คู่มือการใช้งานสำหรับนักพัฒนาระบบ

PAdES Signer (Java)

โครงการ จ้างที่ปรึกษาเพื่อบริหารโครงการปรับเปลี่ยนบริการภาครัฐที่เกี่ยวกับการออกใบอนุญาต หรือหลักฐานสำคัญ ให้เป็นดิจิทัล ด้วยมาตรฐานที่จำเป็น

สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์

23 เมษายน 2564

Strategy Consulting & Digital Transformation

บริษัท ฟรอนทิส จำกัด

สารบัญ

การกำหนดค่าสำหรับ Library 3

1. Environment และ Software ที่เกี่ยวข้อง 3
2. การกำหนด Dependencies 3

ข้อมูลรายละเอียด Library 4

1. Class and method 4

การใช้งานและการ Deploy library 10

1. การเตรียม Project 10
2. การเรียกใช้งานสำหรับการทดสอบ (Debug) 16
3. การ Deploy library (Executable jar) 17

การเรียกใช้งานผ่าน Command-line interface 19

1. รายละเอียด Argument 19
2. ตัวอย่างการเรียกใช้งาน 20

การกำหนดค่าสำหรับ Library

1. Environment และ Software ที่เกี่ยวข้อง

Library นี้พัฒนาด้วยภาษา Java ซึ่งมี environment และ software ที่จำเป็นในการใช้พัฒนา ดังนี้

1. Java JRE 8 (ติดตั้งทั้ง 32-bit และ 64-bit)
2. Eclipse (Editor สำหรับใช้การพัฒนา)

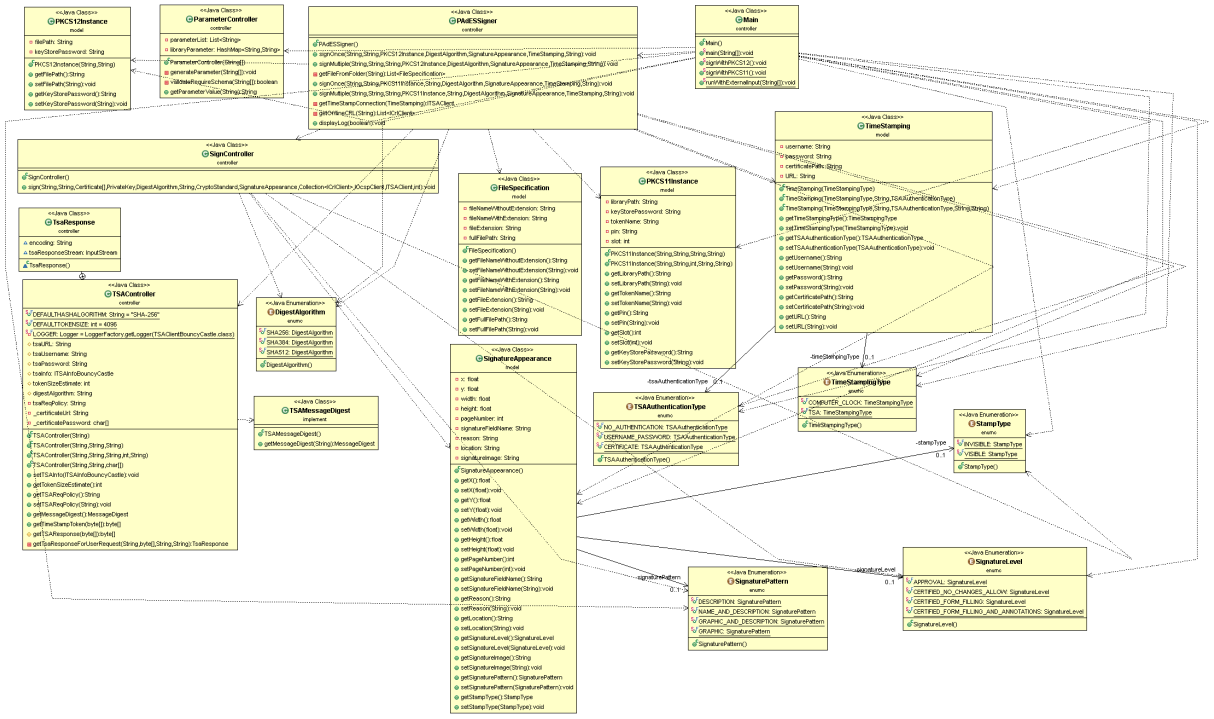
2. การกำหนด Dependencies

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

#	groupId	artifactId	version
1	com.itextpdf	itext7-core	7.1.7
2	com.itextpdf	sign	7.1.4
3	com.itextpdf	kernel	7.1.14
4	com.itextpdf	io	7.1.14
5	org.slf4j	slf4j-api	1.7.30
6	org.bouncycastle	bcprov-jdk16	1.46
7	org.bouncycastle	bcmail-jdk15	1.46
8	org.bouncycastle	bctsp-jdk15	1.46
9	org.apache.logging.log4j	log4j-api	2.13.3
10	org.apache.logging.log4j	log4j-core	2.13.3
11	org.apache.logging.log4j	log4j-slf4j-impl	2.13.3
12	commons-io	commons-io	2.4

ข้อมูลรายละเอียด Library

1. Class and method



Class PAdESSigner

Method name	Parameter in	Return	Remark
PAdESSigner	-	-	Class constructor
signOnce	**String** inputFilePath, **String** outputFilePath, **IPKCSInstance** pkcsInstance, **DigestAlgorithm** digestAlgorithm, **SignatureAppearance** signatureAppearance, **TimeStamp** timeStamping	-	Sign PDF ไฟล์เดียว
signMultiple	**String** inputFolderPath, **String** outputFolderPath, **String** outputSuffix, **IPKCSInstance** pkcsInstance, **DigestAlgorithm** digestAlgorithm, **SignatureAppearance**	-	Sign PDF หลายไฟล์ (Bulk sign)

Method name	Parameter in	Return	Remark
	signatureAppearance, **TimeStamp** timeStamping		
getFileFromFolder	**String** folderPath	**java.util.List<FileSpecification>**	แสดงรายการไฟล์ทั้งหมดในโฟล์เดอร์ที่กำหนด
getTimeStampConnection	**TimeStamp** timeStamping	ITSAClient	สร้าง TimeStamp instance จากข้อมูลที่หนด
loadKeyStore	**IPKCSInstance** pkcsInstance	**CertificateKeyPack**	load

Class CertificateKeyPack

Method name	Parameter in	Return	Remark
CertificateKeyPack	**-**	-	Class constructor
setCertificateChain	**Certificate[]** certificateChain	-	กำหนดค่า Certificate chain
getCertificateChain	-	**Certificate[]**	คืนค่า Certificate chain
setPrivateKey	**PrivateKey** privateKey	**-**	กำหนดค่า Private key
getPrivateKey	**-**	**PrivateKey**	คืนค่า Private key
setProvider	**String provider**	**-**	กำหนดค่า PKCS provider
getProvider	**-**	**String**	คืนค่า PKCS Provider

Class ParameterController

Method name	Parameter in	Return	Remark
ParameterController	**String[]** args	-	Class constructor
generateParameter	**String[]** args	-	ประมวลผล external จาก Main method
validateRequireParameter	**String[]** args	**boolean**	ตรวจสอบความครบถ้วนของ parameter ที่จำเป็น
getParameterValue	**String** key	**String**	คืนค่าของ parameter ตาม key ที่ส่งค่าเข้ามา

