LLDD BE - Integration with SBP Platform

SBP Mall - ระบบประกันรายได้ | Low Level Design Document

แนวทางการใช้เอกสาร

หัวข้อ	คำอธิบาย
วัตถุประสงค์	ใช้เป็นรายละเอียดระดับพัฒนาสำหรับออกแบบ ลงมือพัฒนา ตรวจทาน และทดสอบขอบเขตที่ระบุในฉบับนี้
ลำดับการอ่าน	เริ่มจาก Overview และ Scope จากนั้นตรวจ Field/Validation, Implementation, Contract, Processing Flow, Acceptance Criteria และ Test Checklist ตามลำดับ
ผลลัพธ์ที่คาดหวัง	ผู้อ่านสามารถระบุ input, ขั้นตอนประมวลผล, output, เงื่อนไขผิดพลาด และหลักฐานการทดสอบได้จากเนื้อหาในฉบับเดียว

คำว่า Input, Progress และ Output ในเอกสารนี้หมายถึงข้อมูลตั้งต้น ลำดับการทำงาน และผลลัพธ์ที่ตรวจสอบได้ของขอบเขตที่กำลังพัฒนา

1. Overview

รายการ	รายละเอียด
Track	BE
Estimate	20 ชั่วโมง (ไม่มี unit test แยก — ดูเหตุผลใน NO_UNIT_TEST_DOCS)
Owner	Tunyatorn <Vava> Kiatkongphongsa
Target repository	`SBP/srm-sps-spsap-store-backend` (NestJS + TypeORM · schema `sps_store`) + `SBP/srm-sps-spsap-sbp-bff` (forward ผ่าน client service · ไม่มี DB) สำหรับเส้นที่ FE เรียก
Objective	กำหนดวิธีที่ SBPGI ต่อกับแพลตฟอร์ม SBP เดิม: BFF header/ตัวตน, response envelope, ไฟล์บน S3, อีเมลผ่าน @gosoft-sbp/email-lib และค่ากำหนดกลางใน mas_param/common_code — เป็น blocker ที่ต้องปิดในสัปดาห์แรก

Common contract reference: ทุกหัวข้อ API/FE ต้องยึด LLDD-BE-API-Common-Contracts.pdf และ LLDD-FE-Integration-Contracts.pdf สำหรับ error/auth/format/pagination/action/RBAC ก่อนลงรายละเอียดเฉพาะหน้าหรือเฉพาะ endpoint

