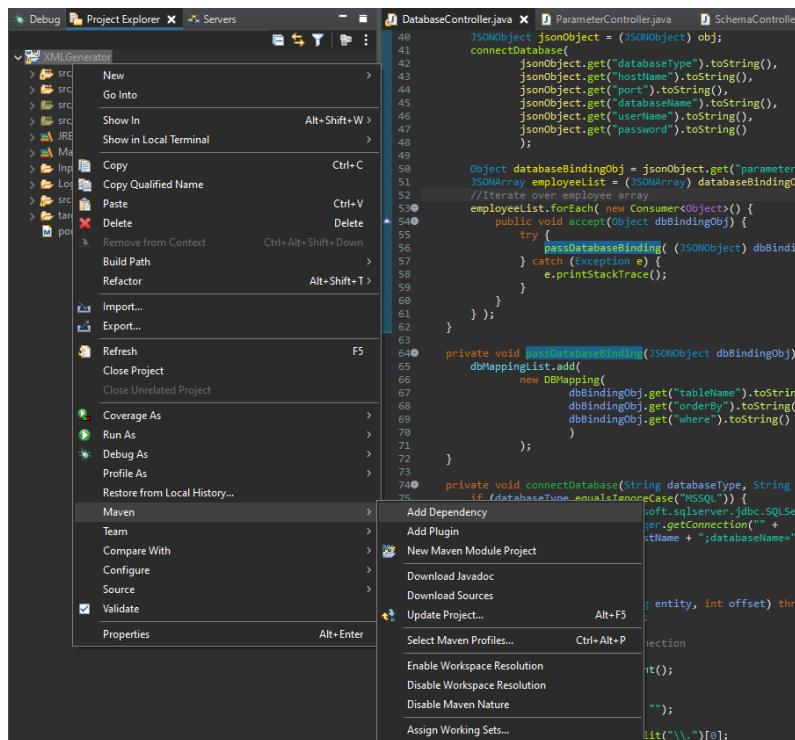
2.1.3 หน้าเว็บจะแสดง Dependency detail ของ library เวอร์ชันนั้น ๆ ซึ่งจะนำไปใช้ในขั้นตอนถัดไป

คู่มือการใช้งานสำหรับนักพัฒนาระบบ XML Generator (Java)

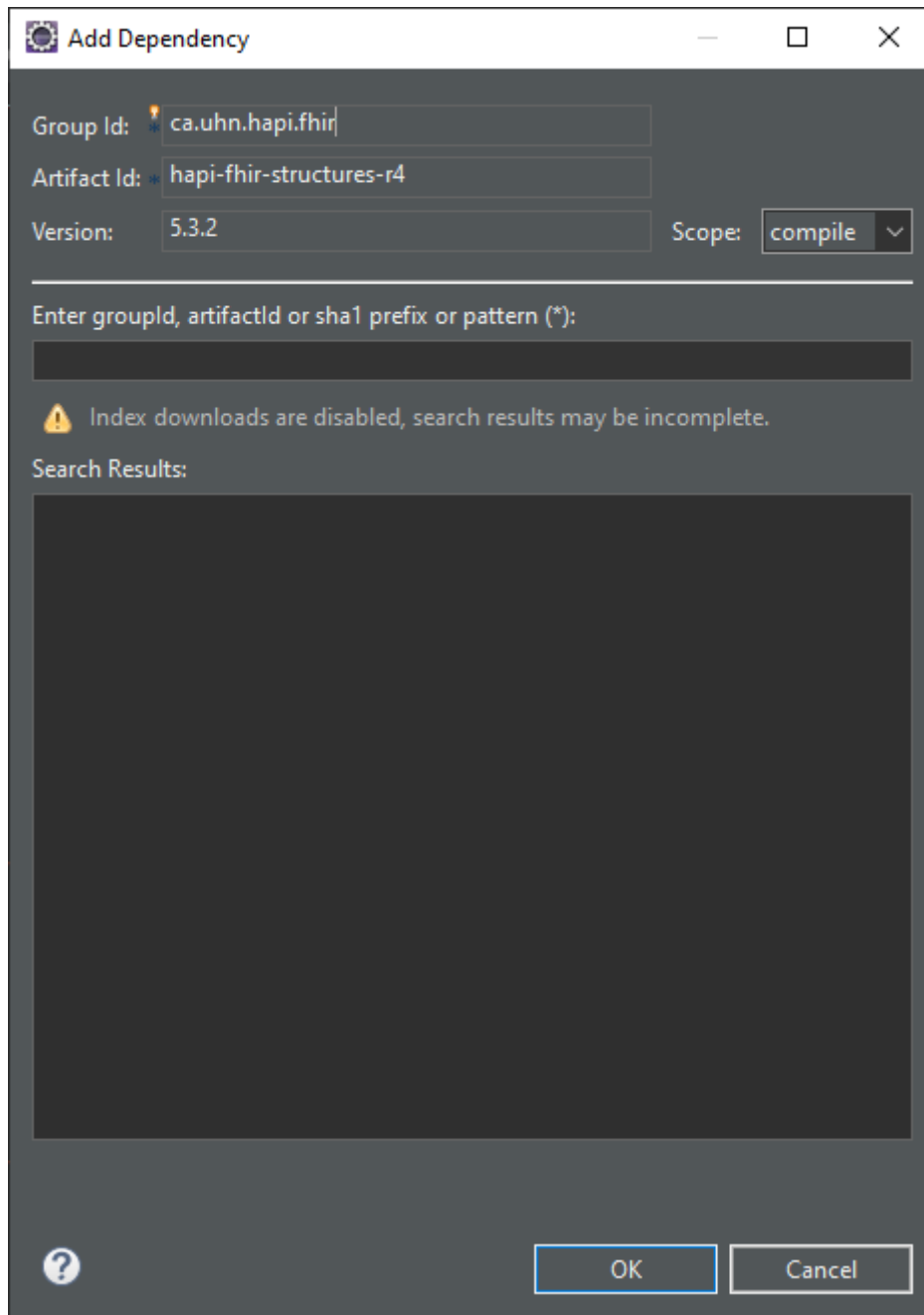
โครงการ “จ้างที่ปรึกษาเพื่อบริหารโครงการปรับเปลี่ยนบริการภาครัฐที่เกี่ยวกับการออกใบอนุญาต หรือหลักฐานสำคัญ ให้เป็นดิจิทัล ด้วยมาตรฐานที่จำเป็น”

The screenshot shows the Maven Repository website. The main content area displays the details for the artifact **HAPI FHIR Structures FHIR R4 » 5.3.2**. The license is Apache 2.0, dated Apr 14, 2021. It is a bundle (37 KB) available in the Central repository, used by 45 artifacts. The page includes a Maven dependency snippet and a checkbox for "Include comment with link to declaration".

