LLDD BE - Workflow Engine Definition

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	24 ชั่วโมง (ไม่มี unit test แยก — ดูเหตุผลใน NO_UNIT_TEST_DOCS)
Owner	Aphiwit <Bank> Khammoon
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	สร้างข้อมูลนิยาม workflow ลงฐานข้อมูลของ engine — ระบุว่า flow ของ SBPGI มีกี่ step แต่ละ step ทำอะไร ใครทำได้ กดปุ่มไหนแล้วไป state ใด โดย register version/state/status/event/route/group/part ของ `@srm/glb-workflow` ตามสัญญาในเอกสารของ lib เอง (TSM SRM LLDD SBP workflow ไฟล์1.2 full — แปลงจาก `SBP/TSM-SRM-LLDD SBP workflow 1.2.xlsx`) · เป็นงานตั้งต้นที่ต้องเสร็จก่อน ฝั่ง BE คนอื่นจึงจะเรียก `initializeWorkflow` และ `eventWorkflow` (trigger event) ได้ — blocker ของสัปดาห์แรก

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

- ลงทะเบียน workflow version ของ SBPGI 1 version (url_main + url_param_mapping)
- ผลลัพธ์ที่ส่งมอบคือ seed script/มัยเกรชันของข้อมูลนิยาม ไม่ใช่โค้ดเรียก engine — ทีมอื่นเรียก engine ต่อจากนิยามชุดนี้
- จำนวน step ที่ต้องสร้าง = 6 state — 5 ขั้นทำงาน (`06` รอฝ่าย SBP DSA → `08` รอเจ้าหน้าที่ SBP DSA → `01` รอหน่วยงานส่งเสริมธุรกิจ SBP → `02` รอ GM → `03` รอ AVP) + 1 state จบ (`99` เสร็จสิ้นดำเนินการ) · `state_id` เป็น running ตาม version ตามกติกาของ engine (v1 → 10001+)
- จำนวน route ที่ต้องสร้าง = 12 เส้น ตาม Canonical Workflow Transition Matrix ใน `LLDD-BE-API-Document-Workflow-Actions.pdf` §5.1 (รวมเส้นข้ามขั้น 06→01 · เส้นจบทันทีเมื่อ เห็นควรไม่ชดเชย ที่ 01/02 · เส้นแตกตามวงเงิน 100,000 ที่ 02 และเส้นส่งกลับ)
- ตารางที่ต้อง seed = 10 ตาราง จาก 13 ตารางของ engine (`sps_store`) ตาม TSM SRM LLDD SBP workflow ไฟล์1.2 §4 — อีก 3 ตารางเป็นรันไทม์ที่ engine เขียนเอง
- ขอบเขตหยุดที่ข้อมูลนิยาม: ไม่รวม `initializeWorkflow` (เปิด instance · อยู่ใน LLDD-BE-API-Workflow-Instances.pdf) และ ไม่รวม `eventWorkflow`/trigger event (อยู่ใน LLDD-BE-API-Document-Workflow-Actions.pdf)
- นิยาม state/status 5 ขั้น 06 -> 08 -> 01 -> 02 -> 03 และปลายทางจบ flow
- นิยาม route ของทุกปุ่ม · การแตก route ตามวงเงินอนุมัติ เกณฑ์เดียว 100,000 เขียนเป็นตัวอย่างทางเลือก B เท่านั้น — แหล่งเก็บวงเงินยังไม่ตัดสิน (มติเดิมคือ common_code · ดูข้อค้าง 5.6)
- สำรวจทางเลือกผู้อนุมัติ: workflow_group / workflow_group_map เทียบกับ addPreApprover รายคน — ยังไม่ตัดสิน (ดูข้อค้าง 5.6)
- สำรวจทางเลือก workflow_part / workflow_part_display สำหรับคุมการแสดงผลรายส่วน — ยังไม่ตัดสิน ว่าจะใช้แทน data-editrole ของ SBPGI หรือไม่ (ดูข้อค้าง 5.5/5.6)
- ความเสี่ยงและข้อค้างของ engine — ชื่อ function ปิดแล้ว 2026-08-14 (ยึด 8 API ตามชีต Detail ของ LLDD ฝั่ง lib) · ที่ยังค้างคือ DP-2 `workflow_transaction` ไม่มี PK/index