Class SignController

Method name	Parameter in	Return	Remark
sign	**String** src, **String** dest, **Certificate[]** chain, **PrivateKey** pk, **DigestAlgorithm** digestAlgorithm, **String** provider, **PdfSigner.CryptoStandard** subfilter,	-	ทำการเรียกใช้ third-party library เพื่อ sign เอกสาร PDF ด้วยรูปแบบและข้อมูลที่กำหนด

	SignatureAppearance signatureAppearance, **Collection<ICrlClient>** crlList, **IOcspClient** ocspClient, **ITSAClient** tsaClient, **int** estimatedSize		

Class FileSpecification

Method name	Parameter in	Return	Remark
getFileNameWithoutExtension	-	**String**	คืนค่าชื่อไฟล์แบบไม่มีนามสกุล
setFileNameWithoutExtension	**String** fileNameWithoutExtension	-	กำหนดค่าชื่อไฟล์แบบไม่มีนามสกุล
getFileNameWithExtension	-	**String**	คืนค่าชื่อไฟล์พร้อมนามสกุล
setFileNameWithExtension	**String** fileNameWithExtension	**-**	กำหนดค่าชื่อไฟล์พร้อมนามสกุล
getFileExtension	-	**String**	คืนค่านามสกุลไฟล์
setFileExtension	**String** fileExtension	**-**	กำหนดค่านามสกุลไฟล์
getFullFilePath	-	**String**	คืนค่าตำแหน่งเต็มของไฟล์
setFullFilePath	**String** fullFilePath	**-**	กำหนดค่าตำแหน่งเต็มของไฟล์

Class PKCS11Instance

Method name	Parameter in	Return	Remark
PKCS11Instance	**String** tokenName, **String** libraryPath, **String** pin, **String** keyStorePassword, **String** searchPhase	**-**	Class constructor
PKCS11Instance	**String** tokenName, **String** libraryPath, **int** slot, **String** pin, **String** keyStorePassword, **String** searchPhase	-	Class constructor
getLibraryPath	-	**String**	คืนค่าตำแหน่งของไฟล์ .dll ของ PKCS11

setLibraryPath	**String** libraryPath,	-	กำหนดค่าตำแหน่งของไฟล์ .dll ของ PKCS11
getTokenName	-	**String**	คืนค่า Token name
setTokenName	**String** tokenName	-	กำหนดค่า Token name
getPin	-	**String**	คืนค่า Token Pin
setPin	**String** pin	-	กำหนดค่า Token Pin
getSlot	-	**int**	คืนค่า PKCS11 Slot
setSlot	**int** slot	-	กำหนดค่า PKCS11 Slot
getKeyStorePassword	-	**String**	คืนค่ารหัสผ่านของ KeyStore
setKeyStorePassword	**String** keyStorePassword	-	กำหนดค่ารหัสผ่านของ KeyStore
getSearchPhase	-	**String**	คืนค่าคำค้นหา Certificate
setSearchPhase	**String** searchPhase	-	กำหนดค่าคำค้นหา Certificate

Class PKCS12Instance

Method name	Parameter in	Return	Remark
PKCS12Instance	**String** filePath, **String** keyStorePassword		Class constructor
getFilePath		**String**	คืนค่าตำแหน่งของไฟล์ PFX, P12
setFilePath	**String** filePath		กำหนดค่าตำแหน่งของไฟล์ ไฟล์ PFX, P12
getKeyStorePassword		**String**	คืนค่ารหัสผ่านของ KeyStore
setKeyStorePassword	**String** keyStorePassword		กำหนดค่ารหัสผ่านของ KeyStore

Class SignatureAppearance

Method name	Parameter in	Return	Remark
getX	-	**float**	คืนค่าตำแหน่ง X ของ Signature
setX	**Float** X		กำหนดค่าตำแหน่ง X ของ Signature
getY	-	**float**	คืนค่าตำแหน่ง Y ของ Signature
setY	**Float** Y		กำหนดค่าตำแหน่ง Y ของ Signature
getWidth	-	**float**	คืนค่าความกว้างของ Signature
setWidth	**Float** Width		กำหนดค่าความกว้างของ Signature
getHeight	-	**float**	คืนค่าความสูงของ Signature
setHeight	**Float** Height		กำหนดค่าความสูงของ Signature
getPageNumber	-	**int**	คืนค่าหมายเลขหน้า
setPageNumber	**int** pageNumber		กำหนดค่าหมายเลขหน้า
getSignatureFieldName	-	**String**	คืนค่าชื่อ Signature form
setSignatureFieldName	**String** signatureFieldName		กำหนดค่า Signature form
getReason	-	**String**	คืนค่าเหตุผลในการ Sign
setReason	**String** reason		กำหนดค่าเหตุผลในการ Sign
getLocation	-	**String**	คืนค่าตำแหน่งในการ Sign

Method name	Parameter in	Return	Remark
setLocation	**String** location		กำหนดค่าตำแหน่งในการ Sign
getSignatureLevel	-	**SignatureLevel**	คืนค่าระดับของ Signature
setSignatureLevel	**SignatureLevel** signatureLevel		กำหนดค่า Signature
getSignatureImage	-	**String**	คืนค่ารูป Signature
setSignatureImage	**String** signatureImage		กำหนดค่ารูป Signature
getSignaturePattern	-	**SignaturePattern**	คืนค่ารูปแบบ Signature
setSignaturePattern	**SignaturePattern** signaturePattern		กำหนดค่ารูปแบบ Signature
getSignatureVisibility	-	**SignatureVisibility**	คืนค่าการมองเห็นของ Signature
setSignatureVisibility	**SignatureVisibility** signatureVisibility	-	กำหนดค่าการมองเห็นของ Signature

Class TimeStamp

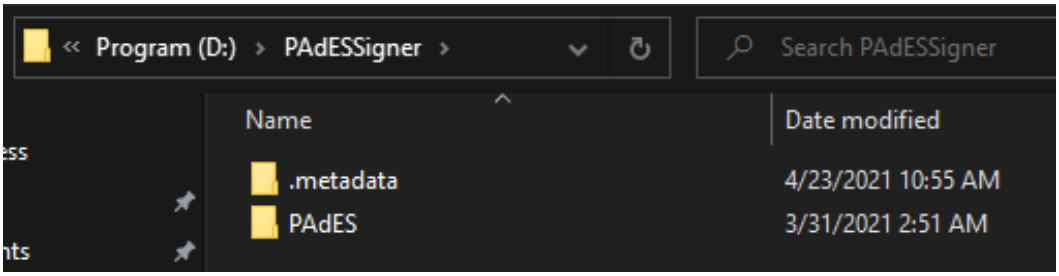
Method name	Parameter in	Return	Remark
TimeStamp	**TimeStampType** timeStampType	-	Class constructor
TimeStamp	**TimeStampType** timeStampType, **String** url, **TSAAuthenticationType** tsaAuthenticationType	-	Class constructor
TimeStamp	**TimeStampType** timeStampType, **String** url, **TSAAuthenticationType** tsaAuthenticationType, **String** username, **String** password	-	Class constructor
getTimeStampingType	-	**TimeStampType**	คืนค่ารูปแบบ TimeStamp
setTimeStampingType	**TimeStampType** timeStampType	-	กำหนดรูปแบบ TimeStamp
getTSAAuthenticationType	-	**TSAAuthenticationType**	คืนค่ารูปแบบ TSA Authentication
setTSAAuthenticationType	**TSAAuthenticationType** tsaAuthenticationType	-	กำหนดรูปแบบ TSA Authentication
getUsername	-	**String**	คืนค่าชื่อผู้ใช้
setUsername	**String** username	-	กำหนดชื่อผู้ใช้
getPassword	-	**String**	คืนค่ารหัสผ่าน
setPassword	**String** password	-	กำหนดรหัสผ่าน
getCertificatePath	-	**String**	คืนค่าตำแหน่ง certificate
setCertificatePath	**String** certificatePath	-	กำหนดตำแหน่ง certificate