2.1.4 ที่ eclipse คลิกขวาที่ project ในหน้าต่าง project explorer เลือกที่ Maven > Add Dependency



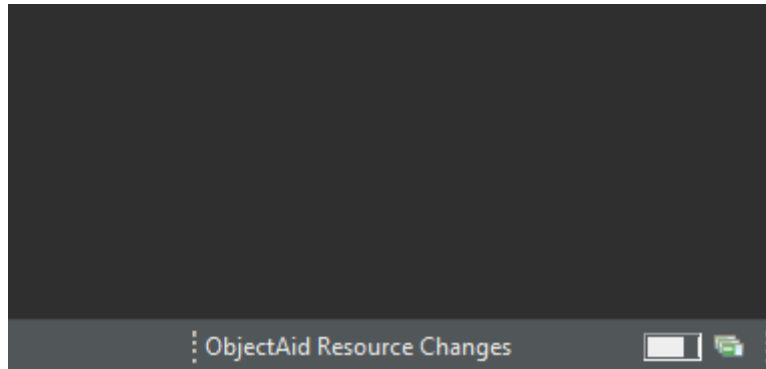
2.1.5 ที่หน้าต่าง Add dependency ใส่รายละเอียด library จากขั้นตอนที่ 2.1.3 จากนั้นกด OK



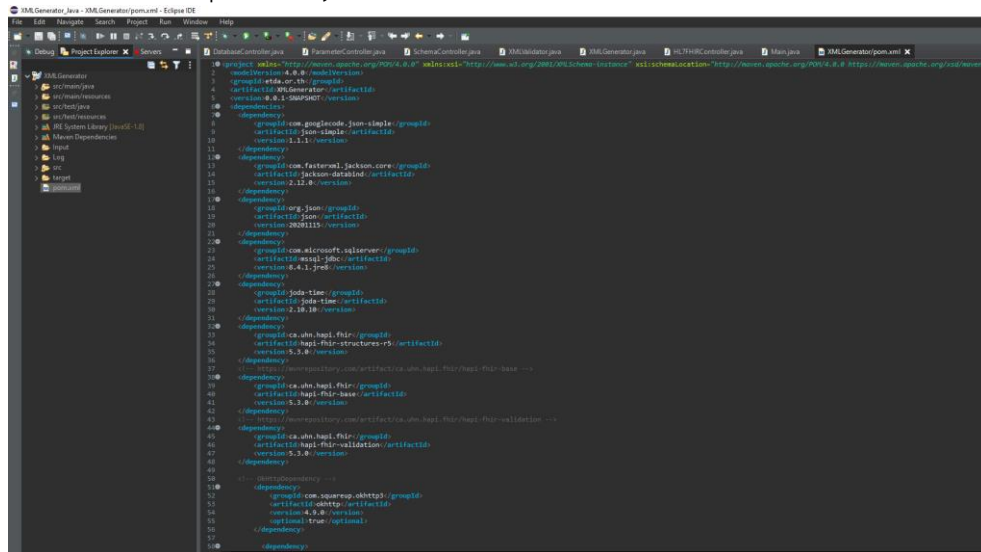
- 2.1.6 โปรแกรมจะทำการติดตั้ง library ดังกล่าวให้ โดยสังเกตได้จาก progress bar ที่หน้าจอมุมขวาล่าง

คู่มือการใช้งานสำหรับนักพัฒนาระบบ XML Generator (Java)

โครงการ “จ้างที่ปรึกษาเพื่อบริหารโครงการปรับเปลี่ยนบริการภาครัฐที่เกี่ยวกับการออกใบอนุญาต หรือหลักฐานสำคัญ ให้เป็นดิจิทัล ด้วยมาตรฐานที่จำเป็น”



## 2.2 ภาพการติดตั้ง Dependency เรียบร้อยแล้ว







Method name	Parameter in	Return	Remark
setTemplateDocument	<b>Document</b> templateDocument	-	กำหนด Template file ในรูปแบบ XML Document
getTemplateFile	-	<b>String</b>	คืนค่าตำแหน่งไฟล์ Template
setTemplateFile	<b>String</b> templateFile	-	กำหนดตำแหน่งไฟล์ Template
getDatabaseController	-	<b>DatabaseController</b>	คืนค่า instance ของ <b>DatabaseController</b>
setDatabaseController	<b>DatabaseController</b> databaseController	-	กำหนด instance ของ <b>DatabaseController</b>
getNodeConfigList	-	<b>List&lt;XMLNode&gt;</b>	คืนค่า config ของ XML ราย node
setNodeConfigList	<b>List&lt;XMLNode&gt;</b> nodeConfigList	-	กำหนดค่า config ของ XML ราย node
getUserParameter	-	<b>UserParameter</b>	คืนค่า instance ของ <b>UserParameter</b>
setUserParameter	<b>UserParameter</b> userParameter	-	กำหนด instance ของ <b>UserParameter</b>

