

## การประยุกต์ใช้ส่วนต่อประสานโปรแกรมประยุกต์กับระบบสารสนเทศทางการแพทย์

เสรี เจริญไทย

ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล กรุงเทพมหานคร 10700

อีเมลของผู้ประพันธ์บรรณกิจ: seri.cha@mahidol.ac.th

### บทคัดย่อ

เทคโนโลยีและนวัตกรรมในปัจจุบันพัฒนาอย่างรวดเร็วแบบก้าวกระโดด ส่งผลทำให้การบริการมีความรวดเร็ว ถูกต้องแม่นยำ และมีความปลอดภัยมากขึ้น ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาลมีการนำเทคโนโลยีสารสนเทศเข้ามาใช้ในการบริการ ด้านการแพทย์ การศึกษาและการวิจัย ส่งผลให้มีระบบสารสนเทศมีจำนวนเพิ่มมากขึ้น จึงต้องการให้ระบบสารสนเทศที่เกี่ยวข้องสามารถทำงานร่วมกันส่งข้อมูลระหว่างระบบได้อย่างถูกต้อง และพร้อมใช้งานตลอดเวลา ดังนั้นส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) จึงเข้ามามีบทบาทที่สำคัญ และเป็นอีกหนึ่งแนวทาง ที่ตอบโจทย์ในการส่งผ่านข้อมูลทำให้การบริการส่งผลให้การเข้าถึงข้อมูลต่างๆ สามารถกระทำได้อย่างรวดเร็ว ปลอดภัย และพร้อมใช้งานได้ตลอดเวลา ส่งผลทำให้คณะแพทยศาสตร์ศิริราชพยาบาลสามารถให้บริการผู้ป่วยได้อย่างต่อเนื่อง ส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) เป็นอีกหนึ่งแนวทางของเทคโนโลยีที่พัฒนามาเพื่อให้สามารถเชื่อมต่อระหว่างระบบหนึ่งไปอีกระบบหนึ่งได้ โดยมี API เป็นตัวกลางในการรับคำสั่งจาก Application และทำการนำคำสั่งเหล่านั้นไปประมวลผลข้อมูล แล้วทำการส่งข้อมูลที่ตรงกันกลับไป และ Application สามารถนำข้อมูลเหล่านั้นไปใช้งานต่อไป ด้วยคุณสมบัติที่ยืดหยุ่นทำให้ปัจจุบันนี้พบว่าการใช้งานอย่างแพร่หลายไม่ว่าจะเป็นองค์กรขนาดใหญ่หรือองค์กรขนาดเล็ก คณะแพทยศาสตร์ศิริราชพยาบาลได้ประยุกต์ใช้ส่วนต่อขยายโปรแกรมประยุกต์สนับสนุนการทำงานระบบฐานข้อมูลเวชภัณฑ์ (Medical Supply Master) ให้เชื่อมโยงกับระบบจ่ายยาและเวชภัณฑ์ผู้ป่วยนอก (SiOP) เมื่อผู้ใช้งานได้ทำการปรับปรุงข้อมูลเวชภัณฑ์ที่ระบบระบบฐานข้อมูลเวชภัณฑ์ ข้อมูลเวชภัณฑ์จะถูกเรียกใช้จากระบบ SiOP ทำให้ระบบได้รับข้อมูลที่ถูกต้องและรวดเร็ว ส่งผลให้การบริการผู้ป่วยมีประสิทธิภาพและสร้าง ความพึงพอใจมากยิ่งขึ้น

**คำสำคัญ:** ส่วนต่อประสานโปรแกรมประยุกต์, การพัฒนาระบบสารสนเทศ, สารสนเทศทางการแพทย์

## Implementation of Application Programming Interface in Medical Information Systems

Seree Charoenthai

*Siriraj Information Technology Department, Faculty of Medicine Siriraj Hospital, Bangkok 10700, Thailand*

Corresponding author's e-mail: seri.cha@mahidol.ac.th

### Abstract

Currently, technology and innovation have been developing rapidly, as well as improving the service quality, accuracy, and security for customers. The Siriraj Information Technology Department of the Faculty of Medicine Siriraj Hospital has applied information technology in medical service, education, and research. This has led to an increase in the number of information systems, thereby necessitating that relevant systems be able to interoperate, accurately exchange data between one another, and remain available for use at all times. Accordingly, the Application Programming Interface (API) has become an important and one of the best practice approaches to exchange or transfer data between information systems. An API results in accessing information rapidly, securely, and reliably. All of these have resulted in the Faculty of Medicine Siriraj Hospital being able to continue serving patients. An Application Programming Interface (API) is the best practice approach to connecting system-to-system by an API gateway from each application and processing data. Therefore, when the data processing is completed, an application is able to continue to use that information. The flexible features of the API are found to be widely implemented in both large and small organizations. The Faculty of Medicine Siriraj Hospital has implemented application extensions to support the Medical Supply Master database system by enabling its integration with the outpatient medication and medical supply dispensing system (SiOP). When users update medical supply information in the Medical Supply Master system, data is retrieved by the SiOP system in a timely and accurate manner. This integration enhances the efficiency of patient care services and contributes to improved patient satisfaction.

**Keywords:** Application Programming Interface (API), Information system development, Medical informatics

## บทนำ

การให้บริการในปัจจุบันนั้นมุ่งเน้น ความรวดเร็ว ความถูกต้องแม่นยำของข้อมูล หลากๆ องค์กรล้วนมีการนำเทคโนโลยีสารสนเทศเข้ามาเพื่อเป็นเครื่องมือในการดำเนินงาน เนื่องจากระบบสารสนเทศสามารถเพิ่มศักยภาพในการให้บริการ ลดกระบวนการในการทำงาน และลดระยะเวลาในการดำเนินงานรวมถึงกิจกรรมต่างๆ ขององค์กรได้เป็นอย่างมาก ซึ่งส่งผลต่อการให้บริการหรือการดำเนินการขององค์กรมีประสิทธิภาพมากขึ้น

ส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) เป็นเทคโนโลยีที่สามารถนำมาช่วยเพิ่มประสิทธิภาพในการแลกเปลี่ยนข้อมูลหรือให้บริการระหว่างผู้ให้บริการให้เป็นระบบอัตโนมัติ รวมทั้งรักษาความปลอดภัยของข้อมูลตามมาตรฐานสากล ส่งเสริมการใช้ทรัพยากรที่จำเป็นร่วมกัน ลดเวลาในการทำงาน และต้นทุนในการพัฒนาระบบ รวมทั้งสนับสนุนการพัฒนาบริการที่หลากหลายเพื่อตอบสนองความต้องการแก่ผู้ใช้บริการทางการได้ดีขึ้น ทั้งนี้ ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาลได้พัฒนาส่วนต่อประสานโปรแกรมประยุกต์หรือ API เพื่อเชื่อมโยงข้อมูลสารสนเทศระหว่างระบบให้เป็นไปตามมาตรฐานขององค์ประกอบพื้นฐานความมั่นคงปลอดภัยสารสนเทศ (CIA) โดยวัตถุประสงค์ของบทความนี้อธิบายถึงการประยุกต์ใช้ส่วนต่อประสานโปรแกรมประยุกต์ (API) กับระบบสารสนเทศทางการแพทย์ โดยมุ่งเน้นการเพิ่มประสิทธิภาพในการแลกเปลี่ยนข้อมูล การรักษาความปลอดภัยของข้อมูลตามมาตรฐานสากล และการสนับสนุนการพัฒนาบริการทางการแพทย์ที่หลากหลายซึ่งจะเป็นประโยชน์ในการเพิ่มประสิทธิภาพการให้บริการทางการแพทย์และการจัดการทรัพยากรที่จำเป็นในอนาคต