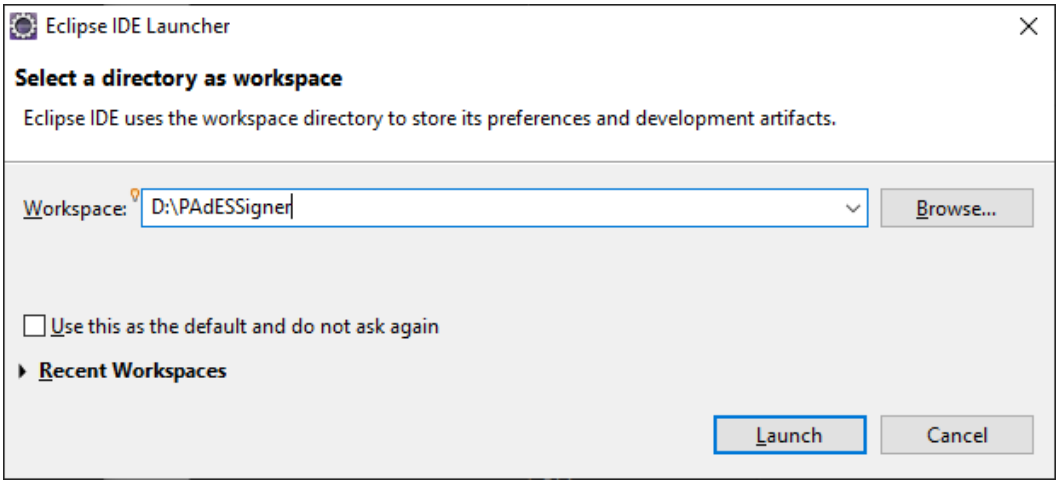
getURL	-	**String**	คืนค่า URL
setURL	**String** uRL	**-**	กำหนด URL