### Class DatabaseController

Method name	Parameter in	Return	Remark
DatabaseController	<b>String</b> databaseConnectionFile	-	Class constructor รับค่าไฟล์ที่กำหนดรายละเอียดการเชื่อมต่อ database
passDatabaseConfig	<b>String</b> databaseConnectionFile	-	อ่านค่ารายละเอียดการเชื่อมต่อ database เข้ามาในรูปแบบ JSON
passDatabaseBinding	<b>JSONObject</b> dbBindingObj	-	อ่านค่ารายละเอียดเงื่อนไขการอ่าน database เข้ามาในรูปแบบ JSON
connectDatabase	<b>String</b> databaseType, <b>String</b> hostName, <b>String</b> port, <b>String</b> databaseName, <b>String</b> userName, <b>String</b> password	-	เชื่อมต่อ database
readDatabase	<b>String</b> entity, <b>int</b> offset	Object	อ่านข้อมูลจาก database ตามเงื่อนไขที่กำหนดในแต่ละรอบ
checkAll	<b>String</b> entity	int	นับจำนวน database row ภายใต้อินเตอร์เฟซที่กำหนด
closeConnection	-	-	ยุติการเชื่อมต่อ database

Method name	Parameter in	Return	Remark
getDatabaseParamList	-	HashMap<String, String>	คืนค่าข้อมูลรายละเอียดการเชื่อมต่อ database
setDatabaseParamList	HashMap<String, String> databaseparamList	-	กำหนดค่าข้อมูลรายละเอียดการเชื่อมต่อ database

### Class ParameterController

Method name	Parameter in	Return	Remark
ParameterController	String[] args	-	Class constructor
generateParameter	String[] args	-	ประมวลผล external จาก Main method
validateRequireSchema	String[] args	boolean	ตรวจสอบความครบถ้วนของ parameter ที่จำเป็น
passUserParameter	String userParameterFilePath	-	อ่านค่า Parameter จากไฟล์ที่กำหนด
parseConfig	JSONObject config	-	ส่งต่อค่า Parameter เข้าสู่ Library
getLibraryParameter	-	HashMap<String, String>	คืนค่า Global parameter ของ Library
getUserParameterList	-	List<UserParameter>	คืนค่า User parameter ของ Library
setUserParameterList	List<UserParameter> userParameterList	-	กำหนดค่า User parameter ของ Library

### Class SchemaController

Method name	Parameter in	Return	Remark
SchemaController	String messagepackage, String templateFile	-	Class constructor กำหนดค่าเริ่มต้นให้ Class
getNameSpaceList	String templateFile	-	อ่าน Prefix และ Namespace ทั้งหมดจาก Template file
getSchema	String messagePackage	-	อ่านไฟล์ XML Schema เข้าสู่ Library
listSchemaFile	File folder	-	แสดงรายการ XML Schema ทั้งหมดที่อยู่ใน Folder ที่กำหนด

Method name	Parameter in	Return	Remark
getComplexNode	<b>Document</b> document, <b>String</b> prefix	-	ค้นหา XML Element ที่เป็น complex node เท่านั้น
getSequenceNode	<b>Node</b> node, <b>String</b> prefix	-	ค้นหา XML Element ที่เป็น sequence node เท่านั้น
getElementNode	<b>Node</b> node, <b>Node</b> parentComplexNode, <b>String</b> prefix	-	ค้นหา XML Element ที่เป็น element node เท่านั้น จากนั้นอ่านรายละเอียดข้อมูลของ node เข้าสู่ Library
getXmlNodeList	-	<b>List&lt;XMLNode&gt;</b>	คืนค่าเป็น XML Node ทั้งหมด

### Class XMLValidator

Method name	Parameter in	Return	Remark
XMLValidator	<b>String</b> xmlFilePath, <b>String</b> schemaFolderPath	-	Class constructor กำหนดค่าเริ่มต้นให้ Class
validate	-	<b>List&lt;SAXParseException&gt;</b>	ตรวจสอบ XML file กับ XML Schema ที่กำหนด
validateWithFHIRCLI	<b>String</b> filePath	-	ตรวจสอบ XML file กับ Validator library เฉพาะ HL7 FHIR
listSchemaFile	<b>File</b> folder	-	แสดงรายการ xml schema ทั้งหมดใน folder ที่กำหนด
getXmlFilePath	-	<b>String</b>	คืนค่า Path ของ XML File
setXmlFilePath	<b>String</b> xmlFilePath	-	กำหนดค่า Path ของ XML File
getSchemaFolderPath	-	<b>String</b>	คืนค่า path ของ Schema folder
setSchemaFolderPath	<b>String</b> schemaFolderPath	-	กำหนดค่า path ของ Schema folder

## Class HL7FHIRController

Method name	Parameter in	Return	Remark
convertAndValidate	<b>String</b> filePath, <b>String</b> fhirProfile	-	Convert XML ให้อยู่ใน รูปแบบมาตรฐานของ FHIR และทำการ validate ด้วย profile ที่กำหนดเอง
convertAndValidate	<b>String</b> filePath	-	Convert XML ให้อยู่ใน รูปแบบมาตรฐานของ FHIR และทำการ validate ด้วย profile มาตรฐาน
convert	<b>String</b> filePath	<b>String</b>	Convert XML ให้อยู่ใน รูปแบบมาตรฐานของ FHIR
validate	<b>String[]</b> args	-	HL7 FHIR Implementation method
destinationDirectoryValid	<b>String</b> dest	boolean	HL7 FHIR Implementation method
shouldDisplayHelpToUser	<b>String[]</b> args	boolean	HL7 FHIR Implementation method
doLeftRightComparison	<b>String[]</b> args, <b>CliContext</b> cliContext, <b>TimeTracker</b> tt	-	HL7 FHIR Implementation method
doValidation	<b>TimeTracker</b> tt, <b>TimeTracker.Session</b> tts, <b>CliContext</b> cliContext	-	HL7 FHIR Implementation method

## Class DBMapping

Method name	Parameter in	Return	Remark
DBMapping	<b>String</b> tableName, <b>String</b> orderBy, <b>String</b> where	-	Class constructor
getTableName	-	<b>String</b>	คืนค่าชื่อ database table
setTableName	<b>String</b> tableName	-	กำหนดค่าชื่อ database table
getFieldList	-	<b>String</b>	คืนค่าชื่อ table field
setFieldList	<b>String</b> fieldList	-	กำหนดค่าชื่อ table field
getOrderBy	-	<b>String</b>	คืนค่าการจัดเรียงผลลัพธ์
setOrderBy	<b>String</b> orderBy	-	กำหนดค่าการจัดเรียงผลลัพธ์
getWhere	-	<b>String</b>	คืนค่าเงื่อนไขในการดึงผลลัพธ์
setWhere	<b>String</b> where	-	กำหนดค่าเงื่อนไขในการดึงผลลัพธ์

