# บริการ TEDA e-Timestamping

Version: 2.1

October 6, 2021





# Change History

Date	Version	Description	Author
01/02/2019	0.1	รวบรวมจากคู่มือที่มีอยู่ เพื่อเป็น version เริ่มต้น	Jariya S.
		1) คู่มือ_OpenSSL for Windows_20181024.docx	
		2) คู่มือ_Runnable jar for	
		PDFTimestamping_20181227_win.docx	
		3) Timestamp API Specification_V2.0.docx	
11/02/2019	1.0	Baseline document	Jariya S.
25/04/2019	1.1	Change test.time to time-test	Pragasit C.
09/10/2019	1.2	Update authentication step for TSA-API	Natham T.
12/02/2020	1.3	Correct test TSA API url and add negative	Natham T.
		response example for API key	
11/06/2020	1.4	Remove invalid API URL	Natham T.
03/02/2021	1.5	1) Update P12 creation step to support	Natham T.
		certificate that has intermediate chain	
		2) Update sample timestamp source code	
		screenshot	
17/02/2021	1.6	เพิ่มส่วน "การขอใช้บริการ", "ขั้นตอนที่ 4 การติดตั้ง	Jariya S.
		KeyStoreFile เพื่อทดสอบฯ" และ "การติดตั้ง	
		Timestamp Proxy Server"	
03/03/2021	1.7	เพิ่มภาคผนวก ตัวอย่างหนังสือขอใช้บริการ	Avatsada C.
10/08/2021	1.8	แก้ไขเนื้อหา แก้ไขคำผิด และปรับปรุงรูปจากระบบ TEDA	Napon S.
		Web Validation Portal	
27/08/2021	2.0	Baseline document	Jariya S.
04/10/2021	2.1	เพิ่ม <u>support-center@etda.or.th</u> ในบทที่ 1 การขอใช้	Jariya S.
		บริการ	

#### ETDA www.etda.or.th

ภาพรวม	4
บทที่ 1 การขอใช้บริการ e-Timestamping	5
การขอทดสอบใช้บริการ	5
การขอใช้บริการบนระบบปฏิบัติการจริง	5
บทที่ 2 การยืนยันตัวตนผู้ขอใช้บริการด้วยรูปแบบ Certificate Authentication	7
ขั้นตอนที่ 2.1: การติดตั้ง OpenSSL	7
ขั้นตอนที่ 2.2: การสร้าง Certificate Signing Request (CSR) โดยใช้ OpenSSL บน Windows	8
ขั้นตอนที่ 2.3: การรวม certificate กับ private key โดยใช้ OpenSSL บน Windows	10
การทดสอบเรียก timestamp ผ่าน Adobe Acrobat Reader DC	11
บทที่ 3 การยืนยันตัวตนผู้ขอใช้บริการด้วยรูปแบบ API key	19
บทที่ 4 การติดตั้ง Timestamp Proxy Server	20
บทที่ 5 การสร้าง <sub>Runnable JAR file</sub> เพื่อประทับรองรองเวลาไฟล์ pDF	21
ขั้นตอนที่ 5.1: การ import project PDFTimestamping	21
ขั้นตอนที่ 5.2: การสร้าง JAR file	24
ขั้นตอนที่ 5.3: การนำ Runnable JAR file มาใช้	26
บทที่ 6 Timestamp API Specification	32
HTTP Request : Timestamp request	33
HTTP Response : Timestamp response	35
Example Request and Response:	37
บทที่ 7 การตรวจสอบความถูกต้องของการประทับรับรองเวลา ด้วย TEDA Web Validation Portal	43
ภาคผนวก	45



# ภาพรวม

e-Timestamping คือ บริการประทับรับรองเวลาอิเล็กทรอนิกส์ เพื่อรับรองการมีอยู่ของเอกสาร ณ เวลานั้น ๆ และสามารถใช้ในการตรวจสอบว่าเอกสารอิเล็กทรอนิกส์ที่ได้รับการประทับรับรองเวลาแล้วนั้น ถูก แก้ไขหรือไม่

ระบบที่ให้บริการประทับรับรองเวลาอิเล็กทรอนิกส์ ของ ETDA หรือ Timestamping Authority (TSA) มีการตรวจสอบความถูกต้องของเวลาบนเครื่องแม่ข่ายที่ใช้อ้างอิง (NTP Server) เทียบกันจาก 3 แหล่ง คือ สถาบันมาตรวิทยาแห่งชาติ, กรมอุทกศาสตร์ กองทัพเรือ และ สพธอ.

หลักการทำงานโดยย่อของบริการนี้คือ ผู้ใช้บริการส่ง hash ของสิ่งที่ต้องการทำ timestamp เข้ามา แล้วระบบจะส่ง timestamp token ที่คู่กับ hash นั้น กลับไปให้ ซึ่งผู้ใช้ฯสามารถนำ timestamp token ไป ใช้งานได้ ทั้งนำไปใช้เป็นเวลาในการลงลายมือชื่อดิจิทัล (แทนที่จะใช้เวลาจากเครื่องที่ลงลายมือชื่อดิจิทัล) หรือประกอบเข้ากับสิ่งที่ต้องการประทับรับรองเวลา

โดยหากต่อมามีการแก้ไขเนื้อหาของสิ่งที่ได้ทำ timestamp ไปแล้ว ค่า hash ที่คำนวณใหม่ (โดยใช้ ฟังก์ชั่นเดียวกับตอนที่คำนวณเพื่อส่งไปทำ timestamp) ก็จะเปลี่ยนแปลงไปด้วย หากทำการตรวจสอบ ก็จะ ทราบได้ว่าเกิดการแก้ไขขึ้น ซึ่ง ETDA ได้จัดทำ website และให้บริการ API ที่ใช้ในการตรวจสอบ โดย ปัจจุบันรองรับไฟล์ประเภท PDF และ XML ดูรายละเอียดได้ที่บท "<u>การตรวจสอบความถูกต้องของการ</u> <u>ประทับรับรองเวลา ด้วย TEDA Web Validation Portal</u>"

# บริการ e-Timestamping มีอยู่ 2 รูปแบบคือ

 การรับ-ส่ง message ตามรูปแบบของ RFC3161: สามารถ download ตัวอย่าง source code (JAVA) ได้จาก project PDFTimestamping ใน ETDA GitHub ที่ https://github.com/ETDA/PDFTimestamping หรือหากต้องการประทับรับรองเวลาบนไฟล์ PDF ก็ สามารถสร้าง Runnable JAR file จาก project นี้ แล้วเรียกใช้งานได้ โดยดูขั้นตอนได้ที่บท "<u>การสร้าง</u> <u>Runnable JAR file เพื่อปรับทับรับรองเวลาลงบนไฟล์ PDF</u>"

2. HTTP API: ดูรายละเอียดได้ที่บทที่ 6 "<u>Timestamp API Specification</u>"

การยืนยันตัวตนของผู้ขอใช้บริการ สำหรับบริการ e-Timestamping ในรูปแบบ RFC3161 จะใช้ การยืนยันตัวตนด้วยรูปแบบ Certificate Authentication ส่วนบริการ e-Timestamping ในรูปแบบ HTTP API จะใช้การยืนยันตัวตนด้วยรูปแบบ API key แทนซึ่งสามารถดูรายละเอียดขั้นตอนได้ที่บทต่อๆไป



# บทที่ 1 การขอใช้บริการ e-Timestamping

สพธอ. ขอสงวนสิทธิ์ในการให้บริการ e-Timestamping แก่หน่วยงานรัฐเท่านั้น โดยไม่คิด ค่าบริการ (ข้อมูล ณ เดือน ต.ค. 2564) ซึ่งการขอใช้บริการมีขั้นตอนดังนี้

# การขอทดสอบใช้บริการ

- 1. ดาวน์โหลดไฟล์ เพื่อกรอก<u>แบบคำขอทดสอบบริการ (TEDA Test Application Form)</u>
- ทำการสร้าง Certificate Signing Request (CSR) file ตามรายละเอียดในบท "<u>การยืนยันตัวตนผู้ขอ</u> <u>ใช้บริการด้วยรูปแบบ Certificate Authentication</u>" แต่หากเลือกใช้บริการในรูปแบบ HTTP API ให้ข้ามขั้นตอนนี้ไป
- ส่งแบบคำขอทดสอบบริการที่กรอกแล้ว มาทางอีเมลถึง support-center@etda.or.th และ eservice@etda.or.th