การใช้งานและการ Deploy library

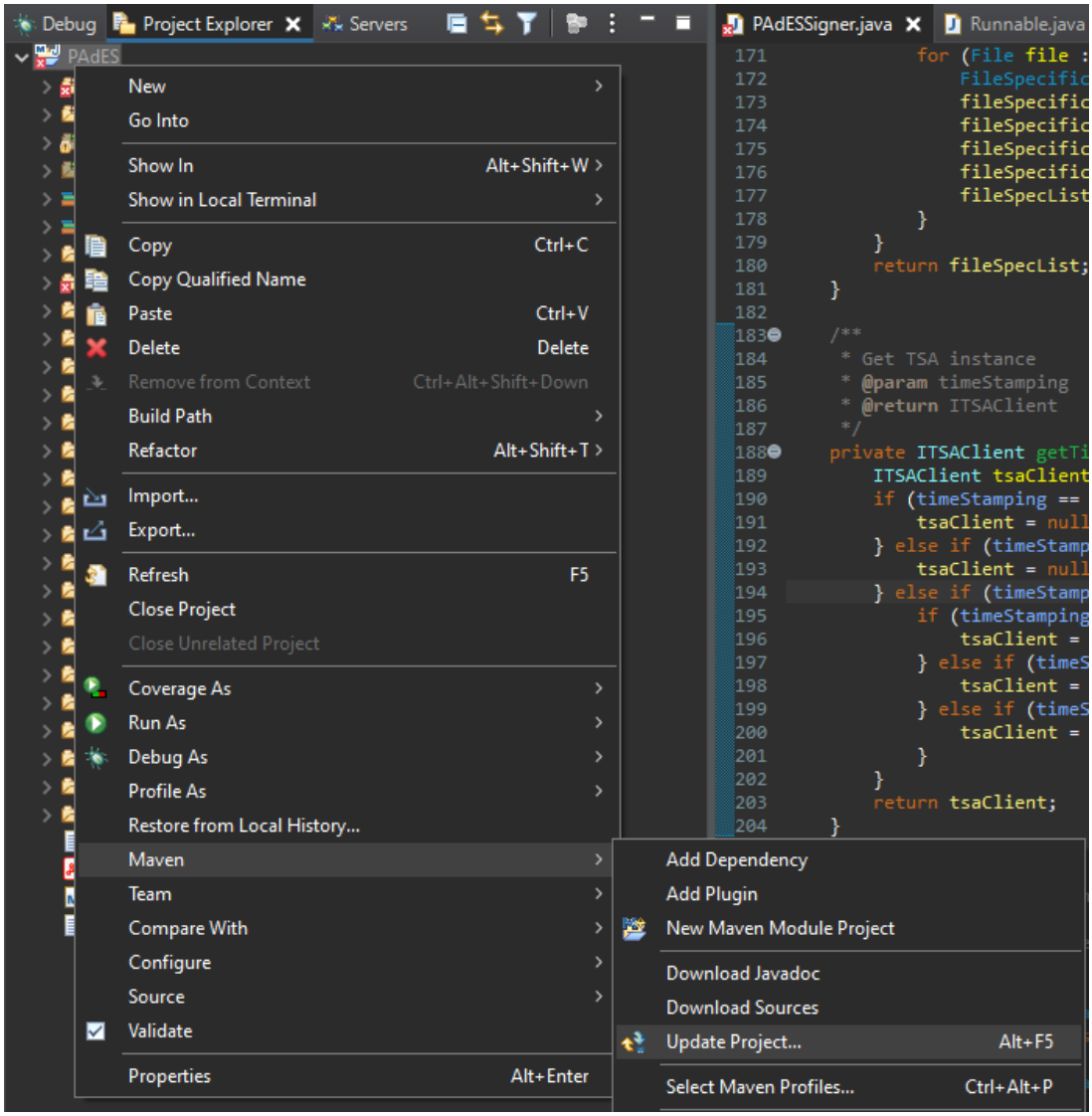
1. การเตรียม Project

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

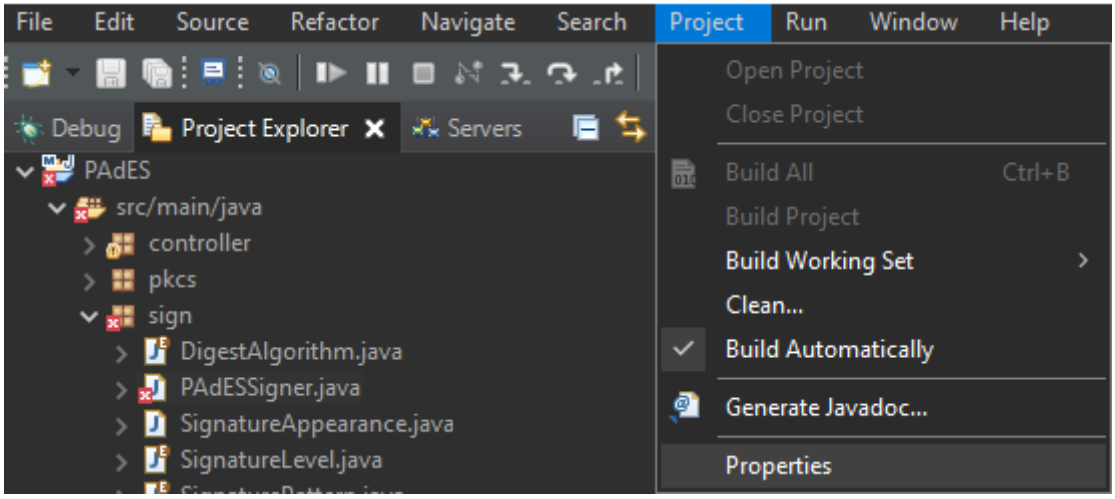




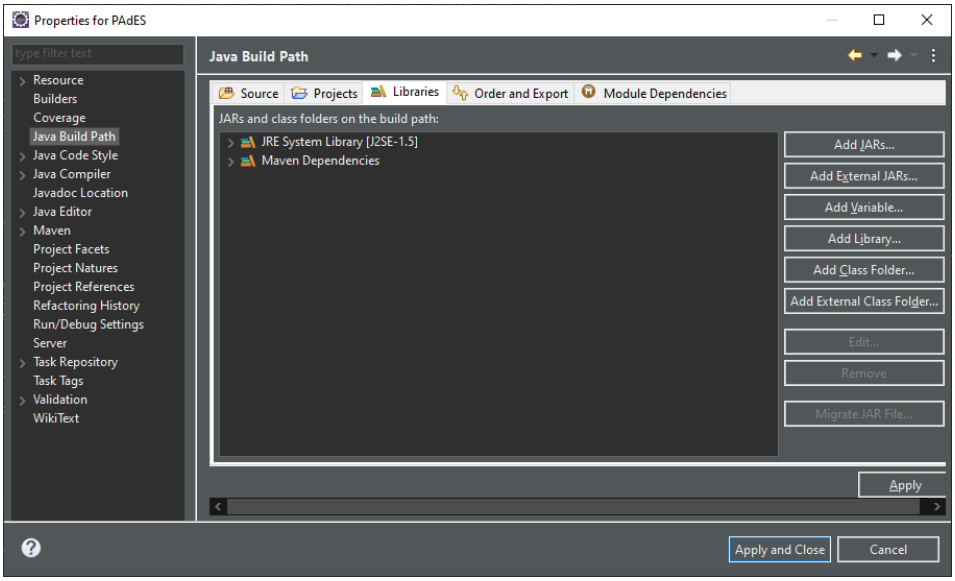
2. หลังจากเปิด project แล้ว ที่หน้าต่าง Project explorer ให้คลิกขวาที่ root folder ของ project แล้วเลือกไปที่ Maven > Update project จากนั้นรอจนกว่า project จะติดตั้ง dependency ที่จำเป็นเสร็จ



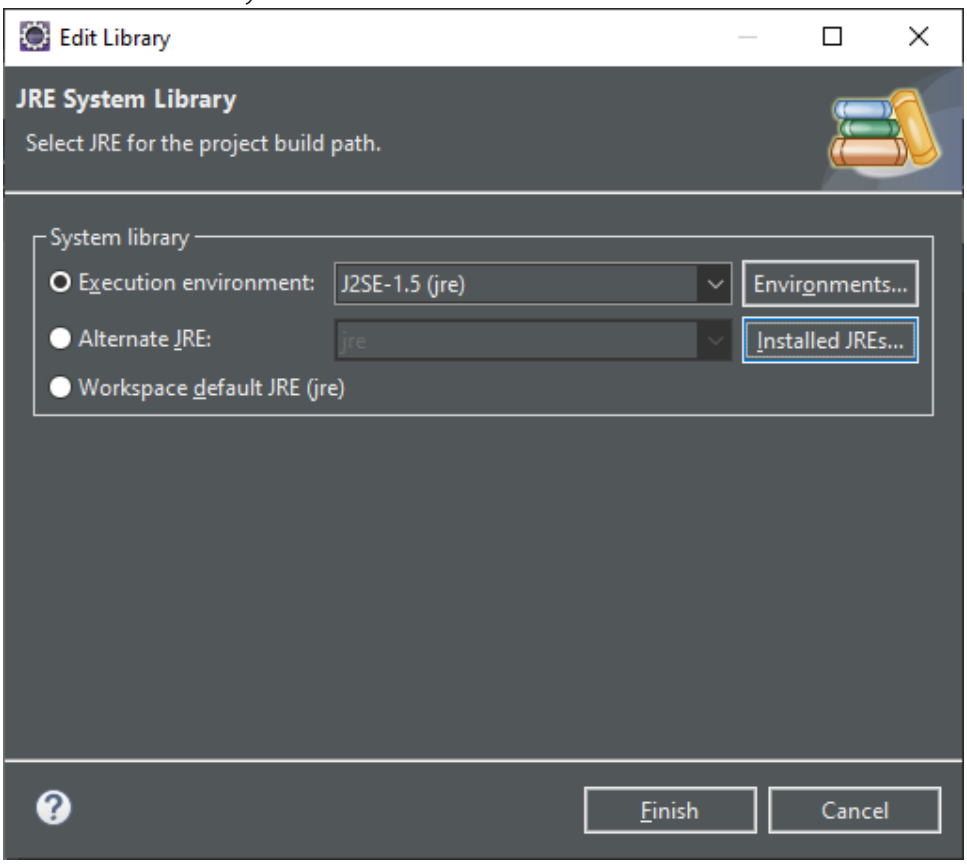
3. ที่ Menu bar เลือกที่ Project > Properties



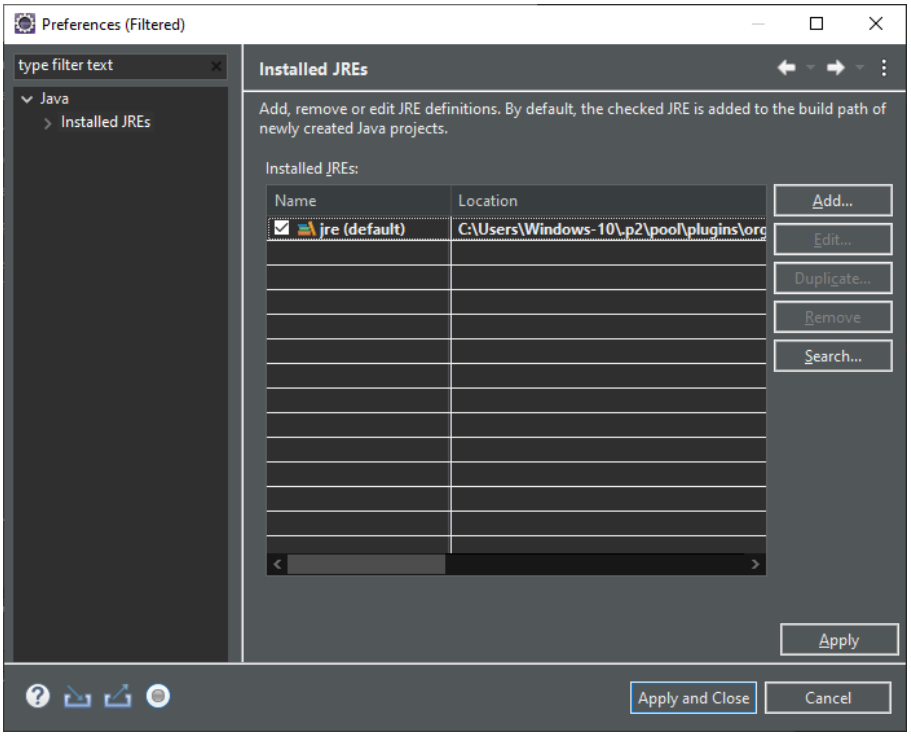
4. เลือกหัวข้อ Java build path จากนั้นเลือกที่ tab libraries แล้ว double click ที่ JRE System Library