## ความสำคัญ

ในอดีตฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาลได้พัฒนาแอปพลิเคชัน (Application) และ เว็บแอปพลิเคชัน (Web application) โดยผ่านกระบวนการจัดซื้อจัดจ้างบริษัทภายนอก (Outsource) เข้ามาร่วมในการพัฒนาระบบสารสนเทศทางการแพทย์ ในกระบวนการในการพัฒนาระบบสารสนเทศนั้น บริษัทภายนอกมีความจำเป็นต้องมีสิทธิ์ในการเข้าถึงฐานข้อมูลประวัติส่วนตัวของผู้ป่วยและการรักษาประวัติการรักษาพยาบาลของผู้ป่วยจึงทำให้เกิดความเสี่ยงที่ข้อมูลประวัติส่วนตัวและดังกล่าวถูกนำไปเผยแพร่ผู้ไม่หวังดี ทั้งโดยตั้งใจ หรือไม่ตั้งใจจะส่งผลทำให้ทำให้คณะแพทยศาสตร์ศิริราชพยาบาล ในฐานะที่เป็นผู้ดูแลและกำกับข้อมูลดังกล่าวเกิดความสูญเสียความเชื่อมั่นอาจจะถูกฟ้องร้องเรียกค่าเสียหายและดำเนินคดีจากผู้ป่วยที่มาทำการรักษาพยาบาล ดังนั้น ส่วนต่อประสานโปรแกรมประยุกต์ที่ได้ทำการพัฒนาขึ้นมาเพื่อเชื่อมโยงข้อมูลที่สำคัญระหว่างระบบจึงเป็นอีกหนึ่งแนวทางที่ตอบโจทย์การรักษาความมั่นคงปลอดภัยของข้อมูลสารสนเทศและแยกการบริหารจัดการเป็นส่วนๆ ทำให้การเข้าถึงข้อมูลสามารถกระทำได้อย่างรวดเร็ว การนำเทคโนโลยีส่วนต่อประสานโปรแกรมประยุกต์มาช่วยในการบริหารจัดการข้อมูลทำให้องค์กรภายนอกหรือหน่วยงานภายนอกที่เข้าร่วมในการพัฒนาระบบสารสนเทศไม่สามารถเข้าถึงฐานข้อมูลการรักษาของผู้ป่วยได้ และสามารถกำหนดขอบเขตของการแลกเปลี่ยนข้อมูลได้ เช่น ผู้ป่วยต้องการทราบว่าตนเองมีนัดพบแพทย์ในวันและเวลาใด หรือคลินิกใดเป็นต้น ทำให้ผู้ป่วยสามารถตรวจสอบข้อมูลดังกล่าวผ่านแอปพลิเคชันบนสมาร์ตโฟนได้

## ประเภทของส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API)

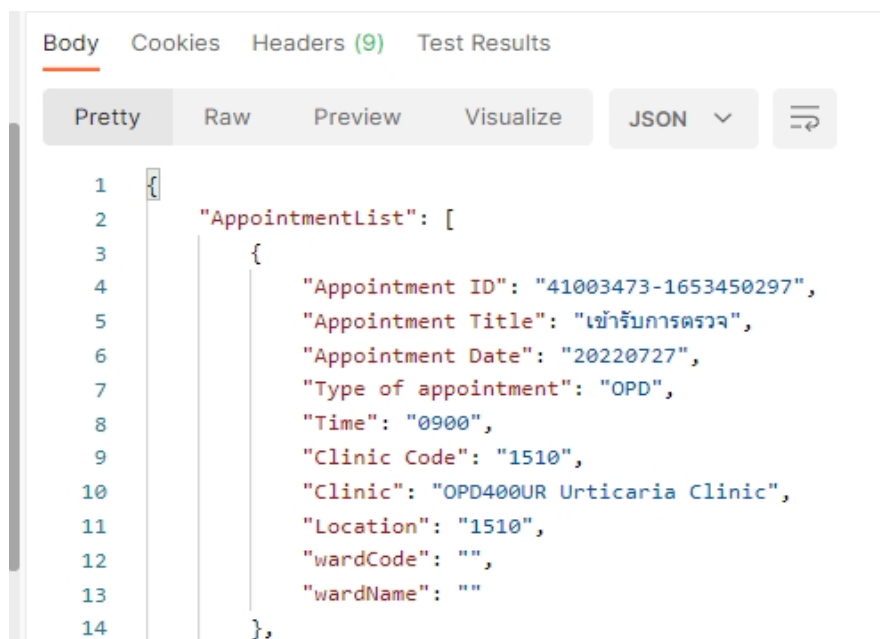
ส่วนต่อประสานโปรแกรมประยุกต์ที่ถูกสร้างเพื่อใช้งานตามวัตถุประสงค์ที่แตกต่างกัน โดยมีอยู่ 4 ประเภท ดังนี้

1. ส่วนต่อประสานโปรแกรมประยุกต์สาธารณะ (Public API) คือ ส่วนต่อประสานโปรแกรมประยุกต์ที่เปิดสาธารณะ ที่พร้อมให้นักพัฒนาหรือองค์กรอื่นสามารถนำไปใช้งานได้ (Richardson et al., 2013) มีหลายองค์กรที่ได้พัฒนาเช่น Microsoft สามารถหารายได้โดยการเปิดส่วนต่อประสานโปรแกรมประยุกต์สาธารณะ เพื่อให้นักพัฒนาสามารถเข้าถึงโปรแกรมคอมพิวเตอร์หรือข้อมูลที่ต้องการจากระบบปฏิบัติการได้อาจมีหรือไม่มีค่าใช้จ่ายแล้วแต่ตามข้อตกลง เช่น ส่วนต่อประสานโปรแกรมประยุกต์ของระบบปฏิบัติการวินโดวส์ หรือ Windows API ที่นักพัฒนาสามารถเขียนโปรแกรมที่เข้าถึงระบบของระบบปฏิบัติการได้ ดังแสดงในภาพที่ 1

2. ส่วนต่อประสานโปรแกรมประยุกต์สำหรับคู่ค้า (Partner API) คือ ส่วนต่อประสานโปรแกรมประยุกต์ที่มีให้สำหรับคู่ค้าทางธุรกิจ เพื่อเป็นการอำนวยความสะดวกให้คู่ค้าทางธุรกิจเข้าถึงข้อมูลฝ่ายตรงข้ามได้ โดยมีข้อกำหนดในการเข้าถึงข้อมูลได้เฉพาะที่ต้องการเท่านั้น (Bondel et al., 2021) เช่น ระบบ Siriraj Connect เป็นความร่วมมือระหว่างคณะแพทยศาสตร์ศิริราชพยาบาลกับธนาคารไทยพาณิชย์ จำกัด (มหาชน) แอปพลิเคชันสามารถแจ้งเตือนผู้ป่วยในกรณีที่มีนัดพบแพทย์ล่วงหน้าก่อนวันนัดหมาย เช่น การแจ้งเตือนคิวเพื่อรับบริการเจาะเลือด การแจ้งเตือนคิวเพื่อชำระค่ายาและเวชภัณฑ์ทางการแพทย์ แจ้งเตือนคิวรับยาและเวชภัณฑ์ทางการแพทย์ ให้แก่ผู้ป่วยที่มาใช้บริการที่คณะแพทยศาสตร์ศิริราชพยาบาล

3. ส่วนต่อประสานโปรแกรมประยุกต์ภายในองค์กร (Internal API) คือส่วนต่อประสานโปรแกรมประยุกต์ที่ใช้เฉพาะภายในองค์กรเพื่อเชื่อมต่อข้อมูลและระบบอย่างปลอดภัย (Faruk et al., 2022) องค์กรขนาดใหญ่ที่มีหลายแผนกมักใช้ Internal API เพื่อเข้าถึงข้อมูลของระบบอื่นในองค์กร เช่น ระบบการรับชำระเงินของคณะแพทยศาสตร์ศิริราชพยาบาลที่เชื่อมข้อมูลผู้ป่วยกับระบบเวชระเบียน โดยผู้ใช้งานกรอกหมายเลขผู้ป่วยหรือสแกนบาร์โค้ดเพื่อเชื่อมข้อมูลสำหรับการชำระค่ารักษาพยาบาล เพื่อความสะดวกและความถูกต้องของข้อมูล

4. ส่วนต่อประสานโปรแกรมประยุกต์แบบผสมผสาน (Composite API) คือรูปแบบการผสมผสานส่วนต่อประสานโปรแกรมประยุกต์ตั้งแต่ 2 บริการขึ้นไปเข้าด้วยกัน ส่วนต่อประสานแบบผสมผสานนี้สามารถช่วยให้เรียกใช้ข้อมูลพร้อมกันได้ครั้งเดียว (Gong et al., 2021) มีประโยชน์ในการช่วยลดปัญหาการใช้ส่วนต่อประสานโปรแกรมประยุกต์ซ้ำซ้อนและปรับปรุงประสิทธิภาพแพลตฟอร์มในเรื่องความเร็วได้ด้วย