To: support-center@etda.or.th; eservice@etda.or.th

Subject: ขอทดสอบการใช้งานระบบ e-Timestamping

โดยหากใช้บริการในรูปแบบ RFC3161 ให้แนบ CSR file จากขั้นตอนที่ 2 มาด้วย จากนั้น ETDA จะ สร้าง credential ในการเข้าใช้ระบบส่งกลับไปให้ทางอีเมล พร้อม URL ของระบบทดสอบ

4. เมื่อทำการติดตั้ง credential ที่ได้รับจาก ETDA ลงในระบบของผู้ใช้บริการแล้ว สามารถทดสอบ บริการโดยส่งไฟล์เข้ามาทำ Timestamp ที่ระบบทดสอบตาม URL ที่แจ้งไว้

# การขอใช้บริการบนระบบปฏิบัติการจริง

- ทำหนังสือขอใช้บริการจากหัวหน้าหน่วยงาน มาถึงผู้อำนวยการสำนักงานพัฒนาธุรกรรมทาง อิเล็กทรอนิกส์ (ดูตัวอย่างได้จาก <u>ภาคผนวก</u>)
- 2. ดาวน์โหลดไฟล์ เพื่อกรอก<u>แบบคำขอใช้บริกาวิ (TEDA Application Form)</u>
- ทำการสร้าง Certificate Signing Request (CSR) file สำหรับใช้บนระบบปฏิบัติการจริง แต่หาก เลือกใช้บริการในรูปแบบ HTTP API ให้ข้ามขั้นตอนนี้ไป
- 4. ส่งแบบคำขอใช้บริการที่กรอกแล้ว พร้อมหนังสือขอใช้บริการ และไฟล์ที่ทำ Timestamp จากระบบ ทดสอบ มาทางอีเมลถึง support-center@etda.or.th **และ** eservice@etda.or.th

To: support-center@etda.or.th; eservice@etda.or.th

Subject: ขอใช้งานระบบ e-Timestamping

โดยหากใช้บริการในรูปแบบ RFC3161 ให้แนบ CSR file จากขั้นตอนที่ 3 มาด้วย จากนั้น ETDA จะ สร้าง credential ในการเข้าใช้ระบบส่งกลับไปให้ทางอีเมล พร้อม URL ของระบบปฏิบัติการจริง



5. เมื่อทำการติดตั้ง credential ที่ได้รับจาก ETDA ลงในระบบของผู้ใช้บริการแล้ว สามารถใช้บริการ โดยส่งไฟล์เข้ามาทำ Timestamp ได้ที่ระบบปฏิบัติการจริงตาม URL ที่แจ้งไว้



# บทที่ 2 การยืนยันตัวตนผู้ขอใช้บริการด้วยรูปแบบ Certificate Authentication

การเรียกใช้บริการ e-Timestamping จะต้องมีขั้นตอนในการยืนยันตัวตนสำหรับเข้าใช้บริการ โดย ในส่วนนี้จะเป็นการอธิบายขั้นตอนในการสร้าง private key และ Certificate Signing Request (CSR) file (เพื่อส่งให้ ETDA ในการลงทะเบียนใบรับรอง) โดยใช้ OpenSSL บน Windows ซึ่งมีขั้นตอนดังต่อไปนี้

# ขั้นตอนที่ 2.1: การติดตั้ง OpenSSL

- 1. เปิดลิงก์ต่อไปนี้ในเบราว์เซอร์ของคุณ <u>https://slproweb.com/products/Win32OpenSSL.html</u>
- 2. เลื่อนไปที่หัวข้อ "Download Win32 OpenSSL"
- ในคู่มือนี้ เลือกเป็น Win64 OpenSSL v.1.1.1 (ในตัวอย่างเลือกเป็นแบบ 64 บิต ซึ่งอาจเลือกให้ตรง กับคอมพิวเตอร์ของคุณได้ โดยการเลือก 32 บิตหรือ 64 บิต จะมีผลต่อชื่อโฟลเดอร์ ในขั้นตอนที่ 2.2 และ ขั้นตอนที่ 2.3 และขอให้ใช้ edition อื่นๆที่ไม่ใช่ "Light" edition)

nload Win32 OpenSSL today u	sing the links be	low!
File	Туре	Description
Win64 OpenSSL v1.1.1 Light <u>EXE   MSI (experimental)</u>	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.1.1 (Recommended for users by the creators of <u>OpenSSL</u> ). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.1.1 EXE   <u>MSI (experimental)</u>	43MB Installer	Installs Win64 OpenSSL v1.1.1 (Recommended for software developers by the creators of <u>OpenSSL</u> ). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.1 Light <u>EXE   MSI (experimental)</u>	3MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.1.1 (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.1 EXE   MSI (experimental)	30MB Installer	Installs Win32 OpenSSL v1.1.1 (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

 คลิกไปที่ "EXE" เพื่อทำการดาวน์โหลดโปรแกรม OpenSSL v.1.1.1 และทำการติดตั้งโปรแกรมโดย ใช้ค่าเริ่มต้น



# ขั้นตอนที่ 2.2: การสร้าง Certificate Signing Request (CSR) โดยใช้ OpenSSL บน Windows

- 1. คลิกไปที่ปุ่ม **Start** พิมพ์ว่า CMD (Command Prompt)
- 2. คลิกขวาที่ โปรแกรม CMD และเลือกไปที่ "Run as Administrator"
- 3. จะปรากฏโปรแกรม CMD ขึ้น
- 4. พิมพ์คำสั่งต่อไปนี้ และกด Enter

# cd C:\Program Files\OpenSSL-Win64\bin

- 5. จะเห็นว่าบรรทัดที่พิมพ์ เปลี่ยนเป็น C:\Program Files\OpenSSL-Win64\bin
- 6. พิมพ์คำสั่งต่อไปนี้ และกด Enter

# openssl genrsa -out private-key.key 2048

7. พิมพ์คำสั่งต่อไปนี้ และกด Enter

# openssl req -new -key private-key.key -out csr.txt -config cnf\openssl.cnf

- 8. กรอกข้อมูลองค์กรในฟิลด์ที่ต้องการ:
  - Country Name: \_\_\_\_\_ (ตัวอย่าง: *TH*)
  - State or Province: \_\_\_\_\_(ตัวอย่าง: Bangkok)
  - Locality or City: \_\_\_\_\_ (ตัวอย่าง: Huai Khwang)
  - Company: \_\_\_\_\_ (ตัวอย่าง: A Test Company Ltd.)
  - Organizational Unit: \_\_\_\_\_ (ตัวอย่าง: Information Technology Department)
  - Common Name: \_\_\_\_\_ (ตัวอย่าง: IT Dept.)
  - Email: \_\_\_\_\_ (ตัวอย่าง: itadmin@atest.co.th)
  - Challenge Password: เว้นว่างและกด Enter
  - Optional company name: เว้นว่างและกด Enter
- Private key และ Public key จะถูกสร้างขึ้น โดย Private key (ตัวอย่างในที่นี้ กำหนดชื่อไฟล์เป็น private-key.key) จะถูกสร้างเพื่อใช้สำหรับการถอดรหัส และ Public keys ซึ่งอยู่ในรูปแบบ Certificate Signing Request (ตัวอย่างในที่นี้ กำหนดชื่อไฟล์เป็น csr.txt) จะถูกสร้างเพื่อใช้ในการลงทะเบียน ใบรับรอง



10. สร้างโฟลเดอร์ใหม่ (ตัวอย่างในที่นี้ กำหนดชื่อโฟลเดอร์เป็น certificate) โดยนำไปไว้ในไดรฟ์ C

		—
🏪 C:\	ٽ ~	Search Windows (C:)
Name		Туре
certificate		File folder

 11. ย้าย private key (ตัวอย่างในที่นี้ ไฟล์อยู่ที่ C:\Program Files\OpenSSL-Win64\bin\privatekey.key) and CSR file (ตัวอย่างในที่นี้ ไฟล์อยู่ที่ C:\Program Files\OpenSSL-Win64\bin\csr.txt) ไปไว้ที่โฟลเดอร์ใหม่ที่เพิ่งสร้างขึ้น (ตัวอย่างในที่นี้ กำหนดเป็น C:\certificate)