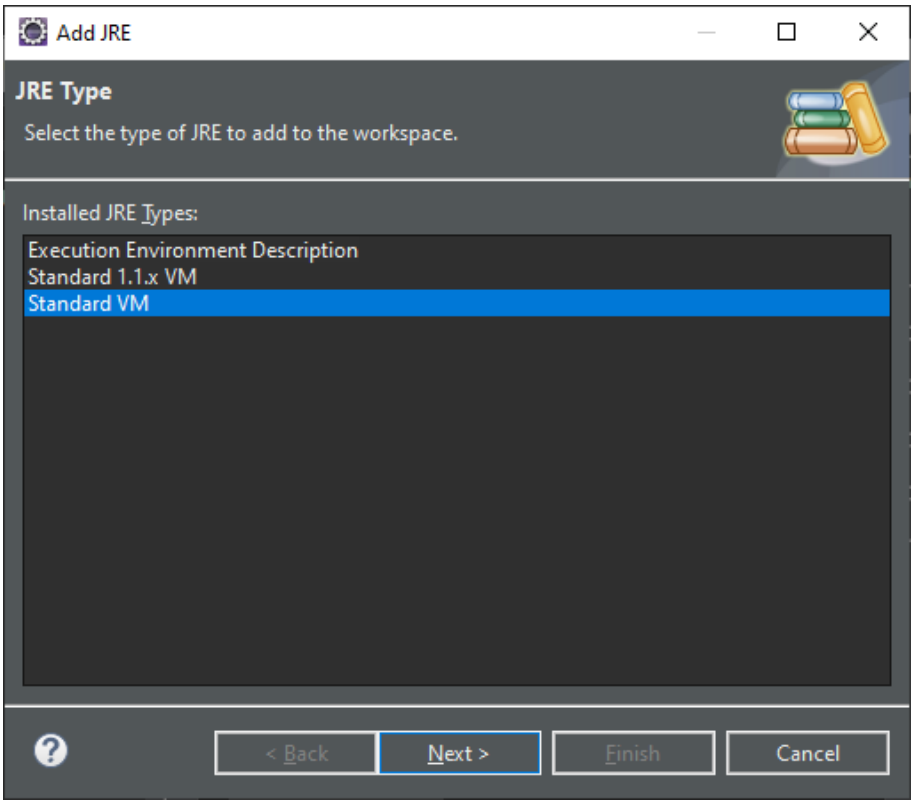
5. ที่หน้าต่าง Edit library เลือกที่ Installed JREs...



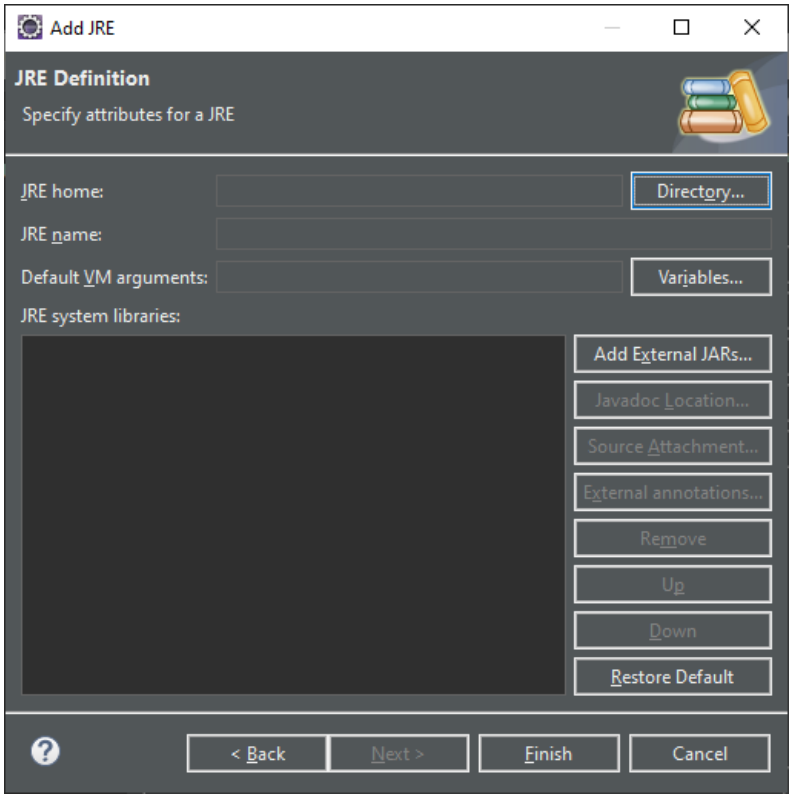
6. ที่หน้าต่าง Preferences (Filtered) เลือกที่ Add...