3. Screenshot Reference

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

4. Implementation Flow Diagram (Reference)

รูปที่ 1: Implementation flow reference: LLDD BE - Workflow Engine Definition

5. Field, Format, and Validation

Field / UI	Format	Validation	Behavior
versionId	integer	1 ระบบ = 1 version	SBPGI ขอ version ใหม่จากทีมเจ้าของ library
referenceId	string unique	required ตอน initializeWorkflow	☐ DP-1 ปิดแล้ว 2026-08-17 = `compensation_documents.id` (surrogate) แปลงเป็น string — `reference_id` ของ engine เป็น varchar(255)
state_id	integer running ตาม version	1 state มีได้หลาย status	map 5 ขั้นของ SBPGI: 06/08/01/02/03 + state จบ
event	save|submit|approve|reject|cancel|sendback	ค่าเริ่มต้นของ engine	ปุ่มไทยของ SBPGI map ลง event เหล่านี้ผ่าน common_code (code_type=SBPGI_DECISION) — ตาราง decisions ถูกตัดตามมติ DP-9 (2026-08-10)
condition_json	{"field","operator","value"}	operator: == != > < >= <=	ใช้ {"field":"amount","operator":"<","value":100000} แยก route GM/AVP
eventParam	object	ส่งมาพร้อม event	SBPGI ส่ง {"amount": ยอดชดเชยรวม} ให้ engine เลือก route เอง

Field / UI	Format	Validation	Behavior
part_display_type	READ | WRITE	ต้องยืนยันค่าจริงกับทีม library	ไฟล์ต้นฉบับสะกดว่า WRTIE ทุกแถวของชีต sample data
url_main / url_param_mapping	string	required ตอน register version	ทำให้ inbox กลาง (GET /api/workflow/pending) ลิงก์กลับหน้าเอกสารของ SBPGI ได้

5.0 ทำไมเอกสารฉบับนี้ต้องปิดเป็นฉบับแรก

เอกสารฉบับนี้ ไม่มี endpoint ของตัวเอง — ผลลัพธ์คือชุดนิยาม state/status/route/part ที่เอกสารอื่น เอาไปใช้ต่อ จึงต้องจบก่อนผู้บริโภคทั้งหมดเริ่ม (ปรับลำดับ 2026-08-10: เดิมถูกจัดไว้ท้ายกลุ่ม API ทำให้ `BE-API-Document-Workflow-Actions` และ `BE-API-Workflow-Instances` เริ่มก่อนเอกสารที่นิยาม สิ่งที่มันต้องใช้)

เอกสารที่รอ	รออะไรจากฉบับนี้
BE-API-Document-Workflow-Actions	รหัส event ต่อปุ่ม · route ของแต่ละ state · เงื่อนไขแตกสายตามวงเงิน
BE-API-Workflow-Instances	โครง version/state/status ที่จะ query และรูปแบบ payload ของ engine
BE-Job-8b-StartInternalWorkflow	ลำดับเรียก initialize -> addPreApprover และค่า `referenceId`
FE-Document-Detail (5 ฉบับ role)	`workflow_part_display` READ/WRITE ต่อ state ที่คุมการแสดงผลรายส่วน

5.1 Engine คือของกลาง 13 ตาราง ใน schema `sps_store`

`@srm/glb-workflow` เป็น library กลางที่ทุกระบบใน SBP platform import ไปใช้ (ต้นฉบับ: TSM SRM LLDD SBP workflow ไฟล์1.2 v1.2 ลงวันที่ 29/04/2026) · SBPGI ใช้ engine ตัวนี้ตามมติ 2026-08-06 ที่ตัดตาราง workflow ของตัวเองทิ้งทั้งหมด

ตัวเลขที่เอกสารรุ่นก่อนเขียนผิด 2 จุด (แก้แล้ว 2026-08-07): (1) engine มี 13 ตาราง ไม่ใช่ 10 · (2) engine ตัวที่ใช้งานจริงอยู่ schema `sps_store` ไม่ใช่ `sps_auth` — ทั้งสอง schema มีครบ 13 ตารางเหมือนกันแต่เป็นคนละชุดและคนละเวอร์ชัน (`workflow_state` ของ `sps_auth` มี 3 คอลัมน์ · ของ `sps_store` มี 4 คอลัมน์) · `sps_auth.workflow_transaction` มีแค่ 55 แถว (route 41 · state 10) ซึ่งเป็นชุดของ auth-backend คนละเรื่องกัน