## Class XMLNode

Method name	Parameter in	Return	Remark
getName	-	String	กำหนด/คืนค่า ชื่อ element
setName	String name	-	
getType	-	String	กำหนด/คืนค่า ชนิดของ element
setType	String type	-	
getMinOccur	-	String	กำหนด/คืนค่า จำนวน element ต่ำสุด
setMinOccur	String minOccur	-	
getMaxOccur	-	String	กำหนด/คืนค่า จำนวน element สูงสุด
setMaxOccur	String maxOccur	-	
getParentName	-	String	กำหนด/คืนค่า ชื่อ parent element
setParentName	String parentName	-	
isRoot	-	bool	กำหนด/คืนค่า ว่า element เป็น root หรือไม่
setRoot	bool isRoot	-	
isMandatory	-	bool	กำหนด/คืนค่า ว่า element เป็น mandatory หรือไม่
setMandatory	bool isMandatory	-	
isOptional	-	bool	กำหนด/คืนค่า ว่า element เป็น optional หรือไม่
setOptional	bool isOptional	-	
isRepeatable	-	bool	กำหนด/คืนค่า ว่า element เป็น repeatable หรือไม่
setRepeatable	bool isRepeatable	-	
isArray	-	bool	กำหนด/คืนค่า ว่า element เป็น Array หรือไม่
setArray	bool isArray	-	
isComplexType	-	bool	กำหนด/คืนค่า ว่า element เป็น Complex element หรือไม่
setComplexType	bool isComplexType	-	
validateProperties	-	-	verify property ป้องกันการ conflict ของการตั้งค่า

## Class UserParameter

Method name	Parameter in	Return	Remark
UserParameter	-	-	Class constructor
getOutputFileName	-	string	คืนค่าไฟล์ Output
setOutputFileName	string outputFileName	-	กำหนดค่าไฟล์ Output
addDatabaseParameter	string key, string value	-	เพิ่มค่า database parameter
removeDatabaseParameter	string key	-	ลบค่า database parameter
clearDatabaseParameter	-	-	เพิ่มค่า database parameter ทั้งหมด

Method name	Parameter in	Return	Remark
getDatabaseParameter	string key	string	คืนค่า database parameter
getDatabaseParameterList	-	HashMap<string, string>	กำหนดค่า database parameter

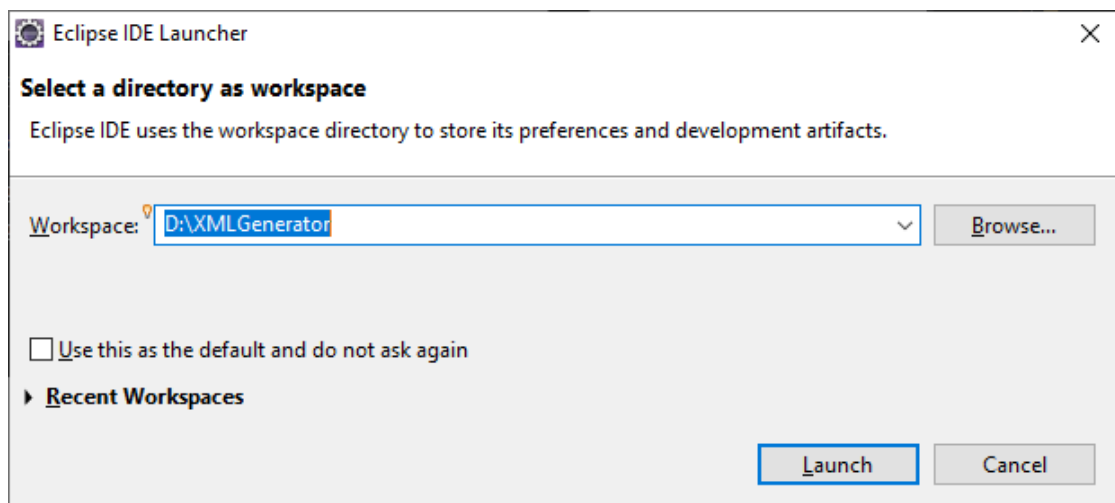
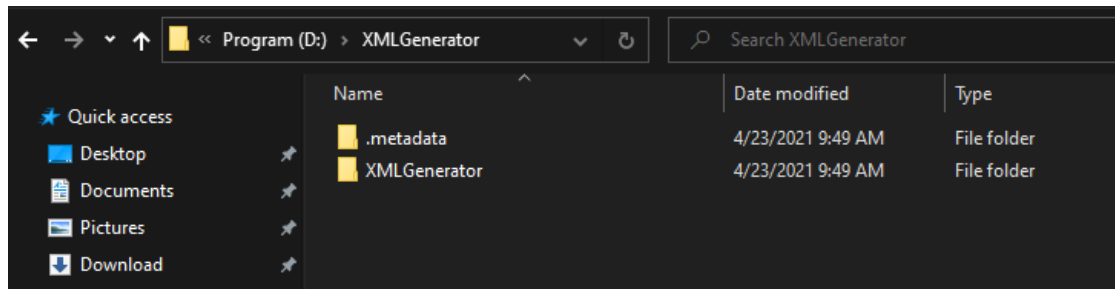
### Class NamespaceModel

Method name	Parameter in	Return	Remark
getPrefix		String	คืนค่า namespace prefix
setPrefix	String prefix		กำหนดค่า namespace prefix
getValue		String	คืนค่า namespace value
setValue	String value		กำหนดค่า namespace value

