# คู่มือประกอบการพัฒนาระบบ

## Administrative Manual

## การพัฒนาระบบสารสนเทศภาครัฐให้เชื่อมโยงกับระบบยืนยันตัวบุคคลกลางแบบรวมศูนย์ (Single Sign-On)

สำนักนายกรัฐมนตรี

รุ่นเอกสาร 1.0

11 เมษายน 2562

**สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน)**

# ประวัติการแก้ไขเอกสาร

| วันที่ | รุ่นเอกสาร | รายละเอียดการแก้ไข | อ้างอิง |
|---|---|---|---|
| 28 มีนาคม 2556 | 1.0 | - | - |
| 25 มีนาคม 2557 | 1.0 | เพิ่มหัวข้อ 11 การพัฒนาโมบายแอพพลิเคชั่นให้สามารถใช้งานร่วมกับระบบยืนยันตัวบุคคลกลาง | http://openid.egov.go.th/publish/DevMain.aspx |
| 11 เมษายน 2562 | 1.0 | เปลี่ยนชื่อหน่วยงานจากสำนักงานรัฐบาลอิเล็กทรอนิก (องค์การมหาชน) (สรอ.) เป็นสำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) (สพร.) | https://www.dga.or.th/th/profile_history/ |
| 17 เมษายน 2562 | 1.0 | ปรับแก้ url จาก openid.egov.go.th เป็น accounts.egov.go.th | |
| | | | |
| | | | |

# สารบัญ

หน้า

# สารบัญรูป

หน้า

## 1. หลักการและเหตุผล

ในปัจจุบัน ระบบสารสนเทศแต่ละระบบมักมีฐานข้อมูลบัญชีผู้ใช้และรหัสผ่านเป็นของตนเอง ซึ่งก่อให้เกิดปัญหาต่อผู้ใช้งานที่จำเป็นต้องเข้าใช้งานหลายระบบ เนื่องจากต้องมีและจดจำชื่อผู้ใช้ (Login) และรหัสผ่าน (Password) หลายชุด บางท่านจึงเลือกที่จะใช้ชื่อผู้ใช้และรหัสผ่านเดียวสำหรับทุกระบบ หรือใช้รหัสผ่านที่จำง่าย ซึ่งเสี่ยงต่อการถูกขโมยชื่อผู้ใช้และรหัสผ่านไปใช้เพื่อขโมยข้อมูล หรือนำไปใช้ในกิจกรรมผิดกฎหมายต่าง ๆ ได้

สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) (สพร.) ทราบถึงปัญหาดังกล่าว จึงได้จัดทำ *ระบบยืนยันตัวบุคคลกลาง (e-Authentication Service)* ขึ้นเพื่อให้ประชาชน และเจ้าหน้าที่ของหน่วยงานภาครัฐสามารถเข้าถึงระบบสารสนเทศต่าง ๆ ของรัฐ ทั้งที่เป็นระบบบริการอิเล็กทรอนิกส์ภาครัฐ (e-Service) และระบบงานภายในของภาครัฐ (Back Office) โดยมีการควบคุม รักษาความปลอดภัยด้วยมาตรการอันเหมาะสม นอกจากนี้ระบบยืนยันตัวบุคคลดังกล่าวยังรองรับการเข้าถึงระบบงานแบบรวมศูนย์ (Single Sign-On: SSO) กล่าวคือ ผู้ใช้งานสามารถลงชื่อเข้าใช้งานระบบ (Login) ครั้งเดียว แล้วสามารถเข้าใช้งานระบบหลายระบบได้โดยไม่ต้องลงชื่อเข้าใช้งานซ้ำอีก

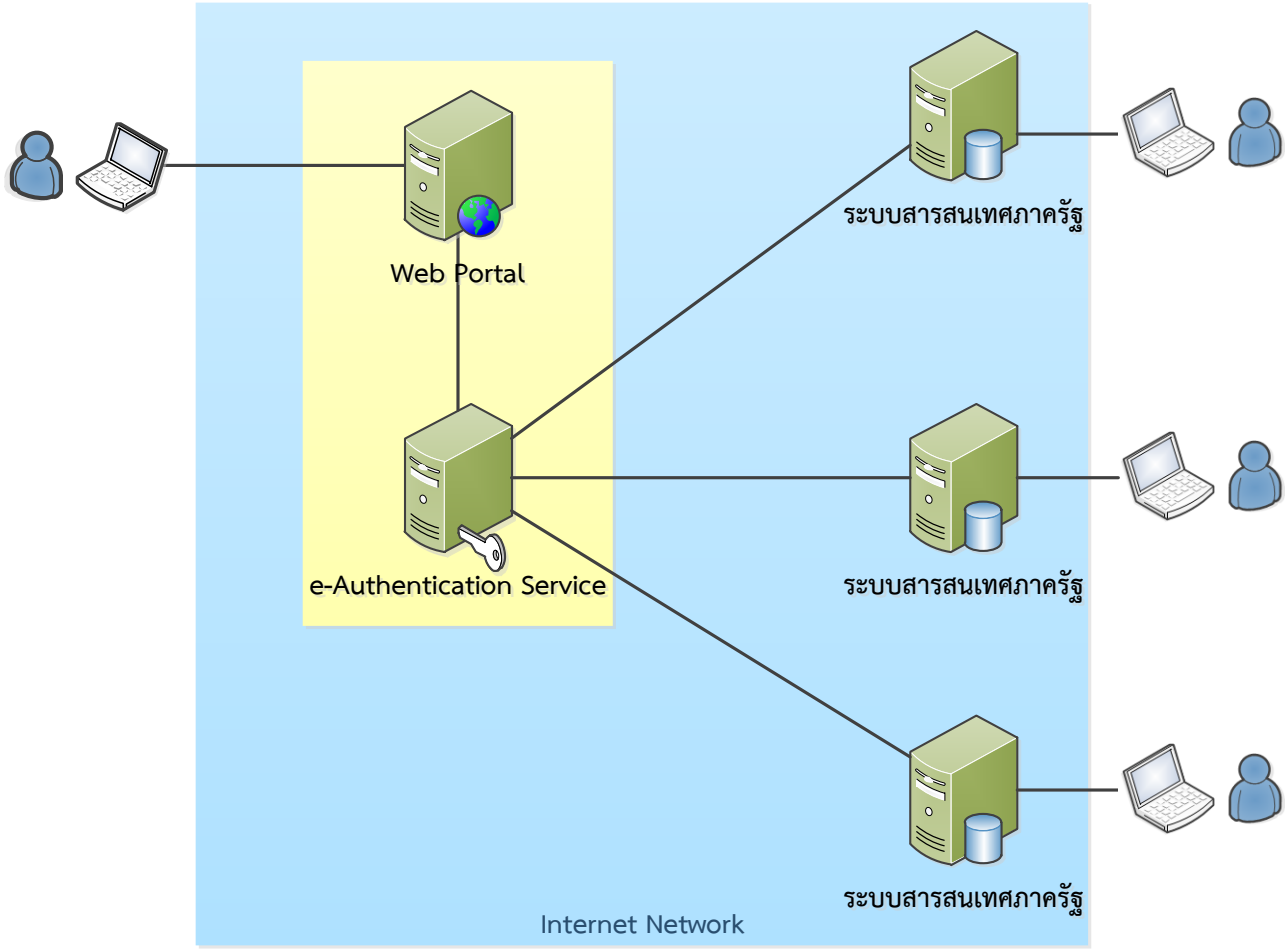
ระบบยืนยันตัวบุคคลกลางเป็นก้าวสำคัญไปสู่การให้และรับบริการแบบเบ็ดเสร็จ (One Stop Service) ในลักษณะ Single Window Entry อันจะเป็นการอำนวยความสะดวกต่อประชาชนในการเข้าถึงบริการของรัฐได้อย่างบูรณาการ นอกจากนี้ ในอนาคตถ้าหน่วยงานภาครัฐสามารถใช้ฐานข้อมูลผู้ใช้งานร่วมกันได้แล้ว ก็จะเป็นการเพิ่มความสะดวกต่อหน่วยงานอีกด้วย เนื่องจากไม่ต้องจัดเก็บ บริหารจัดการ และรักษาความปลอดภัยฐานข้อมูลดังกล่าวเอง

## 2. ประโยชน์ที่คาดว่าจะได้รับ

- ผู้ใช้งานทั้งที่เป็นประชาชนทั่วไป และเจ้าหน้าที่ของรัฐได้รับความสะดวกในการเข้าถึงระบบสารสนเทศต่าง ๆ ทั้งที่เป็นระบบบริการอิเล็กทรอนิกส์ภาครัฐ (e-Service) และระบบงานภายในของภาครัฐ (Back Office) โดยอาศัยหลักฐานอ้างอิง (Authentication Credential) ชุดเดียว โดยหลักฐานอ้างอิงดังกล่าวอาจเป็น Login/ Password หรือในอนาคตอาจเป็นใบรับรองอิเล็กทรอนิกส์ก็ได้
- หน่วยงานภาครัฐไม่จำเป็นที่จะต้องพัฒนาระบบสำหรับยืนยันตัวบุคคลด้วยตนเอง ซึ่งเป็นการลดค่าใช้จ่ายในการพัฒนาและดูแลรักษาฐานข้อมูลผู้ใช้งาน และระบบการยืนยันตัวบุคคล

## 3. สถาปัตยกรรมของระบบในภาพรวม

องค์ประกอบหลักในการเชื่อมโยงระบบสารสนเทศภาครัฐกับระบบยืนยันตัวบุคคลกลางแบบ Single Sign-On สามารถแสดงได้ ดังรูปที่ 3-1




รูปที่ 3-1 องค์ประกอบหลักที่เกี่ยวข้องกับการเชื่อมโยงระบบสารสนเทศภาครัฐกับระบบยืนยันตัวบุคคลกลาง

- **เว็บไซต์ท่าสำหรับเข้าถึงข้อมูลและระบบงานต่าง ๆ ของภาครัฐ (Web Portal)** - เว็บไซต์ซึ่งรวบรวมข้อมูลข่าวสารระบบงานต่าง ๆ ไว้ในเว็บไซต์เดียว เพื่ออำนวยความสะดวกให้กับประชาชนหรือเจ้าหน้าที่ของรัฐในการเข้าถึงข้อมูล และระบบงานต่าง ๆ ดังกล่าว โดยเว็บไซต์ดังกล่าวเป็นได้ทั้ง
  - ระบบเว็บไซต์กลางบริการอิเล็กทรอนิกส์ภาครัฐ (e-Government Portal) ซึ่งรวบรวมข้อมูล และบริการต่าง ๆ สำหรับประชาชนทั่วไป
  - เว็บไซต์หน่วยงาน ซึ่งมักจะรวบรวมลิงค์ไปยังระบบงานต่าง ๆ ที่เจ้าหน้าที่ของหน่วยงานนั้น ๆ จำเป็นต้องใช้ อาทิเช่น ระบบเว็บเมล์ ระบบอินทราเน็ต หรือระบบงานภายในอื่น ๆ เป็นต้น
- **ระบบยืนยันตัวบุคคลกลาง (e-Authentication Service)** เป็นระบบที่ใช้ในการยืนยันตัวบุคคลโดยอาศัยหลักฐานอ้างอิง (Authentication Credential) ที่เหมาะสม โดยหลักฐานอ้างอิงดังกล่าวอาจเป็น Login/ Password หรือในอนาคตอาจเป็น One Time Password หรือใบรับรองอิเล็กทรอนิกส์ก็ได้
- **ระบบสารสนเทศภาครัฐ** – ครอบคลุมถึงระบบต่าง ๆ ดังนี้
  - **ระบบบริการอิเล็กทรอนิกส์ภาครัฐ (e-Service)** หมายถึง ระบบของหน่วยงานภาครัฐซึ่งให้บริการอิเล็กทรอนิกส์สำหรับประชาชน ผู้ประกอบการ หรือชาวต่างชาติ โดยบริการดังกล่าวอาจจะเป็นใน

ลักษณะของการให้ข้อมูล (Information) มีการปฏิสัมพันธ์กับประชาชน (Interaction) รองรับการดำเนินธุรกรรมภาครัฐ (Interchange Transaction) หรืออยู่ในระดับของการบูรณาการ (Integration) ก็ได้

- **ระบบงานภายในของภาครัฐ (Back Office)** หมายถึง ระบบของหน่วยงานภาครัฐซึ่งใช้ในการบริหารจัดการ สนับสนุนงานตามภารกิจของหน่วยงาน อาทิเช่น ระบบบัญชี ระบบบริหารงานบุคคล ระบบงบประมาณ ระบบยุทธศาสตร์ แผนงาน โครงการ เป็นต้น

❖ **การลงชื่อเข้าใช้งานแบบรวมศูนย์ (Single Sign-On)** หมายถึง การที่ผู้ใช้งานสามารถลงชื่อเข้าใช้งานระบบ (Login) ครั้งเดียว แล้วสามารถเข้าใช้งานระบบหลายระบบได้โดยไม่ต้องลงชื่อเข้าใช้งานซ้ำอีก ทั้งนี้

- ถึงแม้ผู้ใช้งานจะไม่ต้อง Login ซ้ำอีกครั้งที่ระบบถัด ๆ ไป แต่ระบบดังกล่าวก็ยังคงต้องเป็นผู้ตรวจสอบสิทธิในการเข้าถึงข้อมูลและบริการของผู้ใช้งานนั้น ๆ
- ในกรณีที่ระบบยังต้องการข้อมูลเพิ่มเติมจากผู้ใช้งานเพื่อการยืนยันตัวบุคคล ระบบนั้น ๆ ก็ยังสามารถทำได้
- ผู้ใช้งานยังสามารถ Login เพื่อระบบต่าง ๆ ได้โดยตรงโดยไม่ต้องผ่านระบบเว็บไซต์กลาง

## 4. ประเภทของผู้ใช้ที่รองรับ

ระบบยืนยันตัวบุคคลกลางรองรับผู้ใช้งาน 5 ประเภท ดังนี้

| | ประเภทผู้ใช้ที่รองรับ | วิธีการสมัครเพื่อขอใช้งาน |
|---|---|---|
| 1 | ประชาชน/ บุคคลธรรมดา | สมัครด้วยตนเองแบบออนไลน์ |
| 2 | นิติบุคคล | สมัครด้วยตนเองแบบออนไลน์ |
| 3 | ชาวต่างชาติ | สมัครด้วยตนเองแบบออนไลน์ |
| 4 | ข้าราชการ/ เจ้าหน้าที่รัฐที่ไม่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai | สมัครด้วยตนเองแบบออนไลน์ |
| 5 | ข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai | 1) หน่วยงานต้นสังกัดสมัครขอใช้บริการ MailGoThai กับ สพร.<br>2) เจ้าหน้าที่ขอบัญชีผู้ใช้งานกับผู้ดูแลระบบของหน่วยงาน |

## 5. ระดับของการยืนยันตัวบุคคล

ปัจจุบัน รัฐบาลไทยยังไม่มีการออกข้อกำหนดเรื่องระดับของการยืนยันตัวบุคคลเพื่อใช้ในการทำธุรกรรมอิเล็กทรอนิกส์กับหน่วยงานภาครัฐ ซึ่งข้อกำหนดดังกล่าวเป็นสิ่งจำเป็น ถ้าต้องการให้เกิดรูปแบบการทำธุรกรรม-อิเลิกทรอนิกส์กับหน่วยงานภาครัฐที่เป็นมาตรฐาน และสามารถเชื่อมโยงธุรกรรมอิเล็กทรอนิกส์ระหว่างหน่วยงานได้

ดังนั้น สพร. จึงกำหนดแนวทางการจำแนกบริการตามระดับความเสียหายที่ยอมรับได้หากเกิดความผิดพลาดในการยืนยันตัวบุคคล โดยได้ประยุกต์เนื้อหาจากเอกสาร “Registration and Authentication: e-Government Strategy Framework Policy and Guidelines” เวอร์ชั่น 3.0 จัดทำโดยรัฐบาลอังกฤษ กันยายน 2545 โดยจำแนกระดับความเสียหายไว้ ดังนี้

**ระดับ 0**: เหมาะสมกับธุรกรรมอิเล็กทรอนิกส์ภาครัฐที่*ไม่ก่อให้เกิดความเสียหาย* ถ้าเกิดความผิดพลาดในการยืนยันตัวบุคคล – นั่นคือ ความผิดพลาดในการยืนยันตัวบุคคล อย่างมากที่สุดแล้ว อาจก่อให้เกิด

- ✓ ไม่ก่อให้เกิดความไม่สะดวกต่อผู้ที่เกี่ยวข้อง
- ✓ ข้อมูลส่วนตัวหรือข้อมูลที่สามารถนำไปใช้ประโยชน์เชิงพาณิชย์ได้ไม่เกิดการรั่วไหลไปยังบุคคลภายนอก
- ✓ ไม่ก่อให้เกิดความเสียหายด้านร่างกายหรือทรัพย์สินต่อบุคคลใด
- ✓ ไม่ก่อให้เกิดความเสื่อมเสียต่อบุคคลใด
- ✓ ไม่ก่อให้เกิดอุปสรรคหรือความล่าช้าต่อการตรวจพบอาชญากรรม

<u>ตัวอย่าง</u>

- ผู้ใช้งานอ่านหรือดาวน์โหลดข้อมูลที่เปิดเผยต่อสาธารณะจากเว็บไซต์ของหน่วยงาน

<u>วิธีการยืนยันตัวบุคคลขั้นต่ำที่ต้องการ</u>

- ผู้ใช้งานไม่จำเป็นที่จะต้องลงทะเบียน และไม่จำเป็นที่จะต้องยืนยันตัวบุคคล

**ระดับ 1**: เหมาะสมกับธุรกรรมอิเล็กทรอนิกส์ภาครัฐที่*อาจก่อให้เกิดความเสียหายเล็กน้อย* ถ้าเกิดความผิดพลาดในการยืนยันตัวบุคคล – นั่นคือ ความผิดพลาดในการยืนยันตัวบุคคล อย่างมากที่สุดแล้วอาจก่อให้เกิด

- ✓ อาจก่อให้เกิดความไม่สะดวกต่อผู้ที่เกี่ยวข้องเล็กน้อย
- ✓ ข้อมูลส่วนตัว หรือข้อมูลที่สามารถนำไปใช้ประโยชน์เชิงพาณิชย์ได้ไม่เกิดการรั่วไหลไปยังบุคคลภายนอก
- ✓ ไม่ก่อให้เกิดความเสียหายด้านร่างกายต่อผู้ที่เกี่ยวข้อง
- ✓ อาจก่อให้เกิดความเสียหายด้านทรัพย์สินต่อผู้ที่เกี่ยวข้องเล็กน้อย
- ✓ อาจก่อให้เกิดความเสื่อมเสียต่อผู้ที่เกี่ยวข้องเล็กน้อย
- ✓ ไม่ก่อให้เกิดอุปสรรคหรือความล่าช้าต่อการตรวจพบอาชญากรรม

<u>ตัวอย่าง</u>

- ผู้ใช้งานร้องขอข้อมูลจากหน่วยงานภาครัฐผ่านอินเตอร์เน็ต
- ผู้ใช้งานนัดหมายเจ้าหน้าที่ของหน่วยงานผ่านทางเว็บไซต์

<u>วิธีการยืนยันตัวบุคคลขั้นต่ำที่ต้องการ</u>

- ผู้ใช้งานจำเป็นที่จะต้องลงทะเบียนกับระบบเพื่อขอใช้บริการ โดยระบุรายละเอียด ชื่อ นามสกุล และรายละเอียดที่ติดต่อได้ (อาทิเช่น อีเมล์) เป็นอย่างน้อย
- ผู้ใช้งานจำเป็นที่จะต้องยืนยันตัวบุคคล โดยใช้ Login/ Password เป็นอย่างน้อย

**ระดับ 2**: เหมาะสมกับธุรกรรมอิเล็กทรอนิกส์ภาครัฐที่*อาจก่อให้เกิดความเสียหายพอสมควร* ถ้าเกิดความผิดพลาดในการยืนยันตัวบุคคล – นั่นคือ ความผิดพลาดในการยืนยันตัวบุคคล อย่างมากที่สุดแล้ว อาจก่อให้เกิด

- ✓ อาจก่อให้เกิดความไม่สะดวกต่อผู้ที่เกี่ยวข้องพอสมควร

- ✓ ข้อมูลส่วนตัวหรือข้อมูลที่สามารถนำไปใช้ประโยชน์เชิงพาณิชย์ได้อาจเกิดการรั่วไหลไปยังบุคคลภายนอก
- ✓ ไม่ก่อให้เกิดความเสียหายด้านร่างกายต่อผู้ที่เกี่ยวข้อง
- ✓ อาจก่อให้เกิดความเสียหายด้านทรัพย์สินต่อผู้ที่เกี่ยวข้องพอสมควร
- ✓ อาจก่อให้เกิดความเสื่อมเสียต่อผู้ที่เกี่ยวข้องพอสมควร
- ✓ อาจก่อให้เกิดอุปสรรคหรือความล่าช้าต่อการตรวจพบอาชญากรรม

ตัวอย่าง

- ผู้ใช้งานขอยืนภาษีเงินได้บุคคลธรรมดากับกรมสรรพากร
- ผู้ใช้งานขอตรวจสอบสิทธิประกันสุขภาพของตน
- ผู้ใช้งานขอจดทะเบียนนิติบุคคล

วิธีการยืนยันตัวบุคคลขั้นต่ำที่ต้องการ

- ผู้ใช้งานจำเป็นที่จะต้องลงทะเบียนกับระบบเพื่อขอใช้บริการ โดยระบุรายละเอียดเลขประจำตัว 13 หลัก ชื่อ นามสกุล วัน-เดือน-ปีเกิด โดยการลงทะเบียนต้องผ่านการตรวจสอบความถูกต้องจากหน่วยงานที่น่าเชื่อถือ อาทิเช่น กรมการปกครอง
- ผู้ใช้งานจำเป็นที่จะต้องส่งเอกสารเพิ่มเติมไปยังหน่วยงานที่เกี่ยวข้องเพื่อยืนยันการทำธุรกรรม

**ระดับ 3**: เหมาะสมกับธุรกรรมอิเล็กทรอนิกส์ภาครัฐที่*อาจก่อให้เกิดความเสียหายอย่างมาก* ถ้าเกิดความผิดพลาดในการยืนยันตัวบุคคล – นั่นคือ ความผิดพลาดในการยืนยันตัวบุคคล อย่างมากที่สุดแล้ว อาจก่อให้เกิด

- ✓ อาจก่อให้เกิดความไม่สะดวกต่อผู้ที่เกี่ยวข้องอย่างมาก
- ✓ ข้อมูลส่วนตัว หรือข้อมูลที่สามารถนำไปใช้ประโยชน์เชิงพาณิชย์ได้อาจเกิดการรั่วไหลไปยังบุคคลภายนอก
- ✓ อาจก่อให้เกิดความเสียหายด้านร่างกายต่อผู้ที่เกี่ยวข้อง
- ✓ อาจก่อให้เกิดความเสียหายด้านทรัพย์สินต่อผู้ที่เกี่ยวข้องอย่างมาก
- ✓ อาจก่อให้เกิดความเสื่อมเสียต่อผู้ที่เกี่ยวข้องอย่างมาก
- ✓ อาจก่อให้เกิดอุปสรรคหรือความล่าช้าต่อการตรวจพบอาชญากรรม

ตัวอย่าง

- ผู้ใช้งานต้องการขอบัตรประจำตัวประชาชนใบใหม่

วิธีการยืนยันตัวบุคคลขั้นต่ำที่ต้องการ

- ผู้ร้องขอบริการจำเป็นต้องไปยื่นคำร้องด้วยตนเองกับเจ้าหน้าที่ ณ จุดบริการ

ระบบยืนยันตัวบุคคลกลางที่ สพร. พัฒนาขึ้นรองรับระดับการยืนยันตัวบุคคลในหลายระดับ ดังนี้

| | ประเภทผู้ใช้ที่รองรับ | ข้อมูลเพื่อใช้ตรวจสอบ (Idenfication) | วิธีการยืนยันตัวบุคคล/ ยืนยันความถูกต้อง (Verification) | ระดับการยืนยันตัวบุคคล |
|---|---|---|---|---|
| 1 | ประชาชน/ บุคคลธรรมดา | อีเมล์ | ส่งอีเมล์ให้ผู้ใช้งานคลิกลิงค์จากอีเมล์ดังกล่าว เพื่อยืนยันความเป็นเจ้าของอีเมล์ | ระดับที่ 1 |
| | | เลขประจำตัวประชาชน | ให้ผู้ใช้งานกรอกข้อมูลส่วนบุคคล อาทิ ชื่อบิดา-มารดา และตรวจสอบข้อมูลดังกล่าวกับกรมการปกครองแล้ว | ระดับที่ 2 |
| 2 | นิติบุคคล | เลขทะเบียนนิติบุคคล | ยังไม่สามารถตรวจสอบผ่านระบบออนไลน์ได้ | ระดับที่ 1 |

| | ประเภทผู้ใช้ที่รองรับ | ข้อมูลเพื่อใช้ตรวจสอบ (Idenfication) | วิธีการยืนยันตัวบุคคล/ ยืนยันความถูกต้อง (Verification) | ระดับการยืนยันตัวบุคคล |
|---|---|---|---|---|
| 3 | ชาวต่างชาติ | หมายเลขหนังสือเดินทาง (Passport Number) | ยังไม่สามารถตรวจสอบผ่านระบบออนไลน์ได้ | ระดับที่ 1 |
| 4 | ข้าราชการ/ เจ้าหน้าที่รัฐที่ไม่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai | อีเมล์ | ส่งอีเมล์ให้ผู้ใช้งานคลิกลิงค์จากอีเมล์ดังกล่าว เพื่อยืนยันความเป็นเจ้าของอีเมล์ | ระดับที่ 1 |
| | | เลขประจำตัวประชาชน | ให้ผู้ใช้งานกรอกข้อมูลส่วนบุคคล อาทิ ชื่อบิดา-มารดา และตรวจสอบข้อมูลดังกล่าวกับกรมการปกครองแล้ว | ระดับที่ 2 |
| 5 | ข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai | ไม่มี | หน่วยงานต้นสังกัดเป็นผู้บริหารจัดการบัญชีผู้ใช้งานของผู้ใช้งานในหน่วยงานตนเอง จึงมีความน่าเชื่อถือในระดับหนึ่ง | ระดับที่ 2 |