🔒 bin				- 0	×
$\leftarrow \rightarrow \land \uparrow$	C:\Program Files\OpenSSL-Win64\bin		✓ O Search bin		Q
A Quick of	Name	Туре	Size		^
Des Des	csr.txt	Text Document KEY File	2 KB 2 KB		
Doc 🖈	CA.pl	PL File Application extens	8 KB 68 KB		
■ Pict # ✓	Selected	Application extens	44 KB		↓
15 terns - Fitern	Secto				<u> </u>
certificate			_		×
$\leftrightarrow$ $\rightarrow$ $\star$	↑ 📙 C:\certificate	~ Ū	Search certificate		Q
🖈 Ouick a	A Name		Туре	Size	
Dec 🖈	csr.bxt		Text Document		2 KB
🚽 Dov 🖈	private-key.key		KEY File		2 KB
🚆 Doc 🖈					
📰 Pict 🖈	v <				>
2 items					:==

12. CSR file (ตัวอย่างในที่นี้ กำหนดชื่อไฟล์เป็น csr.txt) ในขณะนี้พร้อมแล้วที่จะทำการลงทะเบียนใบรับรอง
 13. ส่ง CSR file ไปให้ ETDA เพื่อทำการลงทะเบียนใบรับรอง



# ขั้นตอนที่ 2.3: การรวม certificate กับ private key โดยใช้ OpenSSL บน Windows

- เมื่อ ETDA ส่งไฟล์ certificate สำหรับ authentication (ตัวอย่างในที่นี้ กำหนดชื่อไฟล์เป็น eservice.cer ) และไฟล์ certificate ของ CA ที่ออก certificate สำหรับ authentication (ตัวอย่างใน ที่นี้ กำหนดชื่อไฟล์เป็น authenCA.cer) กลับมา ให้ย้าย certificate ที่ ETDA ส่งมา ไปไว้ที่โฟลเดอร์ เดียวกับที่เก็บ private key (ตัวอย่างในที่นี้ กำหนดเป็น C:\certificate)
- 2. คลิกไปที่ปุ่ม Start พิมพ์ว่า CMD (Command Prompt)
- 3. คลิกขวาที่ โปรแกรม CMD และเลือกไปที่ "Run as Administrator"
- 4. จะปรากฏโปรแกรม CMD ขึ้น
- 5. พิมพ์คำสั่งต่อไปนี้ และกด Enter
  - cd C:\Program Files\OpenSSL-Win64\bin
- 6. จะเห็นว่าบรรทัดที่พิมพ์เปลี่ยนเป็น C:\Program Files\OpenSSL-Win64\bin
- 7. พิมพ์คำสั่งต่อไปนี้ และกด Enter

openssl pkcs12 -export -in c:\certificate\eservice.cer -certfile c:\certificate

\authenCa.cer -inkey c:\certificate\private-key.key -out eservice.p12

- 8. กรอก Export Password และยืนยัน Export password อีกครั้ง
- certificate กับ private key จะถูกรวมกันเป็น keystoreFile (ตัวอย่างในที่นี้ กำหนดชื่อ keystoreFile เป็น eservice.p12)

# หมายเหตุ:

 โปรดเก็บรหัสผ่าน (Export Password/keystorePassword) ไว้ในที่ปลอดภัย และเก็บแยกกับ keystoreFile (.P12)

 หากทำการเรียกใช้บริการ e-Timestamping แล้วพบ ข้อผิดพลาดแจ้ง "could not create SSL/TLS secure channel." โปรดทำตาม <u>ขั้นตอนที่ 2A.2 การติดตั้ง Trust Root CA</u> กับเครื่องที่ใช้บริการ e-Timestamping



# การทดสอบเรียก timestamp ผ่าน Adobe Acrobat Reader DC

KeyStoreFile ที่ได้มาจาก ขั้นตอนที่ 2.3 จะใช้เป็น certificate ในการยืนยันตัวตนผู้ใช้บริการ ในการ เรียกใช้ timestamp ไปที่ server ของ ETDA โดยหัวข้อนี้ มีจุดประสงค์เพื่อเป็น quick guide เพื่อทดสอบให้เห็น ว่าจาก client เครื่องหนึ่งๆ สามารถเรียกใช้ timestamp ได้อย่างไร ผ่านการเรียกโปรแกรมที่มีการใช้งาน แพร่หลายอย่าง Adobe Acrobat Reader DC แต่หากหน่วยงานท<u>ำการติดตั้ง Timestamp Proxy Server</u> เอง ให้ข้ามขั้นตอนที่ 2A.1 – 2A.2 ไปขั้นตอนที่ 2A.3 ได้เลย

# ขั้นตอนที่ 2A.1: การติดตั้ง KeyStoreFile

1. ดับเบิ้ลคลิก KeyStoreFile

Name	~	Туре	Size
🏂 eservice.p12		Personal Inform	nati

คลิก Next

Welcome to the Certificate Import Wizard
This wizard helps you copy certificates, certificate trust lists, and certificate revocation ists from your disk to a certificate store.
A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.
Store Location
Current User
○ <u>L</u> ocal Machine
To continue, dick Next.



3. ตรวจสอบ Path File (ซึ่งจะแสดงค่าเริ่มต้นเป็น path ที่วางไฟล์นั้นอยู่) แล้วคลิก Next



4. กรอกรหัสผ่าน (keystorePassword) จากขั้นตอนที่ 2.3 แล้วคลิก Next

	Private key protection	
	To maintain security, the private key was protected with a password.	
67	Type the password for the private key.	
	Password -	
	Display Password	
	Import options:	
	Enable strong private key protection. You will be prompted every time the	
	Mark this key as exportable. This will allow you to back up or transport your	
	keys at a later time.	
	Protect private key using virtualized-based security(Non-exportable)	
	☑ Indude <u>a</u> ll extended properties.	



# 5. คลิก Next

		×
~	🚰 Certificate Import Wizard	
	<b>Certificate Store</b> Certificate stores are system areas where certificates are l	kept.
	Windows can automatically select a certificate store, or you the certificate.	u can specify a location for
	$\textcircled{\begin{tabular}{ll} \bullet \\ \bullet $	e type of certificate
	$\bigcirc$ Place all certificates in the following store	
	Certificate store:	Browse
		Next
6 ดลิก Finish เพื่อสื่	ขึ้นสุดการติดตั้ง KayStoreFile จะบีการแล่	
0. 1611111151161106		
🗲 🛛 🐖 Certificate Import Wizard	^	
Completing the C	artificato Import Wizard	
completing the c		
The certificate will be import	ed after you dick Finish.	
You have specified the follow	vina settinas:	
Certificate Store Selected	Automatically determined by the wizard	
Content File Name	PFX E:\Users\napon\Desktop\ดูมือ\20210216\cert\newCAauthent	
		Certificate Import Wizard X
<	>	
	72	
		Ine Import was successful.
	<u>F</u> inish Cancel	ОК



# ขั้นตอนที่ 2A.2 : การติดตั้ง Trust Root CA

1. ดับเบิ้ลคลิก Root Certificate (ตัวอย่างในที่นี้ กำหนดชื่อ Root Certificate เป็น authenCA.cer)



2. คลิก Install Certificate...

	1				
This cer	Certifica	cate Infor te cannot authority.	mation be verified up	to a tri	usted
_	Issued t	o: ETDA C	CA CA - G2		
	Issued b	Y: ETDA R	oot CA – G1		
	Valid fro	m 22/8/25	62 <b>to</b> 17/8/258	81	
			Install Certifica	ite	Issuer <u>S</u> tatement

3. คลิก Next

~

Certificate Import Wizard	
Welcome to the Certificate Import	Wizard
This wizard helps you copy certificates, certificate trust ists from your disk to a certificate store.	lists, and certificate revocation
A certificate, which is issued by a certification authority, and contains information used to protect data or to esta connections. A certificate store is the system area wher	, is a confirmation of your identit ablish secure network e certificates are kept.
Store Location	
To continue, click Next.	