กลุ่ม	ตาราง	หน้าที่
นิยาม flow (config)	workflow · workflow_version · workflow_state · workflow_status · workflow_event · workflow_route	ตั้งครั้งเดียวต่อระบบ · `workflow_version.url_main` / `url_param_mapping` ทำให้ inbox กลางลิงก์กลับหน้าเอกสารของ SBPGI ได้
กลุ่มผู้อนุมัติ	workflow_group · workflow_group_map	`map_table` ว่าง = เทียบกับ field ของ user ตรง ๆ · ระบุ map_table = ต้องเป็น view ที่ where ด้วย user_id/group_id ได้
ข้อมูลรันไทม์	workflow_transaction · workflow_history · workflow_approver	19,283 / 38,010 / 96,542 แถวใน sps_store (ตรวจ 2026-08-07)
คุมการแสดงผล	workflow_part · workflow_part_display	`part_display_type` = READ / WRITE ต่อ state — คืนมากับ getPermissionEvents

5.2 ความเสี่ยงที่ต้องคุยกับทีมเจ้าของ library

ความเสี่ยง	ข้อเท็จจริงที่ตรวจแล้ว	ผลกระทบต่อ SBPGI	สิ่งที่ต้องทำ
`sps_store.workflow_transaction` ไม่มี PK และไม่มี index เลย	มี 19,283 แถวแต่ schema dump ไม่พบ PK/index ใด ๆ (ตารางชื่อเดียวกันใน `sps_auth` มี PK `transaction_id` ปกติ) · `workflow_state` / `workflow_event` / `workflow_part_display` ของ `sps_store` ก็ไม่มี PK เช่นกัน	ทุกครั้งที่เปิดเอกสารหรือกด action ต้อง seq-scan 19,283 แถวเพื่อหา `reference_id` · ไม่มีอะไรกัน initialize ซ้ำแม้ระดับ application · จะแย่ลงเมื่อ SBPGI เพิ่มอีกราวหมื่นแถวต่อปี	ยื่นเรื่องขอ sign-off เพิ่ม PK + UNIQUE(version_id, reference_id) + index กับทีมเจ้าของ `@srm/glb-workflow` · ระหว่างรอ ให้กันซ้ำ + เก็บ mapping ที่ฝั่ง SBPGI (ทางเลือกที่จะใช้จริงยังไม่ตัดสิน — DP-2)
`part_display_type` สะกดว่า `WRTIE` ในไฟล์ต้นฉบับ	สะกดผิดทุกแถวของชีต `sample data`	ถ้า SBPGI เขียนค่า `WRITE` แล้ว engine เทียบกับ `WRTIE` การแสดงผลจะเพี้ยนทั้งหน้า	ยืนยันค่าจริงในระบบกับทีม library ก่อนลงทะเบียน part

ความเสี่ยง	ข้อเท็จจริงที่ตรวจแล้ว	ผลกระทบต่อ SBPGI	สิ่งที่ต้องทำ
`workflow_route` มี 2 นิยามในไฟล์เดียวกัน	ชีต `sample data` มีคอลัมน์ `group_id` แต่ entity ที่แนบมาใช้ `approver` และตั้งชื่อ property ว่า `approverRoleId`	เขียนโค้ดผูกผู้อนุมัติผิดคอลัมน์	ยืนยัน schema จริงของ route กับทีม library
ไม่มี API ถอน/แก้ผู้อนุมัติล่วงหน้า	มีแต่ `addPreApprover`	เคสเปลี่ยนตัวผู้อนุมัติ (ลาออก/รักษาการ) ทำไม่ได้ผ่าน library	ถามทีม library ว่าจะเพิ่มให้หรือให้ SBPGI แก้ตารางตรง

5.3 API ของ engine — 8 function (ยึด LLDD ของ lib · ปิดข้อค้าง 2026-08-14)

แหล่งความจริงคือชีต `Detail` ของ `SBP/TSM-SRM-LLDD SBP workflow 1.2.xlsx` ซึ่งเป็น เอกสารของ lib เอง · ชื่อที่เคยนับว่าขัดกันไม่ใช่ชื่อ API: *Trigger Event* เป็นชื่อหัวข้อของขั้นตอนภายใน `eventWorkflow` ในชีต 2 ส่วน `TriggerEventUseCase` / `AddPreparedApproverUseCase` / `GetPendingFlowUseCase` เป็น UseCase class ที่ store-backend ห่อไว้ใช้เอง ไม่ใช่ API ของ lib