ภาพที่ 1 กระบวนการแลกเปลี่ยนข้อมูลของ API

### กระบวนการทำงานของส่วนต่อประสานโปรแกรมประยุกต์

#### (Application Programming Interface: API) ประกอบด้วย 5 ขั้นตอนหลักดังนี้

1. เริ่มทำการรายการ (Initiate) เป็นกระบวนการทำงานของส่วนต่อประสานโปรแกรมประยุกต์ (API) โดยทั่วไปเริ่มจากโปรแกรมจากเครื่องคอมพิวเตอร์ลูกข่าย (Client Application) ที่ต้องการเรียกข้อมูลหรือใช้บริการจาก API ความต้องการนี้อาจเกิดจากการตัดสินใจอัตโนมัติของโปรแกรมหรือการสั่งการของผู้ใช้บริการ หากโปรแกรมจากเครื่องลูกข่ายเป็นของหน่วยงานภายนอก ขั้นตอนนี้จะมีความเสี่ยงจากมาตรการรักษาความปลอดภัยที่ไม่เพียงพอของหน่วยงานนั้น ดังนั้นผู้ให้บริการ API จึงต้องระมัดระวังความเสี่ยงในขั้นตอนนี้เป็นพิเศษ

2. ส่งคำสั่งการทำการรายการไปยังผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ (API Provider) โปรแกรมที่อยู่ในเครื่องคอมพิวเตอร์ลูกข่าย (Client Application) สร้างชุดข้อมูลการร้องขอส่วนต่อประสานโปรแกรมประยุกต์ (API Request) เพื่อเรียกข้อมูลหรือใช้บริการจากผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ตามข้อกำหนดกฎเกณฑ์ที่ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์กำหนดไว้และส่งผ่านระบบเครือข่ายไปให้ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์

3. ประมวลผลรายการ (Process) เมื่อผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ (API) ได้รับการร้องขอ (API Request) จะดำเนินการประมวลผลโดยระบบต่าง ๆ ที่เกี่ยวข้อง เพื่อจัดเตรียมผลลัพธ์ข้อมูลหรือให้บริการตามที่ Client Application ร้องขอ ขั้นตอนนี้มีความเสี่ยงหลายประการ เช่น การเปิดเผยข้อมูลเกินความจำเป็น (Excessive Data Exposure) การบริหารจัดการวงจรชีวิตที่ไม่รัดกุม (Improper API Lifecycle Management) และการตั้งค่าความมั่นคงปลอดภัยที่ผิดพลาด (Security Misconfiguration) ผู้ให้บริการ API จึงต้องระมัดระวังและจัดการความเสี่ยงในด้านเหล่านี้อย่างเคร่งครัด

4. ส่งผลลัพธ์การทำรายการกลับไปยัง API Consumer (Respond) ผู้ให้บริการ API สร้างชุดข้อมูล API Response ที่ประกอบด้วยข้อมูลผลลัพธ์จากการประมวลผลในขั้นตอนก่อนหน้า เพื่อส่งผลลัพธ์ของข้อมูลที่ร้องขอหรือผลลัพธ์ของการให้บริการไปยัง Client Application

5. นำผลลัพธ์ที่ได้มาแสดงหรือใช้งาน (Output) เมื่อ Client Application ได้รับ API Response แล้ว Client Application จะดำเนินการแสดงผลข้อมูลให้ผู้ให้บริการทราบ หรือใช้งานผลลัพธ์ที่ได้มาต่อไปในทุกขั้นตอน ตั้งแต่ขั้นตอนที่ 1 ถึงขั้นตอนที่ 5 มีความเสี่ยงจากการลักลอบเข้าถึงข้อมูลหรือใช้งานส่วนต่อประสานโปรแกรมประยุกต์ โดยไม่ได้รับสิทธิ์ (Unauthorized Access) และการให้บริการหยุดชะงัก (Service Disruption) ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ จึงให้ความสำคัญในการระมัดระวังความเสี่ยงทั้ง 2 ด้านนี้ ในทุกขั้นตอน (Geewax, 2021)

กระบวนการทำงานของส่วนต่อประสานโปรแกรมประยุกต์ที่กล่าวมาแล้วนั้นจำเป็นต้องอ้างอิงตามมาตรฐานความมั่นคงปลอดภัย เพื่อให้การทำงานของระบบนั้นมีความน่าเชื่อถือและมีมาตรฐาน

### มาตรฐานขององค์ประกอบพื้นฐานความมั่นคงปลอดภัยสารสนเทศ

มาตรฐานขององค์ประกอบพื้นฐานความมั่นคงปลอดภัยสารสนเทศประกอบด้วย ความลับของข้อมูล (Confidentiality) ความถูกต้องครบถ้วนของข้อมูล (Integrity) และ ความพร้อมใช้ของข้อมูล และระบบสารสนเทศ (Availability) ซึ่งเป็นลักษณะที่จำเป็นในการรักษาความมั่นคงปลอดภัยของข้อมูลสารสนเทศ ดังแสดงในภาพที่ 2



ภาพที่ 2 องค์ประกอบพื้นฐานความมั่นคงปลอดภัยสารสนเทศ (CIA)

ความลับของข้อมูล (Confidentiality) เป็นองค์ประกอบสำคัญของการรักษาความมั่นคงปลอดภัยระบบสารสนเทศ ผู้บริหารต้องมีนโยบายและมาตรการตรวจสอบสิทธิในการเข้าถึงข้อมูล โดยผู้ที่มีสิทธิหรือได้รับอนุญาตเท่านั้นที่สามารถเข้าถึงข้อมูลได้ การยืนยันตัวตนในการเข้าถึงระบบสารสนเทศและข้อมูลมักใช้รหัสผ่าน (Password) ส่วนบุคคล นอกจากนี้การกำหนดชั้นความลับของเอกสารเป็นอีกปัจจัยสำคัญที่ทำให้โครงการความมั่นคงปลอดภัยระบบสารสนเทศประสบความสำเร็จ โดยแบ่งชั้นความลับออกเป็น 3 ประเภทตามระเบียบว่าด้วยการรักษาความลับ

ของราชการ พ.ศ. 2544 (สำนักนายกรัฐมนตรี, 2544) ประกอบด้วย ระดับชั้นลับที่สุด (Top Secret) ระดับชั้นลับมาก (Secret) ระดับชั้นลับ (Confidential) และระดับชั้นข้อมูลที่ถูกกำหนดขึ้นเพิ่มเติม ประกอบด้วย ระดับชั้นข้อมูลสำหรับใช้ภายในองค์กร (Internal Use) และระดับชั้นสาธารณะ (Public) (ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล, 2562) ในปัจจุบันการปกป้องความลับของข้อมูล เช่น การเข้ารหัส (Encryption) ถูกใช้แพร่หลายเพื่อปกป้องสารสนเทศที่ต้องการการดูแลอย่างเข้มงวด ตัวอย่างเช่น เอกสารคุณภาพของฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล ซึ่งจัดเป็นเอกสารชั้นความลับ ถูกเก็บรักษาโดยเจ้าหน้าที่ผู้ดูแลเอกสารคุณภาพ (Document Control) ในตู้เอกสารที่อยู่ในห้องมั่นคงที่ทนความร้อนและกันไฟไหม้ ผู้ที่มีสิทธิ์เข้าถึงเอกสารเหล่านี้คือเจ้าหน้าที่ที่ได้รับมอบหมายเท่านั้น ในการรักษาความลับของข้อมูลการรักษาพยาบาลของผู้ป่วย รวมถึงข้อมูลต่างๆ ในระบบสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล มีมาตรฐานในการป้องกันการเข้าถึงข้อมูลและการกำหนดสิทธิ์ตามภาระหน้าที่ความรับผิดชอบในแต่ละตำแหน่ง เมื่อมีการแก้ไขข้อมูลในระบบสารสนเทศ จะมีการจัดเก็บประวัติการแก้ไขเพื่อการตรวจสอบในกรณีจำเป็น สำคัญของการกำหนดสิทธิ์ในการเข้าถึงข้อมูลในระบบสารสนเทศคือการป้องกันไม่ให้ข้อมูลที่สำคัญถูกแก้ไขหรือทำสำเนาโดยไม่ได้รับอนุญาต หากข้อมูลเหล่านี้ถูกแก้ไขหรือทำสำเนาและถูกนำไปใช้ในทางที่ผิด อาจเกิดความเสียหายร้ายแรงต่อคณะแพทยศาสตร์ศิริราชพยาบาล