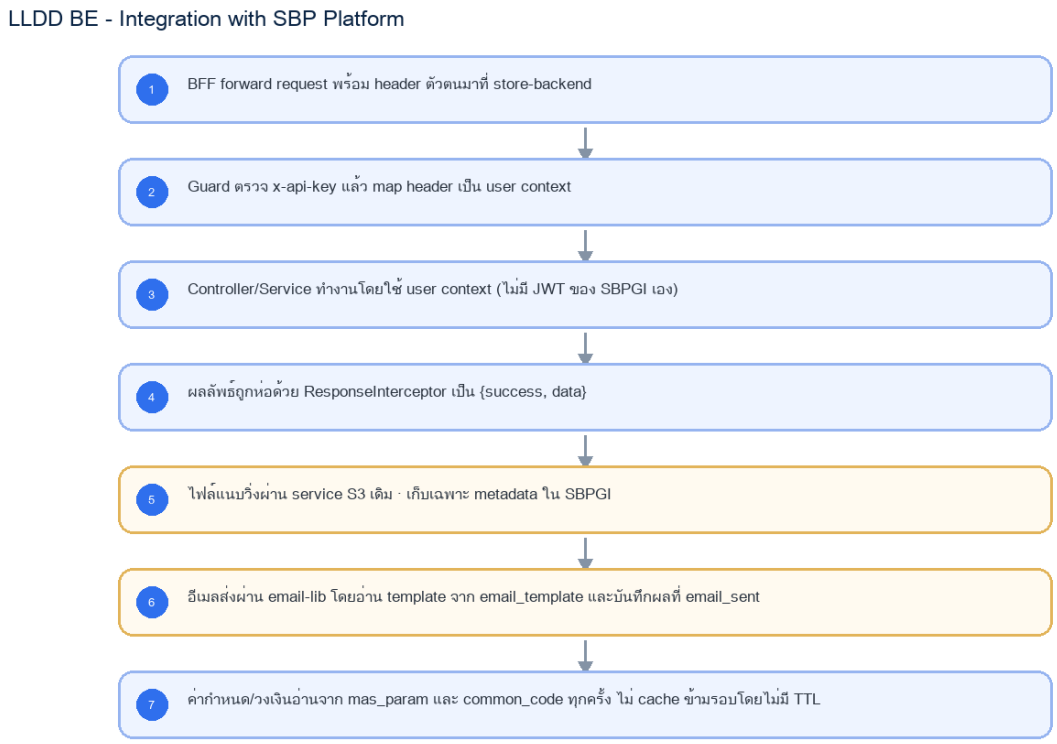
2. Screen / Functional Scope

- ตัวตนผู้ใช้จาก BFF header (x-api-key, x-user-id, x-user-group-id, x-user-permissions)
- Response envelope ของ store-backend: {success, data} / {success:false, data:null, error:{code,message}}
- ไฟล์แนบผ่าน service S3 เดิม (POST /statement/upload-file-aws · download-file-aws)
- อีเมลผ่าน @gosoft-sbp/email-lib + ตาราง email_template / email_sent
- ค่ากำหนดกลางที่ mas_param และ common_code (รวม SBPGI_APPROVE_LIMIT)
- การใช้ตาราง master ของระบบเดิม (store/mas_store · business_user · common_code) และปริมาณข้อมูลจริง

3. Screenshot Reference

ไม่มีภาพหน้าจอสำหรับหัวข้อนี้ — เป็นเอกสารฝั่ง Backend/Batch ที่ไม่มี UI (ภาพหน้าจอทั้งหมดอยู่ในเอกสารชุด FE)

4. Implementation Flow Diagram (Reference)



รูปที่ 1: Implementation flow reference: LLDD BE - Integration with SBP Platform

5. Field, Format, and Validation

Field / UI	Format	Validation	Behavior
x-api-key	string	required ทุก request จาก BFF	ตรวจที่ guard ของ store-backend ก่อนเข้า controller
x-user-id	string	required สำหรับ endpoint ของผู้ใช้	ใช้เป็น current_approver/create_by ของ workflow และเป็น updated_by ของ master
x-user-group-id	string	required	ใช้เทียบสิทธิ์แบบกลุ่ม (approve_type = group ของ engine)
x-user-permissions	string (serialized)	required	สิทธิ์เมนูจาก auth-backend — SBPGI ไม่คำนวณสิทธิ์เมนูเอง
envelope	{success, data}	บังคับทุก endpoint	ResponseInterceptor ห่อให้แล้ว — service ห้ามห่อซ้ำ
error	{success:false, data:null, error:{code,message}}	message ภาษาไทย verbatim ตามข้อกำหนดทางธุรกิจ	โยนผ่าน HttpException เท่านั้น
mas_param	key-value ของระบบเดิม	read-only สำหรับ SBPGI	93,752 แถว — ต้อง filter ด้วย key prefix ของ SBPGI เสมอ
common_code / common_code_type	code master ของระบบเดิม	read-only สำหรับ SBPGI	2,609 / 376 แถว — วงเงินอนุมัติอยู่ code_type = SBPGI_APPROVE_LIMIT

5.1 User Context จาก BFF

SBPGI ไม่มีระบบ login ของตัวเอง — ตัวตนมาจาก BFF ผ่าน header · guard ของ store-backend แปลง header เป็น user context แล้วส่งต่อให้ service ทุกชั้น

```
// src/common/guards/bff-user.guard.ts (ยึด convention ของ store-backend)
@Injectable()
export class BffUserGuard implements CanActivate {
  canActivate(ctx: ExecutionContext): boolean {
    const req = ctx.switchToHttp().getRequest();
    const apiKey = req.headers['x-api-key'];
    // TODO: เทียบ apiKey กับค่าใน Secret Manager (ห้าม hardcode / ห้ามอยู่ใน .env ที่ commit)
    if (!apiKey) throw new UnauthorizedException('ไม่พบสิทธิ์การเข้าใช้งาน');
    req.user = {
      userId: req.headers['x-user-id'],
      groupId: req.headers['x-user-group-id'],
      permissions: req.headers['x-user-permissions'],
    };
    if (!req.user.userId) throw new UnauthorizedException('ไม่พบสิทธิ์การเข้าใช้งาน');
    return true;
  }
}
```