#	function	พารามิเตอร์ (ชีต Detail)	SBPGI ใช้ที่ไหน
1	`initializeWorkflow`	version, userId, referenceId	เปิด flow ให้เอกสารใหม่ (Job 8b · `POST /sbpgi/workflow/instances`)
2	`eventWorkflow`	version, referenceId, event, eventParam, remark, userId + userData · userFullname · nextApproverId (ส่วนขยาย 29/04 · 20/05 · 16/06/2026 — ยึดชุดนี้เวลาเขียนโค้ด)	`POST /sbpgi/document/{docNo}/actions`
3	`getPermissionEvents`	version, referenceId, userData	ปุ่ม/ผลพิจารณาที่ user กดได้ในหน้าเอกสาร
4	`getHistory`	version, referenceId	`GET /sbpgi/document/{docNo}/timeline`
5	`getTransaction`	version, referenceId	สถานะ + ผู้ถืองานปัจจุบันของเอกสาร
6	`getPendingFlowByUser`	userData	หน้า เอกสาร → รอดำเนินการ + reminder รายสัปดาห์
7	`getWorkflowsByUser`	userData	หน้า เอกสาร → ที่เกี่ยวข้อง (รวมที่ยังไม่ถึงคิวและที่อนุมัติไปแล้ว)
8	`addPreApprover`	version, userId, referenceId, state_id, approver, seq	ตั้งผู้อนุมัติล่วงหน้าของขั้นถัดไป

5.4 นิยาม flow ของ SBPGI ที่ต้อง register

state	ชื่อสถานะเอกสาร	event ที่ทำได้	ปลายทาง
06	รอฝ่าย SBP DSA ดำเนินการ	submit (ส่งเจ้าหน้าที่ SBP DSA) · reject (เห็นควรไม่ชดเชย) · cancel (หยุดชดเชย) · submit (ส่งหน่วยงานส่งเสริมธุรกิจ SBP)	08 หรือ 01 หรือจบ flow
08	รอเจ้าหน้าที่ SBP DSA ดำเนินการ	submit (คำนวณเงินชดเชยเรียบร้อย)	01
01	รอหน่วยงานส่งเสริมธุรกิจ SBP ดำเนินการ	approve (เห็นควรชดเชย) · reject (เห็นควรไม่ชดเชย → จบ flow ทันที) · sendback (ฝ่าย SBP DSA ดำเนินการ)	02 · จบ flow · 06
02	รอ GM ส่งเสริมธุรกิจ SBP ดำเนินการ	approve (เห็นควรชดเชย) · reject (เห็นควรไม่ชดเชย → จบ flow ทันที) · sendback	จบ flow เมื่อยอด < 100,000 · ไป 03 เมื่อ ≥ 100,000 · 01
03	รอผู้บริหารสำนักบริหาร SBP ดำเนินการ	approve (เห็นควรชดเชย) · sendback	จบ flow · 02

```
-- ☐☐ ตัวอย่างนี้คือ **ทางเลือก B ของข้อค้าง 5.6 (ยังไม่ตัดสิน) — ห้าม seed ลงจริงก่อนได้ข้อสรุป**
-- มติเดิม (ทางเลือก A) คือเก็บวงเงินที่ `common_code` (code_type = SBPGI_APPROVE_LIMIT) แล้ว "อ่านทุกครั้ง ห้าม hardcode"
-- ตามที่ LLDD-BE-Integration-SBP-Platform.pdf / LLDD-Database.pdf ระบุไว้ · กติกาที่แน่นอนคือ **ห้ามเก็บสองที่**
-- ถ้าเลือกทางเลือก A: route ยังแตกสองเส้นเหมือนเดิม แต่ SBPGI เป็นผู้เทียบยอดกับ common_code
-- แล้วส่งผลลัพธ์ (เช่น eventParam = {"limitTier":"GM"|"AVP"}) ให้ engine เลือก route โดยไม่ฝังตัวเลขใน condition_json
--
-- ตัวอย่างทางเลือก B (ฝังวงเงินใน condition_json ตามความสามารถของ engine):
-- SBPGI ส่ง eventParam = {"amount": <ยอดชดเชยรวมของเอกสาร>} แล้วให้ engine เลือก route เอง
-- seq = ลำดับที่ engine ใช้ไล่ตรวจ condition_json (ตัวแรกที่ตรงชนะ)
-- ตัวเลข 100000 ด้านล่างเป็นค่า **ตัวอย่าง** ของเกณฑ์เดียว (มติ 2026-08-18) ไม่ใช่ค่าที่ตกลงให้ hardcode
INSERT INTO sps_store.workflow_route
  (version_id, from_state_id, event, to_state_id, to_status_id, seq, condition_json, approver_type, group_id)
VALUES
  (:v, :state_02, 'approve', :state_end, :status_done, 1,
   '{"field":"amount","operator":"<","value":100000}', 'group', :group_none),
  (:v, :state_02, 'approve', :state_03, :status_wait_avp, 2,
   '{"field":"amount","operator":">=","value":100000}', 'group', :group_avp);
-- ☐ เกณฑ์เดียวจึงไม่มีช่องโหว่ปลายบน: ทุกยอดตั้งแต่ 100,000 ขึ้นไปวิ่งเข้า AVP เส้นเดียว
```