ความถูกต้องสมบูรณ์ของข้อมูล (Integrity) เป็นสิ่งที่สะท้อนถึงความน่าเชื่อถือของระบบสารสนเทศของคณะแพทยศาสตร์ศิริราชพยาบาล ส่งผลถึงความน่าเชื่อถือข้อมูลต้องไม่ถูกแก้ไขโดยผู้ที่ไม่ได้รับอนุญาตในระหว่างและหลังจากการบันทึก ทั้งนี้ข้อมูลมีความถูกต้องสมบูรณ์เพียงใดนั้น ต้องมีกลไกการตรวจสอบสิทธิ์หรือการได้รับอนุญาตให้ดำเนินการเปลี่ยนแปลงแก้ไขหรือกระทำการใดต่อข้อมูล เช่นการเก็บประวัติการแก้ไขข้อมูล เพื่อให้เจ้าหน้าที่ปฏิบัติงานสามารถตรวจสอบและติดตามการแก้ไขข้อมูลได้หากข้อมูลถูกเปลี่ยนแปลงหรือแก้ไขโดยผู้ไม่ประสงค์ดีส่งผลเสียต่อเจ้าของข้อมูล ความน่าเชื่อถือต่อคณะแพทยศาสตร์ศิริราชพยาบาลลดน้อยลงและอาจนำไปสู่การฟ้องร้องทางกฎหมาย

ความพร้อมใช้ของระบบสารสนเทศ (Availability) คือการมั่นใจว่าระบบสารสนเทศที่เกี่ยวข้องกับการให้บริการผู้ป่วยสามารถใช้งานได้เมื่อต้องการ คณะแพทยศาสตร์ศิริราชพยาบาลมีนโยบายและแผนการกู้คืนระบบจากความเสียหาย (Disaster Recovery Plan) อย่างทันทั่วทั้งที่ ระบบคอมพิวเตอร์ ระบบเครือข่าย และระบบไฟฟ้ามีความจำเป็น หากอุปกรณ์ใดขัดข้องและไม่สามารถกู้คืนได้ทันที จะทำให้การให้บริการหยุดชะงัก เหตุการณ์ที่ทำให้ระบบไม่พร้อมใช้งาน ได้แก่ การปฏิเสธการให้บริการ (Denial of Service) และระบบคอมพิวเตอร์ทำงานด้อยประสิทธิภาพ (Loss of data processing capability) ดังนั้นต้องตรวจสอบและประเมินสมรรถนะของระบบสารสนเทศและอุปกรณ์สนับสนุนอย่างสม่ำเสมอเพื่อให้มั่นใจว่าระบบพร้อมใช้งานอยู่เสมอ

## มาตรฐานด้านความมั่นคงปลอดภัยของส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API)

แนวปฏิบัติที่ดีสำหรับมาตรฐานด้านความมั่นคงปลอดภัยของส่วนต่อประสานโปรแกรมประยุกต์ เพื่อให้การเชื่อมโยงข้อมูลมีความถูกต้อง รวดเร็วและแม่นยำ เป็นการเปิดกว้างเพื่อเป็นส่วนต่อขยายสำหรับการพัฒนา

ระบบสารสนเทศให้รวดเร็วและมีความปลอดภัยมากขึ้น (ธนาคารแห่งประเทศไทย, 2565) ประกอบไปด้วยกระบวนการดังนี้

1. กระบวนการยืนยันตัวตน (Authentication) ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องจัดให้มีกระบวนการในการยืนยันตัวตนของผู้ใช้บริการ ส่วนต่อประสานโปรแกรมประยุกต์ ที่รัดกุม ปลอดภัย เหมาะสมกับลักษณะและความเสี่ยงของบริการที่ใช้ เพื่อควบคุมการเข้าถึงข้อมูลของผู้ใช้บริการ โดยต้องมีกระบวนการในการบริหารจัดการเกี่ยวกับการยืนยันตัวตน เช่น การสร้าง บริหารจัดการ การแสดงผล เพิกถอน ต่ออายุ และการให้สิทธิ์แก่ปัจจัยและข้อมูลที่ใช้ในการยืนยันตัวตนอย่างเหมาะสมปลอดภัย เพื่อป้องกันไม่ให้ข้อมูลและปัจจัยในการยืนยันตัวตนถูกแก้ไขเปลี่ยนแปลงหรือผู้อื่นแอบอ้างใช้งานได้ ในที่นี้จะกล่าวรวมถึงการยืนยันตัวตนทั้งด้านผู้ให้บริการและด้านระบบผู้ให้บริการ

กระบวนการยืนยันตัวตนทางด้านผู้ให้บริการ (Client Authentication) การยืนยันตัวตนทางฝั่งผู้ให้บริการหรือระบบต้นทางที่ต้องการเชื่อมต่อไปอีกระบบหนึ่ง เพื่อขอใช้บริการต่างๆ แนวทางที่พัฒนาใช้ส่วนใหญ่มีวิธีดังนี้

1.1 การยืนยันตัวตนโดยใช้ชื่อผู้ใช้งาน (Username) และรหัสผ่าน (Password) คือรูปแบบการยืนยันตัวตนที่ง่ายที่สุด อย่างไรก็ตามการใช้ชื่อผู้ใช้งาน (Username) และรหัสผ่าน (Password) ต้องมีการบริหารจัดการให้มีความมั่นคงปลอดภัยตั้งแต่การสร้างรหัสผ่านต้องคาดเดาได้ยากและการเปลี่ยนรหัสผ่านอย่างสม่ำเสมอ หรือเมื่อพบว่าข้อมูลรหัสผ่านมีความเสี่ยงจากเหตุการณ์ข้อมูลรั่วไหลและการป้องกันความลับของข้อมูลรหัสผ่านขณะส่ง ด้วยเหตุนี้จึงต้องใช้ร่วมกับใบรับรองอิเล็กทรอนิกส์ (Certificate) เพื่อให้เกิดการรักษาความมั่นคงปลอดภัยของข้อมูลที่ใช้ในการยืนยันตัวตน

1.2 การยืนยันตัวตนโดยใช้ Token-based คือรูปแบบการยืนยันตัวตนเปรียบเสมือนลายเซ็นดิจิทัลที่ใช้รหัส Token แทนการส่งด้วยข้อมูลชื่อผู้ใช้งาน (Username) และรหัสผ่าน (Password) โดยตรง เช่น JWT (JSON Web Tokens), API Key เป็นต้น ข้อดีของวิธีการยืนยันตัวตนโดยใช้ Token-based คือ การยืนยันตัวตนมีความมั่นคงปลอดภัยมากขึ้น เนื่องจากระบบจะมีการสร้างค่า Token สำหรับใช้งานแต่ละครั้งและสามารถกำหนดการใช้งานในระยะเวลาที่จำกัด ทั้งนี้ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องดูแลข้อมูลที่ใช้ยืนยันตัวตนสำหรับการใช้งานส่วนต่อประสานโปรแกรมประยุกต์ของผู้ให้บริการด้วย

ทั้งนี้ต้องไม่ประกอบด้วยข้อมูลที่มีความอ่อนไหว (Sensitive Data) หรือข้อมูลที่บุคคลอื่นสามารถเข้าถึงได้ การรับส่งข้อมูลที่ใช้ยืนยันตัวตนต้องทำผ่านช่องทางที่มีความมั่นคงปลอดภัย เช่น โพรโทคอลความปลอดภัยที่เข้ารหัสอีเมลเพื่อความเป็นส่วนตัว โดยป้องกันการเข้าถึงอีเมลโดยไม่ได้รับอนุญาตเมื่อส่งอีเมลผ่านการเชื่อมต่ออินเทอร์เน็ต (Transport Layer Security: TLS) เป็นต้น ทั้งนี้ต้องมีกลไกในการตรวจสอบว่าข้อมูลหรือปัจจัยที่ใช้ในการยืนยันตัวตนดังกล่าวไม่ถูกเพิกถอนหรือหมดอายุแล้ว นอกจากนี้ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องมีการบริหารจัดการ Session ที่เหมาะสม มีระยะเวลาสั้นเท่าที่จำเป็นต่อการทำธุรกรรมและตรวจสอบเป็นประจำเพื่อยกเลิก Session ที่ไม่ถูกใช้งานแล้วที่ค้างอยู่ในระบบ

1.3 กระบวนการยืนยันตัวตนทางด้านผู้ให้บริการ (Server Authentication) กระบวนการที่ผู้ให้บริการตรวจสอบความถูกต้องของผู้ให้บริการว่าเป็นผู้ให้บริการตัวจริง สำหรับการขอใช้บริการโดยแนวทางที่พัฒนาใช้กัน