ทั้งนี้ ในปัจจุบัน ข้าราชการ/ เจ้าหน้าที่ของรัฐที่ใช้ระบบ MailGoThai ยังไม่สามารถระบุและยืนยันเลขประจำตัวประชาชน (13 หลัก) ของตนได้ เนื่องจาก สพร. ให้สิทธิหน่วยงานในการบริหารจัดการบัญชีผู้ใช้ระบบ MailGoThai ของเจ้าหน้าที่ภายใต้หน่วยงานตน และแต่ละหน่วยงานก็มีวิธีการที่แตกต่างกันในการบริหารจัดการ บางหน่วยงานก็มีการบันทึกข้อมูลเลขประจำตัวฯ 13 หลักของเจ้าหน้าที่ของตน บางหน่วยงานก็ไม่มี บางหน่วยงานก็อนุญาตให้เจ้าหน้าที่แก้ไขข้อมูลส่วนตัวได้ บางหน่วยงานก็ไม่ได้ เป็นต้น ดังนั้น สพร. จึงขอเสนอว่า

1) บัญชีผู้ใช้ระบบ MailGoThai – ให้ใช้กับระบบที่เป็นระบบภายใน (Back Office) ของหน่วยงานเป็นหลัก
2) หากถ้าข้าราชการ/ เจ้าหน้าที่ของรัฐ ต้องการใช้งานระบบบริการของรัฐ (e-Service หรือ Front Office) ให้เจ้าหน้าที่สมัครใช้บริการระบบยืนยันตัวบุคคลกลางในฐานะ *ประชาชน/ บุคคลธรรมดา* อีกบัญชีหนึ่ง

ในอนาคต ระบบอาจได้รับการขยายผลให้รองรับการยืนยันตัวบุคคลในรูปแบบอื่นเพิ่มเติม ซึ่งมีความน่าเชื่อถือในระดับสูงขึ้น อาทิ

- การยืนยันตัวบุคคลโดยการมาแสดงตัวตนกับเจ้าหน้าที่: ในอนาคตกระทรวงฯ อาจจัดเตรียมสถานที่ซึ่งผู้ใช้งานสามารถมาแสดงตน พร้อมหลักฐานต่าง ๆ อาทิ บัตรประจำตัวประชาชน หนังสือเดินทาง เพื่อขอใช้บริการต่าง ๆ ของรัฐผ่านอิเล็กทรอนิกส์ได้ ซึ่งผู้ที่ผ่านการยืนยันตัวบุคคลด้วยวิธีนี้ควรจะสามารถเข้าถึงบริการระดับ 3 ได้ (อาทิเช่น การขอจดทะเบียนนิติบุคคล)
- การยืนยันตัวบุคคลโดยใช้ One-Time Password: ในกรณีนี้ก่อนที่ผู้ใช้งานจะทำธุรกรรมสำคัญทุกครั้ง (อาทิเช่น การโอนเงินผ่านธนาคาร) ระบบจะส่ง One-Time Password ให้กับผู้ใช้งานทางช่องทางที่เชื่อถือได้ (อาทิเช่น ทางโทรศัพท์มือถือที่ได้ลงทะเบียนกับหน่วยงาน/ ทางการไว้แล้ว) ผู้ใช้งานต้องระบุ One-Time Password ดังกล่าวกับระบบ จึงจะสามารถยืนยันและดำเนินการธุรกรรมดังกล่าวได้ การยืนยันตัวบุคคลด้วยวิธีนี้เป็นที่แพร่หลายในวงการพาณิชย์อิเล็กทรอนิกส์ (อาทิเช่น e-Banking) และมีความน่าเชื่อถือพอสมควร แต่ยังไม่เป็นที่แพร่หลายในวงการรัฐบาลอิเล็กทรอนิกส์

## 6. สมมติฐานการเชื่อมโยงระบบสารสนเทศของหน่วยงานเข้ากับระบบยืนยันตัวบุคคลกลาง

- ระบบสารสนเทศภาครัฐที่ต้องใช้งานระบบยืนยันตัวบุคคลกลางจะต้องเป็นระบบเว็บแอพพลิเคชั่นที่สามารถเข้าถึงได้ผ่านเครือข่ายอินเตอร์เน็ต
- ระบบสารสนเทศภาครัฐที่ต้องใช้งานระบบยืนยันตัวบุคคลกลางควรจะต้องมีระบบสมาชิก และผู้ใช้งานจะต้องสมัครสมาชิกก่อนที่จะเข้าถึงข้อมูลและบริการต่าง ๆ ในระบบได้
- เทคนิคการยืนยันตัวบุคคลที่กล่าวถึงในเอกสารนี้ได้ออกแบบบนข้อสมมุติฐานที่ว่าระบบของหน่วยงานภาครัฐจะต้องเชื่อถือ (Trust) การยืนยันตัวบุคคลที่ดำเนินการโดยระบบยืนยันตัวบุคคลกลาง และอนุญาตให้ผู้ใช้งานที่ยืนยันตัวตนกับระบบยืนยันตัวบุคคลกลางแล้ว สามารถเข้าถึงบริการและข้อมูลต่าง ๆ ตามสิทธิที่หน่วยงานกำหนดไว้สำหรับบุคคลประเภทนั้นได้

## 7. ขั้นตอนการขอเชื่อมโยงระบบสารสนเทศของหน่วยงานกับระบบยืนยันตัวบุคคลกลางแบบ Sigle Sign-On

ส่วนราชการที่มีระบบสารสนเทศอยู่แล้ว และต้องการที่จะเชื่อมโยงระบบของตนกับระบบยืนยันตัวบุคคลกลาง ต้องดำเนินการตามขั้นตอนต่อไปนี้

1) ประเมินว่าระบบของตนยอมรับความเสียหายจากการยืนยันตัวผิดพลาดได้ในระดับใด (ดูหัวข้อที่ 5) เนื่องจากในปัจจุบันระบบเว็บไซต์กลางมีการยืนยันตัวบุคคลที่ระดับ 1-2 เท่านั้น ซึ่งอาจไม่เหมาะกับระดับความปลอดภัยที่หน่วยงานต้องการ
2) ในกรณีที่ส่วนราชการเห็นว่าระบบของตนควรจะเชื่อมโยงกับระบบยืนยันตัวบุคคลกลาง เจ้าหน้าที่สารสนเทศของหน่วยงานควรศึกษารายละเอียดทางเทคนิคเพิ่มเติม (ดูหัวข้อที่ 8) รวมถึงการดาวน์โหลด Software Library ต่าง ๆ ที่จำเป็น และศึกษาตัวอย่าง Source Code ต่าง ๆ
3) ประสานงานกับสำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) (สพร.) เพื่อขอเชื่อมโยง ท่านจะได้รับ ServiceCode (OAuth Consumer Key) และ Passcode (OAuth Consumer Secret) เพื่อใช้ในการเรียกข้อมูลต่าง ๆ ผ่าน API ที่กำหนดได้ (ดูหัวข้อ 12.1)
4) ทำการพัฒนาหน้าเว็บเพจที่เกี่ยวข้อง ได้แก่ หน้า SSOLogin และ/หรือ หน้า SSORegister ตามที่ได้กล่าวถึงในหัวข้อที่ 10
5) ทดสอบการเชื่อมโยง

| ประเภทผู้ใช้งาน | Url ทดสอบ |
|---|---|
| ประชาชน/ บุคคลธรรมดา | http://testopenid.ega.or.th/ |
| นิติบุคคล | http://testopenid.ega.or.th/ |
| ชาวต่างชาติ | http://testopenid.ega.or.th/ |
| ข้าราชการ/ เจ้าหน้าที่ของรัฐที่ไม่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai | http://testopenid.ega.or.th/ |
| ข้าราชการ/ เจ้าหน้าที่ของรัฐ ที่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai | http://govid.ega.or.th/ |

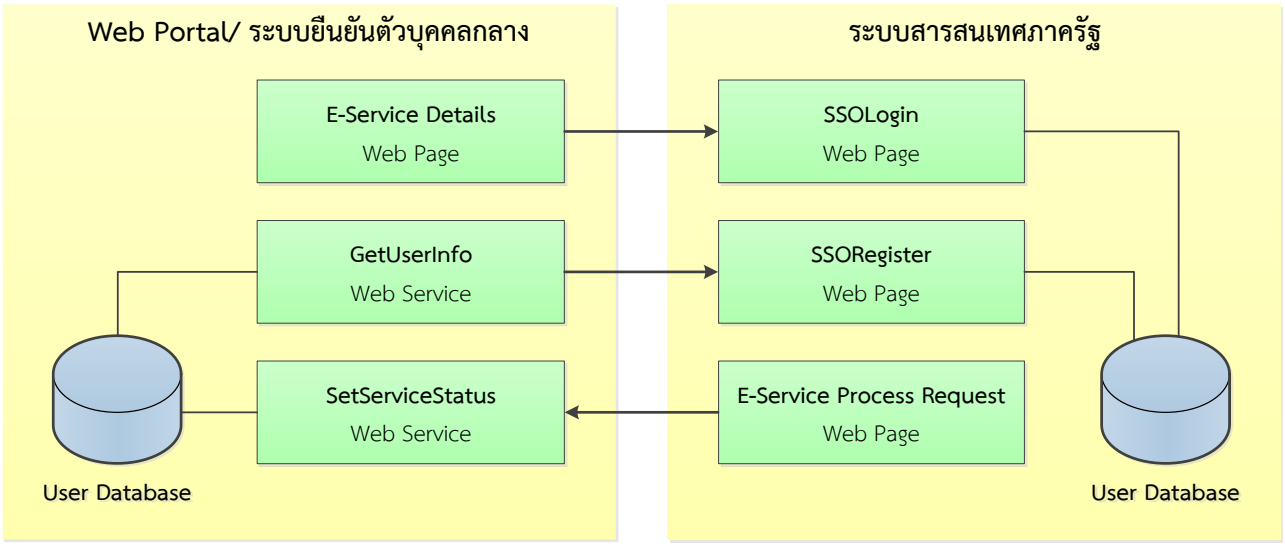
6) ปรึกษาทีมงานสนับสนุนของสำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) (สพร.) ถ้าเกิดปัญหาในการพัฒนาหรือในการเชื่อมโยง (ดูหัวข้อ 12.1)

## 8. รายละเอียดด้านเทคนิค

ในส่วนนี้เป็นรายละเอียดเชิงเทคนิคของการเชื่อมโยงระบบสารสนเทศภาครัฐแบบ Single Sign-On ซึ่งมีเป้าหมายให้ผู้ที่พัฒนาระบบสารสนเทศของหน่วยงานภาครัฐสามารถศึกษารายละเอียดในหัวข้อนี้ ประกอบกับการศึกษาตัวอย่าง Source Code ต่าง ๆ เพิ่มเติมจากเว็บไซต์ที่กำหนด เพื่อประโยชน์ต่อการพัฒนาระบบสารสนเทศของหน่วยงานให้สามารถเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางแบบ Single Sign-On ได้

### 8.1 องค์ประกอบหลักของระบบ

องค์ประกอบสำคัญที่เกี่ยวข้องกับการเชื่อมโยงระบบสารสนเทศภาครัฐเข้ากับระบบยืนยันตัวบุคคลกลางแบบ Single Sign-On นั้น สามารถแบ่งออกเป็น 2 กลุ่ม ได้แก่ กลุ่มเว็บไซต์ท่าและระบบยืนยันตัวบุคคลกลาง และกลุ่มระบบสารสนเทศภาครัฐ ดังรูปที่ 8-1




รูปที่ 8-1 องค์ประกอบที่สำคัญการเชื่อมโยงระบบสารสนเทศภาครัฐแบบ Single Sign-On

1) **กลุ่มเว็บไซต์ท่าและระบบยืนยันตัวบุคคลกลาง** – เป็นองค์ประกอบที่ สพร. จะพัฒนาขึ้นเพื่อรองรับการเชื่อมโยงระบบสารสนเทศภาครัฐเข้ากับระบบแบบ Single Sign-On ประกอบด้วย
   1.1) E-Service Details - เป็นหน้าจอรายละเอียดบริการของหน่วยงานที่รวบรวมไว้ในเว็บไซต์ท่า ถ้าผู้ใช้งานสนใจสามารถคลิกปุ่มเพื่อขอเข้าไปใช้บริการนี้แบบ Single Sign-On ได้
   1.2) GetUserInfo - เป็นโปรแกรมซึ่งระบบสารสนเทศของหน่วยงานสามารถใช้ร้องขอข้อมูลผู้ใช้งานในแต่ละประเภท (บุคคลธรรมดา ข้าราชการ/ เจ้าหน้าที่รัฐ) ในแบบ Web Service ได้
   1.3) SetServiceStatus - เป็นโปรแกรมซึ่งระบบสารสนเทศของหน่วยงานสามารถใช้แจ้งสถานะล่าสุดของคำร้องขอบริการมายังเว็บไซต์ท่าในแบบ Web Service ได้
2) **กลุ่มระบบสารสนเทศภาครัฐ** - ระบบสารสนเทศของหน่วยงานเป็นองค์ประกอบซึ่งผู้พัฒนาของหน่วยงานภาครัฐต่าง ๆ จะต้องพัฒนาขึ้นเพื่อให้สามารถเชื่อมโยงระบบของตนเข้ากับระบบยืนยันตัวบุคคลกลางในแบบ Single Sign-On ได้

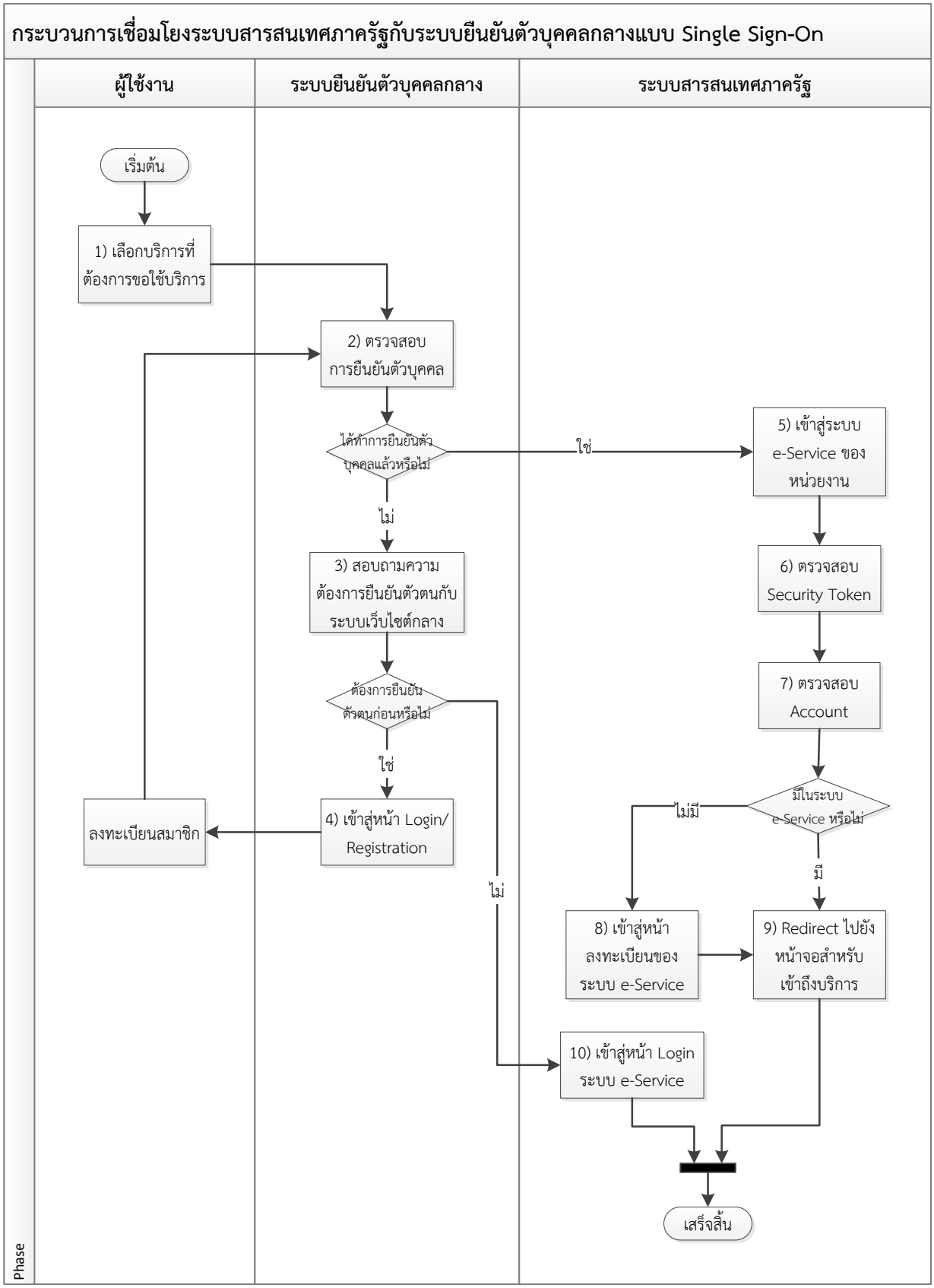
2.1) SSOLogin เป็นหน้าจอสำหรับรับคำร้องขอใช้งานระบบสารสนเทศภาครัฐจากเว็บไซต์ท่าแบบ Single Sign-On หน้าจอนี้จะตรวจสอบรายละเอียดผู้ใช้งาน และ Redirect ผู้ใช้งานไปยังหน้าสำหรับขอใช้บริการหรือหน้าลงทะเบียนขอใช้บริการตามความเหมาะสม

2.2) SSO Register เป็นหน้าจอสำหรับรองรับการลงทะเบียน/ สมัครสมาชิกใหม่ที่ทำ Single Sign-On มาจากเว็บไซต์ท่า

2.3) E-Service Process Request เป็นหน้าจอที่เจ้าหน้าที่หน่วยงานใช้ในการบันทึกสถานะความก้าวหน้าในการจัดการคำร้องขอบริการแต่ละคำร้อง - ซึ่งเมื่อมีการจัดการคำร้องแล้วหน้าจอนี้จะต้องแจ้งผลไปยังเว็บไซต์ท่า (ผ่าน SetServiceStatus Web Service) เพื่อให้ผู้ร้องขอบริการได้รับทราบสถานะล่าสุดของคำร้อง

## 8.2 ขั้นตอนการทำงาน

ขั้นตอนการทำงานในภาพรวมของการเชื่อมโยงระบบสารสนเทศภาครัฐแบบ Single Sign-On สรุปได้ ดังนี้




รูปที่ 8-2 ภาพรวมการทำงานของการเชื่อมโยงระบบสารสนเทศภาครัฐแบบ Single Sign-On

1) ผู้ใช้งาน (ทุกประเภท) เลือกบริการที่ต้องการขอใช้บริการจากเว็บไซต์ท่า (หน้าจอ E-Service Details) โดยบริการดังกล่าวรองรับการเชื่อมโยงแบบ Single Sign-On กับระบบยืนยันตัวบุคคลกลาง
2) ระบบยืนยันตัวบุคคลกลางตรวจสอบว่าผู้ใช้งานดังกล่าวได้ยืนยันตัวบุคคลกับระบบแล้วหรือไม่
   2.1) ถ้าผู้ใช้งานยังไม่ได้ยืนยันตัวเองกับระบบยืนยันตัวบุคคลกลางให้ไปที่ขั้นตอนที่ 3)
   2.2) ถ้าผู้ใช้งานได้ทำการยืนยันตัวแล้ว ให้ไปที่ขั้นตอนที่ 5)
3) ผู้ใช้งานยังไม่ได้ยืนยันตัวกับระบบยืนยันตัวบุคคลกลาง ระบบจะสอบถามผู้ใช้งานว่าต้องการยืนยันตัวตนกับระบบยืนยันตัวบุคคลกลาง (แบบ Single Sign-On) ก่อนหรือไม่
   3.1) ถ้าผู้ใช้งานไม่ต้องการยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลาง (เช่น ผู้ใช้อาจจะยังไม่ได้ลงทะเบียนกับระบบยืนยันตัวบุคคลกลางไว้) ให้ไปขั้นตอนที่ 10)
   3.2) ถ้าผู้ใช้งานต้องการยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลางให้ไปที่ขั้นตอนที่ 4)
4) ผู้ใช้งานต้องการยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลาง ระบบจะแสดงหน้าจอ Login/ Registration เพื่อขอ Username และ Password จากผู้ใช้งาน ซึ่งเมื่อผู้ใช้งานระบุ Username และ Password และยืนยันการเข้าสู่ระบบแล้ว จะกลับไปยังขั้นตอนที่ 2)
   หมายเหตุ: กรณีที่ผู้ใช้งานต้องการยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลาง แต่ยังไม่ได้สมัครเป็นสมาชิกกับระบบยืนยันตัวบุคคลกลาง ระบบจะนำผู้ใช้งานไปสู่หน้าจอลงทะเบียน/ สมัครสมาชิก เพื่อขอใช้งานระบบ
5) ผู้ใช้งานได้ยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลางเรียบร้อยแล้ว ระบบจะเปิดหน้าเว็บบราวเซอร์หน้าใหม่ พร้อมนำผู้ใช้งานไปยังหน้าจอเพื่อเข้าถึงบริการที่ผู้ใช้ต้องการแบบ Single Sign-On
6) ระบบสารสนเทศของหน่วยงาน (หน้าจอ SSOLogin) ตรวจสอบสถานะการยืนยันตัวบุคคล และดึงข้อมูลรายละเอียดของผู้ใช้งานนั้น ๆ จากระบบยืนยันตัวบุคคลกลาง (ถ้าต้องการ) (GetUserInfo Web Service)
   6.1) กณีที่ระบบสารสนเทศของหน่วยงานมีระบบสมาชิกรองรับ ให้ดำเนินการตามขั้นตอนที่ 7)
   6.2) กณีที่ระบบสารสนเทศของหน่วยงานไม่มีระบบสมาชิกรองรับ เช่น ระบบ e-Form ต่าง ๆ ระบบอาจทำการดึงข้อมูลรายละเอียดผู้ใช้งานที่ต้องการมาแสดงที่หน้าฟอร์มได้เลย เป็นอันสิ้นสุดขั้นตอนการเชื่อมโยงกับระบบยืนยันตัวบุคคลกลาง
7) ระบบสารสนเทศของหน่วยงานตรวจสอบผู้ใช้งานดังกล่าวว่ามีบัญชีผู้ใช้งานอยู่กับระบบสารสนเทศของหน่วยงานแล้วหรือไม่
   7.1) ถ้าไม่มี ให้ระบบเข้าสู่ขั้นตอนการ Enroll (ขั้นตอนที่ 8)
   7.2) ถ้ามีอยู่แล้ว จะเข้าสู่ขั้นตอนที่ 9)

   การตรวจสอบข้อมูล[1] ของผู้ใช้งานแต่ละประเภทนั้นมีการตรวจสอบข้อมูลที่แตกต่างกัน ดังนี้
   - ✓ ประชาชน/ บุคคลธรรมดา: ให้ตรวจสอบจากเลขประจำตัวประชาชน 13 หลัก
   - ✓ นิติบุคคล: ให้ตรวจสอบจากเลขทะเบียนนิติบุคคล
   - ✓ ชาวต่างชาติ: ให้ตรวจสอบจากหมายเลขหนังสือเดินทาง (Passport)
   - ✓ ข้าราชการ/ เจ้าหน้าที่รัฐ (ที่ไม่มีบัญชีผู้ใช้งาน MailGoThai): ให้ตรวจสอบจากเลขประจำตัวประชาชน 13 หลัก