5.5 `workflow_part_display` ทับซ้อนกับกลไกของ prototype

หน้า `Document Detail screen` ของ prototype คุมสิทธิ์แก้ไขรายส่วนด้วย `data-editrole` / `data-roleonly` / `.edit-only` ฝั่ง client เอง · engine มีกลไกเดียวกันให้อยู่แล้วผ่าน `workflow_part` + `workflow_part_display` ที่คืนมาใน `display[]` ของ `getPermissionEvents` (รูปแบบ `{partId, partName, stateId, partDisplayType, partSeq}`)

บันทึกเป็นข้อสังเกต ยังไม่ตัดสิน ว่า SBPGI จะลงทะเบียน part ของทุกส่วนในหน้าเอกสารแล้วให้ FE อ่าน `display[]` แทนการ hardcode สิทธิ์ต่อ role หรือไม่ · ถ้าเลือกทางนี้จะกระทบ `LLDD-FE-Document-Detail.pdf` + role pack 5 ฉบับ ที่ปัจจุบันอ่าน `visibleSections/editableSections` จาก API ของ SBPGI เอง · ทางเลือกเต็มอยู่ที่ SBPGI vs existing system หัวข้อ 4

5.6 ข้อค้างตัดสินใจที่กระทบ engine (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-1 · `reference_id`	`doc_no` — ตกไป (บังคับออกเลขตั้งแต่ initialize และแก้ภายหลังไม่ได้)	เลือก surrogate id (`compensation_documents.id` · ส่งเป็น string เพราะ `reference_id` เป็น varchar(255)) — ตรงกับที่ cooperation-request/inform-evaluate ทำจริงทุกจุด	☐ ปิดแล้ว 2026-08-17 — ยืนยันตามระบบเดิม
DP-2 · `workflow_transaction` ไม่มี PK/index	ขอ sign-off จากทีม library ให้เพิ่ม PK + UNIQUE + index	ไม่แตะตารางของ library · กันซ้ำและทำ index ที่ฝั่ง SBPGI	ยังไม่ตัดสิน ☐
วงเงินอนุมัติเก็บที่ไหน	`common_code` (SBPGI_APPROVE_LIMIT) ตามมติเดิม	`workflow_route.condition_json` ตามความสามารถของ engine	ยังไม่ตัดสิน · กติกาที่แน่นอนคือ ห้ามเก็บสองที่
DP-5 · ใครเรียก email-lib ☐ ปิดแล้ว 2026-08-14	engine ส่งเอง — ตกไป (ไม่มี `mailTo/param` ใน `triggerEvent`)	SBPGI ส่งเอง โดยใช้เลข template จาก `workflow_route.email_id`	ปิดแล้ว
ผู้อนุมัติของ SBPGI	`workflow_group_map` ผ่าน view (ต้องยืนยันว่า view where ด้วย user_id/group_id ได้)	`addPreApprover` ระบุรายคน	ยังไม่ตัดสิน
`workflow_part_display` แทน `data-editrole`	ลงทะเบียน part แล้วให้ FE อ่าน `display[]`	คงกลไกของ SBPGI เอง (`visibleSections/editableSections`)	ยังไม่ตัดสิน · กระทบ role pack 5 ฉบับ

5.9 Input / Progress / Output Contract

Stage	Contract for implementation
Input	ไม่มี endpoint ของตัวเอง — input คือ request ที่เอกสารอื่นส่งเข้ามา พร้อม user context จาก BFF header (ดู 5.1) และค่ากำหนดกลางที่อ่านจากระบบเดิม

