มาตรฐานรายการตรวจสอบการพัฒนา API เพื่อรองรับการพัฒนาระบบสารสนเทศให้เป็นไปตามมาตรฐานการรักษาความมั่นคงปลอดภัยทางไซเบอร์

แบบฟอร์มตรวจสอบการพัฒนา API ตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
1	การพิสูจน์ตัวตนและการอนุญาต (Authentication & Authorization) มาตรฐานที่อ้างอิง : OWASP 2021 (A01, A07), Zero Trust			
1.1	ใช้ JWT (JSON Web Tokens) ที่มีการกำหนดอายุการใช้งาน Token เช่น 15 นาที และมีระบบ Refresh Token เพื่ออัปเดต Token โดยไม่ต้องเข้าสู่ระบบใหม่			
1.2	รองรับ OAuth 2.0 และ OpenID Connect			
1.3	ใช้ API Keys สำหรับ Service-to-Service Authentication และต้องไม่เก็บ API Keys ไว้ในโค้ดหรือ Repository โดยตรง แต่ใช้ระบบจัดการความลับ (Secret Management) เช่น HashiCorp Vault, Bitwarden, Keycloak หรือ CyberArk Conjur			
1.4	กำหนด Rate Limiting เพื่อป้องกันการใช้งานเกินขีดจำกัด			

แบบฟอร์มตรวจสอบการพัฒนา API ตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
1.5	มีระบบ API Gateway สำหรับการจัดการการเข้าถึง			
2	การจัดการข้อมูล (Data Handling) มาตรฐานที่อ้างอิง : OWASP 2021 (A03), Cybersecurity Framework			
2.1	ตรวจสอบและกรองข้อมูล Input ทั้งหมด โดยนำ Library ในการพัฒนาระบบของ Framework ที่ใช้ หรือติดตั้ง Component มาใช้งานเพื่อป้องกัน SQL Injection และ Cross Site Scripting (XSS)			
2.2	เข้ารหัสข้อมูลที่ส่งผ่าน API ด้วย HTTPS			
2.3	จำกัดขนาดของ Request และ Response			
2.4	กำหนดรูปแบบข้อมูลที่ชัดเจนด้วย Schema Validation			
2.5	ป้องกันการรั่วไหลของข้อมูลในข้อความแจ้งข้อผิดพลาด ห้ามแสดงข้อมูลระบบ เช่น รายละเอียดข้อผิดพลาด (Stack Trace), โครงสร้างฐานข้อมูล (Database Schema) หรือชื่อเครื่องเซิร์ฟเวอร์ (Server Hostname)			
3	การป้องกันการโจมตี (Security Controls) มาตรฐานที่อ้างอิง : OWASP 2021 (A02, A05), Zero			

แบบฟอร์มตรวจสอบการพัฒนา API ตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
	Trust			
3.1	ใช้ Security Headers ที่เหมาะสม เช่น CORS หรือ CSP			
3.2	ป้องกัน API-specific Attacks เช่น Mass Assignment โดยการใช้ Allowlist สำหรับฟิลด์ที่อนุญาตให้ปรับปรุงลงใน Request Body			
3.3	มีระบบป้องกัน Brute Force Attacks			
3.4	ตรวจสอบและป้องกัน API-specific Injection Attacks เช่น ใช้ Parameterized Queries สำหรับการ Query ฐานข้อมูล หรือตรวจสอบ Input ด้วย Regular Expression เพื่อป้องกัน Command Injection			
3.5	มีระบบป้องกัน Man-in-the-Middle Attacks เช่น ต้องใช้งาน HTTPS พร้อมรองรับ TLS 1.2 เป็นอย่างต่ำ, ใช้ Certificate Pinning สำหรับ Mobile API			
4	การตรวจสอบและเฝ้าระวัง (Monitoring & Logging) มาตรฐานที่อ้างอิง : Cybersecurity Framework, Zero Trust			
4.1	บันทึก Log การเรียกใช้ API ทั้งหมด			

แบบฟอร์มตรวจสอบการพัฒนา API ตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
4.2	ติดตามและแจ้งเตือนการใช้งานที่ผิดปกติ โดยใช้ SIEM (Security Information and Event Management) เช่น OSSIM (Open Source Security Information Management), ELK Stack (Elasticsearch Logstash Kibana), Wazuh หรือกำหนด Threshold สำหรับแจ้งเตือน เช่น เรียก API เกิน 100 ครั้ง/นาที, เรียก API Endpoint เดียวกันเกิน 500 ครั้ง/วินาที			
4.3	ใช้ Unique Request IDs สำหรับการติดตามการใช้งาน			
4.4	มีระบบวิเคราะห์พฤติกรรมการใช้งาน API			
4.5	จัดเก็บ Metrics สำหรับการวิเคราะห์ประสิทธิภาพและความปลอดภัย เช่น Latency, Error Rate, Request Count หรือ API Usage by Endpoint			
5	การจัดการเวอร์ชัน (API Versioning) มาตรฐานที่อ้างอิง : OWASP 2021 (A04)			
5.1	กำหนดนโยบายการจัดการเวอร์ชันที่ชัดเจน			
	5.1.1 กำหนดรูปแบบการใช้ Version เช่น /api/v1/resource			

แบบฟอร์มตรวจสอบการพัฒนา API ตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
	5.1.2 มีแผนการยกเลิกการใช้งาน API เวอร์ชันเก่า			
	5.1.3 กำหนดระยะเวลารองรับเวอร์ชันเก่าล่วงหน้าเมื่อทราบว่าจะสิ้นสุดการใช้บริการ			
	5.1.4 แจ้งเตือนผู้ใช้งานล่วงหน้า 3 เดือน ก่อนยกเลิกเวอร์ชันเก่า			
5.2	รักษาความเข้ากันได้ย้อนหลัง (Backward Compatibility)			
5.3	จัดทำเอกสาร API ที่ครบถ้วนสำหรับทุกเวอร์ชัน			
6	การทดสอบ API (API Testing) มาตรฐานที่อ้างอิง : OWASP 2021 (A10)			
6.1	ทำการทดสอบความปลอดภัยของ API อย่างสม่ำเสมอ			
6.2	ใช้เครื่องมือทดสอบอัตโนมัติสำหรับ API Security เช่น OWASP ZAP, Postman หรือ Burp Suite			
6.3	ทดสอบการทำงานของ Rate Limiting และการป้องกันการโจมตี			
6.4	ทดสอบประสิทธิภาพและความทนทานของ API			
6.5	จำลองสถานการณ์การโจมตีเพื่อทดสอบการป้องกัน เช่น			

แบบฟอร์มตรวจสอบการพัฒนา API ตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
	ส่ง Malicious Payload เพื่อทดสอบ SQL Injection ทดสอบ DDoS ด้วยเครื่องมือ เช่น JMeter หรือ Locust			