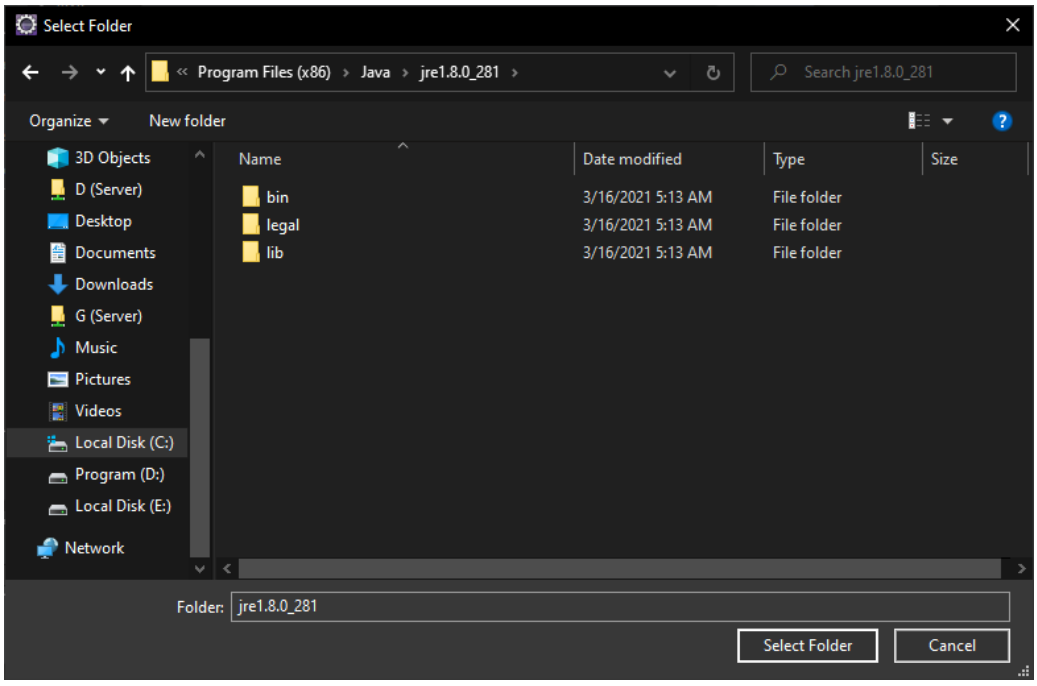
7. เลือก Standard VM จากนั้นกด Next



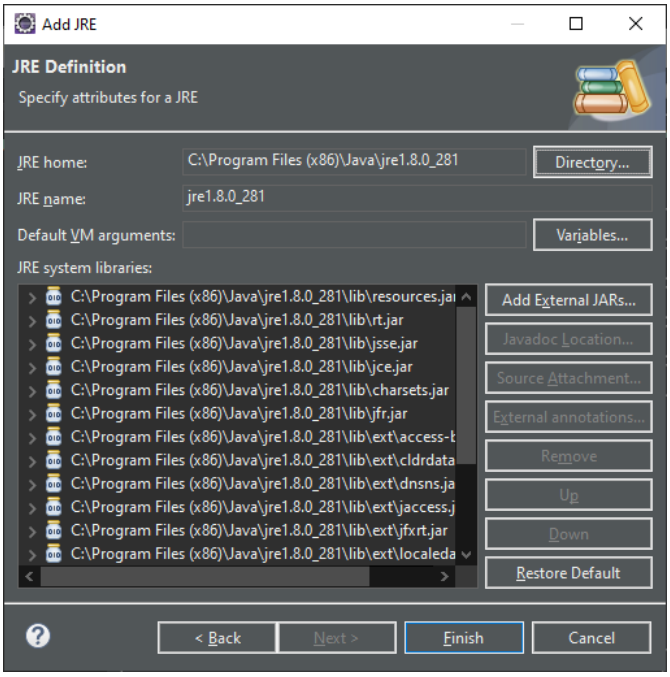
8. กดเลือก Directory...



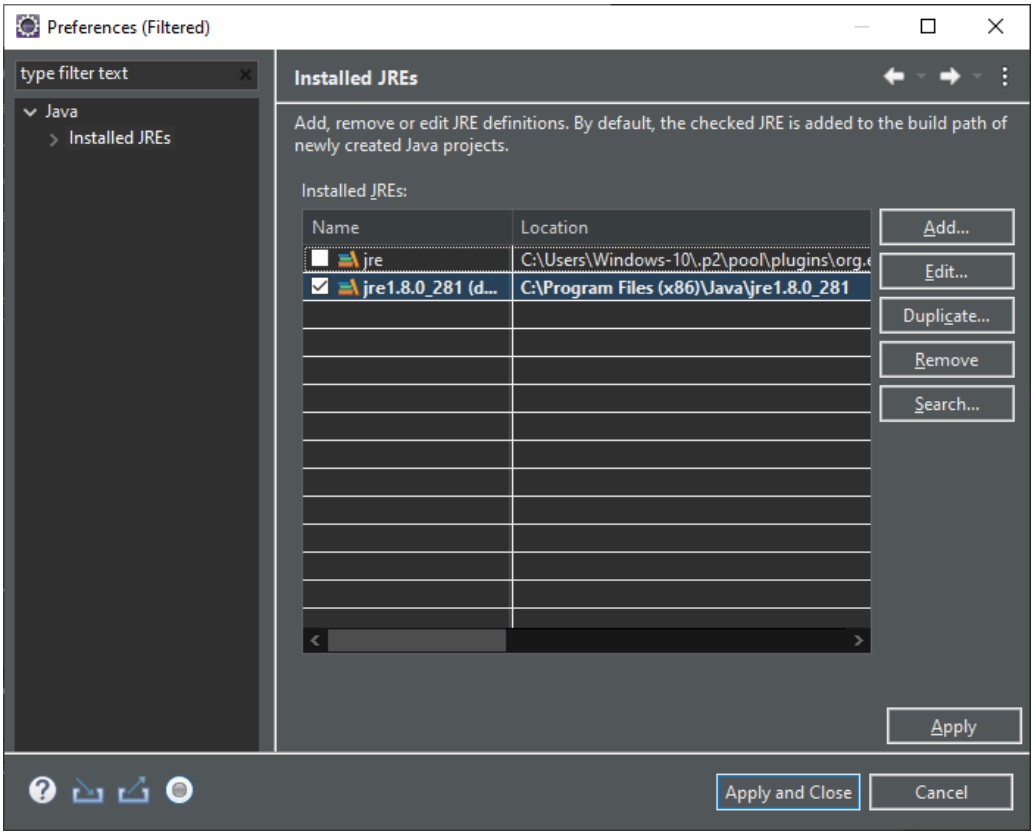
9. เลือกไปยังสถานที่ที่ติดตั้ง JRE 8 32-Bit ไว้ จากนั้นเลือก Select folder



10. ที่หน้าต่าง Add JRE จะปรากฏ System library ขึ้น กดปุ่ม Finish เพื่อยืนยันการเลือก

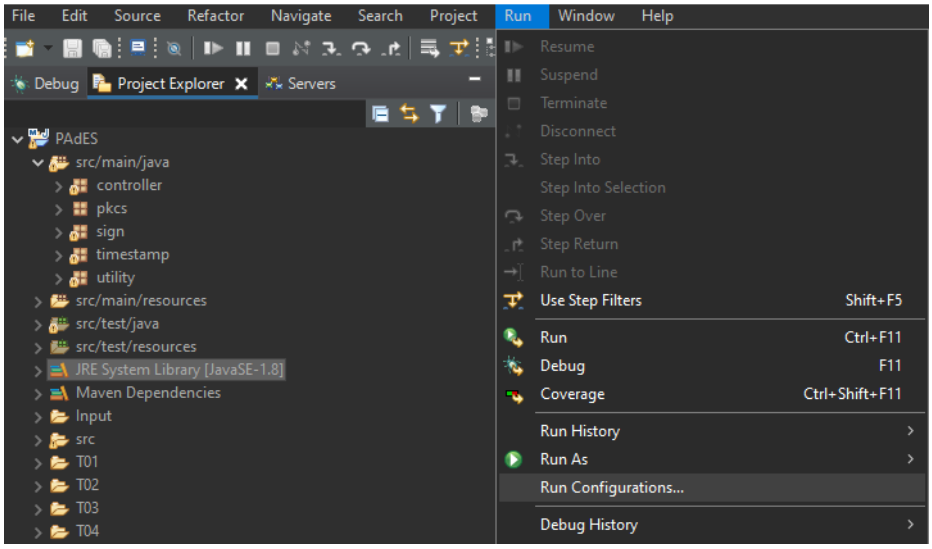


11. เมื่อกลับมาที่หน่าต่าง Preferences (Filtered) ให้เลือก JRE ที่เพิ่มเข้ามาใหม่ จากนั้นกด Apply and Close