5.2 Response Envelope

```
// ResponseInterceptor ของ store-backend ห่อให้แล้ว — service ห้ามห่อซ้ำ
// success : { success: true, data: <payload> }
// error : { success: false, data: null, error: { code, message } }
// message ต้องเป็นภาษาไทย verbatim ตาม ข้อกำหนดทางธุรกิจ และโยนผ่าน HttpException เท่านั้น
```

5.3 ไฟล์แนบผ่าน S3 ของระบบเดิม

ขั้นตอน	ปลายทาง	สิ่งที่ SBPGI เก็บเอง
อัปโหลด	POST /statement/upload-file-aws (ระบบ SBP เดิม)	objectKey + ชื่อไฟล์ + ขนาด + content type + section_code
ดาวน์โหลด	POST /statement/download-file-aws (ระบบ SBP เดิม)	stream ผ่าน BE · ห้ามคืน objectKey ให้ FE
ลบ/purge	lifecycle ของ S3 + flag ใน document_attachments	purge_flag / storage_delete_status

แก้ความเข้าใจผิดเดิม (ตรวจ schema จริง 2026-08-07): เอกสารรุ่นก่อนเคยเขียนว่าถ้าใช้ตาราง `upload_general` ของระบบเดิมจะติด FK `job_id` — ไม่จริง `job_id` และ `audit_log_id` เป็น nullable ทั้งคู่ · เหตุผลจริงที่ SBPGI ต้องมี `document_attachments` ของตัวเองคือ `upload_general` ไม่มีคอลัมน์ `file_size` · `content_type` · `section_code` · `upload_status` · `purge_flag`

5.4 อีเมล

ส่งผ่าน `@gosoft-sbp/email-lib` โดยอ่านเนื้อหาจาก `email_template` (85 แถว) และบันทึกผลที่ `email_sent` (5,214 แถว)

แก้ความเข้าใจผิดเดิม (ตรวจ schema จริง 2026-08-07): เอกสารรุ่นก่อนเคยเขียนว่าระบบเดิม ไม่มีที่เก็บ CC ของอีเมล — ไม่จริง มีอยู่ 3 ที่: `email_sent.mail_cc` · `fcs_reminder_log.reminder_cc` · `fml_email_account`

5.5 ค่ากำหนดกลาง

ค่า	อยู่ที่	กติกา
วงเงินอนุมัติ เกณฑ์เดียว 100,000	common_code · code_type = SBPGI_APPROVE_LIMIT	อ่านทุกครั้ง ห้าม hardcode · ถ้าเลือกเก็บที่ workflow_route.condition_json แทน ต้องเก็บที่เดียว (ดูข้อค้าง)
รัศมีผลกระทบ 1 กม. (กทม./ปริมณฑล) / 2 กม. (ต่างจังหวัด)	mas_param	อ่านตอนคำนวณ ไม่ hardcode
เกณฑ์ยอดขัง 60 วัน · growth rate -10%	mas_param	ใช้กับธงข้อมูลผิดปกติและ Gen Flow Gate

5.6 ข้อค้างตัดสินใจที่กระทบ integration (ยังไม่ตัดสิน)

รายการต่อไปนี้ ยังไม่ตัดสิน ทั้งหมด · ทางเลือกและผลกระทบเต็มอยู่ที่ SBPGI vs existing system หัวข้อ 4 การตัดสินใจขั้นสุดท้ายเป็นของเจ้าของโครงการ — เอกสาร LLDD ฉบับนี้บันทึกไว้เป็นข้อค้าง และห้าม dev เลือกทางใดทางหนึ่งเองระหว่าง implement