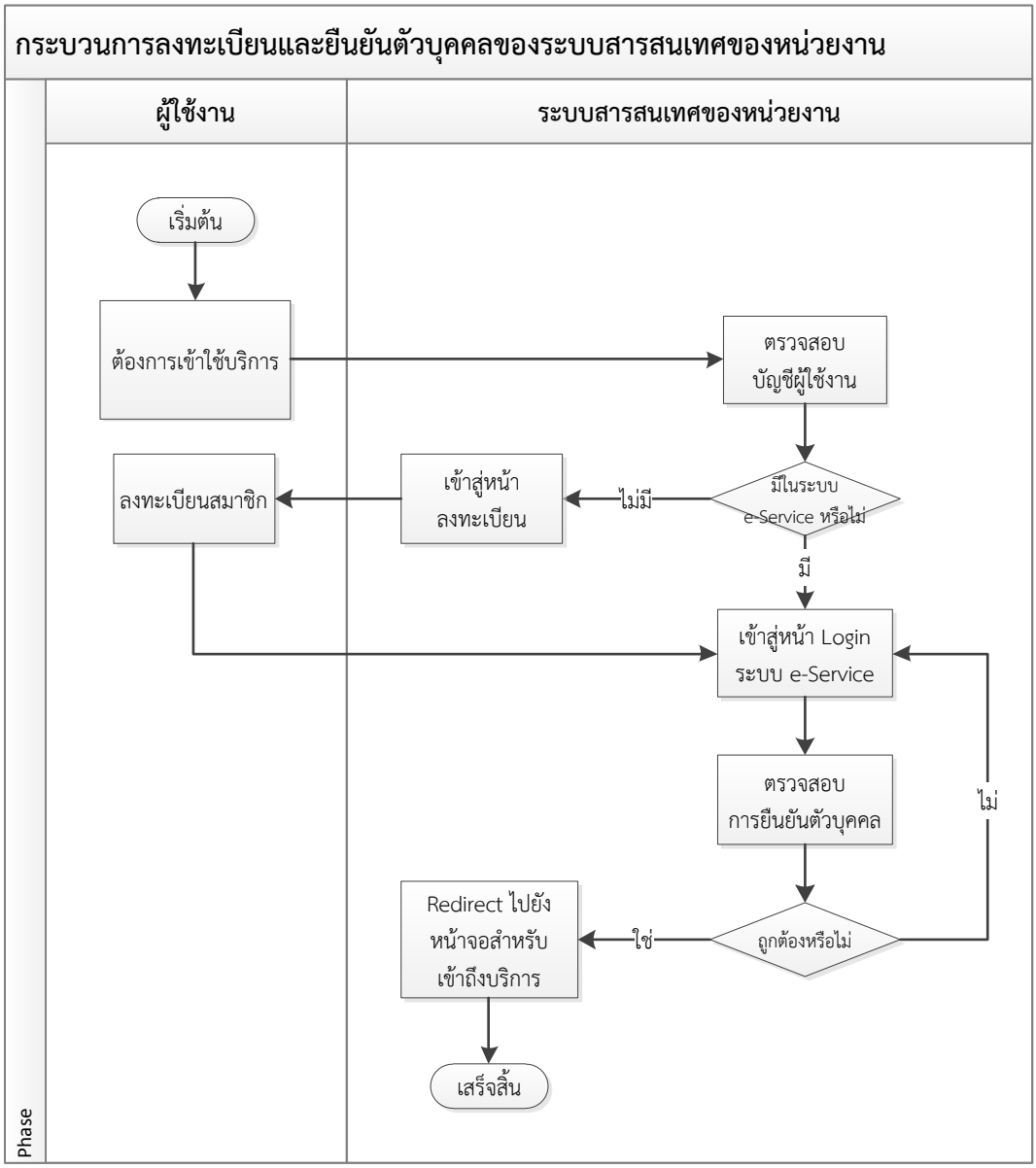
[1] ระบบสารสนเทศของหน่วยงานสามารถดูรายละเอียดสถานะของผู้ใช้งานเพิ่มเติมได้ เพื่อประกอบการตัดสินใจว่าจะ “เชื่อใจ” ให้ผู้ใช้งานท่านนั้นว่าเป็นบุคคลที่กล่าวอ้างหรือไม่ โดยศึกษาเพิ่มเติมได้จากหัวข้อ “วิธีการยืนยันตัวบุคคล”

- ✓ ข้าราชการ/ เจ้าหน้าที่รัฐ (ที่มีบัญชีผู้ใช้งาน MailGoThai): ให้ตรวจสอบจากเลขประจำตัวประชาชน 13 หลัก

หมายเหตุ: ปัจจุบันทาง สพร. ยังไม่มีขั้นตอนการตรวจสอบข้อมูลของผู้ใช้งานประเภทนิติบุคคล ชาวต่างชาติ และข้าราชการ/ เจ้าหน้าที่ (ที่มีบัญชีผู้ใช้งาน MailGoThai) ซึ่งขั้นตอนการตรวจสอบข้อมูลเหล่านั้นจะถูกพัฒนาในลำดับต่อไป ทาง สพร. จึงไม่แนะนำให้นำข้อมูลเหล่านี้ไปใช้งานในการยืนยันตัวบุคคลจริง แต่ระบบยืนยันตัวบุคคลกลางได้รองรับการจัดเก็บข้อมูลเหล่านั้นในระบบ

8) ระบบสารสนเทศของหน่วยงานนำผู้ใช้งานไปยังหน้าลงทะเบียน/ สมัครสมาชิกเพื่อขอเข้าใช้บริการ (หน้าจอ SSORegister) พร้อมทั้งแสดงรายละเอียดต่าง ๆ ที่เรียกได้จากระบบยืนยันตัวบุคคลกลางไว้ในช่องต่าง ๆ เช่น ชื่อ ที่อยู่ อีเมล์ ฯลฯ เพื่อที่ผู้ใช้งานไม่ต้องกรอกข้อมูลต่าง ๆ ดังกล่าวอีกครั้ง
   8.1) ผู้ใช้งานระบุรายละเอียดต่าง ๆ ที่ระบบสารสนเทศของหน่วยงานนั้น ๆ ต้องการเพิ่มเติมก่อนที่จะยืนยันการสมัครสมาชิกกับระบบ
   8.2) เมื่อระบบสารสนเทศของหน่วยงานตรวจสอบข้อมูลผู้ใช้งานและบันทึกข้อมูลดังกล่าวไว้ในฐานข้อมูลของตนแล้ว ให้ไปที่ขั้นตอนที่ 9
9) ผู้ใช้งานดังกล่าวมีบัญชีผู้ใช้งานอยู่กับระบบสารสนเทศของหน่วยงานแล้ว ระบบสารสนเทศของหน่วยงานจะทำการตรวจสอบข้อมูลของผู้ใช้งาน ตรวจสอบสิทธิในการเข้าถึงข้อมูลและบริการต่าง ๆ และนำผู้ใช้งานไปยังหน้าจอหลักเพื่อเข้าใช้บริการที่ผู้ใช้ต้องการ
10) ผู้ใช้งานไม่ต้องการยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลาง (เช่น ผู้ใช้อาจจะยังไม่ได้ลงทะเบียนกับเว็บไซต์ท่า) ระบบยืนยันตัวบุคคลกลางจะเปิดหน้าเว็บราวเซอร์หน้าใหม่พร้อมนำผู้ใช้งานไปยังหน้า Login ของระบบสารสนเทศหน่วยงาน เพื่อขอเข้าใช้บริการ

สำหรับผู้ใช้งานที่ต้องการเข้าใช้บริการจากหน้าระบบสารสนเทศของหน่วยงานโดยตรง ขั้นตอนการสมัครสมาชิกและการยืนยันตัวบุคคลก็จะดำเนินการในระบบสารสนเทศของหน่วยงานนั้น ๆ ทั้งหมด โดยไม่ต้องเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางแต่อย่างใด ดังรูปที่ 8-3




รูปที่ 8-3 กระบวนการลงทะเบียนและยืนยันตัวบุคคลของระบบสารสนเทศของหน่วยงาน

# 9. มาตรฐานและเทคโนโลยีที่เกี่ยวข้อง

สพร. ได้นำโปรโตคอล (Protocols) ที่เป็นมาตรฐานเปิด (Open Standard) มาใช้ เพื่อให้เว็บไซต์ท่าและระบบสารสนเทศภาครัฐสามารถเชื่อมโยงกันได้ในรูปแบบรวมศูนย์ (Single Sign-On) โดยโปรโตคอลดังกล่าว คือ OpenID และ OAuth

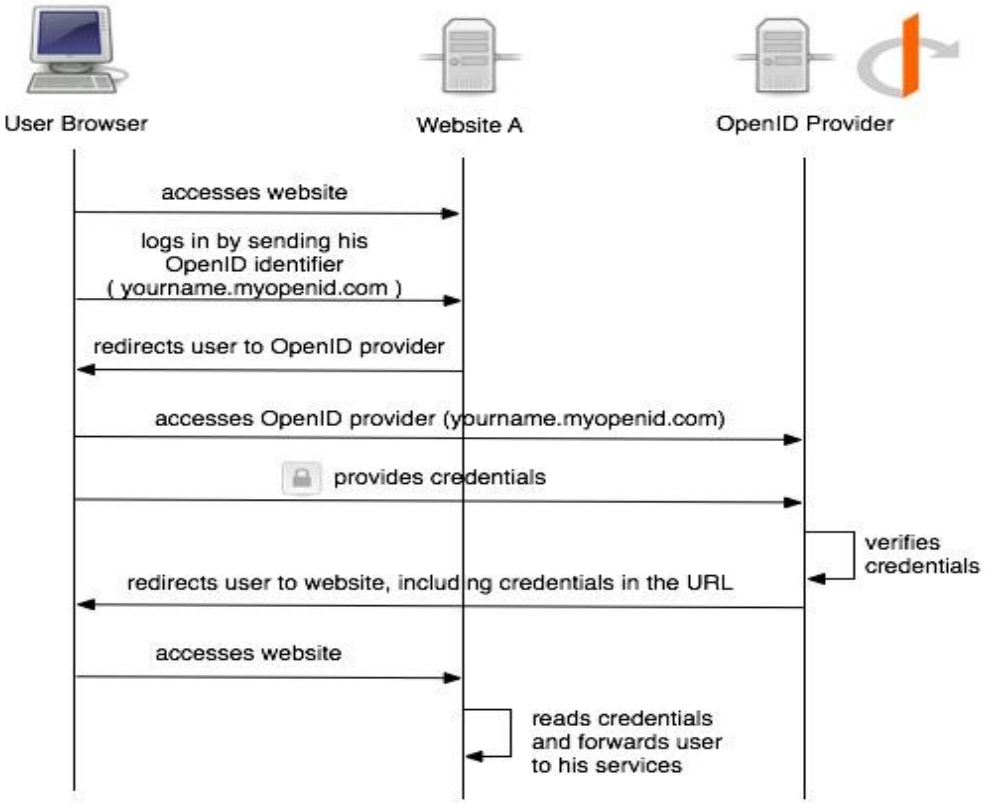
## 9.1 OpenID

เทคโนโลยี OpenID ซึ่งเป็นเทคโนโลยีที่ได้รับการนำไปใช้อย่างกว้างขวาง (พันล้านบัญชีผู้ใช้ ใช้ได้กับสี่หมื่นกว่าเว็บไซต์) กับเว็บไซต์ชั้นนำ อาทิเช่น AOL, BBC, Facebook, Google, IBM, Microsoft, MySpace, Orange, PayPal, VeriSign, Yahoo! รวมถึง US Government

นอกจากนั้น ระบบสารสนเทศภาครัฐที่ต้องการเชื่อมโยงกับระบบเว็บไซต์กลางแบบ Single Sign-On ด้วยเทคโนโลยี OpenID นี้ไม่จำเป็นที่ต้องติดตั้งโปรแกรมเพิ่มเติม แต่ต้องทำการปรับเปลี่ยนระบบสารสนเทศของหน่วยงานเพียงเล็กน้อยก็สามารถเชื่อมโยงข้อมูลกับเว็บไซต์ท่าได้ โดยเทคโนโลยีนี้จะใช้ในการยืนยันตัวบุคคล โดยระบบยืนยันตัวบุคคลกลาง (e-Authentication Service) ทำหน้าที่เป็น OpenID Provider เพื่อให้ระบบสารสนเทศภาครัฐสามารถนำข้อมูลการยืนยันตัวตนนี้ไปอนุญาตให้ผู้ใช้งานสามารถเข้าใช้ระบบได้ โดยที่ระบบสารสนเทศภาครัฐจำเป็นต้องทำการพัฒนาชุดคำสั่งหรือระบบ OpenID Relying Party สำหรับใช้ในการส่ง-รับข้อมูลจากระบบยืนยันตัวบุคคลกลาง

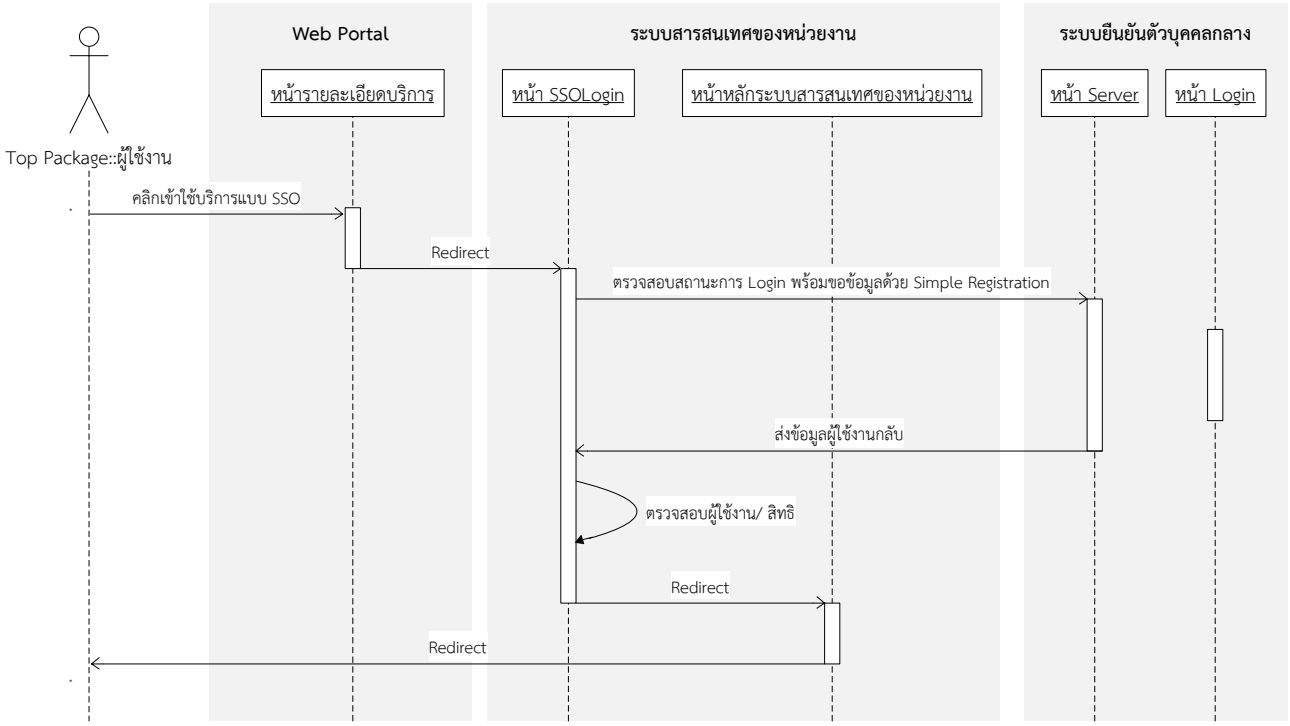
ขั้นตอนการทำงานของเทคโนโลยี OpenID มีดังนี้

1) ผู้ใช้งานต้องการยืนยืนยันตัวกับเว็บไซต์/ ระบบสารสนเทศของหน่วยงาน
2) ผู้ใช้งานระบุรหัส OpenID เพื่อเข้าใช้งานเว็บไซต์/ ระบบสารสนเทศ
3) เว็บไซต์ Redirect ผู้ใช้งานไปยัง OpenID Provider
4) ผู้ใช้งานทำการ Login ณ เว็บไซต์ของ OpenID Provider
5) OpenID Provider ทำการตรวจสอบการยืนยันตัวของผู้ใช้งาน (Username และ Password) ถ้าถูกต้อง OpenID Provider จะ Redirect ผู้ใช้งานกลับไปยังเว็บไซต์/ ระบบสารสนเทศของหน่วยงานที่ผู้ใช้งานต้องการ พร้อมทั้งส่ง Credential เพื่อยืนยันว่าผู้ใช้งานดังกล่าวว่าผ่านการยืนยันตัวบุคคลเรียบร้อยแล้ว
6) เว็บไซต์/ ระบบสารสนเทศของหน่วยงานตรวจสอบ Credential ที่ได้รับ ตรวจสอบสิทธิในการเข้าถึงข้อมูลและบริการ และ Redirect ผู้ใช้งานไปสู่หน้าจอเพื่อขอใช้บริการ




รูปที่ 9-1 ขั้นตอนการทำงานของเทคโนโลยี OpenID

เมื่อนำเทคโนโลยี OpenID มาประยุกต์กับระบบสารสนเทศของหน่วยงานเพื่อเชื่อมโยงเข้ากับเว็บไซต์ท่า (ระบบยืนยันตัวบุคคล) แบบ Single Sign-On สามารถอธิบายได้ตามแผนภาพ ดังนี้



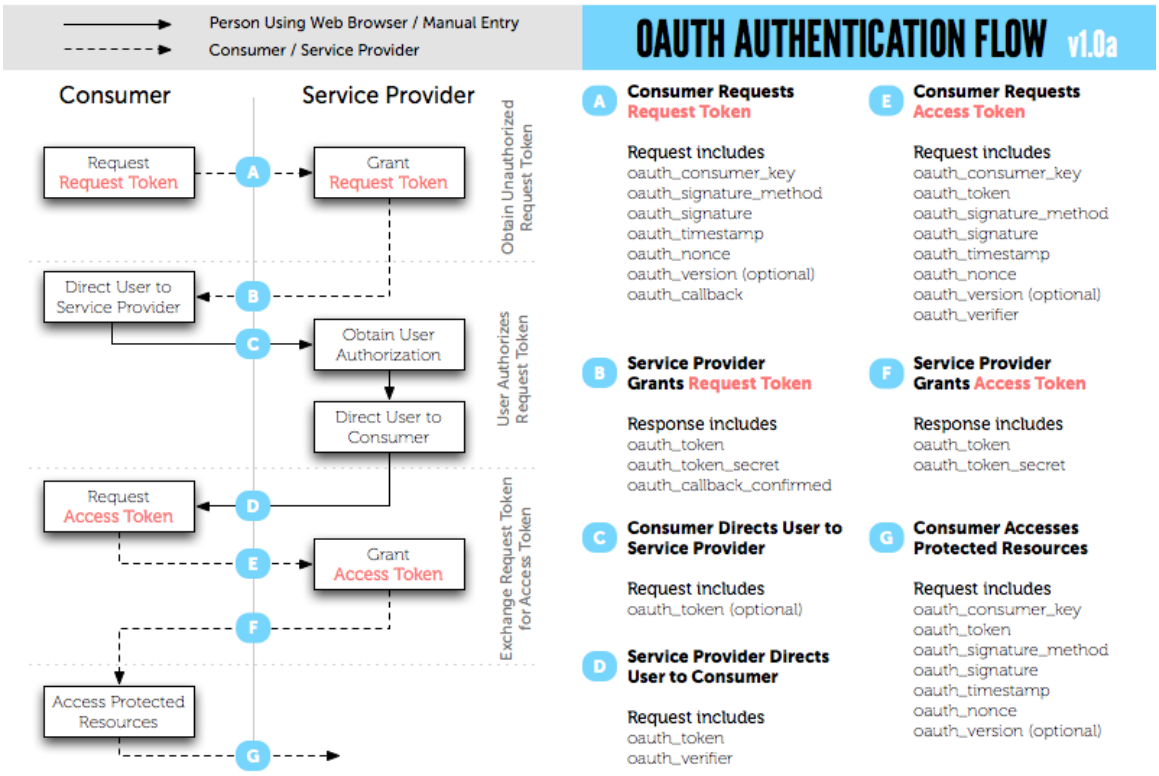

รูปที่ 9-2 แผนภาพของการสมัครสมาชิกและการเข้าใช้งานระบบ

1) เมื่อผู้ใช้งานขอใช้บริการแบบ Single Sign-On จากหน้า Registration/ Login และผ่านการยืนยันตัวบุคคลจากระบบยืนยันตัวบุคคลกลาง
2) ระบบยืนยันตัวบุคคลกลางจะ Redirect ไปยังหน้า Login ของระบบสารสนเทศของหน่วยงาน (หน้า SSOLogin)
3) ระบบสารสนเทศของหน่วยงานทำการ Request ไปยังหน้า Server.aspx ซึ่งเป็น OpenID Endpoint ของระบบยืนยันตัวบุคคลกลางผ่านทาง URL GET Request หรือ POST Request
4) ระบบยืนยันตัวบุคคลกลาง (หน้า Server.aspx) ทำการสกัดข้อมูลออกมาจาก Request ที่ส่งมายังระบบ โดยระบบจะตรวจสอบว่าปัจจุบันผู้ใช้งานได้ทำการลงชื่อเข้าใช้งาน (Login) แล้วหรือยัง ถ้าผู้ใช้งานได้มีการลงชื่อเข้าใช้แล้ว ระบบจะเรียกข้อมูลของผู้ใช้งานจากฐานข้อมูล aspnetdb และส่งข้อมูลกลับไปให้ระบบสารสนเทศของหน่วยงานตามที่ร้องขอด้วย Simple Registration Extension

## 9.2 OAuth

เทคโนโลยี OAuth เป็นเทคโนโลยีการรักษาความปลอดภัยในการแลกเปลี่ยนข้อมูลระหว่างเว็บไซต์ที่รับความนิยมในปัจจุบัน เว็บไซต์ชั้นนำ อาทิ Facebook, Twitter ต่างนำมาตรฐานนี้มาใช้ในการรักษาความปลอดภัยของข้อมูลส่วนตัวสมาชิกที่จะส่งผ่านไปยังเว็บไซต์อื่น หรือโปรแกรมประยุกต์อื่น (Application) ทั้งสิ้น โดย OAuth จะใช้ “Access Token” ในการยืนยันเพื่อแลกเปลี่ยนข้อมูลกัน โดยทั่วไปขั้นตอนการทำงานของ OAuth นั้นมีด้วยกันทั้ง 4 ขั้นตอนหลัก ดังนี้

1) ทำการขอ “Request Token” จาก Server ที่ต้องการจะแลกเปลี่ยนข้อมูลด้วย
2) ขออนุญาตใช้งาน “Request Token” โดยในส่วนนี้จะต้องรอการอนุมัติจากผู้ใช้งาน
3) เปลี่ยน “Request Token” เป็น “Access Token”
4) นำ “Access Token” ไปแลกข้อมูล




รูปที่ 9-3 แผนภาพ Activity Diagram ของ OAuth Protocol

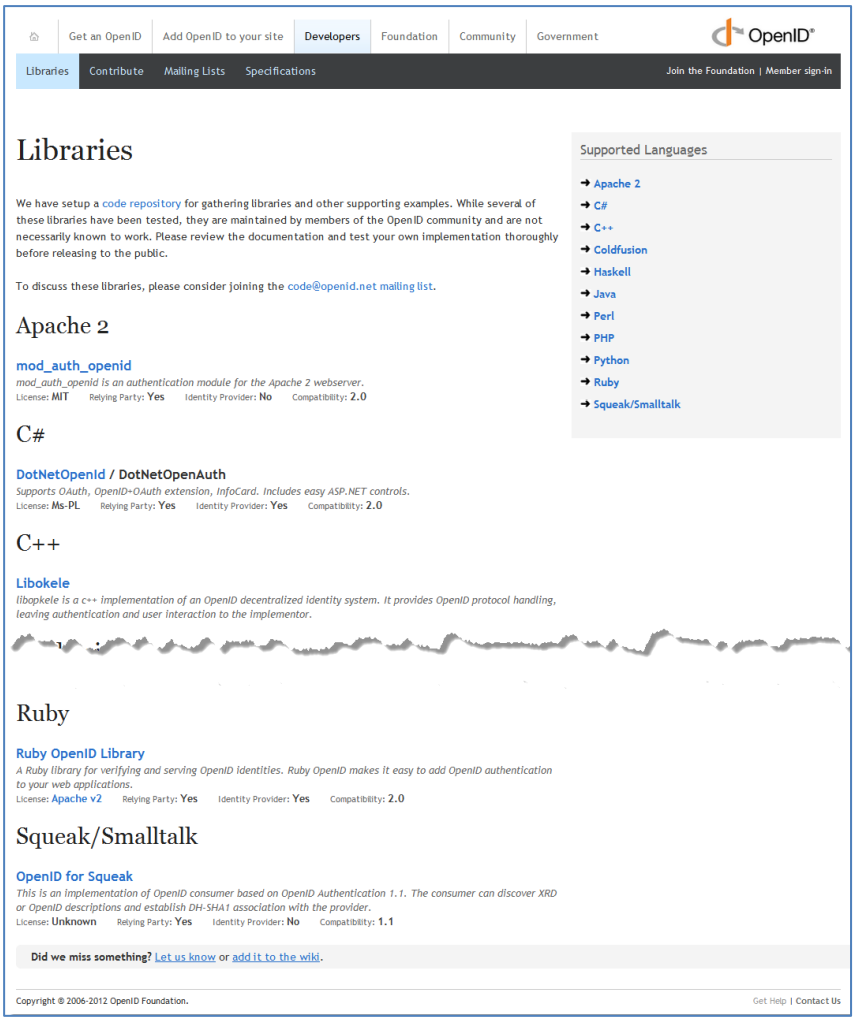
ดังนั้น สพร. จะนำเทคโนโลยีนี้มาประยุกต์ใช้ในการร้องขอข้อมูลส่วนตัวของผู้ใช้บริการ โดยระบบสารสนเทศของหน่วยงานต้องทำการพัฒนาชุดคำสั่งหรือระบบ OAuth Consumer เพื่อใช้ในการร้องขอข้อมูลผู้ใช้งานจากระบบยืนยันตัวบุคคลกลาง (ซึ่งทำหน้าที่เป็น OAuth Provider) โดยระบบจะส่งข้อมูลกลับมาในรูปแบบของ XML