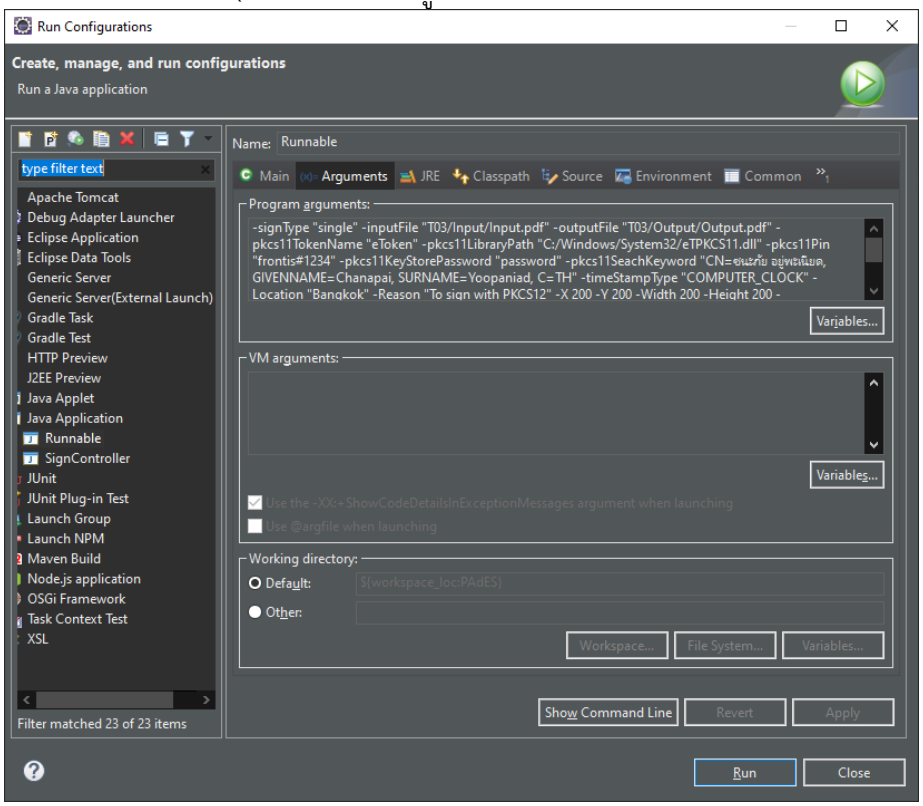


2. การเรียกใช้งานสำหรับการทดสอบ (Debug)

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



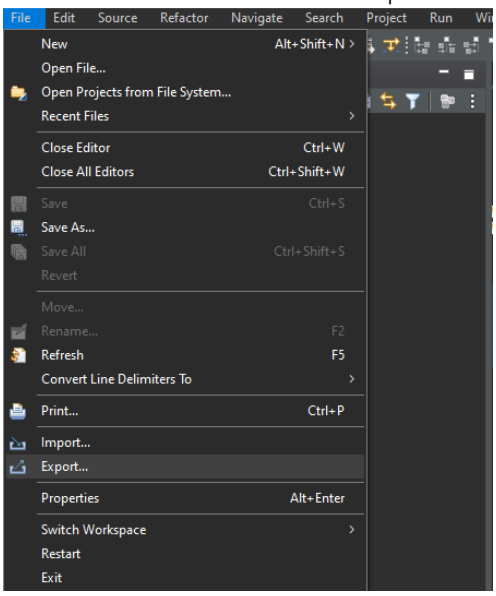
2. เลือกไปที่ Tab Arguments จากนั้นที่หัวข้อ Program arguments สามารถเปลี่ยนเป็น Argument ที่ต้องการทดสอบได้ (โดยรายละเอียดดูได้ที่หัวข้อการเรียกใช้งานผ่าน Command-line interface)



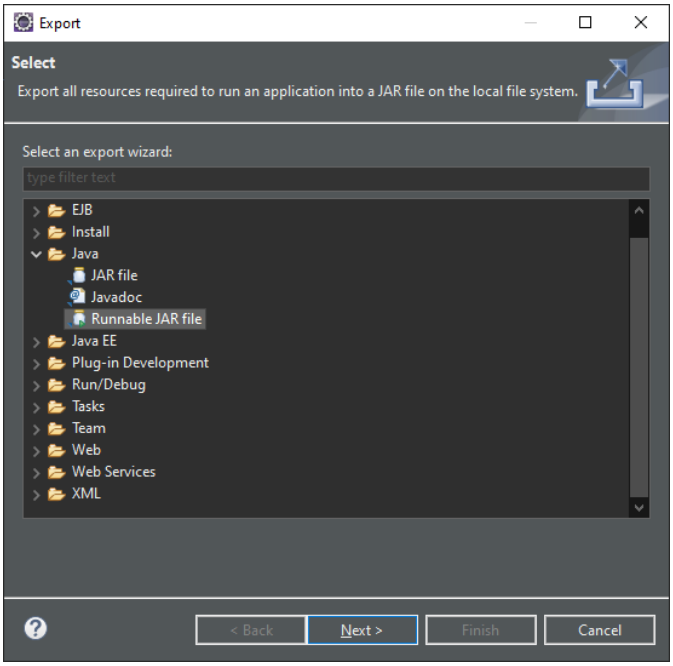
3. กด Run เพื่อดูผลลัพธ์

3. การ Deploy library (Executable jar)

1. ที่ Menu bar เลือกไปที่ File > Export



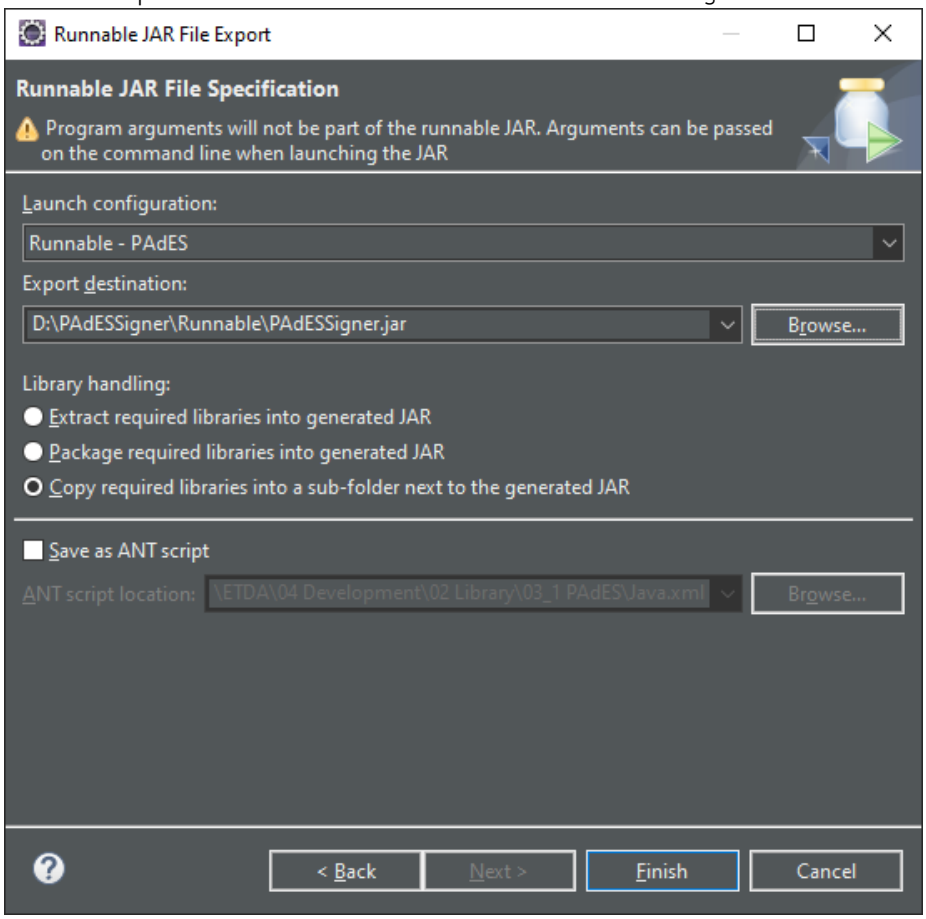
2. เลือกไปที่ Java > Runnable JAR file จากนั้นกด Next