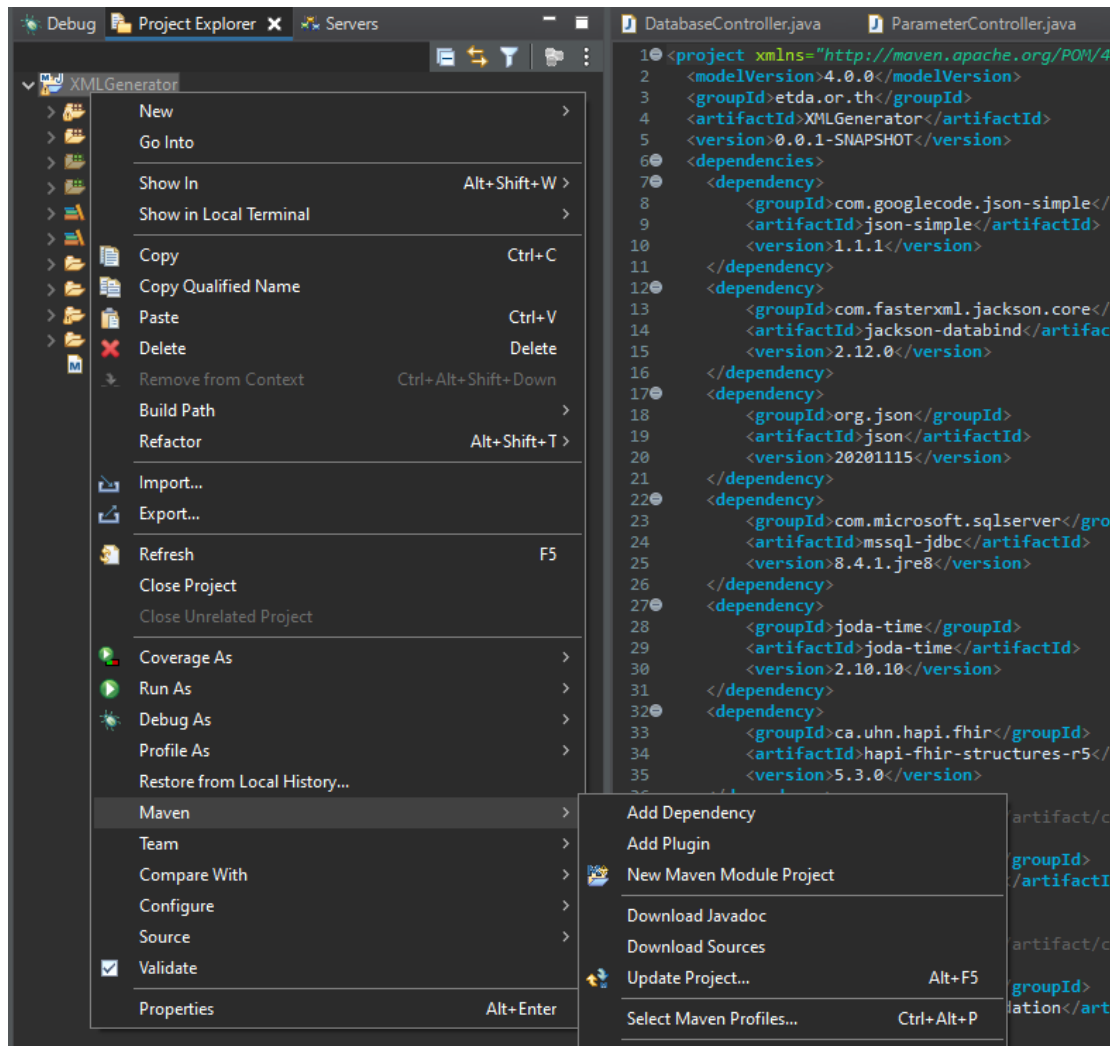
## การใช้งานและการ Deploy library

### 1. การเตรียม Project

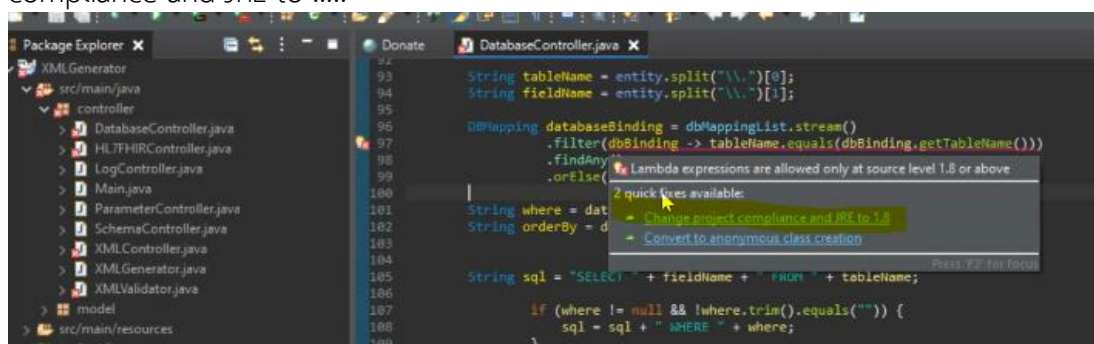
1. เปิดโปรแกรม Eclipse และเลือกไปยังที่ตั้งของโฟลเดอร์ project (วิธีการสังเกตคือต้องมี folder .metadata ด้วยเสมอ)



2. หลังจากเปิด project แล้ว ที่หน้าต่าง Project explorer ให้คลิกขวาที่ root folder ของ project แล้วเลือกไปที่ Maven > Update project จากนั้นรอนกว่า project จะติดตั้ง dependency ที่จำเป็นเสร็จ