Cancel

Next

×

# 4. คลิก Next

rtificate Store Certificate stores are system	areas where certificates are kept.	
Windows can automatically the certificate.	elect a certificate store, or you can	specify a location for
Automatically select t	ne certificate store based on the typ	e of certificate
O <u>P</u> lace all certificates ir	the following store	
Certificate store:		
		Browse

5. คลิก Finish เพื่อสิ้นสุดการติดตั้ง Root Certificate จะมีการแจ้ง import success ตามภาพ

#### ← 😺 Certificate Import Wizard

Completing the Certificate Import Wizard	
The certificate will be imported after you click Finish.	
You have specified the following settings: Certificate Store Selected Automatically determined by the wizard Content Certificate	
	Certificate Import Wizard $ imes$
	The import was successful.
<u>Einish</u> Can	OK



# ขั้นตอนที่ 2A.3: การ configure Timestamp ในโปรแกรม Adobe Acrobat Reader DC

- 1. เปิดโปรแกรม Adobe Acrobat Reader DC
- ไปที่เมนู Edit -> Preference -> Signatures คลิก More... ที่ Configure timestamp server settings

Image: Control of the control of t	Digital Signatures       Creation & Appearance       • Control options for signature creation       • • Effect appearance of signatures within a document       Verification       • • Control how and when signatures are verified       Identifies & Trusted Certificates       • Center and manage identifies (sections)       • Manage credentials used to trust documents       Document Timestamping       • Configure timestamp server settings

3. คลิก New เพื่อตั้งค่า timestamp server โดยใส่ข้อมูลดังนี้

Name: ตัวอย่างในที่นี้ กำหนดเป็น test tsa etda

Server URL: ใส่ URL ระบบ timestamp ของ ETDA ตามที่ได้แจ้งไว้ <u>หรือ</u> URL ของ Timestamp Proxy Server ที่หน่วยงานได้ทำการติดตั้งไว้

แล้วคลิก OK





 เลือกข้อมูล timestamp server จากขั้นตอนก่อนหน้า แล้วคลิก Set Default จะมีรูปดาวขึ้นหน้าชื่อ จากนั้นจึงปิดหน้าตั้งค่า

Directory Servers	📲 New 📝 Edit 🥱 Import 🌈 Export 💿 Remove 📑 Set Default
Time Stamp Servers	Name URL test_tsa_etda https://time-uat.teda.th/signserv
	Configure Time Stamp Servers
	Configure a default Time Stamp Server if you want to embed a trusted time stamp when signing. If you are in a
	computer administrator may provide you with Time Stamp Server configuration information.
	computer administrator may provide you with Time Stamp Server configuration information. Select one of the entries above and click <i>Edit</i> to view or change settings, or click <i>New</i> to add and configure a Tin Select a default server using <i>Set Default</i> . If you have selected a default server then a time stamp will be embedde signature that you create.
	<ul> <li>Configure a default <i>Time Stamp Server</i> in you wint to cinece a distict finite stamp which signing. If you are in a computer administrator may provide you with Time Stamp Server configuration information.</li> <li>Select one of the entries above and click <i>Edit</i> to view or change settings, or click <i>New</i> to add and configure a Time Select a default server using <i>Set Default</i>. If you have selected a default server then a time stamp will be embedde signature that you create.</li> <li>Click <i>Export</i> to share your Time Stamp Server settings with others.</li> </ul>
	<ul> <li>Configure a default <i>Thite Stamp Server</i> in you want to check a distict finite stamp which signing. If you are in a computer administrator may provide you with Time Stamp Server configuration information.</li> <li>Select one of the entries above and click <i>Edit</i> to view or change settings, or click <i>New</i> to add and configure a Tii Select a default server using <i>Set Default</i>. If you have selected a default server then a time stamp will be embedde signature that you create.</li> <li>Click <i>Export</i> to share your Time Stamp Server settings with others.</li> </ul>

 เปิดไฟล์ PDF ที่ต้องการทำ timestamp ไปที่เมนู Tools เลือก icon Certificates จะพบว่ามีแถบ เครื่องมือขึ้นมา ให้คลิกเลือก Time Stamp



- หากมีหน้าต่างให้เลือก Certificate ให้เลือก Certificate ที่ทำการติดตั้งไปในขั้นตอนที่ 2A.1 แล้วทำการ บันทึกไฟล์นี้ ด้วยชื่อที่ต่างออกไป เพื่อให้ทราบว่าเป็นไฟล์ที่มีการทำ timestamp เพิ่ม
- เมื่อเรียก timestamp ได้สำเร็จ ในแถบ Signatures (แถบซ้ายมือ ที่เป็นรูปปากกา) ของเอกสาร จะมี รายละเอียด Timestamp ที่เพิ่งทำไปปรากฏขึ้น





ซึ่งหาก configure timestamp server ในขั้นตอนนี้แล้ว แม้แต่ในการลงลายมือชื่อดิจิทัล ก็จะเรียกใช้เวลาของ timestamp server เป็น signing time (ไม่ใช้เวลาของเครื่องคอมพิวเตอร์ที่เป็น local time) ซึ่งเพิ่มความ น่าเชื่อถือให้เอกสารอิเล็กทรอนิกส์ได้ดียิ่งขึ้น



# บทที่ 3 การยืนยันตัวตนผู้ขอใช้บริการด้วยรูปแบบ API key

การเรียกใช้บริการ e-Timestamping ในรูปแบบ HTTP API ใช้วิธีการยืนยันตัวตนด้วยรูปแบบ API key สำหรับเข้าใช้บริการ โดยในส่วนนี้จะเป็นการอธิบายขั้นตอนในการใช้งาน API key ที่ได้รับหลังจากยื่นใบคำขอ

 ระบุค่า API key ลงใน header ของ HTTP request โดยระบุ key = "apikey" และระบุ value = [ค่า API key ที่ได้รับมา] โดยในตัวอย่างแรก จะแสดงการระบุค่า API key ใน tool สำหรับทดสอบเรียก HTTP ที่ชื่อว่า postman และตัวอย่างที่ 2 จะแสดงการระบุค่า API key ใน HttpPost และ HttpClient ซึ่งเป็น library สำหรับเรียก HTTP request ประเภทหนึ่งของภาษา Java

► TSA API (api-uat.teda.th)							xamples	(0) 🔻	
	post ∨	https://api-uat.t	eda.th/timesta	amp/v1/basic	Params	Send	~	Save	~
Auth	orization 🔵	Headers (2)	Body 🔵	Pre-request Script Tests			(	Cookies	Code
	Key			Value	Description		Bulk Edit	Prese	ets 🔻
~	Content-Type			application/json					
~	apikey			5717ca03-76f7-4365-8221-2436d718a9a6					
	New key			Value					

- 1. HttpPost request = **new** HttpPost(tsaUrl);
- request.addHeader("content-type", "application/json");
- 3. request.addHeader("apikey", "5717ca03-76f7-4365-8221-2436d718a9a6");
- 4. HttpResponse result = httpclient.execute(request);
- 2. เมื่อทำการระบุค่า API key และ parameter ต่างๆ ใน body <u>(รายละเอียดเพิ่ม HTTP Request :</u>

<u>Timestamp request)</u> ของ HTTP request เสร็จเรียบร้อยแล้ว จะสามารถเรียกใช้งาน e-

Timestamping ในรูปแบบ HTTP API ได้

POST	v https://api-uat.teda.th/timestamp/v1/basic	Send	~
Params	Auth Headers (10) Body • Pre-req. Tests Settings	Co	okies
raw ~	V JSON V	Bea	utify
1	8		
2	"messageDigest":"z0PQt16C5ppq+FDEb/1m2AgiqoGGok410Q=sN90yu0ki",		
3	→ "hashAlgo":"1"		
4	8		- 1



# บทที่ 4 การติดตั้ง Timestamp Proxy Server

เพื่อเป็น proxy server สำหรับจัดการ certificate authentication ของหน่วยงานผู้ใช้บริการ และเป็น ตัวกลางในการเรียก timestamp มาที่ ETDA

- ความต้องการขั้นต่ำของ server CPU = 2 cores, RAM = 4 GB, HDD = 50 GB
- 2. ติดตั้งด้วย NGINX (version 1.9.0 ขึ้นไป) และระบุ certificate authentication ไว้ที่ NGINX
- ตั้งค่า NGINX ให้ forward transaction ที่มีการเรียก timestamp มาด้วย port 443 ให้ส่งไปที่ระบบ e-timestamping ของ ETDA

# ตัวอย่าง config nginx

location /upstream {	
proxy_pass	https://backend.example.com; //url ของ tsa etda
proxy_ssl_certificate	<b>/etc/nginx/client.pem;</b> //certificate ที่ทาง etda ส่งให้ (PEM formate)
proxy_ssl_certificate_key	/etc/nginx/client.key; //private key (PEM formate)
proxy_ssl_protocols	TLSv1 TLSv1.1 TLSv1.2;
proxy_ssl_ciphers	HIGH:!aNULL:!MD5;
proxy_ssl_trusted_certificate	e <b>/etc/nginx/trusted_ca_cert.crt;</b> //trust root ที่ทาง etda ส่งให้ (PEM formate)
proxy_ssl_verify on;	
proxy_ssl_verify_depth 2;	
proxy_ssl_session_reuse on	;
}	