# 10. การพัฒนาระบบสารสนเทศของหน่วยงานเพื่อเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางแบบ Single Sign-On

## 10.1 สำหรับผู้ใช้งานประเภทประชาชน/ นิติบุคคล/ ชาวต่างชาติ/ ข้าราชการ (เจ้าหน้าที่รัฐ) ที่ไม่มีบัญชี MailGoThai

การพัฒนาระบบสารสนเทศของหน่วยงานให้สามารถรองรับการเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางแบบ Single Sign-On สำหรับประชาชน/ นิติบุคคล/ ชาวต่างชาติ/ ข้าราชการ (เจ้าหน้าที่รัฐ) ที่ไม่มีบัญชีผู้ใช้งาน MailGoThai ผู้พัฒนาของหน่วยงานจำเป็นต้องดำเนินการ ดังนี้

1) ดาวน์โหลด Software Library เพื่อใช้ในการพัฒนาระบบสารสนเทศของหน่วยงานตนให้สามารถเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางในลักษณะ Single Sign-On ได้ โดยผู้พัฒนาสามารถดาวน์โหลด Libraries ต่าง ๆ ได้ที่ http://openid.net/developers/libraries/ ดังรูปที่ 10-1




รูปที่ 10-1 หน้า Libraries สนับสนุนภาษาต่าง ๆ

2) ผู้พัฒนาของหน่วยงานจะต้องจัดทำโปรแกรมเว็บเพจ (Web Page) ขึ้น 2 หน้า ได้แก่ หน้า SSOLogin และหน้า SSORegister เพื่อรองรับการทำ Single Sign-On

### 10.1.1 การพัฒนาหน้า SSOLogin

SSOLogin เป็นหน้าเว็บเพจสำหรับรับคำร้องขอใช้งานระบบสารสนเทศของหน่วยงานจากเว็บไซต์ท่าแบบ Single Sign-On หน้าเว็บเพจนี้จะตรวจสอบรายละเอียดผู้ใช้งาน และ Redirect ผู้ใช้งานไปยังหน้าสำหรับขอใช้บริการหรือหน้าลงทะเบียนขอใช้บริการตามความเหมาะสม โดยมีรายละเอียดกิจกรรมและการกำหนดค่าพารามิเตอร์ (Parameter) และ Url ดังนี้

| **SSOLogin** | |
|---|---|
| คำอธิบาย | 1) ทำการยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลางด้วย OpenID<br>2) ตรวจสอบข้อมูลผู้ใช้งานโดยเปรียบเทียบกับฐานข้อมูลผู้ใช้งานในระบบสารสนเทศของหน่วยงาน เพื่อใช้ในการจับคู่ (Match) ผู้ใช้งานระหว่างระบบยืนยันตัวบุคคลกลางและระบบสารสนเทศของหน่วยงาน<br>2.1) ในกรณีที่ผู้ใช้งานมีบัญชีผู้ใช้งานอยู่กับระบบสารสนเทศของหน่วยงานให้<br>- ตรวจสอบสิทธิในการเข้าถึงข้อมูลและบริการของผู้ใช้งาน<br>- เรียกดูข้อมูลเฉพาะข้อมูลเพิ่มเติมสำหรับผู้ใช้งานท่านดังกล่าว (ถ้ามี)<br>- Redirect ไปหน้าขอใช้บริการ<br>2.2) ในกรณีที่ผู้ใช้งานไม่มีบัญชีผู้ใช้งานอยู่กับระบบสารสนเทศของหน่วยงานให้ Redirect ไปยังหน้า SSORegister พร้อมส่งข้อมูลต่าง ๆ ของผู้ใช้งานที่ได้จาก Security Token ไปยังหน้าจอดังกล่าว |
| รูปแบบของโปรแกรม | Web Page |

| ประเภทผู้ใช้งาน | Url |
|---|---|
| ประชาชน | https://accounts.egov.go.th/server.aspx? |

| Parameter | Description |
|---|---|
| ***Basic OpenId parameter:*** | |
| openid.mode | - จำเป็นต้องมี<br>- เป็นการบอกให้ OpenId Provider รู้ว่า OpenId Provider นี้สามารถติดต่อกับผู้ใช้งานได้หรือไม่ โดยมีค่าดังนี้<br>o “checked_immediate” - ไม่ให้ติดต่อกับผู้ใช้งาน<br>o “checked_setup” - ให้ติดต่อกับผู้ใช้งานได้ |
| openid.ns | - จำเป็นต้องมี<br>- เป็นการบอกเวอร์ชั่นของ OpenId Request โดยระบบยืนยันตัวบุคคลกลางจะรองรับ OpenID 2.0 ดังนั้นระบบสารสนเทศของหน่วยงานจึงควรใช้ Relying Party ไลบรารี่ที่รองรับ OpenID 2.0 เช่นกัน<br>- ใส่ค่าเป็น “http://specs.openid.net/auth/2.0” |

| Parameter | Description |
|---|---|
| openid.return_to | - ไม่จำเป็นต้องมี<br>- เป็น Url ที่ระบบยืนยันตัวบุคคลกลางจะส่งผู้ใช้งานกลับมาหลังจากผู้ใช้งานทำการลงชื่อเข้าใช้ระบบแล้ว |
| openid.assoc_handle | - ไม่จำเป็นต้องมี<br>- เป็นค่าที่กำหนดขึ้นเพื่อให้ระบบยืนยันตัวบุคคลกลางใช้ในการ Sign OpenID Response<br>- ถ้าค่านี้ไม่ถูกกำหนด ระบบสารสนเทศของหน่วยงานต้องทำการตรวจสอบ OpenID Response กับระบบยืนยันตัวบุคคลกลางอีกครั้ง |
| openid.assoc_handle | - ไม่จำเป็นต้องมี<br>- เป็นค่าที่กำหนดขึ้นเพื่อให้ระบบยืนยันตัวบุคคลกลางใช้ในการ Sign OpenID Response<br>- ถ้าค่านี้ไม่ถูกกำหนด ระบบสารสนเทศของหน่วยงานต้องทำการตรวจสอบ OpenID Response กับระบบยืนยันตัวบุคคลกลางอีกครั้ง |
| openid.claimed_id | - ไม่จำเป็นต้องมี<br>- เป็น ID ของผู้ใช้งานใน OpenID<br>- ถ้าไม่ทราบให้กำหนดค่าเป็น “http://specs.openid.net/auth/2.0/identifier_select” |
| openid.identity | - ไม่จำเป็นต้องมี<br>- เป็น ID ของผู้ใช้งานใน OpenID บนระบบยืนยันตัวบุคคลกลาง<br>- ถ้าไม่ทราบให้กำหนดค่าเป็น “http://specs.openid.net/auth/2.0/identifier_select” |
| openid.realm | - ไม่จำเป็นต้องมี<br>- เป็นค่าที่แจ้งให้ระบบยืนยันตัวบุคคลกลางทราบว่า ผู้ใช้งานคนใดทำการ Login เข้าใช้งานระบบ<br>หมายเหตุ: จำเป็นต้องกำหนดค่านี้ ในกรณีที่ไม่ได้กำหนดพารามิเตอร์ “openid.return_to” ไว้ |
| ***Attribute Exchange Extension*:** | |
| openid.ns.<extension_alias> | - จำเป็นต้องมี<br>- เป็นการบอกระบบยืนยันตัวบุคคลกลางให้ส่งค่าคืนมาในรูปแบบของ Attribute Exchange<br>หมายเหตุ: ระบบยืนยันตัวบุคคลกลาง จะสนับสนุนการส่งค่าคืนในรูปแบบ Attribute Exchange เท่านั้น |

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>
- Attribute Exchange สำหรับประชาชน/ นิติบุคคล/ ชาวต่างชาติ/ เจ้าหน้าที่รัฐที่ไม่มีบัญชีผู้ใช้งาน MailGoThai มีดังนี้
<table>
<tr><th>Attribute</th><th>ค่าที่ส่งคืน</th></tr>
<tr><td>http://axschema.org/contact/email</td><td>e-mail ของผู้ใช้งาน</td></tr>
<tr><td>http://axschema.org/namePerson</td><td>ชื่อ-นามสกุล ของผู้ใช้งาน</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/uuid</td><td>UserID ของผู้ใช้งานในเว็บไซต์ท่า</td></tr>
<tr><td>http://axschema.org/namePerson/friendly</td><td>Username ของผู้ใช้งานในเว็บไซต์ท่า</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/usertype</td><td>ประเภทของผู้ใช้งานในระบบเว็บไซต์ท่า โดย<br>• ค่า Citizen คือ ประชาชน/ บุคคลธรรมดา<br>• ค่า JuristicPerson คือ นิติบุคคล<br>• ค่า Foreigner คือ ชาวต่างชาติ<br>• ค่า GovernmentAgent คือ ข้าราชการ/ เจ้าหน้าที่รัฐที่ไม่มีบัญชีผู้ใช้งาน MailGoThai</td></tr>
<tr><td colspan="2">รหัสยืนยัน โดยรูปแบบขึ้นอยู่กับประเภทของผู้ใช้งาน ดังนี้</td></tr>
<tr><td colspan="2">http://www.egov.go.th/2012/identifier/identity/...</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/identity/citizenid</td><td>บุคคลธรรมดา/ ประชาชน: เลขประจำตัวประชาชน</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/identity/juristicid</td><td>นิติบุคคล: เลขทะเบียนนิติบุคคล</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/identity/passportid</td><td>ชาวต่างชาติ: เลขที่หนังสือเดินทาง</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/identity/governmentagentid</td><td>ข้าราชการ/เจ้าหน้าที่รัฐที่ไม่มีบัญชีผู้ใช้งาน MailGoThai: เลขประจำตัวประชาชน</td></tr>
<tr><td colspan="2">ระดับการตรวจสอบข้อมูล จะขึ้นอยู่กับประเภทผู้ใช้งาน ดังนี้</td></tr>
<tr><td colspan="2">http://www.egov.go.th/2012/identifier/identityverifiedlevel/...</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/citizenidverifiedlevel</td><td>ระดับการตรวจสอบข้อมูลเลขประจำตัวประชาชน</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>http://www.egov.go.th/2012/identifier/juristicidverifiedlevel</td><td>ระดับการตรวจสอบข้อมูลเลขทะเบียนนิติบุคคล</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/passportidverifiedlevel</td><td>ระดับการตรวจสอบข้อมูลเลขที่หนังสือเดินทาง</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/governmentagentidverifiedlevel</td><td>ระดับการตรวจสอบข้อมูลเลขประจำตัวประชาชน</td></tr>
</table>
- ค่านี้ต้องถูกตั้งเป็น “http://openid.net/srv/ax/1.0”
</td></tr>
<tr><td>openid.<extension_alias>.mode</td><td>- จำเป็นต้องมี<br>- เป็นค่าบังคับ เพื่อใช้งาน Attribute Exchange<br>- ค่านี้ต้องถูกตั้งเป็น “fetch_request”</td></tr>
<tr><td>openid.<extension_alias>.required</td><td>- จำเป็นต้องมี<br>- เป็นการบอกระบบยืนยันตัวบุคคลกลางว่าระบบสารสนเทศของหน่วยงานต้องการ Attribute อะไรคืนไปบ้าง โดยทุก Attribute ต้องมีค่าจริงเพื่อให้ใช้งานได้ครอบคลุม</td></tr>
<tr><td>openid.<extension_alias>.type.<attribute_alias></td><td>- จำเป็นต้องมี<br>- เป็นการบอกระบบยืนยันตัวบุคคลกลางว่าระบบสารสนเทศของหน่วยงานมีการส่งชื่อ Attribute อะไรบ้าง โดยชื่อ “attribute_alias” ต้องถูกกำหนดใน openid.<extension_alias>.required ถ้าระบบสารสนเทศของหน่วยงานต้องการ Attribute นั้น</td></tr>
</table>

ตัวอย่าง **OpenID Request**

https://accounts.egov.go.th/server.aspx?
**openid.claimed_id**=http://specs.openid.net/auth/2.0/identifier_select&
**openid.identity**=http://specs.openid.net/auth/2.0/identifier_select&
**openid.assoc_handle**=-Ql1!IAAAAJaadvgtBHXQ-TdyBn02iJjoMBWAgL8NNW5PW_ZK-aP9QQAAAAEFwnb7DfR-3BZgeF9vax1ylJG20soMAF7fm7EtgYA_MkHK5ryfjUvMepv83Pp706BS4cgqL3z0OUkj4afAF6a5&
**openid.return_to**=http://164.115.3.16/test_rp/SSOLogin.aspx?
**dnoa.userSuppliedIdentifier**=https://accounts.egov.go.th /&
**openid.realm**=http://164.115.3.16/&
**openid.mode**=checkid_setup&
**openid.ns**=http://specs.openid.net/auth/2.0&
**openid.ns.alias3**=http://openid.net/srv/ax/1.0&
**openid.alias3.required**=alias1,alias2,alias3,alias4,alias5&
**openid.alias3.mode**=fetch_request&
**openid.alias3.type.alias1**=http://axschema.org/contact/email&
**openid.alias3.count.alias1**=1&
**openid.alias3.type.alias2**=http://axschema.org/namePerson&
**openid.alias3.count.alias2**=1&
**openid.alias3.type.alias3**=http://axschema.org/namePerson/friendly&
**openid.alias3.count.alias3**=1&
**openid.alias3.type.alias4**=http://www.egov.go.th/2012/identifier/citizenid&
**openid.alias3.count.alias4**=1&
**openid.alias3.type.alias5**=http://www.egov.go.th/2012/identifier/usertype&
**penid.alias3.count.alias5**=1

เพื่อไม่ให้ข้อมูลสูญหายเนื่องจากการส่งอักขระพิเศษผ่านทาง Url ทุก Request ที่มีการส่งไปยังระบบยืนยันตัวบุคคลกลาง และ Response ที่ระบบสารสนเทศของหน่วยงานได้คืนมา ต้องทำการเข้ารหัสตามข้อกำหนดที่ 17.13.4 ของ HTML 4.01 Specification[2]

[2] ดูรายละเอียดเพิ่มเติมได้ที่ http://www.w3.org/TR/html401/

ตัวอย่าง **OpenID Response**

http://164.115.3.16 /test_rp /SSOLogin.aspx?
**dnoa.userSuppliedIdentifier**=https://accounts.egov.go.th/&
**openid.claimed_id**=https://accounts.egov.go.th/user/TestUser&
**openid.identity**=https://accounts.egov.go.th/user/TestUser&
**openid.sig**=RC210Sdxlyj2CR6ceplaehaHwdLcOoO3AMJpJPPw4+A=&
**openid.signed**=claimed_id,identity,assoc_handle,op_endpoint,return_to,response_nonce,ns.alias3,alias3.mode,alias3.type.alias1,alias3.value.alias1,alias3.type.alias2,alias3.value.alias2,alias3.type.alias3,alias3.value.alias3,alias3.type.alias4,alias3.value.alias4,alias3.type.alias5,alias3.value.alias5&
**openid.assoc_handle**=7pLA!IAAAACYcN6bXj1kNvIgvjzZdHjEttH7UepsIvLi_rc19CyLYQQAAAAGczdFYool9z_xUyMpCGDSVuScU1jJg_t4p9oh2YTEoq0X0mIir8S9BKYZgo9NMhw-7P9QwQh3R6M473BuY4ccA&
**openid.op_endpoint**=https://accounts.egov.go.th/server.aspx&
**openid.return_to**=http://164.115.5.193/test_rp/SSOLogin.aspx?dnoa.userSuppliedIdentifier=https://testopenid.ega.or.th/&
**openid.response_nonce**=2012-05-24T04:47:03ZwHqUwFMm&
**openid.mode**=id_res&
**openid.ns**=http://specs.openid.net/auth/2.0&
**openid.ns.alias3**=http://openid.net/srv/ax/1.0&
**openid.alias3.mode**=fetch_response&
**openid.alias3.type.alias1**=http://axschema.org/contact/email&
**openid.alias3.value.alias1**=testuser@testuser.com&
**openid.alias3.type.alias2**=http://axschema.org/namePerson&
**openid.alias3.value.alias2**=ทดสอบ ทดสอบจริงๆ&
**openid.alias3.type.alias3**=http://axschema.org/namePerson/friendly&
**openid.alias3.value.alias3**=TestUser&
**openid.alias3.type.alias4**=http://www.egov.go.th/2012/identifier/citizenid&
**openid.alias3.value.alias4**=111111111111&
**openid.alias3.type.alias5**=http://www.egov.go.th/2012/identifier/usertype&
**openid.alias3.value.alias5**=1

เช่นเดียวกับกรณีของการส่ง Request เพื่อไม่ให้ข้อมูลสูญหายเนื่องจากการส่งอักขระพิเศษผ่านทาง Url ทุก Request ที่มีการส่งไปยังระบบยืนยันตัวบุคคลกลาง และ Response ที่ระบบสารสนเทศของหน่วยงานได้คืนมา ต้องทำการเข้ารหัสตามข้อกำหนดที่ 17.13.4 ของ HTML 4.01 Specification

การเตรียม Request และ Response สำหรับใช้ใน OpenID Protocol ค่อนข้างจะมีรายละเอียดซับซ้อน อาทิเช่น จะต้องมีการจัดทำ Digital Signature เพื่อป้องกันการปลอมแปลงข้อมูล ตาม Diffie-Hellman Protocol ดังนั้น หน่วยงานที่สนใจที่จะเชื่อมโยงกับระบบยืนยันตัวบุคคลแบบรวมศูนย์ (Single Sign-On) จึงควรเลือก OpenID ไลบรารี่ที่รองรับการใช้งาน OpenID 2.0 และ ตรงกับภาษาที่ใช้ในการพัฒนาระบบสารสนเทศของหน่วยงาน ซึ่งในปัจจุบันนี้มี ไลบรารี่รองรับอยู่หลายภาษาเช่น PHP, Java, Perl, Python, ASP.NET ฯลฯ โดยสามารถศึกษาข้อมูลเพิ่มเติมได้ที่ http://openid.net/developers/libraries/

### 10.1.2 การพัฒนาหน้า SSORegistrater

SSORegister หน้าเว็บเพจสำหรับรองรับการลงทะเบียน/ สมัครสมาชิกผู้ใช้งานใหม่ที่ทำ Single Sign-On มาจากเว็บไซต์ท่า (ระบบยืนยันตัวบุคคลกลาง) โดยมีรายละเอียดกิจกรรมและการกำหนดค่าพารามิเตอร์ ดังนี้

| **SSORegister** | |
|---|---|
| คำอธิบาย | 1) เรียกข้อมูลรายละเอียดผู้ใช้งานจากเว็บไซต์ท่า เช่น ชื่อ ที่อยู่ หมายเลขโทรศัพท์ ฯลฯ ด้วย OAuth<br>2) หน้าจอ SSORegister แสดงข้อมูลดังกล่าวในหน้าจอลงทะเบียนขอใช้งานระบบสารสนเทศของหน่วยงาน<br>2.1) ผู้ใช้งานสามารถตรวจสอบข้อมูลที่กรอกในแบบฟอร์มโดยอัตโนมัติ<br>2.2) ผู้ใช้งานอาจเพิ่มเติมข้อมูลอื่น ๆ ตามแต่ที่แต่ละระบบสารสนเทศของหน่วยงานนั้น ๆ ต้องการ<br>3) เมื่อผู้ใช้งานตรวจสอบข้อมูลของตนแล้ว ทำการยืนยันการขอสมัครเข้าใช้ระบบสารสนเทศของหน่วยงาน และเริ่มใช้บริการต่อไป |
| รูปแบบของโปรแกรม | Web Page |

รายละเอียดขั้นตอนการทำงานดังแผนภาพ รูปที่ 9-3 โดย Consumer เป็นผู้ที่ต้องการร้องขอข้อมูล (ในที่นี้คือ ระบบสารสนเทศของหน่วยงาน - หน้าเว็บเพจ SSORegister) และ Service Provider เป็นเจ้าของข้อมูล (ในที่นี้คือ ระบบยืนยันตัวบุคคลกลาง - หน้าเว็บเพจ XmlUserInfo.aspx) โดยมีรายละเอียดของพารามิเตอร์ในขั้นตอนต่าง ๆ ดังนี้

**A: Consumer Requests** - เป็นการขอ “Request Token” จากระบบข้อมูลผู้ใช้งาน

| Parameter | Description |
|---|---|
| oauth_consumer_key | รหัสของระบบสารสนเทศของหน่วยงาน (Service Code)<br>หมายเหตุ:<br>o ระบบสารสนเทศของหน่วยงานจำเป็นจะต้องมี Service Code หรือ “oath_consumer_key” ในเบื้องต้นหน่วยงานสามารถติดต่อเพื่อขอรับ Service Code และ Passcode นี้ได้ที่ สพร. (ดูหัวข้อ 12.1)<br>o ระบบสารสนเทศของหน่วยงานที่เชื่อมโยงกับระบบยืนยันตัวบุคคลกลาง แต่ละระบบจะมี Service Code และ Passcode ระบบละ 1 ชุด |
| oauth_signature_method | - วิธีการเข้ารหัส Request<br>- ต้องใส่ค่าเป็น “HMAC-SHA1” |
| oauth_signature | - ค่าที่ได้จากขั้นตอนการเข้ารหัสตาม oauth_signature_method<br>- โดยทั่วไปพารามิเตอร์นี้จะถูกตั้งค่าให้โดยอัตโนมัติในขั้นตอนสร้าง Request ของแต่ละไลบรารี่ |
| oauth_timestamp | เวลาที่ทำการ Request |
| oauth_nonce | เป็นชุดของตัวหนังสือภาษาอังกฤษที่ถูกสุ่มขึ้นมาให้ไม่ซ้ำกันในแต่ละ Request ของแต่ละระบบสารสนเทศภาครัฐ เพื่อตรวจสอบว่า Request นี้เป็น Request ที่ไม่เคยถูกใช้มาก่อน และป้องกันการโจมตีผ่าน HTTP |
| oauth_version | เวอร์ชั่นของ OAuth |
| oauth_callback | Url ที่จะให้ส่ง “Request Token” กลับไป |
| Scope | เป็นพารามิเตอร์ที่ สพร. กำหนดขึ้น โดยให้ใส่พารามิเตอร์นี้เข้าไปใน Request ด้วย แต่ไม่ต้องกำหนดค่าใด ๆ |

**B: Service Provider Grants** - ระบบข้อมูลผู้ใช้งานจะส่ง “Request Token” กลับไปให้ระบบสารสนเทศของหน่วยงาน

| Parameter | Description |
|---|---|
| oauth_token | “Request Token” จากระบบข้อมูลผู้ใช้งาน |
| oauth_token_secret | - เป็นค่าที่ระบบข้อมูลผู้ใช้งานส่งมาพร้อมกับ “Request Token” เพื่อใช้ในการตรวจสอบ “Request Token”<br>- ค่านี้จะไม่ซ้ำกันในแต่ละ “Request Token” |
| oauth_callback_confirmed | เป็น True ถ้าได้รับการยืนยันจากระบบข้อมูลผู้ใช้งาน |

C: **Consumer Direct User to Service Provider** - ระบบสารสนเทศของหน่วยงานส่งผู้ใช้งานไปยังระบบข้อมูลผู้ใช้งานเพื่อทำการยืนยันตัวตน (ในกรณีที่ผู้ใช้งานยังไม่ได้ทำการลงชื่อเข้าใช้กับระบบยืนยันตัวบุคคลกลาง) และให้ผู้ใช้งานตัดสินใจว่าจะอนุญาตให้ระบบสารสนเทศของหน่วยงานสามารถเข้าถึงข้อมูลของผู้ใช้งานได้หรือไม่

| Parameter | Description |
|---|---|
| oauth_token | “Request Token” จากขั้นตอน B |

D: **Service Provider Directs User to Consumer** - ระบบข้อมูลผู้ใช้งานส่งผู้ใช้งานกลับไปยังระบบสารสนเทศของหน่วยงานพร้อมทั้งพารามิเตอร์ ดังนี้

| Parameter | Description |
|---|---|
| oauth_token | “Request Token” จากขั้นตอน B (ในขั้นตอนนี้ “Request Token” ได้รับการอนุญาตให้ใช้งานได้จากระบบข้อมูลผู้ใช้งานแล้ว) |
| oauth_verifier | - เป็นค่าที่ระบบข้อมูลผู้ใช้งานส่งมาพร้อมกับ “Request Token”<br>- ค่านี้จะมีความเชื่อมโยงกับระบบสารสนเทศของหน่วยงาน โดยค่านี้จะถูกใช้ในขั้นตอน E เพื่อยืนยันว่าระบบสารสนเทศของหน่วยงานที่จะขอ “Access Token” นั้นเป็นระบบเดียวกับที่ขอ “Request Token” |

E: **Consumer Request** – ระบบสารสนเทศของหน่วยงานส่ง Request ไปยังระบบข้อมูลผู้ใช้งานเพื่อขอเปลี่ยน “Request Token” เป็น “Access Token”

| Parameter | Description |
|---|---|
| oauth_consumer_key | รหัสของระบบสารสนเทศของหน่วยงาน |
| oauth_token | “Request Token” ในขั้นตอน D |
| oauth_signature_method | - วิธีการเข้ารหัส Request<br>- ต้องใส่ค่าเป็น “HMAC-SHA1” |
| oauth_signature | - ค่าที่ได้จากขั้นตอนการเข้ารหัสตาม oauth_signature_method โดยค่าในขั้นตอนนี้จะไม่เหมือนค่าในขั้นตอน A<br>- โดยทั่วไปพารามิเตอร์นี้จะถูกตั้งค่าให้อัตโนมัติในขั้นตอนสร้าง Request ของแต่ละไลบรารี่ |
| oauth_timestamp | เวลาที่ทำการ Request |
| oauth_nonce | เป็นชุดของตัวหนังสือภาษาอังกฤษที่ถูกสุ่มขึ้นมาให้ไม่ซ้ำกันในแต่ละ Request ของแต่ละระบบสารสนเทศภาครัฐ เพื่อเอาไว้ตรวจสอบว่า Request นี้เป็น Request ที่ไม่เคยถูกใช้มาก่อน และป้องกันการโจมตีผ่าน HTTP |
| oauth_version | เวอร์ชั่นของ OAuth |
| oauth_verifier | ค่าที่ได้จากระบบข้อมูลผู้ใช้งานในขั้นตอน D |