Stage	Contract for implementation
Progress	ขอ workflow version ใหม่จากทีมเจ้าของ @srm/glb-workflow (1 ระบบ = 1 version); ลงทะเบียน state/status 5 ขั้นของ SBPGI + state จบ flow; ลงทะเบียน route ทุกเส้น พร้อม seq · ส่วน condition_json ของวงเงินอนุมัติให้ใส่ ก็ต่อเมื่อเจ้าของโครงการเลือกทางเลือก B ของข้อค้าง 5.6 (ทางเลือก A = อ่านวงเงินจาก common_code SBPGI_APPROVE_LIMIT ตามมติเดิม แล้วให้ route แยกด้วยค่าที่ SBPGI ส่งมาแทน); [conditional — ทำเมื่อเลือกทางเลือก A ของข้อค้าง "ผู้อนุมัติของ SBPGI" ใน 5.6] ลงทะเบียน workflow_group / workflow_group_map สำหรับผู้อนุมัติแบบกลุ่ม · ถ้าเลือกทางเลือก B ให้ใช้ addPreApprover ระบุรายคนแทน
Output	ไม่มีตารางที่เอกสารนี้เขียนเอง — output คือ response ตาม envelope กลาง `{success, data}` และร่องรอยที่ตรวจย้อนได้ (log / consideration_logs / workflow_history ของ engine)

5.90 Endpoint Implementation Contract

Endpoint	Use-case owner	Service/repository behavior	Definition of done
Internal service	สร้างข้อมูลนิยาม workflow ลงฐานข้อมูลของ engine — ระบุว่า flow ของ SBPGI มีกี่ step แต่ละ step ทำอะไร ใครทำได้ กดปุ่มไหนแล้วไป state ใด โดย register version/state/status/event/route/group/part ของ `@srm/glb-workflow` ตามสัญญาในเอกสารของ lib เอง (TSM SRM LLDD SBP workflow ไฟล์1.2 full — แปลงจาก `SBP/TSM-SRM-LLDD SBP workflow 1.2.xlsx`) · เป็นงานตั้งต้นที่ต้องเสร็จก่อน ฝั่ง BE คนอื่นจึงจะเรียก `initializeWorkflow` และ `eventWorkflow` (trigger event) ได้ — blocker ของสัปดาห์แรก	เรียกจาก use case ภายในเท่านั้น	SBPGI ไม่สร้างตาราง workflow ของตัวเองเลย — ใช้ engine 13 ตารางใน schema sps_store

5.91 Backend Execution Sequence

Step	Behavior specific to this LLDD	Failure/test evidence
1	ขอ workflow version ใหม่จากทีมเจ้าของ @srm/glb-workflow (1 ระบบ = 1 version)	initializeWorkflow ซ้ำด้วย referenceId เดิมต้องไม่เกิด transaction ที่สอง
2	ลงทะเบียน state/status 5 ขั้นของ SBPGI + state จบ flow	ยอดชดเชย 99,999 ต้องจบที่ GM · 100,000 ต้องวิ่งต่อ AVP (เกณฑ์เดียว · มติ 2026-08-18)
3	ลงทะเบียน route ทุกเส้น พร้อม seq · ส่วน condition_json ของวงเงินอนุมัติให้ใส่ ก็ต่อเมื่อเจ้าของโครงการเลือกทางเลือก B ของข้อค้าง 5.6 (ทางเลือก A = อ่านวงเงินจาก common_code SBPGI_APPROVE_LIMIT ตามมติเดิม แล้วให้ route แยกด้วยค่าที่ SBPGI ส่งมาแทน)	ผู้ใช้ที่ไม่ใช่ current_approver ต้องไม่ได้ปุ่มใด ๆ จาก getPermissionEvents
4	[conditional — ทำเมื่อเลือกทางเลือก A ของข้อค้าง "ผู้อนุมัติของ SBPGI" ใน 5.6] ลงทะเบียน workflow_group / workflow_group_map สำหรับผู้อนุมัติแบบกลุ่ม · ถ้าเลือกทางเลือก B ให้ใช้ addPreApprover ระบุรายคนแทน	[conditional · เฉพาะทางเลือก A ของ workflow_part_display] display[] ของ state 01 ต้องเปิด WRITE เฉพาะส่วนที่ section 01 แก้ได้
5	[conditional — ทำเมื่อเลือกทางเลือก A ของข้อค้าง `workflow_part_display` ใน 5.6] ลงทะเบียน workflow_part + workflow_part_display ของส่วนต่าง ๆ ในหน้าเอกสาร · ถ้าเลือกทางเลือก B ให้คงกลไก visibleSections/editableSections ของ SBPGI	getPendingFlowByUser ต้องคืน url_main ที่เปิดกลับหน้าเอกสารได้จริง