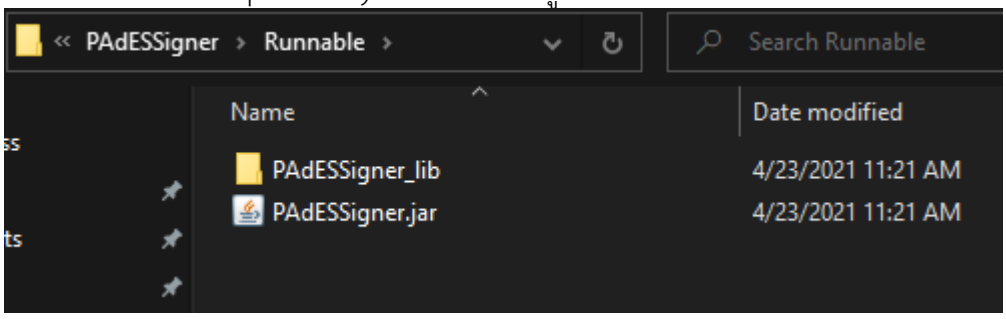
3. ใส่ค่าต่าง ๆ ดังนี้

- 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



การเรียกใช้งานผ่าน Command-line interface

1. รายละเอียด Argument

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

- **-signType "single | multiple"** กำหนดการ sign แบบไฟล์เดียวหรือหลายไฟล์
- **-inputFile "<PATH_TO_FILE>"** กำหนดไฟล์นำเข้า
- **-outputFile "<PATH_TO_FILE>"** กำหนดไฟล์ผลลัพธ์
- **-inputFolder "<PATH_TO_FOLDER>"** กำหนดโฟลเดอร์นำเข้า
- **-outputFolder "<PATH_TO_FOLDER>"** กำหนดโฟลเดอร์ผลลัพธ์
- **-outputSuffix "<ANY_Text>"** กำหนดชื่อต่อท้ายไฟล์ที่ sign แล้ว
- **-pkcs11TokenName "<NAME>"** กำหนดชื่อ PKCS11 Token
- **-pkcs11LibraryPath "<PATH_TO_FILE>"** กำหนดชื่อ PKCS11 Token
- **-pkcs11Pin "<PASSWORD>"** Password ของ Token
- **-pkcs11KeyStorePassword "<PASSWORD>"** password ของ Ketstore
- **-pkcs11SeachKeyword "<ANY_TEXT>"** คำค้นหา Certificate ใน Token
- **-pkcs12FilePath "<PATH_TO_FILE>"** ตำแหน่งของไฟล์ P12, PFX
- **-pkcs12Password "<PASSWORD>"** Password ของไฟล์ P12, PFX
- **-timeStampingType "TSA | COMPUTER_CLOCK"** รูปแบบ Timestamp
- **-tsaURL "<URL>"** URL ของ TSA
- **-tsaAuthenticationType "<NO_AUTHENTICATION | USERNAME_PASSWORD | CERTIFICATE>"** รูปแบบการ Authentication ของ TSA
- **-tsaUsername "<USERNAME>"** Username ของ TSA
- **-tsaPassword "<PASSWORD>"** Password ของ TSA
- **-tsaPKCS12File "<PATH_TO_FILE>"** ตำแหน่งของไฟล์ P12, PFX
- **-tsaPKCS12Password "<PASSWORD>"** Password ของไฟล์ P12, PFX
- **-Location “<ANY_TEXT>"** เหตุผลในการ Sign
- **-Reason “<ANY_TEXT>"** เหตุผลในการ Sign
- **-X <NUMBER>** กำหนดแหน่งของ Signature
- **-Y <NUMBER>** กำหนดแหน่งงของ Signature
- **-Width <NUMBER>** กำหนดความสูงของ Signature

- **-Height <NUMBER>** กำหนดความสูงของ Signature
- **-SignatureFieldName "<NAME>"** กำหนด Signature form
- **-PageNumber <NUMBER>** กำหนดหน้าที่ต้องการแสดง Signature
- -**SignatureLevel "<APPROVAL | NO_CHANGE | FORM_FILLING | FORM_FILLING_AND_ANNOTATION>"** กำหนดระดับของการ Sign
- **-SignatureVisibility "<VISIBLE | INVISIBLE>"** กำหนดการมองเห็นของ Signature
- **-SignaturePattern "<DESCRIPTION | NAME_AND_DESCRIPTION | GRAPHIC_AND_DESCRIPTION | GRAPHIC>"** กำหนด signature graphic
- **-digestAlgorithm "<SHA256 | SHA384 | SHA512>"** กำหนด hash function

2. ตัวอย่างการเรียกใช้งาน

- กรณี Sign PDF ไฟล์เดียว

```
java -jar PAdESSigner.jar -signType "single" -inputFile "Input.pdf" -outputFile "Output.pdf" -pkcs12FilePath "User_Certificate.p12" -pkcs12Password "password" -timeStampType "COMPUTER_CLOCK" -Location "Bangkok" -Reason "To sign with PKCS12" -X 200 -Y 200 -Width 200 -Height 200 -SignatureFieldName "Test_SIG" -PageNumber 1 -SignatureLevel "APPROVAL" -SignatureVisibility "VISIBLE" -SignaturePattern "DESCRIPTION" -digestAlgorithm "SHA256"
```

- กรณี sign แบบ Bulk (Sign ทั้ง folder)

```
java -jar PAdESSigner.jar -signType "multiple" -inputFolder "Input/" -outputFolder "Output.pdf" -outputSuffix "_Signed" -pkcs12FilePath "User_Certificate.p12" -pkcs12Password "password" -timeStampType "COMPUTER_CLOCK" -Location "Bangkok" -Reason "To sign with PKCS12" -X 200 -Y 200 -Width 200 -Height 200 -SignatureFieldName "Test_SIG" -PageNumber 1 -SignatureLevel "APPROVAL" -SignatureVisibility "VISIBLE" -SignaturePattern "DESCRIPTION" -digestAlgorithm "SHA256"
```

- กรณี Sign โดยใช้ TSA Server

```
java -jar PAdESSigner.jar -signType "single" -inputFile "Input.pdf" -outputFile " Output.pdf" -pkcs12FilePath "User_Certificate.p12" -pkcs12Password "password" -timeStampType "TSA" -tsaURL "https://TSA_URL" -tsaAuthenticationType “CERTIFICATE” -tsaPKCS12File "TSA_Certification.p12" -tsaPKCS12Password "password" -Location "Bangkok" -Reason "To sign with PKCS12" -X 200 -Y 200 -Width 200 -Height 200 -SignatureFieldName "Test_SIG" -PageNumber 1 -SignatureLevel "APPROVAL" -SignatureVisibility "VISIBLE" -SignaturePattern "DESCRIPTION" -digestAlgorithm "SHA256"
```