**F: Service Provider Grants** - ระบบข้อมูลผู้ใช้งานส่ง “Access Token” ไปให้ระบบสารสนเทศของหน่วยงาน

| Parameter | Description |
|---|---|
| oauth_token | “Access Token” ที่ได้รับจากระบบข้อมูลผู้ใช้งาน |
| oauth_token_secret | - เป็นค่าที่ระบบข้อมูลผู้ใช้งานส่งมาพร้อมกับ “Access Token” เพื่อใช้ในการตรวจสอบ “Access Token”<br>- ค่านี้จะไม่ซ้ำกันในแต่ละ “Access Token” |

**G: Consumer Access Protected Resources** - ระบบสารสนเทศของหน่วยงานนำ “Access Token” ที่ได้ไปเข้าถึงข้อมูลของผู้ใช้งาน โดยทาง สพร. ได้ปรับวิธีการเข้าถึงข้อมูลเพื่อให้ได้ข้อมูลกลับมาในรูปแบบของ XML โดยผู้พัฒนาสามารถเข้าถึง XML ผ่าน Url: http://123.242.139.6/eAuthenticationService/XmlUserInfo.aspx?AccessToken={Access Token ที่ระบบบริการภาครัฐได้รับ}

** โดยที่ “Access Token” มีอายุการใช้งาน 10 นาที

เช่นเดียวกับการเข้าใช้งานระบบสารสนเทศภาครัฐแบบ Single Sign-On ด้วย OpenID Protocol การเตรียม Request และ Response สำหรับใช้ใน Oauth Protocol ค่อนข้างจะมีรายละเอียดซับซ้อน ดังนั้นหน่วยงานที่สนใจที่จะเชื่อมโยงกับระบบยนยันตัวบุคคลกลางแบบ Single Sign-On จึงควรเลือก OAuth ไลบรารี่ที่รองรับการใช้งาน “OAuth v1.0a” และ ตรงกับภาษาที่ใช้ในการพัฒนาระบบสารสนเทศของหน่วยงาน ซึ่งในปัจจุบันนี้มีไลบรารี่รองรับอยู่หลายภาษาเช่น PHP, Java, Perl, Python, ASP.NET ฯลฯ โดยสามารถศึกษาข้อมูลเพิ่มเติมได้ที่ Url: http://oauth.net/code/

ทั้งนี้ ทาง สพร. ได้บรรจุตัวอย่าง Source Code ในภาษา ASP.NET ไว้ในภาคผนวก ก. ซึ่งท่านสามารถนำไปใช้เป็นตัวอย่างในการพัฒนาต่อไป รวมทั้งสามารถศึกษารายละเอียดการพัฒนา ดาวน์โหลด Source Code/ Libraries ที่ใช้ในการพัฒนา และทดสอบระบบได้ที่ https://testopenid2.ega.or.th/ (สำหรับผู้ใช้งานประเภทประชาชน/ นิติบุคคล/ ชาวต่างชาติ/ ข้าราชการ (เจ้าหน้าที่รัฐ) ที่ไม่มีบัญชีผู้ใช้งาน MailGoThai)

## 10.2 สำหรับผู้ใช้งานประเภทข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งาน MailGoThai

การพัฒนาระบบสารสนเทศของหน่วยงานให้สามารถรองรับการเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางแบบ Single Sign-On สำหรับข้าราชการ/ เจ้าหน้าที่รัฐผู้ที่มีบัญชีผู้ใช้งาน MailGoThai ผู้พัฒนาของหน่วยงานจำเป็นต้องดำเนินการ ดังนี้

1) ดาวน์โหลด Software Library เพื่อใช้ในการพัฒนาระบบสารสนเทศของหน่วยงานตนให้สามารถเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางในลักษณะ Single Sign-On ได้ โดยผู้พัฒนาสามารถดาวน์โหลด Libraries ต่าง ๆ ได้ที่ http://openid.net/developers/libraries/
2) ผู้พัฒนาของหน่วยงานจะต้องจัดทำโปรแกรมเว็บเพจ (Web Page) ขึ้นเพียงหน้าเดียว คือ SSOLogin เพื่อรองรับการทำ Single Sign-On ผ่านระบบยืนยันตัวบุคคลกลาง

### 10.2.1 การพัฒนาหน้า SSOLogin

SSOLogin เป็นหน้าเว็บเพจสำหรับรับคำร้องขอใช้งานระบบสารสนเทศของหน่วยงานจากเว็บไซต์ท่าแบบ Single Sign-On โดยหน้าเว็บเพจนี้จะตรวจสอบรายละเอียดผู้ใช้งานที่เป็นข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งาน MailGoThai และ Redirect ผู้ใช้งานไปยังหน้าสำหรับขอใช้บริการ โดยมีรายละเอียดกิจกรรมและการกำหนดค่าพารามิเตอร์ (Parameter) และ Url ดังนี้

| SSOLogin | |
|---|---|
| คำอธิบาย | 1) ทำการยืนยันตัวบุคคลกับระบบยืนยันตัวบุคคลกลางด้วย OpenID<br>2) ตรวจสอบข้อมูลผู้ใช้งานโดยเปรียบเทียบกับฐานข้อมูลผู้ใช้งานในระบบสารสนเทศของหน่วยงาน เพื่อใช้ในการจับคู่ (Match) ผู้ใช้งานระหว่างระบบยืนยันตัวบุคคลกลางและระบบสารสนเทศของหน่วยงาน<br>2.1) ในกรณีที่ผู้ใช้งานมีบัญชีผู้ใช้งานอยู่กับระบบสารสนเทศของหน่วยงานให้<br>- ตรวจสอบสิทธิในการเข้าถึงข้อมูลและบริการของผู้ใช้งาน<br>- เรียกดูข้อมูลเฉพาะข้อมูลเพิ่มเติมสำหรับผู้ใช้งานท่านดังกล่าว (ถ้ามี)<br>- Redirect ไปหน้าขอใช้บริการ<br>2.2) ในกรณีที่ผู้ใช้งานไม่มีบัญชีผู้ใช้งานอยู่กับระบบสารสนเทศของหน่วยงาน (ดูข้อเสนอแนะ)<br>**ข้อเสนอแนะ: ในกรณีที่ผู้ใช้งานไม่มีบัญชีผู้ใช้งานอยู่กับระบบสารสนเทศของหน่วยงาน โดยผู้พัฒนาอาจเลือกพัฒนา ดังนี้<br>- ให้ทำการแจ้งผู้ใช้งานว่า ผู้ใช้งานดังกล่าวไม่มีสิทธิ์ใช้งานในระบบสารสนเทศหน่วยงานนั้น<br>- ให้ Redirect ไปยังหน้า SSORegister (ดูการพัฒนาหน้า SSORegister ได้จากหัวข้อ10.1.2) พร้อมส่งข้อมูลต่าง ๆ ของผู้ใช้งานที่ได้จาก Security Token ไปยังหน้าจอดังกล่าว โดยให้ผู้ใช้งานทำการลงทะเบียนไว้กับระบบสารสนเทศของหน่วยงาน<br>- ให้ระบบสารสนเทศหน่วยงานทำการจัดเก็บข้อมูลผู้ใช้งานที่ได้จากระบบยืนยันตัวบุคคลลงในฐานข้อมูลระบบของหน่วยงาน และยอมให้ผู้ใช้งานเข้าสู่ระบบโดยอาจจะให้สิทธิ์ในระดับ Viewer และผู้ดูแลระบบของหน่วยงานจะเป็นผู้กำหนดสิทธิ์การเข้าใช้งานให้กับผู้ใช้งานดังกล่าวนั้นในภายหลัง |

| รูปแบบของโปรแกรม | Web Page |
|---|---|
| **ประเภทผู้ใช้งาน** | **Url** |
| ข้าราชการ/ เจ้าหน้าที่รัฐ ที่มีบัญชีผู้ใช้งาน MailGoThai | https://govid.egov.go.th/server.aspx? |

| Parameter | Description |
|---|---|
| ***Basic OpenId parameter:*** | |
| openid.mode | - จำเป็นต้องมี<br>- เป็นการบอกให้ OpenId Provider รู้ว่า OpenId Provider นี้สามารถติดต่อกับผู้ใช้งานได้หรือไม่ โดยมีค่าดังนี้<br>o “checked_immediate” - ไม่ให้ติดต่อกับผู้ใช้งาน<br>o “checked_setup” - ให้ติดต่อกับผู้ใช้งานได้ |
| openid.ns | - จำเป็นต้องมี<br>- เป็นการบอกเวอร์ชั่นของ OpenId Request โดยระบบยืนยันตัวบุคคล-กลางจะรองรับ OpenID 2.0 ดังนั้นระบบสารสนเทศของหน่วยงานจึงควรใช้ Relying Party ไลบรารี่ที่รองรับ OpenID 2.0 เช่นกัน<br>- ใส่ค่าเป็น “http://specs.openid.net/auth/2.0” |
| openid.return_to | - ไม่จำเป็นต้องมี<br>- เป็น Url ที่ระบบยืนยันตัวบุคคลกลางจะส่งผู้ใช้งานกลับมาหลังจากผู้ใช้งานทำการลงชื่อเข้าใช้ระบบแล้ว |
| openid.assoc_handle | - ไม่จำเป็นต้องมี<br>- เป็นค่าที่กำหนดขึ้นเพื่อให้ระบบยืนยันตัวบุคคลกลางใช้ในการ Sign OpenID Response<br>- ถ้าค่านี้ไม่ถูกกำหนด ระบบสารสนเทศของหน่วยงานต้องทำการตรวจสอบ OpenID Response กับระบบยืนยันตัวบุคคลกลางอีกครั้ง |
| openid.assoc_handle | - ไม่จำเป็นต้องมี<br>- เป็นค่าที่กำหนดขึ้นเพื่อให้ระบบยืนยันตัวบุคคลกลางใช้ในการ Sign OpenID Response<br>- ถ้าค่านี้ไม่ถูกกำหนด ระบบสารสนเทศของหน่วยงานต้องทำการตรวจสอบ OpenID Response กับระบบยืนยันตัวบุคคลกลางอีกครั้ง |
| openid.claimed_id | - ไม่จำเป็นต้องมี<br>- เป็น ID ของผู้ใช้งานใน OpenID<br>- ถ้าไม่ทราบให้กำหนดค่าเป็น “http://specs.openid.net/auth/2.0/identifier_select” |
| openid.identity | - ไม่จำเป็นต้องมี<br>- เป็น ID ของผู้ใช้งานใน OpenID บนระบบยืนยันตัวบุคคลกลาง<br>- ถ้าไม่ทราบให้กำหนดค่าเป็น “http://specs.openid.net/auth/2.0/identifier_select” |

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>openid.realm</td><td>- ไม่จำเป็นต้องมี<br>- เป็นค่าที่แจ้งให้ระบบยืนยันตัวบุคคลกลางทราบว่า ผู้ใช้งานคนใดทำการ Login เข้าใช้งานระบบ<br>หมายเหตุ: จำเป็นต้องกำหนดค่านี้ ในกรณีที่ไม่ได้กำหนดพารามิเตอร์ “openid.return_to” ไว้</td></tr>
<tr><td colspan="2">Attribute Exchange Extension:</td></tr>
<tr><td>openid.ns.<extension_alias></td><td>- จำเป็นต้องมี<br>- เป็นการบอกระบบยืนยันตัวบุคคลกลางให้ส่งค่าคืนมาในรูปแบบของ Attribute Exchange<br>หมายเหตุ: ระบบยืนยันตัวบุคคลกลาง จะสนับสนุนการส่งค่าคืนในรูปแบบ Attribute Exchange เท่านั้น<br>- Attribute Exchange สำหรับข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งาน MailGoThai มีดังนี้
<table>
<tr><th>Attribute</th><th>ค่าที่ส่งคืน</th></tr>
<tr><td>http://axschema.org/contact/email</td><td>e-mail ของผู้ใช้งาน</td></tr>
<tr><td>http://axschema.org/namePerson</td><td>ชื่อ-นามสกุล ของผู้ใช้งาน</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/uuid</td><td>UserID ของผู้ใช้งานในระบบ mail.go.th</td></tr>
<tr><td>http://axschema.org/namePerson/friendly</td><td>Username ของผู้ใช้งานในระบบ mail.go.th</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/citizenid</td><td>เลขประจำตัวประชาชนของข้าราชการ/ เจ้าหน้าที่รัฐ</td></tr>
<tr><td>http://www.egov.go.th/2012/identifier/citizenidverifiedlevel</td><td>ระดับการตรวจสอบข้อมูลเลขประจำตัวประชาชน</td></tr>
</table>
- ค่านี้ต้องถูกตั้งเป็น “http://openid.net/srv/ax/1.0”</td></tr>
<tr><td>openid.<extension_alias>.mode</td><td>- จำเป็นต้องมี<br>- เป็นค่าบังคับ เพื่อใช้งาน Attribute Exchange<br>- ค่านี้ต้องถูกตั้งเป็น “fetch_request”</td></tr>
<tr><td>openid.<extension_alias>.required</td><td>- จำเป็นต้องมี<br>- เป็นการบอกระบบยืนยันตัวบุคคลกลางว่าระบบสารสนเทศของหน่วยงานต้องการ Attribute อะไรคืนไปบ้าง โดยทุก Attribute ต้องมีค่าจริงเพื่อให้ใช้งานได้ครอบคลุม</td></tr>
</table>

| Parameter | Description |
|---|---|
| openid.<extension_alias><br>.type.<attribute_alias> | - จำเป็นต้องมี<br>- เป็นการบอกระบบยืนยันตัวบุคคลกลางว่าระบบสารสนเทศของหน่วยงานมีการส่งชื่อ Attribute อะไรบ้าง โดยชื่อ “attribute_alias” ต้องถูกกำหนดใน openid.<extension_alias>.required ถ้าระบบสารสนเทศของหน่วยงานต้องการ Attribute นั้น |

ตัวอย่าง **OpenID Request**

https://govid.egov.go.th /server.aspx?

**openid.claimed_id**=http://specs.openid.net/auth/2.0/identifier_select&

**openid.identity**=http://specs.openid.net/auth/2.0/identifier_select&

**openid.assoc_handle**=-Ql1!IAAAAJaadvgtBHXQ-TdyBn02iJjoMBWAgL8NNW5PW_ZK-aP9QQAAAAEFwnb7DfR-3BZgeF9vax1yIjG20soMAF7fm7EtgYA_MkHK5ryfjUvMepv83Pp706BS4cgqL3z0OUkj4afAF6a5&

**openid.return_to**=http://164.115.3.16/test_rp/SSOLogin.aspx?

**dnoa.userSuppliedIdentifier**=https://govid.egov.go.th /&

**openid.realm**=http://164.115.3.16/&

**openid.mode**=checkid_setup&

**openid.ns**=http://specs.openid.net/auth/2.0&

**openid.ns.alias3**=http://openid.net/srv/ax/1.0&

**openid.alias3.required**=alias1,alias2,alias3,alias4,alias5&

**openid.alias3.mode**=fetch_request&

**openid.alias3.type.alias1**=http://axschema.org/contact/email&

**openid.alias3.count.alias1**=1&

**openid.alias3.type.alias2**=http://axschema.org/namePerson&

**openid.alias3.count.alias2**=1&

**openid.alias3.type.alias3**=http://axschema.org/namePerson/friendly&

**openid.alias3.count.alias3**=1&

**openid.alias3.type.alias4**=http://www.egov.go.th/2012/identifier/citizenid&

**openid.alias3.count.alias4**=1&

**openid.alias3.type.alias5**=http://www.egov.go.th/2012/identifier/usertype&

**penid.alias3.count.alias5**=1

เพื่อไม่ให้ข้อมูลสูญหายเนื่องจากการส่งอักขระพิเศษผ่านทาง Url ทุก Request ที่มีการส่งไปยังระบบยืนยันตัวบุคคลกลาง และ Response ที่ระบบสารสนเทศของหน่วยงานได้คืนมา ต้องทำการเข้ารหัสตามข้อกำหนดที่ 17.13.4 ของ HTML 4.01 Specification[3]

ตัวอย่าง **OpenID Response**

http://164.115.3.16 /test_rp /SSOLogin.aspx?
**dnoa.userSuppliedIdentifier**=https://govid.egov.go.th/&
**openid.claimed_id**=https://govid.egov.go.th/user.aspx/TestUser&
**openid.identity**=https://govid.egov.go.th/user.aspx/TestUser&
**openid.sig**=RC210Sdxlyj2CR6ceplaehaHwdLcOoO3AMJpJPPw4+A=&
**openid.signed**=claimed_id,identity,assoc_handle,op_endpoint,return_to,response_nonce,ns.alias3,alias3.mode,alias3.type.alias1,alias3.value.alias1,alias3.type.alias2,alias3.value.alias2,alias3.type.alias3,alias3.value.alias3,alias3.type.alias4,alias3.value.alias4,alias3.type.alias5,alias3.value.alias5&
**openid.assoc_handle**=7pLA!IAAAACYcN6bXj1kNvIgvjzZdHjEttH7UepsIvLi_rc19CyLYQQAAAAGczdFYool9z_xUyMpCGDSVuScU1jJg_t4p9oh2YTEoq0X0mIir8S9BKYZgo9NMhw-7P9QwQh3R6M473BuY4ccA&
**openid.op_endpoint**=https://govid.egov.go.th/server.aspx&
**openid.return_to**=http://164.115.5.193/test_rp/SSOLogin.aspx?dnoa.userSuppliedIdentifier=https://testopenid.ega.or.th/&
**openid.response_nonce**=2012-05-24T04:47:03ZwHqUwFMm&
**openid.mode**=id_res&
**openid.ns**=http://specs.openid.net/auth/2.0&
**openid.ns.alias3**=http://openid.net/srv/ax/1.0&
**openid.alias3.mode**=fetch_response&
**openid.alias3.type.alias1**=http://axschema.org/contact/email&
**openid.alias3.value.alias1**=testuser@testuser.com&
**openid.alias3.type.alias2**=http://axschema.org/namePerson&
**openid.alias3.value.alias2**=ทดสอบ ทดสอบจริงๆ&
**openid.alias3.type.alias3**=http://axschema.org/namePerson/friendly&
**openid.alias3.value.alias3**=TestUser&
**openid.alias3.type.alias4**=http://www.egov.go.th/2012/identifier/citizenid&

[3] ดูรายละเอียดเพิ่มเติมได้ที่ http://www.w3.org/TR/html401/

**openid.alias3.value.alias4**=111111111111&

**openid.alias3.type.alias5**=http://www.egov.go.th/2012/identifier/usertype&

**openid.alias3.value.alias5**=1

เช่นเดียวกับกรณีของการส่ง Request เพื่อไม่ให้ข้อมูลสูญหายเนื่องจากการส่งอักขระพิเศษผ่านทาง Url ทุก Request ที่มีการส่งไปยังระบบยืนยันตัวบุคคลกลาง และ Response ที่ระบบสารสนเทศของหน่วยงานได้คืนมา ต้องทำการเข้ารหัสตามข้อกำหนดที่ 17.13.4 ของ HTML 4.01 Specification

การเตรียม Request และ Response สำหรับใช้ใน OpenID Protocol ค่อนข้างจะมีรายละเอียดซับซ้อน อาทิเช่น จะต้องมีการจัดทำ Digital Signature เพื่อป้องกันการปลอมแปลงข้อมูล ตาม Diffie-Hellman Protocol ดังนั้น หน่วยงานที่สนใจที่จะเชื่อมโยงกับระบบยืนยันตัวบุคคลแบบรวมศูนย์ (Single Sign-On) จึงควรเลือก OpenID ไลบรารี่ที่รองรับการใช้งาน OpenID 2.0 และ ตรงกับภาษาที่ใช้ในการพัฒนาระบบสารสนเทศของหน่วยงาน ซึ่งในปัจจุบันนี้มี ไลบรารี่รองรับอยู่หลายภาษาเช่น PHP, Java, Perl, Python, ASP.NET ฯลฯ โดยสามารถศึกษาข้อมูลเพิ่มเติมได้ที่ http://openid.net/developers/libraries/

ทั้งนี้ ทาง สพร. ได้บรรจุตัวอย่าง Source Code ในภาษา ASP.NET ไว้ในภาคผนวก ก. ซึ่งท่านสามารถนำไปใช้เป็นตัวอย่างในการพัฒนาต่อไป รวมทั้งสามารถศึกษารายละเอียดการพัฒนา ดาวน์โหลด Source Code/ Libraries ที่ใช้ในการพัฒนา และทดสอบระบบได้ที่ https://govid.ega.or.th/ (สำหรับผู้ใช้งานประเภทข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งาน MailGoThai)

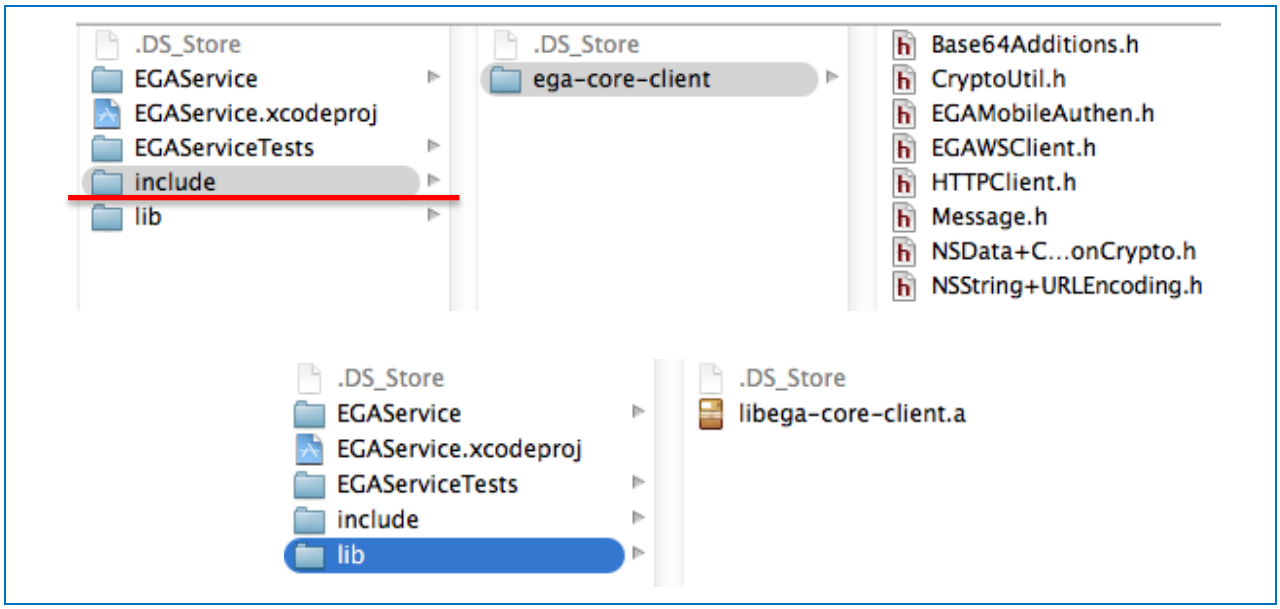
# 11. การพัฒนาโมบายแอพพลิเคชั่นให้สามารถใช้งานร่วมกับระบบยืนยันตัวบุคคลกลาง

การพัฒนาโมบายแอพพลิเคชั่นของหน่วยงานให้สามารถเชื่อมโยงกับระบบยืนยันตัวบุคคลกลางแบบ Single Sign-On เพื่อรองรับการเข้าใช้งาน (Login) และยกเลิกการเข้าใช้งาน (Logout) โมบายแอพพลิเคชั่นของหน่วยงานทั้งที่เป็นระบบปฏิบัติการ IOS และ Andriod ดังหัวข้อ 11.1 และ 11.2 ตามลำดับ รวมทั้งการเรียกขอข้อมูลบุคคลของผู้ใช้งาน (ผ่าน Government API) ในหัวข้อ 11.3

## 11.1 ระบบปฏิบัติการ IOS

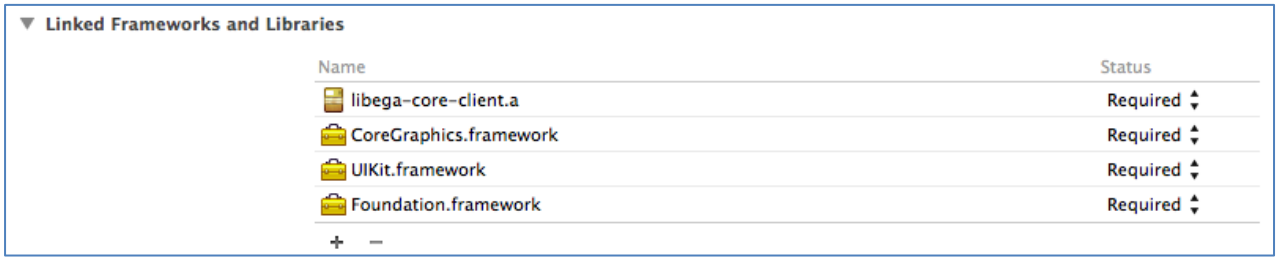
การพัฒนาโมบายแอพพลิเคชั่นในรองรับระบบปฏิบัติการ ISO ให้สามารถใช้งานร่วมกับระบบยืนยันตัวบุคคลกลางโดยวิธีการโหลดหน้า Login ของระบบยืนยันตัวบุคคลกลางด้วย UIWebView ร่วมกับการใช้ URL Schemes เพื่อให้โมบายแอพพลิเคชั่นสามารถใช้งานร่วมกับระบบยืนยันตัวบุคคลกลางได้นั้น จำเป็นต้องดำเนินการตั้งค่าแอพพลิเคชั่นและเพิ่ม Library ที่ชื่อว่า “**libega-core-client**” (ที่ สพร. จัดเตรียมไว้ให้)[4] โดยมีขั้นตอนดังนี้