Step	Behavior specific to this LLDD	Failure/test evidence
6	กรอก url_main / url_param_mapping ให้ inbox กลางลิงก์กลับหน้าเอกสารได้	เดิน flow จนจบแล้ว workflow_history มีครบทุกขั้น
7	ทดสอบเดิน flow ครบทุก route บน environment ทดสอบ	— (ยังไม่มี test เฉพาะขั้นนี้ · ครอบด้วย test รวมของเอกสารในหัวข้อ 11)
8	ส่งความเสี่ยง/ข้อค้างให้ทีมเจ้าของ library และเจ้าของโครงการตัดสิน	— (ยังไม่มี test เฉพาะขั้นนี้ · ครอบด้วย test รวมของเอกสารในหัวข้อ 11)

6. Button / User Action Mapping

Action	Trigger	API / Service	Expected Result
เปิด workflow	Job 8b / สร้างเอกสาร	initializeWorkflow(versionId, userId, referenceId)	สร้าง workflow_transaction ที่ initial state/status
ระบุผู้อนุมัติล่วงหน้า	หลังเปิด workflow	addPreApprover(versionId, referenceId, stateId, approver, seq, userId)	insert workflow_approver (approver_type = user เสมอ)
กดผลพิจารณา	ปุ่มบนหน้าเอกสาร	eventWorkflow(... event, eventParam ...)	เดิน state ตาม route ที่ตรง condition_json แล้วบันทึก workflow_history
อ่านปุ่ม/สิทธิ์แสดงผล	เปิดหน้าเอกสาร	getPermissionEvents(versionId, referenceId, userData)	คืน event[] + display[] (partId/partDisplayType ต่อ state)
อ่านกล่องงาน	หน้ารายการรอดำเนินการ	getPendingFlowByUser(userData, versionId)	คืนงานที่รอ user คนนั้น + url_main
อ่านประวัติ	แท็บประวัติ	getHistory(versionId, referenceId)	คืน timeline ต่อแถว

7. API Contract

เอกสารฉบับนี้ไม่มี endpoint ของตัวเอง — เป็นสัญญา/งานภายในที่เอกสารอื่นเรียกใช้ (ดูขอบเขตใน 5.90 Endpoint Implementation Contract) · รายการ endpoint ทั้ง 29 เส้นของ SBPGI อยู่ที่ LLDD-API.pdf และ `API design`

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
workflow (sps_store)	W ครั้งเดียวตอน setup	1 แถว — `workflow_name` = ระบบประกันรายได้ (SBPGI)
workflow_version (sps_store)	W ครั้งเดียวตอน setup	1 แถว — 1 ระบบ = 1 version · ต้องมี `initial_state_id` (= ขั้น 06), `end_state_id` (= 99), `url_main`, `url_param_mapping` เพื่อให้ inbox กลางลิงก์กลับหน้าเอกสารได้ · ขอเลข version จากทีมเจ้าของ library
workflow_state (sps_store)	W ครั้งเดียวตอน setup	6 แถว = จำนวน step ของ flow — 5 ขั้นทำงาน (06 · 08 · 01 · 02 · 03) + 1 ขั้นจบ (99) · `state_id` running ตาม version (v1 → 10001+)
workflow_status (sps_store)	W ครั้งเดียวตอน setup	6 แถว — ชื่อสถานะเอกสารที่ผู้ใช้เห็น 1:1 กับ state (รอฝ่าย SBP DSA ดำเนินการ / รอเจ้าหน้าที่ SBP DSA ดำเนินการ / รอหน่วยงานส่งเสริมธุรกิจ SBP ดำเนินการ / รอ GM ส่งเสริมธุรกิจ SBP ดำเนินการ / รอผู้บริหารสำนักบริหาร SBP ดำเนินการ / เสร็จสิ้นดำเนินการ) · engine รองรับ 1 state หลาย status