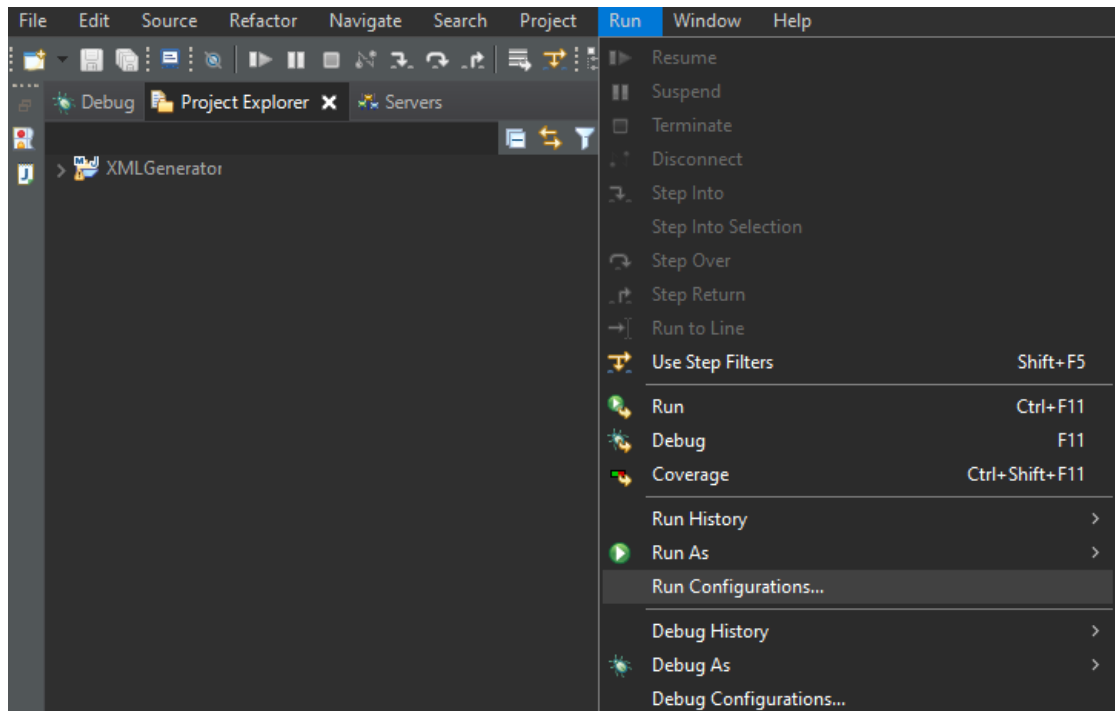
3. กรณีมี error แจ้งให้ update JRE ให้คลิกขวาที่บริเวณ error ดังกล่าว แล้วเลือก change project compliance and JRE to .....



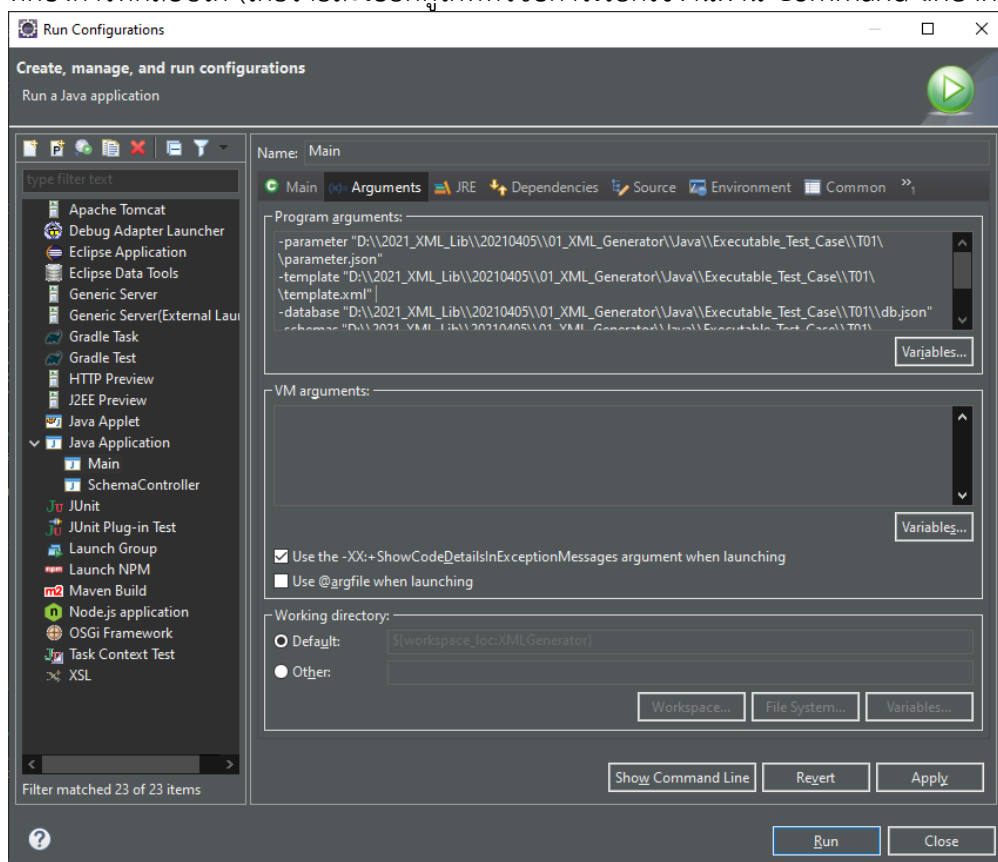
## 2. การเรียกใช้งานสำหรับการทดสอบ (Debug)

1. ที่ menu bar เลือกไปที่ Run > Run configurations...





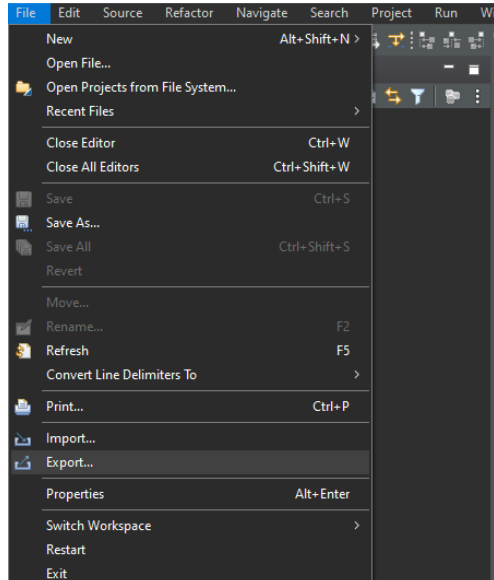
2. เลือกไปที่ Tab Arguments จากนั้นที่หัวข้อ Program arguments สามารถเปลี่ยนเป็น Argument ที่ต้องการทดสอบได้ (โดยรายละเอียดดูได้ที่หัวข้อการเรียกใช้งานผ่าน Command-line interface)