1) สร้างโฟลเดอร์ใน Project แล้วนำ Header File และ Library มาไว้ ดังรูปที่ 11-1




รูปที่ 11-1 การนำ Header File และ Library มาไว้ในโฟลเดอร์ที่สร้างขึ้น

2) จาก Xcode ไปที่ **Project** >> **General** >> **Linked Frameworks and Libraries** แล้วทำการเพิ่ม Library “**libega-core-client**” ดังรูปที่ 11-2

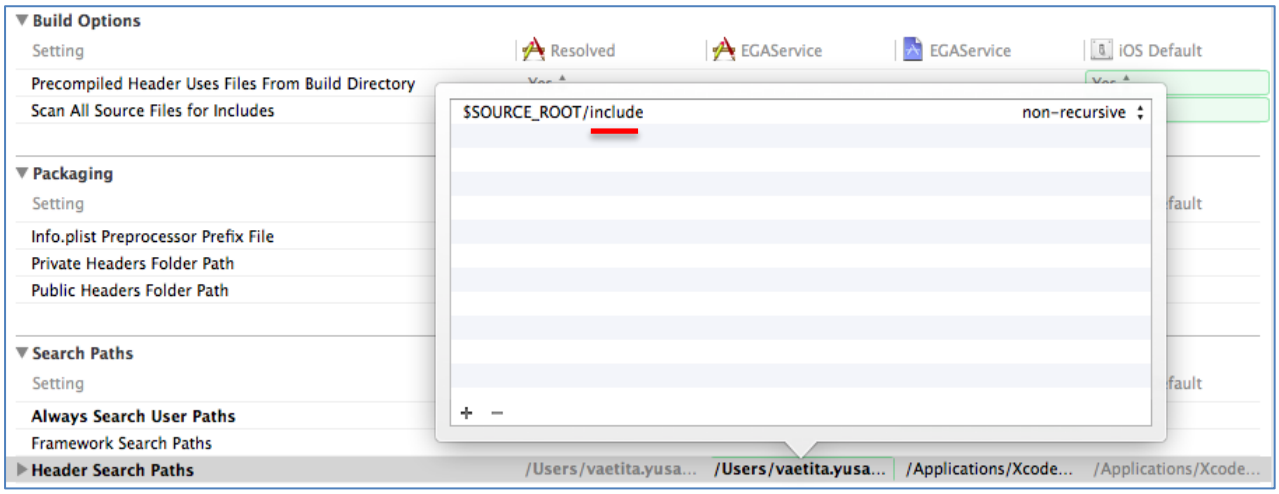



รูปที่ 11-2 การเพิ่ม Library

[4] สามารถดาวน์โหลด Library ได้ที่ **http://openid.egov.go.th/publish/DevMain.aspx**

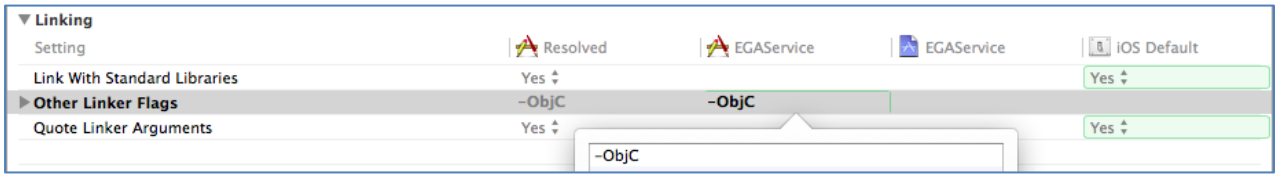
3) จาก Xcode ไปที่ **Project** >> **Build Setting** ให้ทำการตั้งค่าต่อไปนี้

3.1) Header Search Paths ใส่ชื่อ Folder ที่มี Header File ดังรูปที่ 11-3



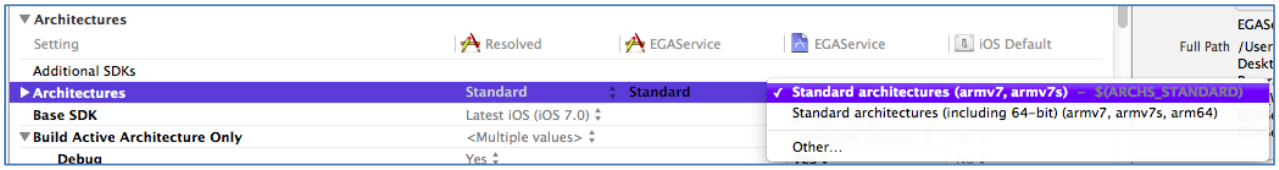

รูปที่ 11-3 ค้นหาโฟลเดอร์ที่ต้องการ

3.2) Other Linker Flags ให้มีค่าเป็น “**-ObjC**” ดังรูปที่ 11-4



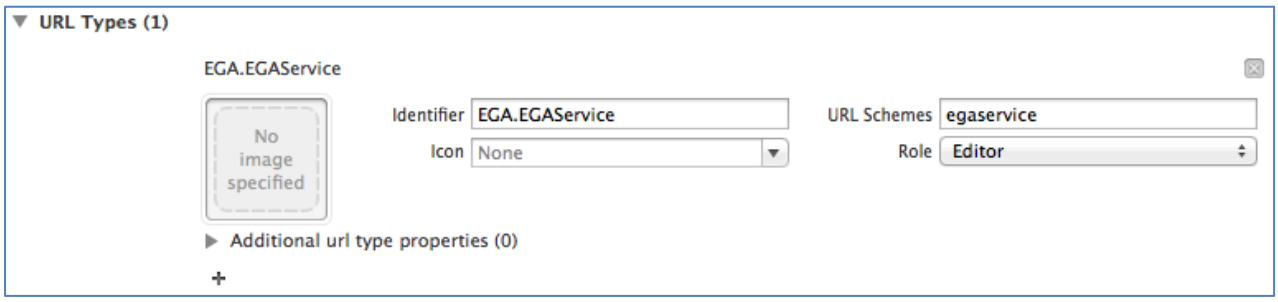

รูปที่ 11-4 ตั้งค่าให้กับ Other Linker Flags

4) เนื่องจาก Library “libega-core-client” รองรับ iDevice ที่ใช้ 32 bit ให้ทำการตั้งค่าดังนี้ จาก Xcode ไปที่ **Project** >> **Build Settings** และตั้งค่าให้ Architectures เป็น “**Standard architectures (armv7, armv7s)**” ดังรูปที่ 11-5




รูปที่ 11-5 ตั้งค่าให้กับ Architectures

5) ทำการตั้งค่าให้ AppDelegate ทำงานเมื่อมีการ Load URL โดยจาก Xcode ให้ไปที่ **Project** >> **Info** จากนั้นให้ทำการเพิ่ม URL Types และทำการกำหนดค่าให้กับ Identifier (ขึ้นอยู่กับแอพพลิเคชั่นของหน่วยงานนั้น ๆ) และ URL Schemes (ใช้กำหนด “ReturnKey” ที่ใช้สำหรับ Login และ Logout ของแอพลิเคชั่นนั้น ๆ) ดังรูปที่ 11-6




รูปที่ 11-6 ตัวอย่างการกำหนดค่าให้กับ URL Types

### 11.1.1 การ Login

โมบายแอพพลิเคชั่นจะต้องทำการเรียก Url ที่ใช้ในการ Login ด้วย WebView โดยสร้าง URL Request โดยใช้ Method “**genLoginMobileURL**” ของ EGAMobileAuthen จาก Library “**libega-core-client**” และ Parameter ดังนี้

| Parameter | Description |
|---|---|
| UrlString | Url ที่ใช้ในการ Login โดยระบุเป็น “**https://accounts.egov.go.th/MobileAuth/Authen/Login.aspx**” |
| OSID | รหัสระบบปฏิบัติการ (OS) ของแอพพลิเคชั่น สำหรับ IOS มี **OSID = 1** |
| ReturnKey | URL Schemes ที่กำหนดในส่วนของ URL Types โดยต้องระบุให้ตรงกับที่กำหนดไว้ (รูปที่ 11-6) |
| ConsumerKey | - ชุดรหัสของโมบายแอพพลิเคชั่น<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |
| ConsumerSecret | - รหัสผ่านสำหรับโมบายแอพพลิเคชั่นนั้น ๆ<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |
| LoginType | ประเภทของการ Login โดยแบ่งเป็น<br>• การ Login โดยข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai จะมี LoginType เป็น “**govid**”<br>• การ Login โดยประชาชน/ บุคคลธรรมดา นิติบุคคล ชาวต่างชาติ และข้าราชการ/ เจ้าหน้าที่รัฐที่ไม่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai จะมี LoginType เป็น “**openid**” |

```
// GOVID
//NSURL *loginURL = [EGAMobileAuthen genLoginMobileURLWithUrlString:SERVICE_URL_LOGIN_GOVID andOSID:OSID
    andReturnKey:RETURN_KEY andConsumerKey:CONSUMER_KEY andConsumerSecret:CONSUMER_SECRET andLoginType:LOGIN_GOVID];

// OpenID
NSURL *loginURL = [EGAMobileAuthen genLoginMobileURLWithUrlString:SERVICE_URL_LOGIN_GOVID andOSID:OSID andReturnKey:RETURN_KE
    andConsumerKey:CONSUMER_KEY andConsumerSecret:CONSUMER_SECRET andLoginType:LOGIN_OPENID];

NSLog(@"LoginURL: %@", [loginURL absoluteString]);

NSURLRequest *requestObj = [NSURLRequest requestWithURL:loginURL cachePolicy:NSURLRequestUseProtocolCachePolicy
    timeoutInterval:60.0];
[self.openidWebView loadRequest:requestObj];
```

เมื่อ Login จาก Webview ผ่าน จะทำให้ Method “**application:openURL:sourceApplication:annotation**” ใน AppDelegate ทำงาน โดยจะทำการถอดค่าจาก Url ให้อยู่ในรูป Key-Value โดยใช้ Method “**handleMobileAuthen**” ของ EGAMobileAuthen จาก Library “**libega-core-client**” และ Parameter ดังนี้

| Parameter | Description |
|---|---|
| ResponseUrl | URL ที่เป็น Parameter ของ Method “application:openURL:sourceApplication:annotation” |
| ReturnKey | URL Schemes ที่กำหนดในส่วนของ URL Types |
| ConsumerKey | - ชุดรหัสของโมบายแอพพลิเคชั่น<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |
| ConsumerSecret | - รหัสผ่านสำหรับโมบายแอพพลิเคชั่นนั้น ๆ<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |

```
-(BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:
    (id)annotation
{
    if (!url) {  return NO; }

    NSDictionary *dic = [EGAMobileAuthen handleMobileAuthenWithResponseUrl:url andReturnKey:RETURN_KEY andConsumerKey:CONSUMER_KE
        andConsumerSecret:CONSUMER_SECRET];
    NSArray *arrayKey = [dic allKeys];
    NSString *statusText = [dic objectForKey:@"status"];
    if ([arrayKey count] == 1) { //logout
        if ([statusText isEqualToString: @"true"]) {
            NSLog(@"Logout complete");
            self.firstView.userInfo = nil;
            [self.userView dismissViewControllerAnimated:YES completion:nil];
        } else {
            NSLog(@"Logout incomplete");
        }
    } else { // login
        if ([statusText isEqualToString: @"true"]) {
            UserInfo *userInfo = [[UserInfo alloc] init];
            userInfo.token = [dic objectForKey:@"token"];
            userInfo.email = [dic objectForKey:@"mail"];
            userInfo.citizenid = [dic objectForKey:@"identifier"];
            userInfo.fullname = [dic objectForKey:@"fullname"];

            self.firstView.userInfo = userInfo;
            [self.loginView dismissViewControllerAnimated:YES completion:nil];
        } else {
            NSString *errMsg = [dic objectForKey:@"message"];
            NSLog(@"Login Incomplete due to : %@", errMsg);
            [self.loginView dismissViewControllerAnimated:YES completion:nil];
        }

    }

    return YES;
}
```

| Key | Description |
|---|---|
| status | สถานะของการ Login ได้แก่<br>• true คือ สำเร็จ<br>• false คือ ไม่สำเร็จ |
| token | Token สำหรับเรียกใช้ Service ของ EGA Web Service |
| mail | E-mail Address ของผู้ทำการ Login |
| identifier | เลขประจำตัวประชาชนของผู้ทำการ Login |

| Key | Description |
|---|---|
| fullname | ชื่อ-นามสกุลของผู้ทำการ Login |
| message | ข้อความแสดงการ Login ไม่สำเร็จ (Login Fail)<br>หมายเหตุ: จะแสดงค่านี้ในกรณีค่าของ status เป็น “false” เท่านั้น |

### 11.1.2 การ Logout

โมบายแอพพลิเคชั่นจะต้องทำการเรียก Url ที่ใช้ในการ Logout ด้วย WebView โดยสร้าง URL Request โดยใช้ Method “**genLogoutMobileURL**” ของ EGAMobileAuthen จาก Library “**libega-core-client**” และ Parameter ดังนี้

| Parameter | Description |
|---|---|
| UrlString | URL ที่ใช้สำหรับ Logout โดยระบุเป็น “**https://accounts.egov.go.th/MobileAuth/Authen/Logout.aspx**” |
| OSID | รหัส OS ของ Application สำหรับ IOS มี **OSID = 1** |
| ReturnKey | URL Schemes ที่กำหนดในส่วนของ URL Types โดยต้องระบุให้ตรงกับที่กำหนดไว้ (รูปที่ 11-6) |

```
NSURL *logoutURL = [EGAMobileAuthen genLogoutMobileURLWithUrlString:SERVICE_URL_LOGOUT_GOVID andOSID:OSID andReturnKey:
    RETURN_KEY];
NSURLRequest *requestObj = [NSURLRequest requestWithURL:logoutURL cachePolicy:NSURLRequestUseProtocolCachePolicy
    timeoutInterval:60.0];
[self.localWebView loadRequest:requestObj];
```

เมื่อ Logout จาก Webview ผ่าน จะทำให้ Method “**application:openURL:sourceApplication:annotation**” ใน AppDelegate ทำงาน โดยจะทำการถอดค่าจาก Url ให้อยู่ในรูป Key-Value โดยใช้ Method “**handleMobileAuthen**” ของ EGAMobileAuthen จาก Library “**libega-core-client**” และ Parameter ดังต่อไปนี้

| Parameter | Description |
|---|---|
| ResponseUrl | URL ที่เป็น Parameter ของ Method “application:openURL:sourceApplication:annotation” |
| ReturnKey | URL Schemes ที่กำหนดในส่วนของ URL Types |
| ConsumerKey | - ชุดรหัสของโมบายแอพพลิเคชั่น<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |
| ConsumerSecret | - รหัสผ่านสำหรับโมบายแอพพลิเคชั่นนั้น ๆ<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |

```
-(BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:
    (id)annotation
{
    if (!url) {  return NO; }

    NSDictionary *dic = [EGAMobileAuthen handleMobileAuthenWithResponseUrl:url andReturnKey:RETURN_KEY andConsumerKey:CONSUMER_KE
        andConsumerSecret:CONSUMER_SECRET];
    NSArray *arrayKey = [dic allKeys];
    NSString *statusText = [dic objectForKey:@"status"];
    if ([arrayKey count] == 1) { //logout
        if ([statusText isEqualToString: @"true"]) {
            NSLog(@"Logout complete");
            self.firstView.userInfo = nil;
            [self.userView dismissViewControllerAnimated:YES completion:nil];
        } else {
            NSLog(@"Logout incomplete");
        }
    } else { // login
        if ([statusText isEqualToString: @"true"]) {
            UserInfo *userInfo = [[UserInfo alloc] init];
            userInfo.token = [dic objectForKey:@"token"];
            userInfo.email = [dic objectForKey:@"mail"];
            userInfo.citizenid = [dic objectForKey:@"identifier"];
            userInfo.fullname = [dic objectForKey:@"fullname"];

            self.firstView.userInfo = userInfo;
            [self.loginView dismissViewControllerAnimated:YES completion:nil];
        } else {
            NSString *errMsg = [dic objectForKey:@"message"];
            NSLog(@"Login Incomplete due to : %@", errMsg);
            [self.loginView dismissViewControllerAnimated:YES completion:nil];
        }

    }

    return YES;
}
```

| Key | Description |
|---|---|
| status | สถานะของการ Logout ได้แก่<br>• true คือ สำเร็จ<br>• false คือ ไม่สำเร็จ |
| message | ข้อความแสดงการ Login ไม่สำเร็จ (Login Fail)<br>หมายเหตุ: จะแสดงค่านี้ในกรณีค่าของ status เป็น “false” เท่านั้น |

## 11.2 ระบบปฏิบัติการ Andriod

การพัฒนาโมบายแอพพลิเคชั่นในรองรับระบบปฏิบัติการ Andriod ให้สามารถใช้งานระบบยืนยันตัวบุคคลกลางนั้น ผู้พัฒนาของหน่วยงานจำเป็นต้องดำเนินการเพิ่ม Library ที่ชื่อว่า “**ega_crypto_lib**” (ที่ สพร. จัดเตรียมไว้ให้)[5] ไว้ใน Project

### 11.2.1 การ Login

เมื่อดำเนินการเพิ่ม Library เรียบร้อยแล้ว จากนั้นจะทำการเรียก Url ที่ใช้ในการ Login ด้วย WebView โดยสร้าง URL Request โดยใช้ Method จาก Library “**ega_crypto_lib**” และ Parameter ดังนี้

| Parameter | Description |
|---|---|
| LOGIN_URL | Url ที่ใช้ในการ Login โดยระบุเป็น “**https://accounts.egov.go.th/MobileAuth/Authen/Login.aspx**” |
| OSID | รหัสระบบปฏิบัติการ (OS) ของแอพพลิเคชั่น สำหรับ Andriod มี **OSID = 2** |
| returnkey | URL Schemes ที่รับค่ากลับมา (โดยส่วนใหญ่จะระบุเป็นชื่อแอพพลิเคชั่นที่หน่วยงานนั้น ๆ กำหนดไว้) |
| type | ประเภทของการ Login โดยแบ่งเป็น<br>• การ Login โดยข้าราชการ/ เจ้าหน้าที่รัฐที่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai จะมี LoginType เป็น “**govid**”<br>• การ Login โดยประชาชน/ บุคคลธรรมดา นิติบุคคล ชาวต่างชาติ และข้าราชการ/ เจ้าหน้าที่รัฐที่ไม่มีบัญชีผู้ใช้งานอยู่กับระบบ MailGoThai จะมี LoginType เป็น “**openid**” |
| Consumerkey | - ชุดรหัสของโมบายแอพพลิเคชั่น<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |
| Consumersecret | - รหัสผ่านสำหรับโมบายแอพพลิเคชั่นนั้น ๆ<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน Service กับ สพร. (ดูหัวข้อ 12) |
| nonce | Timestamp |

[5] สามารถดาวน์โหลด Library ได้ที่ https://accounts.egov.go.th/Home/Developer

หมายเหตุ:

ในการเรียก Url นั้น (Url Request) จะต้องมีการเข้ารหัส (Encode) ข้อมูลด้วย Method จาก Library เสียก่อน กล่าวคือ

- ทำการเรียก Method “**s7 = SecretUtil.encodeSecret(consumersecret)**” เพื่อทำการเข้ารหัส “consunersecret” และนำค่าที่ได้มาเก็บไว้ในตัวแปรที่ชื่อว่า “s7”
- Method “**enSecret = CryptoUtil.encrypt(consumersecret, s7, Long.toString(nonce))**” จะนำ “s7” กับ “nonce” ไปเข้ารหัสกับ “consumersecret” อีกครั้ง และนำไปเก็บไว้ที่ตัวแปร “enSecret”

```
package or.th.ega.demogovid;

import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URLDecoder;
import java.net.URLEncoder;
import java.security.GeneralSecurityException;
import java.security.NoSuchAlgorithmException;
import java.util.HashMap;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.conn.scheme.Scheme;
import org.apache.http.conn.scheme.SchemeRegistry;
import org.apache.http.conn.ssl.SSLSocketFactory;
import org.apache.http.conn.ssl.X509HostnameVerifier;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.impl.conn.SingleClientConnManager;
import org.apache.http.util.EntityUtils;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import th.or.ega.common.utils.CryptoUtil;
import th.or.ega.common.utils.SecretUtil;

import android.net.http.SslError;
import android.os.Bundle;
import android.os.StrictMode;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.Dialog;
import android.graphics.Bitmap;
import android.text.TextUtils;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.View.OnClickListener;
import android.webkit.CookieManager;
import android.webkit.SslErrorHandler;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
```

```
@SuppressLint("SetJavaScriptEnabled")
public class LoginActivity extends Activity {

    WebView webview;
    TextView txt_response;
    private final String consumerKey = "xxxxxxxxxxxxxx";
    private final String consumerSecret = "xxxxxxxxxx";
    private String token = "";
    private final long nonce = System.currentTimeMillis();
    private final String LOGIN_URL =
"https://accounts.egov.go.th/MobileAuth/Authen/Login.aspx?OSID=2&returnkey=xxxxxx&type
=openid";


    private String init(String _url){
        try {
            String s7 = SecretUtil.encodeSecret(consumerSecret);
            String enSecret = Base64.encodeToString(CryptoUtil.encrypt(consumerSecret,
s7, Long.toString(nonce)),android.util.Base64.DEFAULT);


            return LOGIN_URL.concat("&consumerkey=" + URLEncoder.encode(consumerKey)
                    + "&consumersecret=" + URLEncoder.encode(enSecret)
                    + "&nonce=" + nonce);

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        } catch (GeneralSecurityException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        webview = (WebView)findViewById(R.id.webview);
        txt_response = (TextView)findViewById(R.id.txt_response);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.setInitialScale(100);

        webview.setWebViewClient(new WebViewClient() {
            /**
             *
             */
            @Override
            public void onPageStarted(WebView view, String url, Bitmap favicon) {
                HashMap<String, String> para = new HashMap<String, String>();
                String[] splitUrl = url.split("\\?");
                Log.e("url", url);
                Log.e("splitUrl[0]", splitUrl[0]);

                if ("http://mobile.ega.or.th/oscc".equals(splitUrl[0].trim())) {
                    String[] param = splitUrl[1].split("\\&");
                    for (int i = 0; i < param.length; i++) {
                        String[] p = param[i].split("\\=");
                        if (p.length == 1)
                            Log.e("param0", p[0].trim());
                            para.put(p[0].trim(), null);
                        if (p.length == 2)
                            Log.e("param0", p[0].trim());
                            Log.e("param1", p[1].trim());
                            para.put(p[0].trim(), p[1].trim());
                    }
```

```
                    if ("true".equals(para.get("status").toString())) {
                        webview.stopLoading();
                        /**
                         *  if parameter 'status' return true = Login success
                         */
                        Log.e("Login", "SUCCESS");
                        Toast.makeText(LoginActivity.this,"LOGIN SUCCESS",
Toast.LENGTH_LONG).show();

                        /**
                         * decrypt token
                         */
                        try {
                            String s7 = SecretUtil.encodeSecret(consumerSecret);
                            token =
CryptoUtil.decrypt(android.util.Base64.decode(URLDecoder.decode(para.get("token")),and
roid.util.Base64.DEFAULT), s7, para.get("nonce"));
                        } catch (GeneralSecurityException e) {
                            e.printStackTrace();
                        } catch (IOException e) {
                            e.printStackTrace();
                        }

                        // custom dialog
                        final Dialog dialog = new Dialog(LoginActivity.this);
                        dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
                        dialog.setContentView(R.layout.dialog_req_quota);
                        dialog.setTitle("กรุณากรอกเลขบัตรประชาชน");

                        final EditText txt_id = (EditText)
dialog.findViewById(R.id.txt_id);
                        Button btn_ok = (Button) dialog.findViewById(R.id.btn_ok);
                        Button btn_cancel = (Button)
dialog.findViewById(R.id.btn_cancel);

                        btn_ok.setOnClickListener(new OnClickListener() {
                            @Override
                            public void onClick(View v) {
                                dialog.dismiss();
                                String citizenID = txt_id.getText().toString();
                                JSONArray resText = wsGetProfile(token, citizenID,
consumerKey);
                                txt_response.setVisibility(View.VISIBLE);
                                txt_response.setText(resText.toString());
                            }
                        });
                        btn_cancel.setOnClickListener(new OnClickListener() {
                            @Override
                            public void onClick(View v) {
                                // show receiver
                                dialog.dismiss();
                                finish();
                            }
                        });
                        dialog.show();

                    } else if ("false".equals(para.get("status").toString())) {
                        webview.stopLoading();
                        /**
                         * if parameter 'status' return false = Login fail.
                         * and clear cookie in webview
                         */
                        CookieManager cookieManager = CookieManager.getInstance();
                        cookieManager.removeAllCookie();
                        Log.e("Login", "FAIL");
                        Toast.makeText(LoginActivity.this,"LOGIN FAIL",
Toast.LENGTH_LONG).show();
                    }
                }
```

```
            }

            // default method webview component
            @Override
            public boolean shouldOverrideUrlLoading(WebView view, String url) {
                view.loadUrl(url);
                return true;
            }
            // default method webview component
            @Override
            public void onReceivedSslError(WebView view, SslErrorHandler handler,
SslError error) {
                handler.proceed();
            }
        });
        webview.loadUrl(init(LOGIN_URL));
    }

    public JSONArray wsGetProfile(String _token, String _citizenID, String
_consumerKey){
        URI uri = null;
        try {
            uri = new URI("https", "ws.ega.or.th",
"/ws/dopa/personal/profile/normal","CitizenID="+_citizenID, null);
        } catch (URISyntaxException e) {
            e.printStackTrace();
        }

        String wsUrl = uri.toASCIIString();
        Log.e("wsUrl", wsUrl);
        JSONArray result = wsGetJSON(wsUrl, _consumerKey, _token);

        return result;
    }

    private JSONArray wsGetJSON(String _wsUrl, String _consumerKey, String _token) {
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        // Create a new HttpClient  and Header
        DefaultHttpClient httpclient = null;
        if (_wsUrl.startsWith("https")) {
            httpclient = certificateAcceptor();
        }else if (_wsUrl.startsWith("http")) {
            httpclient = new DefaultHttpClient();
        }

        HttpGet httpget = new HttpGet(_wsUrl);
        HttpResponse response = null;
        StringBuilder respText = null;
        JSONArray respJSON = null;
        try {
            httpget.setHeader("Content-type", "application/json");

            if (!TextUtils.isEmpty(_consumerKey)) {
                httpget.setHeader("Consumer-Key", _consumerKey);
            }
            if (!TextUtils.isEmpty(_token)) {
                httpget.setHeader("Token", _token);
            }
            // Execute HTTP Post Request
            response = httpclient.execute(httpget);
            HttpEntity entity = response.getEntity();
            respText = new StringBuilder(EntityUtils.toString(entity));
            // Parse String to JSON
            try {
                if (respText.toString().startsWith("[")) {
                    respJSON = new JSONArray(respText.toString());
                }else if (respText.toString().startsWith("{")) {
```

```
                    respJSON = new JSONArray().put(new
JSONObject(respText.toString()));
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            entity.consumeContent();
        } catch (IOException e) {
            e.printStackTrace();
        }finally{
            httpget.abort();
            respText = null;
        }
        return respJSON;
    }

    // accept all certificates.
    public DefaultHttpClient certificateAcceptor() {

        HostnameVerifier hostnameVerifier =
org.apache.http.conn.ssl.SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER;

        DefaultHttpClient client = new DefaultHttpClient();

        SchemeRegistry registry = new SchemeRegistry();
        SSLSocketFactory socketFactory = SSLSocketFactory.getSocketFactory();
        socketFactory.setHostnameVerifier((X509HostnameVerifier) hostnameVerifier);
        registry.register(new Scheme("https", socketFactory, 443));
        SingleClientConnManager mgr = new SingleClientConnManager(client.getParams(),
registry);
        DefaultHttpClient httpclient = new DefaultHttpClient(mgr,client.getParams());

        // Set verifier
        HttpsURLConnection.setDefaultHostnameVerifier(hostnameVerifier);
        return httpclient;
    }

}
```