ข้อค้าง	ทางเลือก A	ทางเลือก B	สถานะ
DP-5 · อีเมล ☐ ปิดแล้ว 2026-08-14	ให้ engine ส่งเอง — ตกไป เพราะ `triggerEvent` ไม่มี `mailTo/mailCc/param` ที่ `sendEmail` บังคับ	เลือกทางนี้: workflow ให้เลข template ผ่าน `workflow_route.email_id` แล้ว SBPGI เรียก `sendEmail()` ของ email-lib เอง · reminder/escalation ที่ไม่ใช่ transition เก็บเลข template ที่ `mas_param`	ปิดแล้ว · เหลือยืนยันกับทีม engine ว่าไม่ส่งซ้ำ
DP-8 · `document_attachments`	ตารางของ SBPGI เอง (สถานะปัจจุบันของแบบ)	ต่อยอด `upload_general` ของระบบเดิม	ยังไม่ตัดสิน
DP-10 ☐ ปิดแล้ว 2026-08-21 · ที่อยู่ของ SBPGI	เลือกข้อนี้ — โมดูลใน `srm-sps-spsap-store-backend` เดิม	backend ใหม่แยกต่างหาก — ตกไป	☐ ปิดแล้ว — ใช้ guard/interceptor/response envelope ของ store-backend เดิมได้ทันที ไม่ต้องเขียนใหม่
DP-6 · `interface_transactions`	ออกแบบใหม่ตาม DDL ปัจจุบัน	ลอกแพตเทิร์น `statement_summary` ของระบบเดิม	ยังไม่ตัดสิน

5.9 Input / Progress / Output Contract

Stage	Contract for implementation
Input	User action, route/query state, form values, and permission context for this feature.
Progress	BFF forward request พร้อม header ตัวตนมาที่ store-backend; Guard ตรวจ x-api-key แล้ว map header เป็น user context; Controller/Service ทำงานโดยใช้ user context (ไม่มี JWT ของ SBPGI เอง); ผลลัพธ์ถูกห่อด้วย ResponseInterceptor เป็น {success, data}
Output	document_attachments (SBPGI)

5.90 Endpoint Implementation Contract

Endpoint	Use-case owner	Service/repository behavior	Definition of done
Internal service	กำหนดวิธีที่ SBPGI ต่อกับแพลตฟอร์ม SBP เดิม: BFF header/ตัวตน, response envelope, ไฟล์บน S3, อีเมลผ่าน @gosoft-sbp/email-lib และค่ากำหนดกลางใน mas_param/common_code — เป็น blocker ที่ต้องปิดในสัปดาห์แรก	เรียกจาก use case ภายในเท่านั้น	ไม่มี endpoint ใดของ SBPGI ออก/ตรวจ JWT เอง

5.91 Backend Execution Sequence

Step	Behavior specific to this LLDD	Failure/test evidence
1	BFF forward request พร้อม header ตัวตนมาที่ store-backend	ไม่ส่ง x-api-key ต้องได้ 401 ตาม envelope มาตรฐาน
2	Guard ตรวจ x-api-key แล้ว map header เป็น user context	ส่ง x-user-id ที่ไม่มีสิทธิ์เมนูต้องได้ 403
3	Controller/Service ทำงานโดยใช้ user context (ไม่มี JWT ของ SBPGI เอง)	upload ไฟล์ 6MB ต้องถูก block ก่อนขึ้น S3
4	ผลลัพธ์ถูกห่อด้วย ResponseInterceptor เป็น {success, data}	download ไฟล์ของเอกสารที่ผู้ใช้ไม่เกี่ยวข้องต้องถูก block

Step	Behavior specific to this LLDD	Failure/test evidence
5	ไฟล์แนบวิ่งผ่าน service S3 เดิม · เก็บเฉพาะ metadata ใน SBPGI	เปลี่ยนค่า SBPGI_APPROVE_LIMIT ใน common_code แล้ว route อนุมัติเปลี่ยนตามโดยไม่ deploy
6	อีเมลส่งผ่าน email-lib โดยอ่าน template จาก email_template และบันทึกผลที่ email_sent	ส่งอีเมลสำเร็จแล้วมีแถวใน email_sent
7	ค่ากำหนด/วงเงินอ่านจาก mas_param และ common_code ทุกครั้ง ไม่ cache ข้ามรอบโดยไม่มี TTL	ไม่ส่ง x-api-key ต้องได้ 401 ตาม envelope มาตรฐาน