ส่วนใหญ่คือการตรวจสอบใบรับรองอิเล็กทรอนิกส์ (Certificate) ของผู้ให้บริการ ต้องดำเนินการอย่างน้อยในเรื่องดังต่อไปนี้

- ตรวจสอบว่าใบรับรองอิเล็กทรอนิกส์ถูกออกโดย CA (Certification Authority) ที่เชื่อถือได้
- ตรวจสอบว่าใบรับรองอิเล็กทรอนิกส์ไม่หมดอายุ
- ตรวจสอบว่าใบรับรองอิเล็กทรอนิกส์ไม่ถูกเพิกถอนจากเครื่องแม่ข่าย (Server)
- ตรวจสอบว่าใบรับรองอิเล็กทรอนิกส์มีชื่อตรงกับชื่อโดเมนที่ใช้บริการ

ตัวอย่างของมาตรฐานสากลหรือแนวปฏิบัติที่ดีด้านความมั่นคงปลอดภัยที่ใช้อ้างอิง เช่น มาตรฐาน OpenID Connect หรือ SAML (Security Assertion Markup Language) เป็นต้น

2. กระบวนการตรวจสอบสิทธิ์การใช้งาน (Authorization) คือการกำหนดทรัพยากรที่ผู้ใช้สามารถเข้าถึงได้ ผู้ให้บริการ API ต้องจัดการและตรวจสอบสิทธิ์อย่างเข้มงวด ให้ผู้ใช้เข้าถึงเฉพาะข้อมูลหรือฟังก์ชันที่มีสิทธิ์เท่านั้น เช่น ผู้ใช้ที่มีสิทธิ์อ่านอย่างเดียวไม่ควรเข้าถึงฟังก์ชันของผู้ดูแลระบบ ผู้ให้บริการ API ต้องยืนยันตัวตนและให้สิทธิ์แก่ผู้ใช้ทุกครั้งก่อนการเข้าถึงข้อมูลและบริการ โดยเหมาะสมกับลักษณะของผู้ใช้ เช่น ระบบคอมพิวเตอร์ นิติบุคคล หรือบุคคลธรรมดา มาตรฐานที่ใช้ เช่น OAuth 2.0 หากใช้การยืนยันตัวตนและการให้สิทธิ์ด้วย Token-Based ควรคำนึงถึงความปลอดภัยของ Token และป้องกันการโจมตีซ้ำ (Replay Attack) เช่น ใช้การเข้ารหัสที่แข็งแกร่ง กำหนดอายุการใช้งาน Token แบบสั้น หรือใช้ค่า Nonce เพื่อเพิ่มความปลอดภัย

3. การรักษาความลับของข้อมูลและการตรวจสอบความถูกต้องของข้อมูล (Data Confidentiality and Integrity) ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องป้องกันการเปิดเผยข้อมูลเกินความจำเป็น (Excessive Data Exposure) โดยควบคุมข้อมูลที่ส่งผ่านส่วนต่อประสานโปรแกรมประยุกต์ให้เหมาะสมกับการใช้งาน และป้องกันการฝังข้อมูลสำคัญ เช่น รหัสผ่าน หรือกุญแจเข้ารหัสใน Source Code หรือ API Specification แนวทางการรักษาความมั่นคงปลอดภัยของข้อมูลต้องประกอบด้วยหลักการเหล่านี้ (Warkentin & Orgeron, 2020)

3.1 การรักษาความลับของข้อมูล (Data Confidentiality) คือการเก็บข้อมูลเป็นความลับและอนุญาตให้เฉพาะผู้ได้รับอนุญาตเท่านั้นที่สามารถเข้าถึงได้ โดยข้อมูลอ่อนไหวที่รับส่งระหว่างส่วนต่อประสานโปรแกรมประยุกต์ ผู้ให้บริการ และผู้ให้บริการต้องถูกเก็บเป็นความลับด้วยการเข้ารหัสที่รัดกุม ใช้กุญแจเข้ารหัสที่เหมาะสมและควบคุมสิทธิ์การเข้าถึงอย่างเข้มงวดเพื่อป้องกันการเข้าถึงโดยบุคคลที่ไม่ได้รับอนุญาต

3.2 การตรวจสอบความถูกต้องของข้อมูล (Data Integrity) คือการตรวจสอบข้อมูลที่รับส่งระหว่าง ส่วนต่อประสานโปรแกรมประยุกต์สำหรับผู้ให้บริการและผู้ให้บริการข้อมูลต้องมีความถูกต้องและสมบูรณ์ครบถ้วนไม่ถูกเปลี่ยนแปลงแก้ไขจากบุคคลภายนอกหรือบุคคลที่ไม่ได้รับอนุญาต เพื่อความปลอดภัยสำหรับการรักษาความลับของข้อมูลและการตรวจสอบความถูกต้องของข้อมูล คือการรับส่งข้อมูลผ่านช่องทางที่เชื่อมต่อบนด้วยโปรโตคอลที่เข้ารหัสเท่านั้น (HTTPS) เพื่อป้องกันการถูกดักจับและแก้ไขข้อมูลสำคัญระหว่างทางและต้องอนุญาตการเชื่อมต่อเฉพาะเทคโนโลยีที่มีความมั่นคงปลอดภัยตามมาตรฐานสากล เช่น TLS 1.2 สำหรับ Transport Layer Security และสำหรับการรับส่งข้อมูลที่มีความสำคัญสูง (High Risk Data) ควรพิจารณาถึงแนวทางการป้องกันการเข้ารหัสในระดับแอปพลิเคชัน (Application) เพื่อยกระดับการป้องกันข้อมูลที่มีความสำคัญสูงไว้หลาย เช่น ตามมาตรฐาน JWT

(JSON Web Token), JWS (JSON Web Signature), JWE (JSON Web Encryption) และ WS-Security (Web Services Security) สำหรับ Message Level Integrity and Encryption เป็นต้น

4. การรักษาความมั่นคงปลอดภัยด้านการสื่อสาร (Secure Communication) คือ การสื่อสารระหว่างส่วนต่อประสานโปรแกรมประยุกต์ผู้ใช้บริการและผู้ให้บริการควรเกิดขึ้นหลังจากการตรวจสอบความถูกต้องร่วมกันและผ่านช่องทางกาเข้ารหัสที่เป็นมาตรฐาน เช่น การใช้โปรโตคอล Transport Layer Security: TLS ร่วมกัน การติดตั้งใบรับรองอิเล็กทรอนิกส์ (Certificate) ที่ถูกต้องและเป็นปัจจุบันตามนโยบายความมั่นคงปลอดภัยของคณะแพทยศาสตร์ศิริราชพยาบาล บนเครื่องแม่ข่ายส่วนต่อประสานโปรแกรมประยุกต์ (Server API) หรือ API Gateway รวมถึงการบังคับใช้ Transport Layer Security: TLS และอนุญาตเฉพาะชุดการเข้ารหัสที่มีความแข็งแกร่ง Strong Cipher Suites รวมถึงการยกเลิกหรือเลิกใช้ชุดเข้ารหัสที่ไม่ปลอดภัย

5. การพัฒนาโปรแกรมและการกำหนดค่าที่ปลอดภัย (Secure Coding and Configuration) ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องมีแนวทางการพัฒนาโปรแกรมและแนวทางการตั้งค่าความมั่นคงปลอดภัยที่เกี่ยวข้องกับการใช้งานส่วนต่อประสานโปรแกรมประยุกต์โดยยึดหลักตาม Least Functionality (Zeng & Zhu, 2021) เป็นการกำหนดให้มีการใช้งานเฉพาะเท่าที่จำเป็นครอบคลุมหลักการดังต่อไปนี้

5.1 Data Validation ผู้ให้บริการ API ต้องตรวจสอบข้อมูลทั้งในกรณีที่ได้รับมาจากผู้ใช้บริการ (Request) และกรณีที่ส่งออกไปให้ผู้ให้บริการ (Response) เพื่อให้มั่นใจว่าข้อมูลและพารามิเตอร์ถูกต้องและครบถ้วน สอดคล้องกับข้อกำหนดของรูปแบบข้อมูล (Data format validation) และข้อกำหนดของพารามิเตอร์ เช่น การตรวจสอบประเภทของข้อมูลว่าตรงตามข้อกำหนดหรือไม่ วิธีการเหล่านี้ช่วยคัดกรองข้อมูลที่ผิดปกติหรือไม่เข้ากับการทำงานของระบบ เพื่อลดความเสี่ยงจากข้อผิดพลาด และมีแนวทางป้องกันการโจมตีประเภท Injection เช่น SQL Injection, OS Command Injection เป็นต้น