เมื่อ Login จาก Webview ผ่านจะมี Response กลับมา ดังนี้

```
http://mobile.ega.or.th/[returnkey]?status=true&mail=x.xxxxx@ega.or.th&fullname=%e0%b8
%ad%e0%b8%a3%e0%b8%a3%e0%b8%96%e0%b8%9e%e0%b8%a5%20%e0%b8%9a%e0%b8%a3%e0%b8%a3%e0%b8%a
5%e0%b8%b7%e0%b8%ad&token=KmcUhzQ9JSKFyxVrjLUDB3T3zVMlBcBRwVtXdiRL6cX%2b50KqVjWIJ4uTVL
Lb%2bfgD&nonce=1393928493237&identifier=aKVt1JfEqMcsKuxPiDYcRg%3d%3d
```

| Parameter | Description |
|---|---|
| returnkey | URL Schemes ที่รับค่ากลับมา (โดยส่วนใหญ่จะระบุเป็นชื่อแอพพลิเคชั่นที่หน่วยงานนั้น ๆ กำหนดไว้) |
| status | สถานะของการ Login ได้แก่<br>• true คือ สำเร็จ<br>• false คือ ไม่สำเร็จ |
| mail | E-mail Address ของผู้ทำการ Login |
| fullname | ชื่อ-นามสกุลของผู้ทำการ Login |
| token | Token สำหรับเรียกใช้ Service ของ EGA Web Service |
| nonce | Timestamp |
| identifier | เลขประจำตัวประชาชนของผู้ทำการ Login |

### 11.2.2 การ Logout

โมบายแอพพลิเคชั่นจะต้องทำการเรียก Url ที่ใช้ในการ Logout ด้วย WebView โดยสร้าง URL Request โดยใช้ Parameter ดังนี้

| Parameter | Description |
|---|---|
| LOGOUT_URL | Url ที่ใช้ในการ Login โดยระบุเป็น “**https://accounts.egov.go.th/MobileAuth/Authen/Logout.aspx**” |
| OSID | รหัสระบบปฏิบัติการ (OS) ของแอพพลิเคชั่น สำหรับ Andriod มี **OSID = 2** |
| returnkey | URL Schemes ที่รับค่ากลับมา (โดยส่วนใหญ่จะระบุเป็นชื่อแอพพลิเคชั่นที่หน่วยงานนั้น ๆ กำหนดไว้) |

```
package or.th.ega.demogovid;

import java.util.HashMap;

import android.net.http.SslError;
import android.os.Bundle;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.ProgressDialog;
import android.graphics.Bitmap;
import android.util.Log;
import android.webkit.SslErrorHandler;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Toast;

@SuppressLint("SetJavaScriptEnabled")
public class LogoutActivity extends Activity {

    WebView webview;
    private final String LOGOUT_URL =
"https://accounts.egov.go.th/MobileAuth/Authen/Logout.aspx?OSID=2&returnkey=xxxxxx";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        webview = (WebView)findViewById(R.id.webview);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.setInitialScale(100);

        webview.setWebViewClient(new WebViewClient() {
            @Override
            public void onPageStarted(WebView view, String url, Bitmap favicon) {
                super.onPageStarted(view, url, favicon);
                HashMap<String, String> para = new HashMap<String, String>();
                String[] splitUrl = url.split("\\?");
                Log.e("url", url);
                Log.e("splitUrl[0]", splitUrl[0]);

                if ("http://mobile.ega.or.th/oscc".equals(splitUrl[0].trim())) {
                    String[] param = splitUrl[1].split("\\&");
                    for (int i = 0; i < param.length; i++) {
                        String[] p = param[i].split("\\=");
                        if (p.length == 1)
                            para.put(p[0].trim(), null);
                        if (p.length == 2)
```

```
                        para.put(p[0].trim(), p[1].trim());
                    }

                    if ("true".equals(para.get("status").toString())) {
                        webview.stopLoading();
                        /**
                         *  if parameter 'status' return true = Logout success
                         */
                        Toast.makeText(LogoutActivity.this,"LOGOUT SUCCESS",
Toast.LENGTH_LONG).show();

                    } else if ("false".equals(para.get("status").toString())) {
                        webview.stopLoading();
                        /**
                         *  if parameter 'status' return true = Logout fail
                         */
                        Toast.makeText(LogoutActivity.this,"LOGOUT FAIL",
Toast.LENGTH_LONG).show();
                    }
                }
            }

            @Override
            public boolean shouldOverrideUrlLoading(WebView view, String url) {
                view.loadUrl(url);
                return true;
            }

            @Override
            public void onReceivedSslError(WebView view,
                    SslErrorHandler handler, SslError error) {
                handler.proceed();
            }
        });
        webview.loadUrl(LOGOUT_URL);
    }
}
```

และ Response หลังจาก Request เป็นดังนี้

```
http://mobile.ega.or.th/[returnkey]?status=[status]
```

| Key-value | Description |
|---|---|
| returnkey | URL Schemes ที่รับค่ากลับมา (โดยส่วนใหญ่จะระบุเป็นชื่อแอพพลิเคชั่นที่หน่วยงานนั้น ๆ กำหนดไว้) |
| status | สถานะของการ Logout ได้แก่<br>• true คือ สำเร็จ<br>• false คือ ไม่สำเร็จ |

### 11.3 การเรียกขอข้อมูลบุคคลของผู้ใช้งาน

โมบายแอพพลิเคชั่นของหน่วยงานสามารถดำเนินการเรียกขอข้อมูลบุคคลของผู้ใช้งานนั้น ๆ ในรูปแบบ API ได้ โดยมีขั้นตอนการเรียกใช้งานข้อมูลบุคคลผ่าน Government API ดังนี้

1) ทำการ Validate

1.1) ส่วนของ Consumer Key และ Consumer Secret เพื่อขอ Token และเปิด Session ในการเรียกใช้ Service โดยการ HTTP Request โดยวิธี GET

| วิธีการ | Url |
|---|---|
| GET | https://ws.ega.or.th/ws/auth/validate?ConsumerSecret=<ConsumerSecret>&AgentID=<AgentID> |

| Key/ Parameter | Required | Type | Description |
|---|---|---|---|
| ConsumerSecret | Required | String | ชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน API |
| AgentID | Required | String | • เลขบัตรประชาชนของผู้ที่เข้ามาดูข้อมูลในระบบ<br>• เป็นค่าที่สามารถบ่งบอกได้ว่า User ใดที่เข้ามาใช้งาน Service ใน Session นั้น ๆ<br>• เป็นค่า Unique ID ของแต่ละ Session โดยทาง สพร. กำหนดไว้ให้ |

1.2) ส่วนของ HTTP HEADER

| Key | Description |
|---|---|
| Consumer-Key | ชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน API |

หากการทำงานไม่มีข้อผิดพลาด Service จะทำการตอบกลับด้วย http status code: 200 OK และ Token ที่สามารถนำไปใช้ประกอบการพัฒนา API ในการเรียกขอข้อมูลต่าง ๆ ดังนี้

```
200 OK

{
    "Result": "<TOKEN_KEY>"
}
```

| Key | Description |
|---|---|
| TOKEN_KEY | Token ที่ระบบของ สพร. สร้างให้และส่งกลับมา เพื่อนำไปใช้ในการพัฒนา API เรียกใช้ข้อมูล |

โดย Status Code ที่ระบบจะ แจ้งกลับ (Response) มายังต้นทางทราบ มีดังนี้

| Status Code | Status Phrase | Description |
|---|---|---|
| 200 | OK | Service ทำการคืนค่าได้เป็นปกติ |
| 204 | No Content | ไม่พบข้อมูลที่ต้องการ |
| 400 | Bad Request | มีการกำหนด Parameter มาไม่ครบหรือไม่ถูกต้อง |
| 401 | Unauthorized | ยังไม่ได้ทำการ Validate Consumer Key |
| 403 | Forbidden | ไม่ได้รับอนุญาตให้ใช้งาน Service ที่กำลังเรียกใช้ |
| 404 | Not Found | ไม่พบ Service ที่เรียกใช้ |
| 405 | Method Not Allowed | เรียกใช้งาน Service ผ่าน http Method ที่ไม่ถูกต้อง |
| 500 | Internal Server Error | มีข้อผิดพลาดเกิดขึ้นขณะที่กำลังทำงาน |
| 503 | Service Unavailable | Service ปลายทางไม่สามารถให้บริการได้ |

2) ทำการเรียก Request โดยวิธี GET และต้องทำการฝัง HEADER ดังนี้

| Parameter | Value |
|---|---|
| Content-Type | application/x-www-form-urlencoded |
| Consumer-Key | ชุดรหัสที่ สพร. ออกให้ |
| Token | Token ที่ได้รับกลับมาจาก สพร. ในการ Validate ข้างต้น<br>หมายเหตุ: ในการนำ Token มาใช้เรียกขอข้อมูลผ่าน Government API นั้น จำเป็นจะต้องทำการถอดรหัสก่อนนำไปใช้ |

ตัวอย่างการถอดรหัส Token

```
/**
* decrypt token
 */
String s7 = SecretUtil.encodeSecret(consumerSecret);
token =
CryptoUtil.decrypt(android.util.Base64.decode(URLDecoder.decode(para.get("token")),and
roid.util.Base64.DEFAULT), s7, para.get("nonce"));
```

และ Request ขอข้อมูลบุคคลของผู้ใช้งานนั้น ๆ ในรูปแบบ API นั้นจะต้องทำการเรียกผ่าน Url และ Parameter ดังนี้

| วิธีการ | Url |
|---|---|
| GET | https://ws.ega.or.th/ws/dopa/personal/profile/normal?CitizenID=<CitizenID> |

| Parameter | Required | Type | Description |
|---|---|---|---|
| CitizenID | Required | String | หมายเลขบัตรประชาชน 13 หลัก ของผู้ใช้งาน |

หลังจากที่ส่ง Request ข้างต้นไปเรียบร้อยแล้ว ระบบจะทำการส่ง Response กลับมา ดังตัวอย่าง

หมายเหตุ:

การเรียกขอข้อมูลส่วนบุคคลผู้ใช้งานตาม Url ข้างต้น ระบบจะทำการตอบกลับข้อมูล (Response) อันได้แก่

- หมายเลขบัตรประชาชน 13 หลัก
- คำนำหน้าชื่อ (ภาษาไทย)
- ชื่อจริง-ชื่อกลาง-นามสกุล (ภาษาไทย)
- เพศ
- วัน/เดือน/ปี พ.ศ. เกิด
- อายุ
- สถานะการมีชีวิต
- ข้อมูลที่อยู่
- ข้อมูลทำบัตรประชาชน (เช่น วัน/เดือน/ปี พ.ศ. ที่ทำบัตร วัน/เดือน/ปี พ.ศ. ที่บัตรหมดอายุ สถานที่ทำบัตร หน่วยงานที่ทำบัตร)
- ข้อมูลสัญชาติ (เช่น สัญชาติปัจจุบัน สัญชาติเดิม วัน/เดือน/ปี พ.ศ. ที่เปลี่ยนสัญชาติ)

ตัวอย่าง

[{"Address_Moo":"","IssueDate":"","Age":"29","NameTH_First Name":"[illegible]","CitizenID":"[illegible]","Address_Soi ":"[illegible] [illegible]","IssuerID":"","Father_Nationality":null,"Address_Amphur": "เขต[illegible]","Address_Alley":"","NameTH_Prefix":"นาย","Do micileDate":null,"NameEN_FirstName":null,"ExpireDate":"","M other_FirstName":null,"Address_Province":"กรุงเทพมหานคร","NameEN_Prefix":null,"PersonStatus":"มีชีวิตอยู่","Address_No" :"[illegible]","Sex":"ชาย","Address_Tumbol":"[illegible]","NameT H_SurName":"[illegible]","DomicileType":null,"DomicileStatus":n ull,"Address_Road":"","IssuerPlace":"ท้องถิ่นเขต[illegible]","Bir thDate":"[illegible]","Nationality":"ไทย","NameTH_MiddleNa me":"","AddressID":"[illegible]","Nationality_Old":"","Mot her_Nationality":null,"NameEN_MiddleName":null,"IssuerAge ncy":"","Mother_CitizenID":null,"Nationality_ChangeDate":"","Father_CitizenID":null,"NameEN_SurName":null,"Father_First Name":null}]

| Parameter | Description |
|---|---|
| CitizenID | หมายเลขบัตรประจำตัวประชาชน 13 หลัก |
| NameTH_Prefix | คำนำหน้าชื่อ (ภาษาไทย) |
| NameTH_FirstName | ชื่อจริง (ภาษาไทย) |
| NameTH_MiddleName | ชื่อกลาง (ภาษาไทย) |
| NameTH_SurName | นามสกุล (ภาษาไทย) |
| Sex | เพศ |
| BirthDate | วัน/เดือน/ปี พ.ศ. เกิด โดยมีรูปแบบเป็น YYYYMMDD |
| Age | อายุ |
| PersonStatus | สถานะการมีชีวิตอยู่ |
| AddressID | รหัสประจำบ้าน |

| Parameter | Description |
|---|---|
| Address_No | บ้านเลขที่ |
| Address_Moo | หมู่ |
| Address_Alley | ตรอก |
| Address_Soi | ซอย |
| Address_Road | ถนน |
| Address_Tumbol | ตำบล/ แขวง |
| Address_Amphur | อำเภอ/ เขต |
| Address_Province | จังหวัด |
| IssueDate | วัน/เดือน/ปี พ.ศ. ที่ทำบัตร โดยมีรูปแบบเป็น YYYYMMDD |
| ExpireDate | วัน/เดือน/ปี พ.ศ. ที่บัตรหมดอายุ โดยมีรูปแบบเป็น YYYYMMDD |
| IssuerID | รหัสประจำตัวผู้ออกบัตร |
| IssuerPlace | สถานที่ออกบัตร |
| IssuerAgency | หน่วยงานที่ออกบัตร |
| Nationality | สัญชาติ |
| Nationality_Old | สัญชาติเดิม |
| Nationality_ChangeDate | วัน/เดือน/ปี พ.ศ. ที่เปลี่ยนสัญชาติ โดยมีรูปแบบเป็น YYYYMMDD |

## 12. รายละเอียดอื่น ๆ

### 12.1 ตัวแปรบังคับที่ต้องระบุในทุกคำร้อง (Request)

ระบบสารสนเทศ/ แอพพลิเคชั่นของน่วยงานจำเป็นที่จะต้องยืนยันตัวตนกับระบบยืนยันตัวบุคคลกลางก่อนเพื่อที่จะเรียกใช้ API โดยในการยืนยันตัวระบบและแอพพลิเคชั่นที่เรียกใช้ API จะต้องระบุตัวแปรดังนี้

| | Key/ Parameter | Required | Description |
|---|---|---|---|
| 1 | ServiceCode หรือ Consumer Key | ต้องระบุ | - รหัสของระบบสารสนเทศ/ แอพพลิเคชั่นของหน่วยงานนั้น ๆ (OAuth Consumer Key)<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน EGA Web Services |
| 2 | Passcode หรือ Consumer Secret | ต้องระบุ | - รหัสผ่านสำหรับระบบสารสนเทศ/ แอพพลิเคชั่นของหน่วยงานนั้น ๆ (OAuth Consumer Secret)<br>- เป็นชุดรหัสที่ สพร. ออกให้ เพื่อความปลอดภัยในการเรียกใช้งาน EGA Web Services |

หมายเหตุ: ผู้ดูแลระบบสารสนเทศ/ แอพพลิเคชั่น/ โมบายแอพลิเคชั่นของหน่วยงานสามารถลงทะเบียนการขอใช้ EGA Web Service ของ สพร. ได้ที่ระบบจัดการบริการจากบัตรประชาชนสมาร์ทการ์ด (**http://dev.egov.go.th**) หรือสอบถามรายละเอียดเพิ่มเติมได้ที่ **helpdesk@ega.or.th** หรือ โทร. **0-2612-6060**

# ภาคผนวก ก. ตัวอย่าง Source Code

## 6) Asp.net C#

### 1.1) SSOLogin

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.Sql;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using DotNetOpenAuth.OpenId;
using DotNetOpenAuth.OpenId.Extensions.AttributeExchange;
using DotNetOpenAuth.OpenId.Extensions.SimpleRegistration;
using DotNetOpenAuth.OpenId.RelyingParty;
using OpenIdRelyingPartyWebForms;

public partial class SSOLogin : System.Web.UI.Page
{
    #region ตรงนี้เป็น Code ที่ไว้ใช้จัดการกับเรื่องการทำ OpenID

    private const string RolesAttribute = "http://samples.dotnetopenauth.net/sso/roles";

    //บรรทัดนี้ต้องแก้ให้เป็น OpenID Provider ที่ต้องการใช้งาน
    private const string OpenIdProviderURL = "https://testopenid.ega.or.th/";

    private static OpenIdRelyingParty relyingParty = new OpenIdRelyingParty();

    static SSOLogin()
    {
                    // Configure the RP to only allow assertions from our trusted OP endpoint.
        EndpointSelector ep = new EndpointSelector(EndpointFilter);
        relyingParty.EndpointFilter = ep;
           }

    protected static bool EndpointFilter(IProviderEndpoint ipep)
    {
        return ipep.Uri.AbsoluteUri == OpenIdProviderURL + "server.aspx";
    }
```

```
protected void Page_Load(object sender, EventArgs e)
{
    //ทำการสร้าง Url ของหน้าที่ต้องการให้ส่งกลับ โดยในที่นี้ให้ส่งกลับมายังหน้าเดิม
    string[] segments = Request.Url.Segments;
    string pathToPage = "";
    for(int i=0; i<segments.Length-1; i++) {
        pathToPage += segments[i];
    }
    UriBuilder returnToBuilder = new UriBuilder(Request.Url);
    returnToBuilder.Path = pathToPage+"SSOLogin.aspx";
    returnToBuilder.Query = null;
    returnToBuilder.Fragment = null;
    Uri returnTo = returnToBuilder.Uri;
    returnToBuilder.Path = "/";

    //บันทึก Url ลงใน Realm
    Realm realm = returnToBuilder.Uri;

    //ถ้าไม่เคย Request มาก่อน Respones จะเป็น “null”
    IAuthenticationResponse response = relyingParty.GetResponse();
    if (response == null)
    {
        //เพื่อทำ Single Sign-On เราจะส่งผู้ใช้ไปยัง OpenID Provider ของ ICT โดยไม่ให้ผู้ใช้เลือก
        try
        {
            IAuthenticationRequest request = relyingParty.CreateRequest(
                OpenIdProviderURL,
                realm,
                returnTo);

            //Attributes ที่ต้องการร้องขอจาก OpenID Provider ของ ICT
            //โดยใช้หลักการ AttributeExcahnge (AX)
            var ax = new FetchRequest();
            ax.Attributes.Add(new AttributeRequest(WellKnownAttributes.Contact.Email, true));
            ax.Attributes.Add(new AttributeRequest(WellKnownAttributes.Name.FullName, true));
            ax.Attributes.Add(new AttributeRequest(WellKnownAttributes.Name.Alias, true));
            ax.Attributes.Add(new AttributeRequest("http://www.egov.go.th/2012/identifier/citizenid",
true));
            ax.Attributes.Add(new AttributeRequest("http://www.egov.go.th/2012/identifier/usertype",
true));
            request.AddExtension(ax);

            //ส่ง Request ไปยัง OpenID Provider ของ ICT
            request.RedirectToProvider();
        }
```

```
            catch(Exception eNull)
            {
                errorLabel.Text = @"ไม่พบ OpenID end point สาเหตุอาจมาจากการที่ท่านระบุ Url ผิดหรือไม่ได้ทำการ
เพิ่ม SSL Certificate ไปยัง Trusted Root Store บนเครื่อง Server";
            }
        }
        else
        {
            switch (response.Status)
            {
                case AuthenticationStatus.Canceled:
                    //ผู้ใช้ยกเลิก
                    break;
                case AuthenticationStatus.Failed:
                    //ผู้ใช้ Authenticate ไม่ผ่าน
                    break;
                case AuthenticationStatus.Authenticated:

                    //ทำการดึงข้อมูลที่ได้รับมาจาก OpenID Provider ออกมา
                    FetchResponse fetchResponse = response.GetExtension<FetchResponse>();
                    State.FetchResponse = fetchResponse;
                    string claimed = response.FriendlyIdentifierForDisplay;
                    string userName =
State.FetchResponse.Attributes[WellKnownAttributes.Name.Alias].Values[0];
                    string fullName =
State.FetchResponse.Attributes[WellKnownAttributes.Name.FullName].Values[0];
                    string identifier =
State.FetchResponse.Attributes["http://www.egov.go.th/2012/identifier/citizenid"].Values[0];
                    string email =
State.FetchResponse.Attributes[WellKnownAttributes.Contact.Email].Values[0];
                    string userType =
State.FetchResponse.Attributes["http://www.egov.go.th/2012/identifier/usertype"].Values[0];

                    UserMappingAndAuthorization(claimed, userName, fullName, email, identifier,
userType);

                    break;
                default:
                    break;
            }
        }
    }
    #endregion

    #region code ในส่วนนี้ตรงนี้ขึ้นอยู่กับแต่ละระบบสารสนเทศ/ ระบบ e-Service ของหน่วยงานภาครัฐ
```