 ประกาศ domain name ของ Timestamp Proxy Server ภายใน internal network ของหน่วยงาน โดยไม่ควรตั้งให้สามารถเข้าถึง Timestamp Proxy Server จากอินเตอร์เน็ตได้

#### High level network diagram





# บทที่ 5 การสร้าง Runnable JAR file เพื่อประทับรองรองเวลาไฟล์ PDF

เพื่อเพิ่มความสะดวกในการพัฒนาโปรแกรมร่วมกับภาษาอื่น สามารถเรียกใช้บริการ e-Timestamping ผ่าน Runnable JAR file โดยการนำ JAVA library (project PDFTimestamping) มาทำเป็น Runnable JAR file สำหรับการประกอบ timestamp token ที่ได้จากระบบ กับ PDF ไฟล์ที่เลือก

# ความต้องการพื้นฐานบนระบบปฏิบัติการ Windows

- Java 1.8 ขึ้นไป (<u>https://www.java.com/en/download/</u>)
- โปรแกรม Eclipse Oxygen for java EE version 4.7 ขึ้นไป (<u>https://www.eclipse.org/downloads/packages/</u>)
- ไฟล์ Keystore และ Password Keystore (โดยสามารถทำตามขั้นตอน "การสร้าง private key และ CSR file" ในบทก่อนหน้า)

# ขั้นตอนที่ 5.1: การ import project PDFTimestamping

1. เข้าไป Download file project PDFTimestamping จาก GitHub

### (<u>https://github.com/ETDA/PDFTimestamping</u>) และ unzip file

GitHub - ETDA/PDFTimestampir	ng × +				- o ×
$\leftarrow$ $\rightarrow$ $C$ $\cong$ GitHub, Inc	[US]   https://github.com/ETDA/PE	DFTimestamping			९ ☆ <b>Ө</b> :
	ETDA / PDFTimestamping		•	Watch 0 🗙 Star 1 😵 Fork 2	
	↔ Code ① Issues ⓐ 〗	Pull requests 0 🔲 Projects 0 👍 Insight	15		
		Join GitHub to GitHub is home to over 28 million develop and review code, manage projects, and sign up	oday ers working together to host build software together.	Domiss C	
	PDF Timestamping using PDFbo	x			
	14 commits	₽ 1 branch	No releases	LL 1 contributor	
	Branch: master - New pull request			Find file Clone or download •	
	🚺 pragasitcha Formatting		Clone with	h HTTPS 💿	
	resources	0.1.0	Use Git or ch	neckout with SVN using the web URL.	
	🖿 src/main/java	change cert authen detect condition	https://gi	ithub.com/ETDA/PDFTimestamping.	
	.gitignore	support calling tsa by certificate Authen			
	README.md	Formatting	Open in	Desktop Download ZIP	
	pom.xml	add mavon jvm config to 1.8		8 months ago	
	I README.md				
.tps://github.com/ETDA/PDFTimestam	iping/archive/master.zip				



2. เปิดโปรแกรม Eclipse เลือกเมนู File เลือก import เลือกเป็น Maven > Existing Maven Projects



 กดปุ่ม browse แล้วเลือกโฟลเดอร์ของ project PDFTimestamping ที่ทำการ unzip file แล้ว และทำ การกด Finish

Import Mave Maven Project Select Maven p	en Projects <b>s</b> rojects					×
<u>R</u> oot Directory:	C:\Users\napon\eclipse-woo	rkspace\PDFTime	estamping-maste	r v	Brows	e
/pom.x	ml etda.or.th:PDFTimestamp	ing:0.0.1-SNAPSI	HOT:jar		Select Deselect Deselect Refre	<u>A</u> II tt AII Tree t Tree sh
Add project(	s) to working set					
<ul> <li>Advanced</li> </ul>	anthurð					~
?		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inis	Cance	2



4. รอให้ Maven import dependency download เสร็จ



5. จะมี project PDFTimestamping ขึ้นอยู่ด้านข้างของโปรแกรม

eclipse-workspace - Eclipse							-	-		×
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> a	vigate	Se <u>a</u> rch <u>P</u> roject	<u>R</u> un	<u>W</u> indow	He	lp				
📩 🗕 📳 🕲 🖉 🖌 🖗 🗸	0	<b>- Q Q</b>	¥ 0	- 🤌 🖾	• 👌	7 -	刻 - 例	• *:	• († •	⇒ -
							Quick /	Access		봥
😫 Package Explorer 🔀 📃								Т	× -	
E 😒 😜	$\bigtriangledown$							Ć	ॏ ◄	
> 🄛 PDFTimestamping-master								Ŷ.	8: 1	9
4								$\mathbf{x}$	( <b>1</b> ) E	
								-	1	<u> </u>
								~	7	
								F	ind 🔍	► 4
		🗄 Outline 🔀	69	~ - 8		🖹 Р	📃 C 🐹	»2	-	
		An outline is not a	vailable.					e.	<u> </u>	2 -
	>				Ī	No cor	nsoles to di	splay a	at this tin	ne.
PDFTimestamping-master										
		<b></b>								

หมายเหตุ: หากไม่สามารถ import project ได้ สามารถเลื่อนลงไปดูในหัวข้อ "การแก้ปัญหา กรณี import project ไม่สมบูรณ์" ได้



# ขั้นตอนที่ 5.2: การสร้าง JAR file

1. คลิกขวาที่ project PDFTimestamping เลือก Run As > Run Configurations เลือกแถบ Java

#### Application

Run Configurations		×
Create, manage, and run con Main type not specified	figurations	
Image: Content of the second seco	Name: New_configuration  Main Arguments M JRE A Classpath S Source Environment □ C Project  PDFTimestamping-master  Main class:  Include system libraries when searching for a main class Include inherited mains when searching for a main class Stop in main	Search
Filter matched 8 of 16 items	Re <u>v</u> ert	Apply
?	Bun	Close

2. ดูตรง Main Class กดปุ่ม search และเลือกเป็น main – (default package)

Select Main Type			>
Select <u>t</u> ype (? = any character, * = any String, TZ = TimeZone):			
**			
Matching items:			
🕞 main - (default package)			
🗘 Workspace matches		 	
G AddAnnotations - org.apache.pdfbox.examples.pdmodel			
PDFTimestamping-master\target\classes - (default package	e)		
(?)	N OK	Cance	1

3. จากนั้นกด Apply และ Run

ตัว Eclipse จะ return error เพราะไม่พบ การตั้งค่า Argument ไม่ต้องสนใจเพราะเราต้องการตัว execute jar file



4. คลิกขวาที่ project PDFTimestamping เลือก Export เลือก Java และเลือกไปที่ Runnable JAR file

Export -		×
Select Export all resources required to run an application into a JAR file on the local file system		5
Select an export wizard:		
I > È General > È Install ♥ Dava □ JAR file ♥ Javadoc JAR file > E Run/Debug > È Tasks > È Tasks > È Tasks > È Task		
? < Back Next > Finish	Cance	el

5. ทำการเลือก Launch configuration ไปเป็น main – PDFTimestamping และกำหนดตำแหน่งที่ ต้องการวาง JAR file

🖨 Runnable JAR File	e Export							×
Runnable JAR File	Specifica	ation					9	2
<ol> <li>The export destination</li> </ol>	ation will I	be relative to yo	our workspace.					
Launch configuratior	n:							
New_configuration -	- PDFTime	stamping-mas	ter					<u> </u>
Export <u>d</u> estination:								
PDFTimestamping-	master/tin	nestamp.jar				~	B <u>r</u> owse	
Extract required lib     Package required <u>C</u> opy required libr <u>Save as ANT script</u>	braries into libraries in raries into t	o generated JAF ito generated J a sub-folder ne	R AR ext to the gener	ated JAR	1			
ANT script location:	C:\Users\	\napon\eclipse	-workspace			~	Br <u>o</u> wse	
?		< <u>B</u> ack	<u>N</u> ext >		<u>F</u> inish	<b>₽</b>	Cance	1