5.2 Security Header ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องมีแนวทางการตั้งค่าความมั่นคงปลอดภัยต่างๆ ของรายการเรียกใช้และรายการตอบสนอง (Request and Response Headers) ที่เกี่ยวข้องกับการใช้งานส่วนต่อประสานโปรแกรมประยุกต์ให้เหมาะสม ดังนี้

5.2.1 Cross-Origin-Resource-Sharing คือ กลไกการจำกัดการเข้าถึงข้อมูลจากการร้องขอโดเมนอื่นนอกเหนือจากโดเมนที่ให้บริการ

5.2.2 HTTP Verbs คือ การกำหนด HTTP Request Method ที่อนุญาตให้ดำเนินการได้และปิด HTTP Request Method อื่นที่ไม่จำเป็น

5.2.3 การกำหนด Content-Type คือ การตรวจสอบ Content Type ของคำสั่งการเรียกใช้งานต้องระบุอย่างถูกต้อง

5.2.4 Error Message ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องมีกลไกในการจัดการความผิดพลาดของระบบ โดยต้องจัดการการแสดงผลของข้อผิดพลาด (Error) ทั้งหมดที่อาจเกิดขึ้น โดยต้องไม่แสดงข้อมูลรายละเอียดของข้อผิดพลาดมากเกินไปเนื่องจากข้อมูลบางอย่างอาจส่งผลกระทบต่อความมั่นคงปลอดภัยของระบบสารสนเทศ

6. การจัดเก็บข้อมูลบันทึกเหตุการณ์ และการเฝ้าระวัง (Audit Log and Monitoring) ผู้ให้บริการ API ต้องจัดเก็บข้อมูลบันทึกเหตุการณ์ (Audit log) ของการใช้งาน API โดยมีรายละเอียดครบถ้วนเพียงพอสำหรับการตรวจสอบ สามารถระบุตัวบุคคลผู้กระทำได้ เช่น ใครทำอะไร ที่ไหน และเมื่อไหร่ ตามกฎหมายคอมพิวเตอร์จะต้องเก็บข้อมูลเหล่านี้ไม่น้อยกว่า 90 วัน และบันทึกเวลาอิเล็กทรอนิกส์ (Timestamp) อย่างถูกต้อง นอกจากนี้ต้องมีมาตรการป้องกันการแก้ไข Audit log โดยไม่ได้รับอนุญาต รวมถึงการป้องกันไม่ให้ผู้ดูแลระบบปรับเปลี่ยนข้อมูล Audit log ได้ ต้องมั่นใจว่าข้อมูลบันทึกเหตุการณ์จะไม่ถูกแก้ไขและสามารถระบุตัวตนผู้เข้าถึงข้อมูลได้อย่างชัดเจน การจัดเก็บข้อมูลบันทึกเหตุการณ์ที่มีข้อมูลอ่อนไหว (Sensitive Data) หรือข้อมูลลับ ต้องปฏิบัติตามแนวทางการรักษาความปลอดภัยตามระดับชั้นความลับของข้อมูล นอกจากนี้ควรมีมาตรการตรวจจับ แจ้งเตือน และตอบสนองต่อพฤติกรรมผิดปกติ เช่น การพยายามยืนยันตัวตนที่ล้มเหลว การปฏิเสธการเข้าถึง ข้อผิดพลาดในการตรวจสอบอินพุต ข้อผิดพลาดของพารามิเตอร์การเรียกใช้ Token ซ้ำหลายครั้ง พฤติกรรมการโจมตี Session รวมถึงการเกิดเหตุขัดข้อง และเฝ้าระวังปริมาณการเรียกใช้งานจำนวนมากที่อาจส่งผลกระทบต่อการใช้งานบริการ เช่น ช่วงสิ้นเดือนที่มีการใช้งานมาก (Srinivasan, 2022)

7. ความเพียงพอของทรัพยากร (Resource Sufficiency) ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องมีการจัดการทรัพยากรอย่างมีประสิทธิภาพ มีมาตรการตรวจสอบให้แน่ใจว่า API ได้รับการปกป้องจากการใช้งานทรัพยากรที่มากเกินไปและบริการส่วนต่อประสานโปรแกรมประยุกต์จะไม่หยุดชะงัก โดยแนวทางการจัดการทรัพยากรให้เกิดความเพียงพอต้องดำเนินการอย่างน้อยในเรื่องดังต่อไปนี้

7.1 Rate Limit และ Circuit Breaker ในการควบคุมการใช้ทรัพยากรของเครื่องแม่ข่ายอย่างมีประสิทธิภาพ ผู้ให้บริการและผู้ใช้บริการ API ต้องจำกัดจำนวนการส่งข้อมูลต่อวินาทีและขนาดของข้อมูลที่รับและประมวลผลได้ เพื่อป้องกันการใช้ทรัพยากรเกินความสามารถของระบบและป้องกันการโจมตี DoS เช่น การกำหนดปริมาณการเรียกใช้งานรายบุคคลหรือรายบริการ และการกำหนดขนาดของข้อมูลที่รับส่ง นอกจากนี้ ควรมีแนวทางในการตัดการเชื่อมต่อเมื่อระบบมีการเรียกใช้งาน API จากภายนอกที่ตอบสนองช้าหรือไม่ตอบสนอง และการกลับมาให้บริการเมื่อสถานการณ์ปกติ โดยคำนึงถึงความมั่นคงและความเหมาะสมกับการใช้งานธุรกิจ

7.2 DDoS Protection ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องมีแนวทางเชิงป้องกันภัย DDoS Attack ในระดับเครือข่ายหรือควรพิจารณาเพิ่มเติม DDoS Protection ในระดับแอปพลิเคชันสำหรับการเปิดให้บริการส่วนต่อประสานโปรแกรมประยุกต์ที่มีความสำคัญ มีลักษณะความต้องการความพร้อมใช้งาน (Availability) เช่น การใช้ WAF หรือมาตรการอื่น ๆ ที่สามารถป้องกันได้ เพื่อป้องกันระบบถูกโจมตีจนไม่สามารถให้บริการได้ เช่น HTTPs Flood, Application-Level Exploitation เป็นต้น

7.3 Request และ Response Size ผู้ให้บริการส่วนต่อประสานโปรแกรมประยุกต์ต้องกำหนดขนาดของข้อความในการร้องขอบริการ (Request Payload Size) ให้เหมาะสมกับการให้บริการ เพื่อป้องกันการใช้งานทรัพยากรที่มากเกินไปและระบบหยุดชะงัก นอกจากนี้ควรกำหนดปริมาณข้อมูลที่จะส่งกลับไปให้ผู้ใช้บริการในแต่ละคำขอ (Number of Records Per Page to Return) เช่น การใช้ Pagination เพื่อจำกัดจำนวนแถวข้อมูลต่อหน้า ควรหลีกเลี่ยงการใช้มาตรฐานหรือแนวปฏิบัติที่ล้าสมัยหรือมีช่องโหว่ด้านความมั่นคงปลอดภัย และอาจพิจารณาใช้มาตรฐานการควบคุมอื่น ๆ ที่เกี่ยวข้อง เช่น มาตรฐาน PCI-DSS เป็นต้น

## การประยุกต์ใช้ส่วนต่อประสานโปรแกรมประยุกต์ (API)

### ของฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล

การประยุกต์ใช้ API ในปัจจุบันเป็นที่นิยม เนื่องจากอยู่ในกลุ่มของ HTTP และขยายไปสู่รูปแบบ XML และ JSON ที่มักใช้ใน Web Service เช่น SOAP และ REST โดย SOAP (Simple Object Access Protocol) ใช้ XML ในการส่งข้อมูล และ REST (Representational State Transfer) ใช้ทั้ง XML และ JSON ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล ได้พัฒนาส่วนต่อประสานโปรแกรมประยุกต์ API เพื่อสนับสนุนระบบสารสนเทศทางการแพทย์ เช่น ระบบข้อร้องเรียนและรายงานอุบัติการณ์ โดยเรียกใช้ข้อมูลจากระบบต่างๆ ผ่าน API เช่น ข้อมูลส่วนตัวผู้ป่วย จากระบบ MRS, ข้อมูลยา จากระบบ Drug Master ดังภาพที่ 3

The screenshot shows a web form titled "HN Information". It contains the following fields and values:

- HN \***: 526
- เพศ \***: Male (ชาย)
- อายุ (ปี) \***: 74
- อายุ (วัน)**: 0
- ภาควิชา \***: General Internal Medicine (งานทันตกรรม)
- VN/AN**: (Empty)
- ชื่อ-สกุลที่ตรงกับ HN \***: Mr. (นาย)
- อายุ (เดือน)**: 1
- ชื่อแพทย์ \***: Dr. (พว.)
- สาขา \***: Dermatology (เคมีคลินิก)
- ผู้ป่วยหลัก**: (Unchecked checkbox)
- Buttons**: Close, Save

ภาพที่ 3 แสดงการเชื่อมโยงข้อมูลผู้ป่วยจากระบบ MRS

ระบบที่เรียกใช้ข้อมูลจากระบบต้นทางจะได้รับข้อมูลที่เป็นปัจจุบันอยู่เสมอ เนื่องจากไม่มีการจัดเก็บหรือทำซ้ำข้อมูลไว้ในระบบเอง แต่จะทำการเรียกใช้ข้อมูลจากระบบต้นทางโดยตรงทุกครั้งที่มีความต้องการใช้งานข้อมูล วิธีการนี้ช่วยให้ข้อมูลที่ใช้กันระหว่างระบบมีความถูกต้อง สอดคล้องกัน และเป็นข้อมูลล่าสุดอยู่เสมอ อีกทั้งยังลดความเสี่ยงจากความคลาดเคลื่อนของข้อมูลที่เกิดจากการใช้ข้อมูลที่ล้าสมัยหรือซ้ำซ้อน เป็นต้น

สำหรับแนวทางในการประยุกต์ใช้และพัฒนา API สำหรับระบบข้อร้องเรียนและรายงานอุบัติการณ์ ซึ่งมีการเรียกใช้ข้อมูลจากระบบอื่น เช่น ระบบเวชระเบียน (MRS) และระบบฐานข้อมูลยา (Drug Master) จำเป็นต้องมีการวางแผนและดำเนินงานอย่างเป็นขั้นตอน

เริ่มต้นจากการวิเคราะห์ถึงความต้องการของ API เพื่อทำหน้าที่อะไร และต้องใช้ข้อมูลใดจากระบบใดบ้าง เช่น การดึงข้อมูลผู้ป่วยหรือข้อมูลยา จากนั้นจึงทำการ ออกแบบ API โดยกำหนด Endpoints, วิธีการเรียกใช้งาน (GET/POST) และรูปแบบของข้อมูล (JSON)

ในขั้นตอนถัดมาสำหรับการพัฒนา API คือการเลือกใช้เทคโนโลยีและ Framework ที่เหมาะสม พร้อมจัดการด้านความปลอดภัย เช่น การใช้ Token ในการเข้าถึง รวมถึงรองรับการจัดการข้อผิดพลาดอย่างเหมาะสม

หลังจากพัฒนาแล้วต้องมีการทดสอบการทำงาน ทั้งในแง่ของความถูกต้องและประสิทธิภาพ เช่น ตรวจสอบว่า API เชื่อมต่อกับ MRS หรือ Drug Master ได้อย่างเสถียร

สุดท้ายคือการนำ API ไปใช้งานจริง (Deploy) และมีการติดตามผลการทำงานอย่างต่อเนื่อง เพื่อให้มั่นใจว่า API สามารถรองรับการใช้งานจริง และสามารถดูแลปรับปรุงต่อไปในอนาคตได้อย่างมีประสิทธิภาพ

การประยุกต์ใช้ API ในฝ่ายสารสนเทศของคณะแพทยศาสตร์ศิริราชพยาบาลมุ่งพัฒนาเพื่อเชื่อมต่อและแลกเปลี่ยนข้อมูลระหว่างเว็บแอปพลิเคชัน ผ่านโปรโตคอล HTTP เช่น REST API (Representational State Transfer) หรือ SOAP API (Simple Object Access Protocol) การเลือกใช้ API จะพิจารณาจากประสิทธิภาพและปริมาณข้อมูล หากต้องการแลกเปลี่ยนข้อมูลมากหรือมีประสิทธิภาพสูง มักใช้ REST API ที่ใช้โปรโตคอล HTTP (Hypertext Transfer Protocol) เป็นมาตรฐานในการสื่อสารข้อมูลบนเว็บและอินเทอร์เน็ต เพื่อส่งคำขอ เช่น GET, POST, PUT, DELETE

คณะแพทยศาสตร์ศิริราชพยาบาลได้ประยุกต์ใช้งานส่วนต่อขยายโปรแกรมประยุกต์ในการพัฒนาระบบฐานข้อมูลเวชภัณฑ์ (Medical Supply Master) เพื่อสนับสนุนการทำงานของเจ้าหน้าที่ฝ่ายเภสัชกรรม คณะแพทยศาสตร์ศิริราชพยาบาล โดยการเชื่อมโยงข้อมูลเวชภัณฑ์ระหว่างระบบฐานข้อมูลเวชภัณฑ์ และระบบจ่ายยา และเวชภัณฑ์สำหรับผู้ป่วยนอก (SiOP) โดยเภสัชกรที่ใช้งานระบบ SiOP ทำหน้าที่ในการสั่งยาและเวชภัณฑ์สำหรับผู้ป่วยนอกภายในโรงพยาบาลศิริราช ซึ่งการดำเนินการเชื่อมโยงข้อมูลเวชภัณฑ์ผ่านส่วนต่อขยายโปรแกรมประยุกต์ระหว่าง 2 ระบบนี้ เมื่อผู้ใช้งานทำการปรับปรุงข้อมูลเวชภัณฑ์ เช่น ราคาขาย เพดานราคาและคุณสมบัติของเวชภัณฑ์ จากระบบฐานข้อมูลเวชภัณฑ์ (Medical Supply Master) ทำให้ระบบ SiOP เป็นระบบปลายทางได้รับข้อมูลได้อย่างรวดเร็ว ถูกต้องและแม่นยำ ทำให้ผู้ป่วยที่มารับการบริการได้รับความสะดวก รวดเร็ว มีความพึงพอใจในการให้บริการของคณะแพทยศาสตร์ศิริราชพยาบาลมากยิ่งขึ้น

## ประโยชน์ของการของการนำส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API)

ในปัจจุบันนี้คณะแพทยศาสตร์ศิริราชพยาบาลได้มีการพัฒนาระบบสารสนเทศทางการแพทย์อย่างมากมาย และเทคโนโลยีส่วนต่อประสานประยุกต์โปรแกรมได้เข้ามามีบทบาทสำคัญในการเชื่อมโยงข้อมูลระหว่างกันในการพัฒนาระบบสารสนเทศทางการแพทย์ ส่วนต่อประสานประยุกต์โปรแกรมได้มีประโยชน์ดังต่อไปนี้

1. หน่วยงานหรือองค์กรภายนอกที่ต้องการพัฒนาเว็บไซต์หรือแอปพลิเคชัน ที่ต้องการข้อมูลทางการแพทย์ สามารถเรียกใช้ส่วนต่อประสานโปรแกรมประยุกต์ที่คณะแพทยศาสตร์ศิริราชพยาบาลได้มีการพัฒนาไว้แล้วทำให้เกิดความรวดเร็วในการพัฒนาระบบสารสนเทศทางการแพทย์มากขึ้น
2. มีความยืดหยุ่นในกรณีที่มีการเพิ่มเติมหรือปรับเปลี่ยนโครงสร้างของข้อมูล เนื่องจากการเพิ่มเติมปรับเปลี่ยนสามารถได้อย่างสะดวกและมีประสิทธิภาพ ทำให้ช่วยลดเวลาในการพัฒนาระบบสารสนเทศทางการแพทย์ของคณะแพทยศาสตร์ศิริราชพยาบาล

3. มีมาตรฐานความปลอดภัยในการใช้งาน โดยการตรวจสอบสิทธิ์การใช้งาน (Authorization) จะยืนยันตัวบุคคลด้วยการเข้ารหัสที่ได้มาตรฐาน ส่วนต่อประสานโปรแกรมประยุกต์มีกลไกจัดการและตรวจสอบสิทธิ์ก่อนเสมอ ควบคุมการเข้าถึงอย่างเข้มงวดให้ผู้ให้บริการเข้าถึงเฉพาะข้อมูลหรือฟังก์ชันที่มีสิทธิ์เท่านั้น