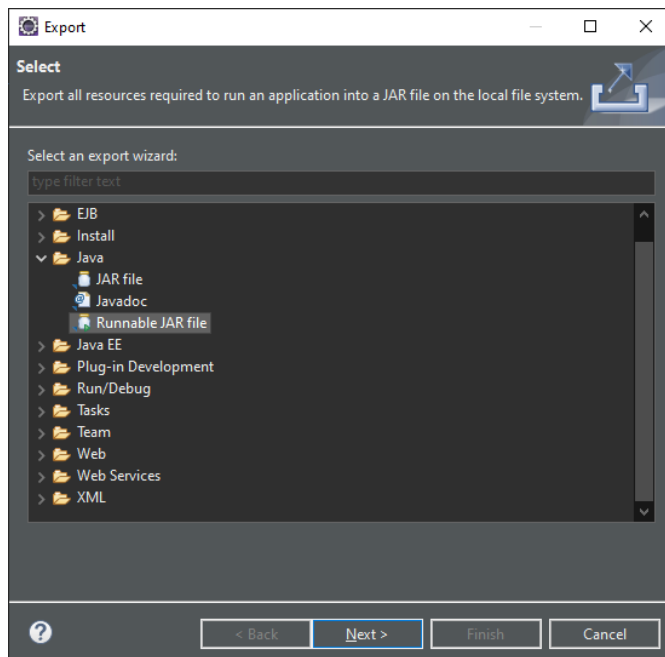
### 3. กด Run เพื่อดูผลลัพธ์

## 3. การ Deploy library (Executable jar)

### 1. ที่ Menu bar เลือกไปที่ File > Export

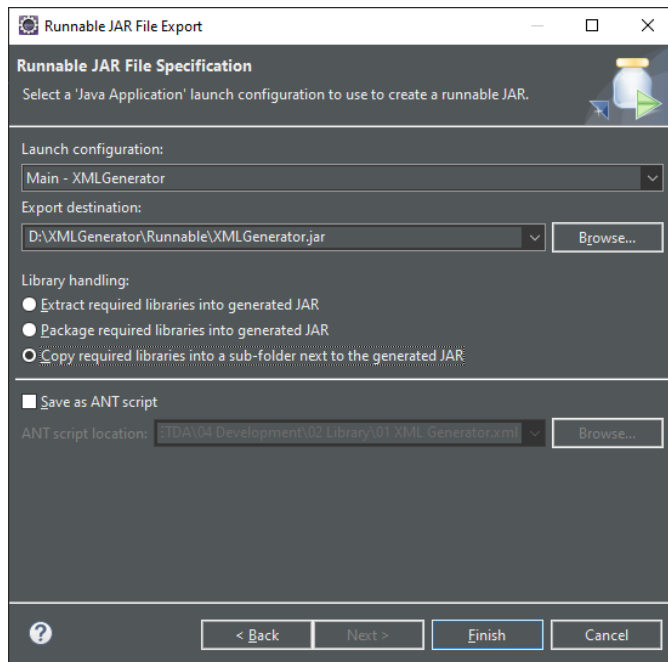


### 2. เลือกไปที่ Java > Runnable JAR file จากนั้นกด Next

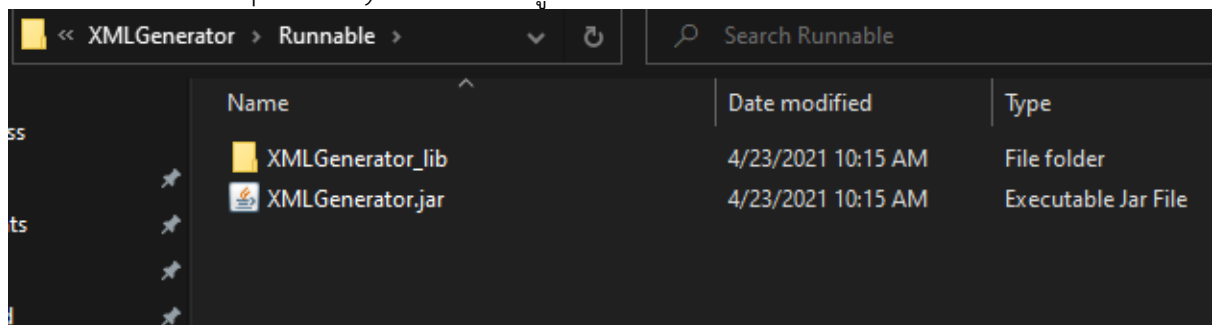


### 3. ใส่ค่าต่าง ๆ ดังนี้

- Launch configuration: เลือก Class ที่มี Method main()
- Export destination: สถานที่สำหรับจัดเก็บ JAR ไฟล์ที่สร้างเสร็จแล้ว
- Library handling: กำหนดรูปแบบการ Package JAR โดยแนะนำให้เลือกเป็น Copy required libraries into a sub-folder next to the generated JAR



4. กด Finish
5. โดยไฟล์ JAR และ Dependency ที่เกี่ยวข้อง จะถูกจัดเก็บใน folder ที่กำหนด



#### 4. การเปลี่ยน database connection

Library สามารถเชื่อมต่อ database อื่น ๆ นอกจาก Microsoft SQL Server โดยการเปลี่ยน Connection string ใน Class DatabaseController

```
private void connectDatabase(String databaseType, String hostName, String port,
    String databaseName, String userName, String password) throws Exception {
    if (databaseType.equalsIgnoreCase("MSSQL")) {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        connection = DriverManager.getConnection("" +
            "jdbc:sqlserver://" + hostName + ";databaseName=" + databaseName + ";user=" +
            userName + ";password=" + password);
    }
}
```

## การเรียกใช้งานผ่าน Command-line interface

### 1. รายละเอียด Argument

ชุดโปรแกรมรองรับการเรียกใช้งานผ่าน Command line สำหรับ Executable program โดยมี Parameter ที่สามารถ input ค่าได้ ดังนี้