6. เมื่อทำการสร้างเสร็จ จะเห็นไฟล์ปรากฏขึ้นในตำแหน่งที่เลือกไว้

,≪ Windows (C:) → Users → napon →	eclipse-workspace > PDFT	imestamping-master
Name	Туре	Size
.settings	File folder	
, resources	File folder	
src	File folder	
h target	File folder	
📄 .classpath	CLASSPATH File	1 KB
📄 .gitignore	Text Document	1 KB
📄 .project	PROJECT File	1 KB
📄 pom.xml	XML Document	1 KB
README.md	MD File	4 KB
🕌 timestamp.jar	Executable Jar File	8,332 KB

# ขั้นตอนที่ 5.3: การนำ Runnable JAR file มาใช้

คือ การนำ JAR file ที่ได้มา run โดยสามารถใช้ภาษาต่างๆเช่น C# .net python มาเรียกใช้

ตัวอย่างเช่น

		Code C#
1 2 3	⊡us _us	ing System; ing System.Diagnostics;
4	⊡na	mespace ConsoleApp2
5	• (	
6 7 8		o references  Class Program  {  Oreferences  Static void Main(string[] args)
9		{
10		String CMD = "/c java -jar \"Timestamp.jar\" \"pdfA3.pdf\" \" <u>https://time-uat.teda.th/signserver/process?workerName=teda\</u> " \"Napon.p12\" \"Napon\" \"PKCS12\" \"2\" ";
11		Process.Start("CMD.exe", CMD);
12		}
13		}
14	1	

โดย process.Start("cmd.exe",CMD) จะเป็นการเรียก cmd.exe และใส่คำสั่งตาม string CMD ซึ่งใน string CMD จะเป็นการเรียก JAR file และใส่ค่า parameter ลงไป คือ

JAR file : ชื่อ JAR file ที่ทำการสร้างออกมา (ตัวอย่างในที่นี้ กำหนดชื่อไฟล์เป็น timestamp.jar) ที่ชื้ ไปยัง JAR file

inputFile : ชื่อไฟล์ PDF ที่ต้องการทำ Timestamp (ตัวอย่างในที่นี้ กำหนดชื่อไฟล์เป็น pdfA3.pdf) TSAURL : URL ของบริการ TSA

โดยทาง ETDA จะส่ง URL ให้พร้อมกับไฟล์ certificate ทางอีเมล หลังมีการขอใช้บริการหรือ ขอทดสอบบริการ

keystoreFile : ชื่อไฟล์ keystore (ตัวอย่างในที่นี้ กำหนดชื่อไฟล์เป็น Napon.p12) keystorePassword : รหัสผ่านของ keystore (ตัวอย่างในที่นี้ กำหนดรหัสผ่านเป็น Napon) keystoreType : ประเภทของไฟล์ keystore (ตัวอย่างในที่นี้ กำหนดประเภทของไฟล์เป็น PKCS12)



logType : ประเภทของการแสดง log มี 2 ประเภท โดยหากกำหนดเป็น 1 จะเป็นการแสดง log ผ่าน console และหากกำหนดเป็น 2 จะเป็นการแสดง log ผ่านทางไฟล์ PDFTimestamp\_log.txt (ตัวอย่าง ในที่นี้ กำหนดประเภทของการแสดง log เป็น **2**)

โดยเมื่อทำการ run เสร็จเรียบร้อย จะได้ output file ที่มีชื่อคล้ายกับ inputFile โดยมีรูปแบบชื่อไฟล์ เป็น **ชื่อ inputFile เดิม\_timestamped** (ตัวอย่างในที่นี้ จะได้ output file ชื่อ pdfA3\_timestamped.pdf) และ log file ชื่อ PDFTimestamp\_log.txt ขึ้น

Co	nsoleApp2 > ConsoleApp2 > bin > Debug	~	- D	Search Debug	
^	Name	Date modified	Туре		Size
	📙 timestamp	19-Oct-18 2:12 PM	File fo	older	
	ConsoleApp2.exe	27-Dec-18 10:57 AM	Appli	cation	5 KB
	ConsoleApp2.exe.config	19-Oct-18 2:08 PM	XML	Configuratio	1 KB
	ConsoleApp2.pdb	27-Dec-18 10:57 AM	PDB F	ile	12 KB
	ConsoleApp2.vshost.exe	27-Dec-18 10:57 AM	Appli	cation	23 KB
	ConsoleApp2.vshost.exe.config	19-Oct-18 2:08 PM	XML	Configuratio	1 KB
	ConsoleApp2.vshost.exe.manifest	29-Sep-17 8:43 PM	MAN	IFEST File	1 KB
	Napon.p12	17-Oct-18 4:03 PM	Perso	nal Informati	5 KB
	pdfA3.pdf	26-Nov-18 4:22 PM	Adob	e Acrobat D	51 KB
	pdfA3_timestamped.pdf	27-Dec-18 10:58 AM	Adob	e Acrobat D	70 KB
	PDFTimestamp_log.txt	27-Dec-18 10:58 AM	Text D	Ocument	1 KB
	🍰 timestamp.jar	27-Dec-18 10:35 AM	Execu	table Jar File	8,334 KB



# <u>การแก้ปัญหา กรณี import project ไม่สมบูรณ์</u>

1. คลิกขวาที่ project เลือก Maven > Update project



2. คลิกเลือก Force Update of Snapshots/Releases แล้วกด OK

Update Maven Project		×
Update Maven Project		
Select Maven projects and update options		
Available Maven Codebases		
PDFTimestamping-master		Select All
		Add out-of-date
		Deselect All
		Expand All
		<u>C</u> ollapse All
Qffline		
Update dependencies		
Force Update of Snapshots/Releases		
Update project configuration from pom.xml		
Refresh workspace resources from local filesystem		
Clean projects		
?	ок	Cancel



3. รอให้ Maven update จนเสร็จ



- 4. หากพบว่า**ปัญหาถูกแก้ไขแล้ว ไม่ต้องทำข้อหลังจากนี้** แต่หาก**ยังพบปัญหาขอให้ทำข้อถัดไป**
- 5. ทำการปิดโปรแกรม Eclipse
- 6. หาโฟลเดอร์ .m2 ใน drive ที่ทำการลงโปรแกรม Eclipse แล้วทำการลบโฟลเดอร์ .m2 ออก

> This PC > Windows (C:) > Users > napon				
Name	Date modified	Туре	Size	
NTUSER.DAT	10/16/2018 6:09 PM	DAT File	5,376 KB	
gitconfig	9/20/2018 1:14 PM	GITCONFIG File	1 KB	
📓 mercurial.ini	9/20/2018 1:14 PM	Configuration sett	1 KB	
m2	10/18/2018 5:54 PM	File folder		
.p2	10/18/2018 5:54 PM	File folder		
eclipse-workspace	10/18/2018 10:18	File folder		
- r	10/0/0010 100 014	ent entre		

- 7. เปิดโปรแกรม Eclipse อีกครั้ง
- 8. ทำ ข้อที่ 1-3 ใหม่อีกครั้ง



# การตรวจสอบ log file ที่เกิดจากการ run JAR File

หลังจากทำเป็น Runnable JAR File แล้ว เมื่อมีการสั่งรัน JAR File จะทำการอ่านค่าจาก พารามิเตอร์ logType ซึ่งเป็นตัวกำหนด โดยหากกำหนดเป็น 1 จะเป็นการแสดง log ผ่าน console และหากกำหนดเป็น 2 จะเป็นการแสดง log ผ่านทางไฟล์ PDFTimestamp\_log.txt

โดยเมื่อกำหนดเป็น 2 แล้วจะทำการตรวจสอบว่ามีไฟล์ PDFTimestamp\_log.txt หรือไม่ หากยังไม่มี ไฟล์ จะทำการสร้างไฟล์ใหม่ขึ้น แต่หากมีไฟล์อยู่แล้ว จะทำการเขียนไฟล์ต่อจากไฟล์เดิม โดยมีรูปแบบเป็น ปีเดือน วัน เวลา inputFile urlTsa keystoreFile และต่อด้วย ข้อความ log โดยไฟล์ PDFTimestamp\_log.txt จะอยู่ที่ โฟลเดอร์เดียวกับ Runnable JAR File