```
    /// <summary>
    /// ประเภทผู้ใช้งาน (userType) ถูกแบ่งออกเป็น 4 ประเภท ดังนี้
    /// กรณีที่ประเภทผู้ใช้งาน = Citizen: ประชาชน/ บุคคลธรรมดา รหัสประจำตัวผู้ใช้งานจะเป็น “เลขประจำตัวประชาชน” (13 หลัก)
    /// กรณีที่ประเภทผู้ใช้งาน = JuristicPerson: นิติบุคคล รหัสประจำตัวผู้ใช้งานจะเป็น “เลขทะเบียนนิติบุคคล”**
    /// กรณีที่ประเภทผู้ใช้งาน = Foreigner: ชาวต่างชาติ รหัสประจำตัวผู้ใช้งานจะเป็น “รหัสประเทศที่ออกตามด้วยเครื่องหมาย “-”และตามด้วยหมายเลขหนังสือเดินทาง (Passport)” เช่น USA-C00001549 เป็นต้น**
    /// กรณีที่ประเภทผู้ใช้งาน = GovernmentAgent: ข้าราชการ/ เจ้าหน้าที่รัฐ รหัสประจำตัวผู้ใช้งานจะเป็น “เลขประจำตัวประชาชน” (13 หลัก)
    ///
    /// ผู้พัฒนาต้อง Implement Method นี้ เพื่อตรวจสอบว่า ผู้ใช้งานที่มีรหัสดังกล่าวเป็นสมาชิกของระบบสารสนเทศของหน่วยงานท่านอยู่แล้วหรือไม่
    /// 1) ถ้าเป็นสมาชิก Method นี้ ก็ควรจะ Load ข้อมูลต่าง ๆ ที่จำเป็นสำหรับผู้ใช้งานนั้น ๆ ใส่ไว้ใน Session หรืออื่น ๆ ตามที่ท่านใช้ในหน้าจอ Login ปกติ
    /// จากนั้นจึง Redirect ผู้ใช้งานไปยังหน้าจอหลักของบริการ
    /// 2) ถ้าไม่เป็นสมาชิก ท่านอาจจะให้ผู้ใช้งานท่านนี้ทำการสมัครสมาชิกเสียก่อน โดย Redirect ไปยังหน้า SSORegister
    ///
    /// </summary>
    /// <param name="userIdentifier"></param>

    protected void UserMappingAndAuthorization(string openId, string userName, string fullName, string email, string userIdentifier, string userType)
    {
        Response.Write(String.Format("OpenId : {0} <br/>", openId));
        Response.Write(String.Format("Username : {0} <br/>", userName));
        Response.Write(String.Format("Name : {0} <br/>", fullName));
        Response.Write(String.Format("E-mail : {0} <br/>", email));
        Response.Write(String.Format("Identifier : {0} <br/>", userIdentifier));
        Response.Write(String.Format("User type : {0} <br/>", userType));
    }

    #endregion


}
```

** หมายเหตุ: ปัจจุบันทาง สพร. ยังไม่มีขั้นตอนการตรวจสอบความถูกต้องของข้อมูลเหล่านี้ ซึ่งขั้นตอนการตรวจสอบนั้นจะถูกพัฒนาในลำดับต่อไป ทาง สพร. จึงไม่แนะนำให้นำข้อมูลนี้ไปใช้งานในการยืนยันตัวบุคคลจริง

1.2) SSORegister

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml.XPath;
using DotNetOpenAuth.OpenId;
using DotNetOpenAuth.OpenId.Extensions.AttributeExchange;
using DotNetOpenAuth.OpenId.RelyingParty;
using DotNetOpenAuth.OpenId.Extensions.SimpleRegistration;
using OpenIdRelyingPartyWebForms;
using DotNetOpenAuth.OAuth;
using DotNetOpenAuth.Messaging;
using DotNetOpenAuth;
using DotNetOpenAuth.ApplicationBlock;
using DotNetOpenAuth.OAuth.ChannelElements;
using DotNetOpenAuth.OAuth.Messages;
//Lib ในการดึงค่าออกจาก Xml ของ สพร.
using GITS_SSO;

public partial class SSORegister : System.Web.UI.Page
{
    //แก้ 2 ตัวนี้ตามชื่อ e-service ต่าง ๆ
    //ซึ่งทาง สพร. จะเป็นคนกำหนด 2 ตัวนี้ให้
    private const string consumerKey = "sampleconsumer";
    private const string consumerSecret = "samplesecret";

    #region ตรงนี้เป็น code ที่ไว้ใช้จัดการกับเรื่องการทำ OAuth นะครับ

     //Url ในการขอ Xml
    private const string xmlUrl = "https://testopenid.ega.or.th/XmlUserInfo.aspx";
    //Url ของ OAuth Provider
    private const string oAuthUrl = "https://testopenid.ega.or.th/OAuth.ashx";

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            //OAuth
            if (Session["WcfTokenManager"] != null)
            {
                //เริ่มสร้าง OAuth Request
                WebConsumer consumer = this.CreateConsumer();
```

```
            AuthorizedTokenResponse accessTokenMessage = consumer.ProcessUserAuthorization();
            if (accessTokenMessage != null)
            {
                Session["WcfAccessToken"] = accessTokenMessage.AccessToken;
                Session.Remove("WcfTokenManager");

                //นำ Access Token ไปแลก Xml แล้ว Bind เข้าสู่ Registeration Form
                string path = String.Format("{0}?AccessToken={1}", xmlUrl,
Server.UrlEncode(accessTokenMessage.AccessToken));
                SSOUserInfo ssoUI = new SSOUserInfo(path);
                this.bindValueToPage(ssoUI);
            }
            else
            {
                UriBuilder callback = new UriBuilder(Request.Url);
                callback.Query = null;
                Dictionary<string, string> requestParams = new Dictionary<string, string>();
                requestParams.Add("scope", "");

                UserAuthorizationRequest response =
consumer.PrepareRequestUserAuthorization(callback.Uri, requestParams, null);
                consumer.Channel.Send(response);
            }
        }
        else
        {
            WebConsumer consumer = this.CreateConsumer();
            UriBuilder callback = new UriBuilder(Request.Url);
            callback.Query = null;

            Dictionary<string, string> requestParams = new Dictionary<string, string>();
            requestParams.Add("scope", "");

            UserAuthorizationRequest response =
consumer.PrepareRequestUserAuthorization(callback.Uri, requestParams, null);
            consumer.Channel.Send(response);
        }
    }
}

private int ConvertToInt(string input)
{
    int result = 0;
    try
    {
```

```
            result = Convert.ToInt32(input);
        }
        catch
        {
            result = 1;
        }
        return result;
    }

    private WebConsumer CreateConsumer()
    {
        InMemoryTokenManager tokenManager = Session["WcfTokenManager"] as
InMemoryTokenManager;
        if (tokenManager == null)
        {
            tokenManager = new InMemoryTokenManager(consumerKey, consumerSecret);
            Session["WcfTokenManager"] = tokenManager;
        }
        MessageReceivingEndpoint oauthEndpoint = new MessageReceivingEndpoint(
            new Uri(oAuthUrl),
            HttpDeliveryMethods.PostRequest);

        ServiceProviderDescription spd = new ServiceProviderDescription();
        spd.RequestTokenEndpoint = oauthEndpoint;
        spd.UserAuthorizationEndpoint = oauthEndpoint;
        spd.AccessTokenEndpoint = oauthEndpoint;
        spd.ProtocolVersion = DotNetOpenAuth.OAuth.ProtocolVersion.V10a;
        spd.TamperProtectionElements = new
DotNetOpenAuth.Messaging.ITamperProtectionChannelBindingElement[] {
                                            new HmacSha1SigningBindingElement(),
                                        };
        WebConsumer consumer = new WebConsumer(spd, tokenManager);

        return consumer;
    }

    #endregion

    #region code ในส่วนนี้ตรงนี้ขึ้นอยู่แต่ละกับระบบสารสนเทศ/ e-Service ภาครัฐ

     /// <summary>
    /// Method bindValueToPage จะรับตัวแปร 1 ตัว คือ SSOUserInfo โดย Method นี้จะถูกเรียกใช้โดย
Page_Load event
    ///
```

```
    /// ผู้พัฒนาต้องนำค่าจาก SSOUserInfo ไปใส่ใน Register Form ที่ผู้พัฒนาทำขึ้น เพื่อให้ผู้ใช้ไม่ต้องกรอกข้อมูลที่มีอยู่แล้วในระบบ e-portal ลงใน Register Form อีก โดยผู้พัฒนาสามารถเข้าถึงข้อมูลผ่าน Properties ใน Class “SSOUserInfo” ที่จะอธิบายต่อไปนี้
    ///
    /// Properities                 Data Type           คำอธิบาย
    /// SSOUserInfo.UserName        ProfileField        ชื่อผู้ใช้ในระบบ e-portal
    /// SSOUserInfo.Title           VerifiedField       คำนำหน้า
    /// SSOUserInfo.FirstName       VerifiedField       ชื่อภาษาไทย
    /// SSOUserInfo.LastName        VerifiedField       นามสกุลภาษาไทย
    /// SSOUserInfo.DateOfBirth     VerifiedField       วัน-เดือน-ปีเกิด
    /// SSOUserInfo.Gender          VerifiedField       เพศ
    /// SSOUserInfo.MemberType      ProfileField        ประเภทผู้ใช้งาน
    /// กรณีที่ประเภทผู้ใช้งาน = Citizen: บุคคลทั่วไป รหัสประจำตัวผู้ใช้งานจะเป็น “เลขประจำตัวประชาชน” (13 หลัก)
    /// กรณีที่ประเภทผู้ใช้งาน = JuristicPerson: นิติบุคคล รหัสประจำตัวผู้ใช้งานจะเป็น “เลขประจำตัวนิติบุคคล”
    /// กรณีที่ประเภทผู้ใช้งาน = Foreigner: ชาวต่างชาติ รหัสประจำตัวผู้ใช้งานจะเป็น “รหัสประเทศที่ออกตามด้วยเครื่องหมาย “-” และตามด้วยหมายเลข Passport” เช่น USA-C00001549
    /// กรณีที่ประเภทผู้ใช้งาน = GovernmentAgent: ข้าราชการ/ เจ้าหน้าที่รัฐ รหัสประจำตัวผู้ใช้งานจะเป็น “เลขประจำตัวประชาชน” (13 หลัก)
    /// SSOUserInfo.IdentificationCode  VerifiedField   เลขแสดงตัวตน โดยแปรผันตามประเภทผู้ใช้งาน
    /// SSOUserInfo.IssuedBy        VerifiedField       สถานที่ออก
    /// SSOUserInfo.IssuedDate      VerifiedField       วัน-เดือน-ปีออก
    /// SSOUserInfo.ExpiryDate      VerifiedField       วัน-เดือน-ปีที่หมดอายุ
    /// SSOUserInfo.Nationality     VerifiedField       สัญชาติ
    /// SSOUserInfo.Occupation      VerifiedField       อาชีพ
    /// SSOUserInfo.HouseNumber     VerifiedField       บ้านเลขที่
    /// SSOUserInfo.VillageName     VerifiedField       ชื่ออาคาร/ หมู่บ้าน
    /// SSOUserInfo.Soi             VerifiedField       ซอย
    /// SSOUserInfo.Road            VerifiedField       ถนน
    /// SSOUserInfo.SubDistrict     VerifiedField       แขวง/ ตำบล
    /// SSOUserInfo.District        VerifiedField       เขต/ อำเภอ
    /// SSOUserInfo.Province        VerifiedField       จังหวัด
    /// SSOUserInfo.PostCode        VerifiedField       รหัสไปรษณีย์
    /// SSOUserInfo.GeoCode         VerifiedField       รหัสพื้นที่ตามกระทรวงมหาดไทย
    /// SSOUserInfo.Phone           VerifiedField       เบอร์โทรศัพท์
    /// SSOUserInfo.MobliePhone     VerifiedField       เบอร์โทรศัพท์เคลื่อนที่
    /// SSOUserInfo.Email           VerifiedField       E-mail Address
    /// SSOUserInfo.OrgName         OrganizationField   หน่วยงาน
    /// SSOUserInfo.Ministry        OrganizationField   กระทรวง
    /// SSOUserInfo.Department      OrganizationField   กรม
    /// SSOUserInfo.Division        OrganizationField   กอง
    ///
    /// ProfileField.StringValue -> คืนค่าเป็น String ของ Properities
    /// VerifiedField.StringValue -> คืนค่าเป็น String ของ Properities
    /// OrganizationField.StringValue -> คืนค่าเป็น String ของ Properities
```

```
    ///
    /// ProfileField.Value -> คืนค่าเป็น Object ของ Properities
    /// VerifiedField.Value -> คืนค่าเป็น Object ของ Properities
    /// OrganizationField.Value -> คืนค่าเป็น Object ของ Properities
    ///
    /// VerifiedField.VerifiedLevel -> คืนค่า VerifiedLevel ของ Properities
    /// OrganizationField.VerifiedLevel -> คืนค่า VerifiedLevelของ Properities
    ///
    /// OrganizationField.Code -> Code ของ กระทรวง/กรม/กอง/หน่วยงาน
    /// </summary>
    /// <param name="userIdentifier"></param>
    private void bindValueToPage(SSOUserInfo ssoUI)
    {
        //ตัวอย่าง
        txt_UserName.Text = ssoUI.IdentificationCode;
        txt_FirstName.Text = ssoUI.FirstName;
        txt_LastName.Text = ssoUI.LastName;
        txt_Citizen.Text = ssoUI.IdentificationCode;
        txt_Email.Text = ssoUI.Email;
        txt_HouseNumber.Text = ssoUI.HouseNumber;
        txt_Village.Text = ssoUI.VillageName;
        txt_Soi.Text = ssoUI.Soi;
        txt_Road.Text = ssoUI.Road;
        txt_PostCode.Text = ssoUI.PostCode;
        txt_Phone.Text = ssoUI.Phone;
        txt_Fax.Text = ssoUI.Fax;
        txt_MPhone.Text = ssoUI.MobliePhone;


    }

    //ลงทะเบียน
    protected void btn_Submit_Click(object sender, EventArgs e)
    {

    }

    #endregion


}
```

# ภาคผนวก ข. ตัวอย่างเอกสารอิเล็กทรอนิกส์ (XML)

1) ผู้ใช้งานประเภทประชาชน/ บุคคลธรรมดา

```
▼<Member type="ประชาชน">
   <UserID>2e56a299-4bde-4016-ab04-b220ccbbd884</UserID>
   <UserName>rapeeTest1</UserName>
   <Title VerifiedLevel="Unverified">นางสาว</Title>
   <FullName VerifiedLevel="Unverified">นางสาว [illegible]</FullName>
   <FirstName VerifiedLevel="Unverified">[illegible]</FirstName>
   <LastName VerifiedLevel="Unverified">[illegible]</LastName>
   <DateOfBirth VerifiedLevel="Unverified"/>
   <Gender VerifiedLevel="Unverified">หญิง</Gender>
  ▼<Identification>
     <Code VerifiedLevel="Unverified">[illegible]</Code>
     <IssueBy VerifiedLevel="Unverified">สำนักงานเขตราชเทวี</IssueBy>
     <IssueDate VerifiedLevel="Unverified">27/3/2556 0:00:00</IssueDate>
     <ExpireDate VerifiedLevel="Unverified">28/3/2561 0:00:00</ExpireDate>
   </Identification>
   <Nationality VerifiedLevel="Unverified">ไทย</Nationality>
   <Occupation VerifiedLevel="Unverified">พนักงานบริษัท</Occupation>
  ▼<Address>
     <HouseNumber VerifiedLevel="Unverified">108</HouseNumber>
     <VillageName VerifiedLevel="Unverified">-</VillageName>
     <Moo VerifiedLevel="Unverified">-</Moo>
     <Soi VerifiedLevel="Unverified">-</Soi>
     <Road VerifiedLevel="Unverified">รางน้ำ</Road>
     <SubDistrict VerifiedLevel="Unverified">ถนนพญาไท</SubDistrict>
     <District VerifiedLevel="Unverified">เขตราชเทวี</District>
     <Province VerifiedLevel="Unverified">กรุงเทพมหานคร</Province>
     <PostCode VerifiedLevel="Unverified">10400</PostCode>
     <GeoCode VerifiedLevel="Unverified">10370200</GeoCode>
   </Address>
  ▼<ContactInfo>
     <Telephone VerifiedLevel="Unverified">026126000</Telephone>
     <Mobilephone VerifiedLevel="Unverified">08[illegible]</Mobilephone>
     <EMail VerifiedLevel="VerifiedLevel1">[illegible]@hotmail.com</EMail>
     <AlternativeEMail VerifiedLevel="Unverified"/>
   </ContactInfo>
 </Member>
```

2) ผู้ใช้งานประเภทนิติบุคคล

```
▼<Member type="นิติบุคคล">
   <UserID>8edb7ce0-4fc0-4ebc-a721-95faaa21522d</UserID>
   <UserName>rapeeTest2</UserName>
   <Name VerifiedLevel="Unverified">บริษัท ทดสอบ จำกัด (มหาชน)</Name>
   <Abbreviation VerifiedLevel="Unverified">ทส บมจ.</Abbreviation>
   <ContactPerson VerifiedLevel="Unverified">[illegible]</ContactPerson>
   <ContactPhoneNumber VerifiedLevel="Unverified">026126000</ContactPhoneNumber>
  ▼<Identification>
     <Code VerifiedLevel="Unverified">1[illegible]9</Code>
   </Identification>
  ▼<Address>
     <HouseNumber VerifiedLevel="Unverified">108</HouseNumber>
     <VillageName VerifiedLevel="Unverified">-</VillageName>
     <Moo VerifiedLevel="Unverified">-</Moo>
     <Soi VerifiedLevel="Unverified">-</Soi>
     <Road VerifiedLevel="Unverified">รางน้ำ</Road>
     <SubDistrict VerifiedLevel="Unverified">ถนนพญาไท</SubDistrict>
     <District VerifiedLevel="Unverified">เขตราชเทวี</District>
     <Province VerifiedLevel="Unverified">กรุงเทพมหานคร</Province>
     <PostCode VerifiedLevel="Unverified">10400</PostCode>
     <GeoCode VerifiedLevel="Unverified">10370200</GeoCode>
   </Address>
  ▼<ContactInfo>
     <Telephone VerifiedLevel="Unverified">026126000</Telephone>
     <Mobilephone VerifiedLevel="Unverified">08[illegible]</Mobilephone>
     <EMail VerifiedLevel="Unverified">[illegible]@gmail.com</EMail>
     <AlternativeEMail VerifiedLevel="Unverified"/>
   </ContactInfo>
 </Member>
```

3) ผู้ใช้งานประเภทชาวต่างชาติ

```
▼<Member type="ชาวต่างชาติ">
   <UserID>aa869d07-0a11-4777-b94d-377ba39bf232</UserID>
   <UserName>rapeeTest5</UserName>
   <Title VerifiedLevel="Unverified">Miss</Title>
   <FullName VerifiedLevel="Unverified">Miss [illegible]</FullName>
   <FirstName VerifiedLevel="Unverified">[illegible]</FirstName>
   <MiddleName VerifiedLevel="Unverified"/>
   <LastName VerifiedLevel="Unverified">[illegible]</LastName>
   <DateOfBirth VerifiedLevel="Unverified"/>
   <Gender VerifiedLevel="Unverified">หญิง</Gender>
  ▼<Identification>
     <Code VerifiedLevel="Unverified">1234567899</Code>
   </Identification>
   <Nationality VerifiedLevel="Unverified">English</Nationality>
   <Occupation VerifiedLevel="Unverified"/>
  ▼<Address>
     <HouseNumber VerifiedLevel="Unverified">108</HouseNumber>
     <VillageName VerifiedLevel="Unverified"/>
     <Moo VerifiedLevel="Unverified"/>
     <Soi VerifiedLevel="Unverified"/>
     <Road VerifiedLevel="Unverified">Rangnum</Road>
     <SubDistrict VerifiedLevel="Unverified"/>
     <District VerifiedLevel="Unverified"/>
     <Province VerifiedLevel="Unverified"/>
     <PostCode VerifiedLevel="Unverified">10400</PostCode>
     <GeoCode VerifiedLevel="Unverified">00000000</GeoCode>
   </Address>
  ▼<ContactInfo>
     <Telephone VerifiedLevel="Unverified"/>
     <Mobilephone VerifiedLevel="Unverified"/>
     <EMail VerifiedLevel="Unverified">[illegible]@hotmail.com</EMail>
     <AlternativeEMail VerifiedLevel="Unverified"/>
   </ContactInfo>
 </Member>
```

4) ผู้ใช้งานประเภทข้าราชการ/ เจ้าหน้าที่รัฐ (ที่ไม่มีบัญชีผู้ใช้งาน MailGoThai)

```
▼<Member type="เจ้าหน้าที่ของรัฐ">
   <UserID>870ebc7e-9534-4963-9b94-a81a0140a98b</UserID>
   <UserName>rapeeTest3</UserName>
   <Title VerifiedLevel="Unverified">นางสาว</Title>
   <FullName VerifiedLevel="Unverified">นางสาว[redacted]</FullName>
   <FirstName VerifiedLevel="Unverified">[redacted]</FirstName>
   <LastName VerifiedLevel="Unverified">[redacted]</LastName>
   <DateOfBirth VerifiedLevel="Unverified"/>
   <Gender VerifiedLevel="Unverified">หญิง</Gender>
  ▼<Identification>
     <Code VerifiedLevel="Unverified">1234567891011</Code>
     <IssueBy VerifiedLevel="Unverified">สำนักงานเขตราชเทวี</IssueBy>
     <IssueDate VerifiedLevel="Unverified">14/3/2556 0:00:00</IssueDate>
     <ExpireDate VerifiedLevel="Unverified">15/3/2562 0:00:00</ExpireDate>
   </Identification>
   <Nationality VerifiedLevel="Unverified">ไทย</Nationality>
   <Occupation VerifiedLevel="Unverified">เจ้าหน้าที่รัฐ</Occupation>
  ▼<Organization>
     <Name VerifiedLevel="Unverified" Code="10009000">สำนักงานรัฐบาลอิเล็กทรอนิกส์ (องค์การมหาชน)</Name>
     <Ministry VerifiedLevel="Unverified" Code="10">กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร</Ministry>
     <Department VerifiedLevel="Unverified" Code="009">สำนักงานรัฐบาลอิเล็กทรอนิกส์ (องค์การมหาชน)</Department>
     <Division VerifiedLevel="Unverified" Code="000"/>
   </Organization>
  ▼<Address>
     <HouseNumber VerifiedLevel="Unverified">108</HouseNumber>
     <VillageName VerifiedLevel="Unverified">-</VillageName>
     <Moo VerifiedLevel="Unverified">-</Moo>
     <Soi VerifiedLevel="Unverified">-</Soi>
     <Road VerifiedLevel="Unverified">รางน้ำ</Road>
     <SubDistrict VerifiedLevel="Unverified">ถนนพญาไท</SubDistrict>
     <District VerifiedLevel="Unverified">เขตราชเทวี</District>
     <Province VerifiedLevel="Unverified">กรุงเทพมหานคร</Province>
     <PostCode VerifiedLevel="Unverified">10400</PostCode>
     <GeoCode VerifiedLevel="Unverified">10370200</GeoCode>
   </Address>
  ▼<ContactInfo>
     <Telephone VerifiedLevel="Unverified">026126000</Telephone>
     <Mobilephone VerifiedLevel="Unverified">08[redacted]</Mobilephone>
     <EMail VerifiedLevel="Unverified">[redacted]@hotmail.com</EMail>
     <AlternativeEMail VerifiedLevel="Unverified">[redacted]@gmail.com</AlternativeEMail>
   </ContactInfo>
 </Member>
```

5) ผู้ใช้งานประเภทข้าราชการ/ เจ้าหน้าที่รัฐ (ที่มีบัญชีผู้ใช้งาน MailGoThai)

```
▼<Member type="เจ้าหน้าที่ของรัฐ">
   <UserID>b70aff9c-e951-102f-8d4b-83b2f2cc5bab</UserID>
   <UserName>[redacted]</UserName>
   <Title VerifiedLevel="Unverified">Miss</Title>
   <TitleTH VerifiedLevel="Unverified">นางสาว</TitleTH>
   <FullName VerifiedLevel="Unverified">Miss [redacted]</FullName>
   <FullNameTH VerifiedLevel="Unverified">นางสาว [redacted]</FullNameTH>
   <FirstName VerifiedLevel="Unverified">[redacted]</FirstName>
   <FirstNameTH VerifiedLevel="Unverified">[redacted]</FirstNameTH>
   <LastName VerifiedLevel="Unverified">[redacted]</LastName>
   <LastNameTH VerifiedLevel="Unverified">[redacted]</LastNameTH>
  ▼<Identification>
     <Code VerifiedLevel="Unverified">System.Byte[]</Code>
   </Identification>
  ▼<Organization>
     <Name VerifiedLevel="Unverified" Code="10009000">สำนักงานรัฐบาลอิเล็กทรอนิกส์ (องค์การมหาชน)</Name>
     <Ministry VerifiedLevel="Unverified" Code="10">กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร</Ministry>
     <Department VerifiedLevel="Unverified" Code="009">สำนักงานรัฐบาลอิเล็กทรอนิกส์ (องค์การมหาชน)</Department>
     <Division VerifiedLevel="Unverified" Code="000"/>
   </Organization>
   <BusinessCategory VerifiedLevel="Unverified">GAD</BusinessCategory>
  ▼<Address>
     <HouseNumber VerifiedLevel="VerifiedLevel3">--</HouseNumber>
     <VillageName VerifiedLevel="Unverified"/>
     <Moo VerifiedLevel="Unverified"/>
     <Soi VerifiedLevel="Unverified"/>
     <Road VerifiedLevel="Unverified"/>
     <SubDistrict VerifiedLevel="Unverified"/>
     <District VerifiedLevel="VerifiedLevel3">[redacted]</District>
     <Province VerifiedLevel="VerifiedLevel3">[redacted]</Province>
     <PostCode VerifiedLevel="Unverified">70110</PostCode>
     <GeoCode VerifiedLevel="Unverified"/>
   </Address>
  ▼<ContactInfo>
     <Telephone VerifiedLevel="Unverified"/>
     <Mobilephone VerifiedLevel="Unverified">08[redacted]</Mobilephone>
     <EMail VerifiedLevel="VerifiedLevel3">[redacted]@ega.or.th</EMail>
     <AlternativeEMail VerifiedLevel="Unverified">-</AlternativeEMail>
   </ContactInfo>
 </Member>
```