- -Parameter “<PATH\_TO\_FILE.json>” สำหรับระบุไฟล์ในการเรียกใช้ในแต่ละรอบการทำงาน ซึ่ง Parameter จะสัมพันธ์กับการประกาศในไฟล์ Database configuration

```
[
  {
    "output": "T01\\Result\\ScoreReport_125685225.xml",
    "@param1": "125685225",
    "@param2": "2344322"
  }
]
```

- -template “<PATH\_TO\_FILE.xml>” สำหรับระบุไฟล์ XML template สำหรับการสร้าง XML ประกอบไปด้วย Tag ต่างๆที่ใช้ในการสร้างไฟล์ XML รวมถึงยังสามารถประกาศชื่อ database table/field name โดยชุดโปรแกรมจะทำการนำค่าที่อ่านจาก database ไปใส่ตาม tag หรือ element ที่ได้กำหนดไว้ (รองรับการใส่ค่า attribute ด้วย)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="../../schematron/ScoreReport_Schematron_lp0.sch" type="application/xml" schematypens="http://purl.oc
<rsm:ScoreReport xmlns:ccts="urn:un:unece:uncefact:documentation:standard:CoreComponentsTechnicalSpecification:2"
  xmlns:qdt="urn:etda:teda:data:QualifiedDataType:1"
  xmlns:ram="urn:un:unece:etda:data:standard:ScoreReport_ReuseableAggregateCoreComponent:1"
  xmlns:udt="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:21"
  xmlns:cct="urn:un:unece:uncefact:documentation:standard:CoreComponentType:2"
  xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
  xmlns:rsm="urn:un:unece:etda:data:standard:ScoreReport:1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:un:unece:etda:data:standard:ScoreReport:1 file:../schema/etda/standard/ScopreReport.xsd">
  <rsm:ScoreReportContext>
    <ram:OID />
    <ram:Version>1.0</ram:Version>
    <ram:Language>TH</ram:Language>
    <ram:ScoreReportID>10112322</ram:ScoreReportID>
    <ram:Name>หนังสือรับรองผลการทดสอบวิชาสามัญ 9 วิชา</ram:Name>
    <ram:TypeCode>-</ram:TypeCode>
    <ram:Purpose>-</ram:Purpose>
    <ram:IssueDateTime>2020-03-30T00:00:00.0</ram:IssueDateTime>
    <ram:EffectiveDateTime>
      <udt:DateTime>2020-03-30T00:00:00.0</udt:DateTime>
    </ram:EffectiveDateTime>
    <ram:ExpireDateTime>
      <udt:DateTime>2020-12-31T23:59:59.0</udt:DateTime>
    </ram:ExpireDateTime>
  </rsm:ScoreReportContext>
  <rsm:Examination>
    <ram:ExaminationType>@t_exam.exam_desc</ram:ExaminationType>
    <ram:ExaminationYear>@t_exam.exam_year</ram:ExaminationYear>
    <ram:ExaminationStartDate>@t_exam.exam_start_date</ram:ExaminationStartDate>
    <ram:ExaminationEndDate>@t_exam.exam_end_date</ram:ExaminationEndDate>
  </rsm:Examination>
```

- -database “<PATH\_TO\_FILE.json>” สำหรับระบุไฟล์เพื่อกำหนดค่าในการเชื่อมต่อ database รวมถึงการจับคู่ค่า Parameter จากไฟล์ Parameter เพื่อใช้ในการกำหนดเงื่อนไขการอ่านค่าจาก database ด้วย

```
{
  "databaseType": "MSSQL",
  "hostName": "localhost",
  "databaseName": "test_course",
  "port": "1433",
  "userName": "user01",
  "password": "password",
  "parameter_biding": [
    {
      "tableName": "m_student",
      "orderBy": "student_id",
      "where": "student_id=@param1"
    },
    {
      "tableName": "t_score",
      "orderBy": "sbj_cd,sbj_name",
      "where": "student_id=@param1"
    },
    {
      "tableName": "t_exam",
      "orderBy": "exam_id",
      "where": "exam_id=@param2"
    }
  ]
}
```

- -schemas “<PATH\_TO\_FOLDER>” สำหรับระบุ schema เพื่อใช้ประกอบในการสร้างไฟล์ XML และ Validate ไฟล์ XML ที่สร้างเสร็จแล้วในขั้นตอนสุดท้าย

## 2. ตัวอย่างการเรียกใช้งาน

- กรณีสร้าง XML ด้วย Schema ตามมาตรฐาน UN/CEFACT

```
java -jar XMLGenerator.jar
-parameter
"D:\2021_XML_Lib\20210405\01_XML_Generator\Java\Executable_Test_Case\T01\parameter.json"
-template
"D:\2021_XML_Lib\20210405\01_XML_Generator\Java\Executable_Test_Case\T01\template.xml"
-database "D:\2021_XML_Lib\20210405\01_XML_Generator\Java\Executable_Test_Case\T01\db.json"
-schemas
"D:\2021_XML_Lib\20210405\01_XML_Generator\Java\Executable_Test_Case\T01\ScoreReport_V1_2021
228_1505"
-wait "true"
```

- กรณีสร้าง XML ด้วย Schema ตามมาตรฐาน HL7 FHIR

```
java -jar XMLGenerator.jar
-parameter
"D:\2021_XML_Lib\20210405\01_XML_Generator\Java\Executable_Test_Case\T08\parameter.json"
-template
"D:\2021_XML_Lib\20210405\01_XML_Generator\Java\Executable_Test_Case\T08\template.xml"
-database "D:\2021_XML_Lib\20210405\01_XML_Generator\Java\Executable_Test_Case\T08\db.json"
```

**-schemas**

"D:\2021\_XML\_Lib\20210405\01\_XML\_Generator\Java\Executable\_Test\_Case\T08\ClinicServicesEstb\_V1\_20201222\_1445\schema"

**-wait** "true"

**-deleteIncomplete** "false" **-enableFHIRValidation** "true"