6. Button / User Action Mapping

Action	Trigger	API / Service	Expected Result
อ่านตัวตนผู้ใช้	ทุก request	guard อ่าน BFF header	req.user = {userId, groupId, permissions}
อัปโหลดไฟล์แนบ	ปุ่มแนบไฟล์	POST /statement/upload-file-aws (ระบบ SBP เดิม)	ได้ objectKey กลับมาเก็บใน document_attachments
ดาวน์โหลดไฟล์แนบ	ปุ่มดาวน์โหลด	POST /statement/download-file-aws (ระบบ SBP เดิม)	stream ไฟล์ผ่าน BE · ห้ามคืน objectKey ให้ FE
ส่งอีเมล	หลัง action สำเร็จ	@gosoft-sbp/email-lib + email_template	บันทึกผลที่ email_sent
อ่านค่ากำหนดกลาง	ตอน bootstrap/ตอนใช้งาน	mas_param / common_code	ห้าม hardcode วงเงินอนุมัติในโค้ด

7. API Contract

8. Reference DB Mapping (No Database Page Work)

ส่วนนี้เป็นข้อมูลอ้างอิงสำหรับการ implement API/Job เท่านั้น ไม่ใช่งานสร้างหน้า Database, ไม่ใช่งานออกแบบ DB page และไม่ถูกนับเป็น deliverable แยกของ FE/BE

Table / Object	R/W	Usage
mas_param (sps_store)	R	ค่ากำหนดกลาง 93,752 แถว
common_code / common_code_type (sps_store)	R	2,609 / 376 แถว · วงเงินอนุมัติ SBPGI_APPROVE_LIMIT
email_template (sps_store)	R	85 แถว · SBPGI/lib อ่านอย่างเดียว — seed 8 แถวของ SBPGI ทำครั้งเดียวตอน migration ไม่ใช่ runtime
email_sent (sps_store)	W (โดย email-lib)	5,214 แถว · lib เขียน log ให้เอง SBPGI ไม่ INSERT เอง (□□ คอลัมน์ผู้ส่งคือ send_by)
business_user (sps_store)	R	12,752 แถว · ข้อมูลผู้ใช้/ผู้อนุมัติ
store / mas_store (sps_store)	R	19,402 / 19,647 แถว · master ร้าน
document_attachments (SBPGI)	R/W	metadata ไฟล์แนบ · ไฟล์จริงอยู่บน S3 ของระบบเดิม

9. Processing Flow

Step	Description
1	BFF forward request พร้อม header ตัวตนมาที่ store-backend
2	Guard ตรวจ x-api-key แล้ว map header เป็น user context
3	Controller/Service ทำงานโดยใช้ user context (ไม่มี JWT ของ SBPGI เอง)
4	ผลลัพธ์ถูกห่อด้วย ResponseInterceptor เป็น {success, data}
5	ไฟล์แนบวิ่งผ่าน service S3 เดิม · เก็บเฉพาะ metadata ใน SBPGI
6	อีเมลส่งผ่าน email-lib โดยอ่าน template จาก email_template และบันทึกผลที่ email_sent
7	ค่ากำหนด/วงเงินอ่านจาก mas_param และ common_code ทุกครั้ง ไม่ cache ข้ามรอบโดยไม่มี TTL

10. Acceptance Criteria

- ไม่มี endpoint ใดของ SBPGI ออก/ตรวจ JWT เอง
- ทุก response ผ่าน envelope เดียวกับ store-backend
- ไม่มี credential ของ S3/SMTP อยู่ในโค้ดหรือ config ของ SBPGI
- วงเงินอนุมัติ เกณฑ์เดียว 100,000 อ่านจาก common_code (SBPGI_APPROVE_LIMIT) ไม่ hardcode
- objectKey ไม่ถูกส่งออกไปที่ FE
- ข้อค้างตัดสินใจเรื่อง email และ attachment ถูกบันทึกเป็นข้อค้าง ไม่ถูกตัดสินในเอกสารนี้

11. Developer Test Checklist

No	Test
1	ไม่ส่ง x-api-key ต้องได้ 401 ตาม envelope มาตรฐาน
2	ส่ง x-user-id ที่ไม่มีสิทธิ์เมนูต้องได้ 403
3	upload ไฟล์ 6MB ต้องถูก block ก่อนขึ้น S3
4	download ไฟล์ของเอกสารที่ผู้ใช้ไม่เกี่ยวข้องต้องถูก block
5	เปลี่ยนค่า SBPGI_APPROVE_LIMIT ใน common_code แล้ว route อนุมัติเปลี่ยนตามโดยไม่ deploy
6	ส่งอีเมลสำเร็จแล้วมีแถวใน email_sent