ในการเขียน log ลง PDFTimestamp\_log.txt แต่ละครั้ง จะเขียน 3 อย่างเพิ่มลงไปคือ

- 1. วันเวลาในการสั่งรัน
- 2. ข้อมูล inputFile, urlTsa, keystoreFile ที่ผู้ใช้ทำการกรอกเข้ามา
- ข้อความ <sub>Log</sub> คำอธิบาย \*\*\*\*\*\*\*\*TimeStamp Done\*\*\*\*\*\*\* สามารถทำ timestamp ได้สำเร็จ หาไฟล์ที่ระบุไว้ไม่เจอ java.io.FileNotFoundException java.io.IOException มีปัญหากับ input/output ใส่พารามิเตอร์ไม่ครบ java.lang.StringIndexOutOfBoundsException ไม่รู้จักกับ url⊤sa ,เชื่อมต่อกับ tsa ไม่ได้ java.net.UnknownHostException keystore มีปัญหา รหัส keystore ผิด/ไม่ถูกต้อง java.security.KeyStoreException java.net.MalformedURLException ไม่มี urlTsa
- ข้อความ log ตัวอย่างเช่น

หมายเหตุ: ในกรณีที่ไม่มีสิทธิ์ในการเขียน PDFTimestamp\_log.txt จะไม่มีการสร้างไฟล์ log หรือเขียนต่อจาก ไฟล์เดิม แต่จะแสดง log ผ่าน console แทน



# **ตัวอย่าง** PDFTimestamp\_log.txt กรณีที่ทำ Timestamp สำเร็จ

#### Input

"resources/pdfA3.pdf"

"https://time-uat.teda.th/signserver/process?workerName=teda"

"resources/Napon.p12"

"Napon"

"PKCS12"

"2"

#### Output: PDFTimestamp\_log.txt

2018/12/07 10:20:30

inputFile: resources/pdfA3.pdf, tsaUrl: https://time-uat.teda.th/signserver/process?workerName=teda, keystoreFile:

resources/Napon.p12

\*\*\*\*\*\*\*TimeStamp Done\*\*\*\*\*\*\*\*

# **ตัวอย่าง** PDFTimestamp\_log.txt กรณีที่ทำ Timestamp แล้วเกิดข้อผิดพลาดขึ้น

Input "
"
"
"
thtps://time-uat.teda.th/signserver/process?workerName=teda"
"
resources/Napon.p12"
"
Napon"
"
PKCS12"
"2"

### Output: PDFTimestamp\_log.txt

2018/12/07 10:20:35

inputFile: , tsaUrl: https://time-uat.teda.th/signserver/process?workerName=teda, keystoreFile: resources/Napon.p12

java.lang.StringIndexOutOfBoundsException: String index out of range: -1

at java.lang.String.substring(Unknown Source)

at main.main(main.java:48)



# บทที่ 6 Timestamp API Specification

Timestamp API เป็นบริการที่สร้างมาเพื่ออำนวยความสะดวกในการเรียก TSA server สำหรับสร้าง Timestamp Token ตามมาตรฐาน RFC3161 โดยฝั่ง API Requester จะส่ง HTTPS request เข้ามาในระบบ พร้อมกับ parameter ที่จำเป็น ได้แก่ค่า hash ของสิ่งที่ต้องการทำ timestamp และ hash algorithm จากนั้น ระบบจะทำการตอบกลับเป็น Timestamp Token ในรูปแบบ Base64String พร้อมกับค่าสถานะของการทำ timestamp สามารถอธิบายลำดับการทำงานได้ตามรูปด้านล่าง

\*\* HTTPS request จำเป็นต้องแนบ API key สำหรับการ authentication (โดยสามารถทำตามขั้นตอน "การ ยืนยันตัวตนผู้ขอใช้บริการด้วยรูปแบบ API key" ในบทก่อนหน้า





### HTTP Request : Timestamp request

Request HTTP Type:

НТТР Туре	Description
POST	POST method สำหรับการส่ง hash ของสิ่งที่ต้องการทำ timestamp

Request URL:

URL
https://[URL]/timestamp/[version]/[endpoint]

### Request HTTP Header:

HTTP Header	Value
Content-Type	application/json
apikey	[ค่า API key ที่ได้รับมาจาก ETDA]

# Request HTTP Body:

Parameter	Туре	Description	M/O
messageDigest	Text	hash ของสิ่งที่ต้องการทำ timestamp โดยจะต้องสร้างด้วย	Mandatory
	(Base64)	hash algorithm เดียวกันกับ parameter hashAlgo	
hashAlgo	Text	ค่า hash algorithm โดยสามารถใส่ค่าได้ดังนี้	Mandatory
		0 : "SHA-1"	
		1 : "SHA-256"	
		2 : "SHA-512"	



# HTTP Status Codes:

Code	Description
200 OK	กรณีที่ใส่ค่ามาครบถูกต้อง
400 Bad Request	กรณีที่ใส่ค่า parameter ผิด หรือไม่ครบ เช่นการ request body มาไม่ตรง
	format ที่กำหนด
401 Unauthorized	กรณีไม่ได้ระบุ certificate authentication
403 Forbidden	กรณีที่ไม่ได้ลงทะเบียนการใช้งานระบบ API
404 Not Found	กรณีที่ใส่ URL ผิด
405 Method Not Allowed	กรณีที่ใส่ HTTP Method ผิด
500 Internal Server Error	กรณีข้อผิดพลาดอื่นๆ

Example Request:

URL

https://api-test.teda.th/timestamp/v1/basic

BODY	
{	
	"messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=",
	"hashAlgo":"1"
}	



### HTTP Response : Timestamp response

Response HTTP Header:

HTTP Header	Value
Content-Type	application/json;charset=UTF-8

Response HTTP Body:

Parameter	Туре	Description	M/O
resultCode	Text	ค่าสถานะการทำ Timestamp	Mandatory
		สามารถดูได้จากตาราง Result code and Result	
		Message	
token	Text	Timestamp token ตามมาตรฐาน RFC3161	Optional
	(Base64)		
		ถ้าหาก resultCode แสดงผลว่าพบ error	
		token จะมีค่าเป็น null	
resultMessage	Text	Return message ของค่า resultCode	Mandatory
		ถ้าหาก result code แสดงผล เป็นค่า P1004 – P1007	
		API จะ return ค่า ID มาในฟิลด์นี้ เพื่ออ้างอิงการตรวจสอบ	
		transaction นั้นในระบบ	



Result code and Result message

resultCode	resultMessage	Description
P1000	Success	สามารถสร้าง Token ได้สำเร็จ
P1001	Request hash is null	parameter messageDigest มีค่าเท่ากับ
		null หรือ parameter name ไม่ถูกต้อง
P1002	Request hash algorithm is null	parameter hashAlgo มีค่าเท่ากับ null
		หรือ parameter name ไม่ถูกต้อง
P1003	Input hash algorithm support only	ใส่ค่า hashAlgo อื่นๆ นอกเหนือจากค่า
	0,1,2 (for SHA-1,SHA-256,SHA-512)	0,1,2
P1004	Input hash is not in Base64 format	Input messageDigest ไม่อยู่ในรูปแบบ
		ของ Base64String
P1005	An error occurred in TSA API.Please	พบ error ในส่วนของ TSA API เช่น ตั้ง
	contact admin with transaction ID	ค่า URL ของ TSA server ไว้ผิดรูปแบบ
P1006	Response timestamp token is null	TSA server มี response message
		ตอบข้อความกลับมาที่ TSA API แต่
		timestamp token ภายใน response
		message จะมีค่าเป็น null
		เมื่อ TSA API return resultCode เป็น
		P1006 TSA API จะแนบ message ที่
		ถูกส่งมาจาก TSA server เพิ่มเติมด้วย
P1007	TSA server connection timeout	ไม่สามารถติดต่อ TSA server ได้ ภายใน
		เวลาที่กำหนดไว้ (ปัจจุบันตั้งค่าไว้ที่ 10
		วินาที)
P9999	Unknown error	พบข้อผิดพลาดอื่นๆ
		เมื่อ TSA API return resultCode เป็น
		P9999 TSA API จะแนบ message ที่
		พบจาก exception เพิ่มเติมด้วย



# Example Request and Response:

Example 1 : Valid request

### Request Body: P1000

"messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=",

"hashAlgo":"1"

}

{

### Response Body: P1000

{

"resultCode": "P1000",

"token": "MIAGCSqGSIb3DQEHAqCAMIIQ9QIBA......bKOQLFtY6N77xmRt+lUZPlSQAAAAA=",

"resultMessage": "Success"