4. ลดความซ้ำซ้อน และความผิดพลาดของข้อมูลทางการแพทย์เป็นข้อมูลที่มีความสำคัญในการให้บริการ

5. การปรับปรุงข้อมูลสามารถทำได้ที่ระบบต้นทางทำให้ข้อมูลที่ถูกนำไปใช้มีความถูกต้องและเป็นปัจจุบัน

## ข้อดีของการนำส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API)

### มาใช้ในฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล

การนำส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) มาใช้ในฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล ช่วยเพิ่มประสิทธิภาพในการพัฒนาและบูรณาการระบบสารสนเทศทางการแพทย์ได้อย่างมีนัยสำคัญ โดย API ช่วยให้การพัฒนาเว็บไซต์และระบบต่างๆ ดำเนินไปอย่างรวดเร็ว ลดความซ้ำซ้อนของข้อมูลระหว่างระบบ เช่น ข้อมูลผู้ป่วยหรือข้อมูลยา อีกทั้งยังรองรับการให้บริการผู้ป่วยแบบทันเวลา เช่น การแจ้งเตือนคิวเจาะเลือด คิวห้องตรวจ คิวรับยา และการชำระเงินผ่านแอปพลิเคชัน ซึ่งช่วยอำนวยความสะดวกให้แก่ผู้ป่วยและบุคลากรทางการแพทย์ นอกจากนี้การเชื่อมโยงข้อมูลผ่าน API ยังสนับสนุนการทำงานเชิงรุก เช่น ระบบข้อร้องเรียนและอุบัติการณ์สามารถแจ้งเตือนเหตุการณ์ความเสี่ยง และเอื้อต่อการขยายระบบในอนาคตได้อย่างยืดหยุ่นและปลอดภัย

## ปัญหาการใช้งานส่วนต่อประสานโปรแกรมประยุกต์

### (Application Programming Interface: API)

การนำส่วนต่อประสานโปรแกรมประยุกต์มาประยุกต์ใช้ในการปฏิบัติงานอาจก่อให้เกิดปัญหาดังต่อไปนี้

1. ข้อจำกัดของรูปแบบข้อมูล รูปแบบข้อมูลที่ซับซ้อนหรือไม่สอดคล้องกับความต้องการของแอปพลิเคชัน ดังนั้นก่อนการนำมาประยุกต์ใช้ต้องมีการแปลงรูปแบบข้อมูลหรือปรับเปลี่ยนโครงสร้างข้อมูลเพื่อให้เข้ากับส่วนต่อประสานโปรแกรมประยุกต์มาประยุกต์ที่จะใช้งาน

2. ความเสถียรของส่วนต่อประสานโปรแกรมประยุกต์มาประยุกต์ บ่อยครั้งมีเมื่อนำมาใช้งานแล้วพบว่ามีปัญหาเกี่ยวกับความเสถียรหรือความพร้อมใช้งาน อาจมีปัญหาคอนเนกชันช้าส่งผลให้แอปพลิเคชันไม่สามารถทำงานได้อย่างเต็มประสิทธิภาพ

3. ข้อจำกัดการเข้าถึง มีการจำกัดการเข้าถึงด้วยการให้สิทธิ์หรือการรับรองตัวตน ทำให้ผู้ใช้งานต้องมีการลงทะเบียนหรือขอสิทธิ์เพิ่มเติม เช่น การใช้ API Keys การตรวจสอบ Token หรือการตรวจสอบรหัสผ่าน

## สรุป

การนำระบบส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) เข้ามาประยุกต์ใช้ให้สอดคล้องกับการดำเนินงานทั้งภายในและภายนอกคณะแพทยศาสตร์ศิริราชพยาบาล เช่น การศึกษา

การรักษาพยาบาล และการวิจัย สอดคล้องกับพันธกิจหลักของคณะแพทยศาสตร์ศิริราชพยาบาลถือเป็นการเพิ่มประสิทธิภาพ ลดความยุ่งยากในการแลกเปลี่ยนข้อมูล และเพิ่มความเร็วในการพัฒนาเว็บไซต์หรือระบบสารสนเทศที่ให้บริการทางด้านการศึกษา การรักษาพยาบาลและการวิจัยทำให้เกิดประโยชน์กับผู้ป่วยและผู้ที่มีรับบริการกับคณะแพทยศาสตร์ศิริราชพยาบาลได้รับความสะดวกสบายเพิ่มขึ้นและเป็นการรักษาความปลอดภัยของข้อมูลสารสนเทศภายในของคณะแพทยศาสตร์ศิริราชพยาบาล เนื่องจากการเชื่อมต่อแบบ HTTPS (Hypertext Transfer Protocol Secure) คือโพรโทคอลการสื่อสารอินเทอร์เน็ตที่ช่วยรักษาความสมบูรณ์ของข้อมูลที่ส่งระหว่างคอมพิวเตอร์ของผู้ใช้กับเว็บไซต์ ข้อมูลที่ส่งด้วย HTTPS ได้รับการรักษาความปลอดภัยผ่านโพรโทคอล Transport Layer Security (TLS)

### ข้อเสนอแนะ

ในอนาคตหากคณะแพทยศาสตร์ศิริราชพยาบาลพัฒนาระบบส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) ที่สามารถให้บริการแก่หน่วยงานภายในและภายนอกเพื่อแลกเปลี่ยนข้อมูล ภายในคณะแพทยศาสตร์ศิริราชพยาบาล เช่น การเชื่อมต่อข้อมูลแพทย์ตารางเวรแพทย์ที่ออกตรวจ, ข้อการใช้ยา ทำให้เกิดความสะดวก รวดเร็ว และผู้ป่วยรวมถึงผู้มารับบริการจะได้รับประโยชน์สูงสุด ส่วนภายนอก เช่น การเชื่อมต่อข้อมูลนักเรียนระดับก่อนปริญญา และระดับบัณฑิตศึกษากับมหาวิทยาลัยมหิดลในการออกรายงานสำหรับผู้บริหารเพื่อตัดสินใจวางแผน และปรับนโยบายให้สอดคล้องกับการบริหารงานของคณะแพทยศาสตร์ศิริราชพยาบาล

### เอกสารอ้างอิง

- ธนาคารแห่งประเทศไทย. (2565). *แนวปฏิบัติการใช้เทคโนโลยี Application Programming Interface (API) ใน การให้บริการทางการเงิน*. ธนาคารแห่งประเทศไทย.
- ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล. (2562). *การจัดระดับชั้นความลับ และการจัดการข้อมูลตาม ระดับชั้นความลับ (Information Classification and Handling Procedure)*. ฝ่ายสารสนเทศ คณะ แพทยศาสตร์ศิริราชพยาบาล.
- สำนักนายกรัฐมนตรี. (2562). *ระเบียบว่าด้วยการรักษาความลับของทางราชการ*. <http://www.gad.moi.go.th/nsk-17-04-62-2366-2367-2368.pdf>
- Bondel, G., Landgraf, A., & Matthes, F. (2021). *API Management patterns for public, partner, and group web API initiatives with a focus on collaboration*. Proceedings of the 26<sup>th</sup> European Conference on Pattern Languages of Programs, Graz, Austria.
- Faruk, M. J. H., Patinga, A. J., Migiro, L., Shahriar, H., & Sneha, S. (2022). *Leveraging healthcare API to transform interoperability: api security and privacy* (pp. 444–445). Proceedings 2022 IEEE 46<sup>th</sup> Annual Computers, Software, and Applications Conference, California, USA.
- Geewax, J. J. (2021). *API design patterns*. Manning Publications Co.

- Gong, W., Lv, C., Duan, Y., Liu, Z., Khosravi, M. R., Qi, L., & Dou, W. (2021). *Keywords-driven web APIs group recommendation for automatic app service creation process*. *Software: Practice and Experience*, 51(11), 2337–2354.
- Richardson, L., Amundsen, M., Amundsen, M., & Ruby, S. (2013). *RESTful web APIs: Services for a changing world*. O'Reilly Media, Inc.
- Srinivasan, K. (2022). *Auditing capabilities in IT monitoring tools for security and compliance*. <https://www.eginnovations.com/blog/auditing-capabilities-in-it-monitoring-tools-for-security-and-compliance/>
- Warkentin, M., & Orgeron, C. (2020). *Using the security triad to assess blockchain technology in public sector applications*. *International Journal of Information Management*, 52, 102090.
- Zeng, M., & Zhu, F. (2021). Secure coding in five steps. *Journal of Cybersecurity Education, Research and Practice*, 2021(1), 5.