}

Example 2: Request hash is null



### Response Body: P1001

{ "resultCode": "P1001", "token": null,

"resultMessage": "Request hash is null"

}



#### Example 3: Request hash algorithm is null

#### Request Body: P1002

"messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=",

"hashAlgo":null

}

{

### Response Body: P1002

{

"resultCode": "P1002",

"token": null,

"resultMessage": "Request hash algorithm is null"

}

Example 4 : Invalid hash algorithm

Request Body: P1003 { "messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=", "hashAlgo":"99" }

### Response Body: P1003

{

```
"resultCode": "P1003",
```

"token": null,

"resultMessage": "ID:3d074e0900cefd77b21546f7898e405e1541576287382,Message:Input

hash algorithm support only 0,1,2 (for SHA-1,SHA-256,SHA-512)"

}



### Example 5 : Input hash is not Base64String format

### Request Body: P1004

```
"messageDigest":"aaa",
```

"hashAlgo":"1"

}

{

{

# Response Body: P1004

"resultCode": "P1004",

"token": null,

"resultMessage": "ID:714d1e81f5df788a000af4a11a7f4ece1541495691180,Message:Input hash

is not in Base64 format"

}

Example 6 : Internal TSA API error

เช่น config ให้ TSA API เรียก TSA Server ด้วย URL ที่ผิด format

```
Request Body: P1005
{
    "messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=",
    "hashAlgo":"1"
}
```

# Response Body: P1005

```
{
    "resultCode": "P1005",
    "token": null,
    "resultMessage": "ID:64d8170196c1ac29a7d7309a04b561f61541496549424,Message:An error
    occurred in TSA API.Please contact admin with transaction ID"
}
```



#### Example 7: Response from TSA server is null

#### Request Body: P1006

"messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=",

"hashAlgo":"0"

}

{

### Response Body: P1006

{

"resultCode": "P1006",

"token": null,

"resultMessage":

"ID:41c82709e34ca448d2e546bb274ebd2d1541496303142,Message:Response timestamp token

is null,Cause:null,Exception:[Error-45008] Failed to validate request - hash length in TSA

request does not match with hash algorithm"

}

### Example 8: TSA server connection timeout

เช่น config ให้ TSA API เรียก TSA Server ด้วย URL ที่ไม่ถูกต้อง ทำให้เกิด TSA Timeout ที่ 10 วินาที

```
Request Body: P1007
{
    "messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=",
    "hashAlgo":"1"
```

}

# Response Body: P1007

{

```
"resultCode": "P1007",
```

```
"token": null,
```



"resultMessage": "ID:ad1ba9f841aeefe7b04053c708f40e091541496943758,Message:TSA server connection timeout"

}

Example 9: Unknown error

Request Body: P9999			
{			
	"messageDigest":"+FDEb/1m2AgiqosN9Oyuz0PQt16C5ppqOkiGGok410Q=",		
	"hashAlgo":"1"		

}

Response Body: P9999		
{		
"resultCode": "P9999",		
"token": null,		
"resultMessage": "ID:93d0d8525249fc3cf24952b7be6e8d2f1541680732734,Message:Unknown		
error,Cause:null,Exception:null"		
}		

Example 10: No API key

Request Header: No API key				
▼ Head	lers (0)			
KEY	VALU	E DESCRIPTION	••• Bulk Edit Presets 🔻	
Key	y Valu	e Description		

Response Body: No API key	
{	
"message": "No API key found in request"	
}	



# Example 11: Invalid API key

Request Header: Invalid API key			
▼ Headers (1)			
	KEY	VALUE	DESCRIPTION •••• Bulk Edit Presets 🔻
$\mathbf{>}$	apikey	abcd1234	
	Key	Value	Description

Response Body: Invalid API key	
{	
"message": "Invalid authentication credentials"	
}	



# บทที่ 7 การตรวจสอบความถูกต้องของการประทับรับรองเวลา ด้วย TEDA Web Validation Portal

- 1. เปิดลิงก์ต่อไปนี้ในเบราว์เซอร์ของคุณ <u>https://validation.teda.th</u>
- 2. เลือกไฟล์เอกสารที่จะทำการตรวจสอบ และทำการติ๊กที่ช่องสี่เหลี่ยม ทำ process reCAPTCHA และกด ปุ่มตรวจสอบ

		₩ info@etda.or.th J 0-2123-1234
ตรวจสอบเอกสารอิเล็กทรอนิกส์ ค่	าถามที่พบบ่อย ติดต่อ เงื่อนไขการให้บริการ	
<b>ตรวจสอบเอกสาร</b> เลือกเอกสารที่ต้องการตรวจสอบ รองรับเอกสาร PDF, XML, เอกสารตาเ	มีครงการ e-Tax Invoice by Email และ e-Tax Invoice and e-Receipt	
	Choose File No file chosen	
<b>หมายเหตุ</b> : การตรวจสอบดังกล่าวเ ประทับรับรองเวลาและ/หรือลงลาย ละเอียดเงื่อนไขการให้บริการได้ที่ ค	เป็นการตรวจสอบการประทับรับรองเวลา ลายมือชื่ออิเล็กทรอนิกส์ และการเปลี่ยนแปลงใ ขมือชื่ออิเล็กทรอนิกส์ แต่ไม่รวมถึงการตรวจสอบความถูกต้องของเนื้อหาในเอกสารนั้น ๆ เลิก	ด ๆ ที่เกิดแก่เอกสารภายหลัง ทั้งนี้ สามารถตรวจสอบราย
		ตรวจสอบ

 เลื่อนลงมาที่หัวข้อ "E TIMESTAMP" เพื่อดูรายละเอียดในของการทำ timestamp ด้วยใบรับรอง อิเล็กทรอนิกส์ที่ระบบรองรับ บนไฟล์เอกสารที่ทำการตรวจสอบ



#### PDF E-Timestamp 🗸

ผลการตรวจสอบการประทับรับรองเวลา (Timestamp Validation Result)	✔ น่าเชื่อถือ
หน่วยงานผู้ประทับรับรองเวลา (Organization Name)	Electronic Transactions Development Agency (Public Organization)
ผู้ประทับรับรองเวลา (Timestamping Authority)	TeDA Time-Stamping Service G3
ผู้ออกใบรับรอง (Certification Authority)	DigiCert SHA2 Assured ID Timestamping CA
วันที่ประทับรับรองเวลา (Timestamping Date)	18 ส.ค. 2564 15:49:16 น. (เวลาประเทศไทย)
วันออกใบรับรอง (Certificate Create Date)	12 มิ.ย. 2563 07:00:00 น. (เวลาประเทศไทย)
วันหมดอายุใบรับรอง (Certificate Expiration Date)	11 มิ.ย. 2566 19:00:00 น. (เวลาประเทศไทย)
สถานะ (Status)	การประทับรับรองเวลามีความน่าเชื่อถือ
ผลการตรวจสอบ LTV (LTV Result)	-
ผลการตรวจสอบ LTA (LTA Result)	-



# ภาคผนวก

# ตัวอย่างหนังสือขอใช้บริการ

เรื่อง ขอความอนุเคราะห์ใช้งานระบบ e-Timestamping เรียน ผู้อำนวยการสำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ สิ่งที่แนบมาด้วย แบบฟอร์มการสมัครขอใช้บริการประทับรับรองเวลา

ด้วย....ชื่อหน่วยงาน...... มีนโยบายในการนำระบบ Electronic Document มาใช้ให้บริการ เพื่อเพิ่มความสะดวกวดเร็ว ในการรับรองเอกสาร สัญญา หรือ ธุรกรรมอื่นใด ที่จะต้องมีการลงประทับรับรองเวลา เพื่อเป็นหลักฐานในการรับรองเอกสารว่าน่าเชื่อถือได้

ในการนี้ เพื่อให้การบริการเป็นไปด้วยความเรียบร้อย จึงขอความอนุเคราะห์การใช้งานระบบ e-Timestamp บนเอกสาร เพื่อสร้างความมั่นใจให้กับผู้ใช้เอกสาร รวมถึงหน่วยงานที่เกี่ยวข้อง สามารถตรวจสอบ การเปลี่ยนแปลงใดๆ และความถูกต้องของเอกสารได้

จึงเรียนมาเพื่อโปรดพิจารณาให้ความอนุเคราะห์ด้วย จะเป็นพระคุณยิ่ง

End of document