Table / Object	R/W	Usage
workflow_event (sps_store)	R (ใช้ค่า default ของ engine)	`save` `submit` `approve` `reject` `cancel` `sendback` — ปุ่มไทยของ SBPGI map ลง 6 event นี้ผ่าน `common_code` (`code_type` = `SBPGI_DECISION`) · ไม่ต้องเพิ่ม event ใหม่
workflow_route (sps_store)	W ครั้งเดียวตอน setup	12 แถว = ทุกเส้นทางของ flow ตาม Canonical Workflow Transition Matrix (`LLDD-BE-API-Document-Workflow-Actions.pdf` §5.1) — รวมเส้นข้ามขั้น 06→01 · เส้นจบทันทีเมื่อ *เห็นควรไม่ชดเชย* ที่ 01/02 · เส้นแตกตามวงเงิน 100,000 ที่ 02 (`condition_json`) และเส้นส่งกลับทุกขั้น · `seq` ห้ามชนกันภายใน from_state เดียวกัน
workflow_group (sps_store)	W ครั้งเดียวตอน setup *(conditional)*	กลุ่มผู้อนุมัติต่อขั้น — ทำเมื่อเลือกทางเลือก A ของข้อค้าง "ผู้อนุมัติของ SBPGI" (5.6) · ถ้าเลือกทางเลือก B (ระบุรายคนด้วย `addPreApprover`) ไม่ต้อง seed
workflow_group_map (sps_store)	W ครั้งเดียวตอน setup *(conditional)*	map กลุ่ม → ผู้ใช้ · ไม่ระบุ `map_table` = เทียบ `userId/groupId` ตรง ๆ · ระบุ `map_table` = ต้องเป็น view ที่ where ด้วย user_id/group_id ได้
workflow_part (sps_store)	W ครั้งเดียวตอน setup *(conditional)*	ชื่อ component ของหน้าเอกสาร + `part_seq` — ทำเมื่อเลือกทางเลือก A ของข้อค้าง `workflow_part_display` (5.6)
workflow_part_display (sps_store)	W ครั้งเดียวตอน setup *(conditional)*	`part_display_type` = READ / WRITE ต่อ (state, part) · ☐☐ ไฟล์ต้นฉบับสะกด `WRTIE` ทุกแถว ต้องยืนยันค่าจริงกับทีม library ก่อน seed
workflow_transaction / workflow_history / workflow_approver (sps_store)	R เท่านั้น (engine เขียนเอง)	ข้อมูลรันไทม์ 19,283 / 38,010 / 96,542 แถว (ตรวจ 2026-08-07) — ☐ ห้าม INSERT/UPDATE ตรง ต้องผ่าน `initializeWorkflow` / `eventWorkflow` / `addPreApprover` ของ lib เท่านั้น · DP-2: `workflow_transaction` ไม่มี PK และไม่มี index

9. Processing Flow

Step	Description
1	ขอ workflow version ใหม่จากทีมเจ้าของ @srm/glb-workflow (1 ระบบ = 1 version)
2	ลงทะเบียน state/status 5 ขั้นของ SBPGI + state จบ flow
3	ลงทะเบียน route ทุกเส้น พร้อม seq · ส่วน condition_json ของวงเงินอนุมัติให้ใส่ ก็ต่อเมื่อเจ้าของโครงการเลือกทางเลือก B ของข้อค้าง 5.6 (ทางเลือก A = อ่านวงเงินจาก common_code SBPGI_APPROVE_LIMIT ตามมติเดิม แล้วให้ route แยกด้วยค่าที่ SBPGI ส่งมาแทน)
4	[conditional — ทำเมื่อเลือกทางเลือก A ของข้อค้าง "ผู้อนุมัติของ SBPGI" ใน 5.6] ลงทะเบียน workflow_group / workflow_group_map สำหรับผู้อนุมัติแบบกลุ่ม · ถ้าเลือกทางเลือก B ให้ใช้ addPreApprover ระบุรายคนแทน
5	[conditional — ทำเมื่อเลือกทางเลือก A ของข้อค้าง `workflow_part_display` ใน 5.6] ลงทะเบียน workflow_part + workflow_part_display ของส่วนต่าง ๆ ในหน้าเอกสาร · ถ้าเลือกทางเลือก B ให้คงกลไก visibleSections/editableSections ของ SBPGI
6	กรอก url_main / url_param_mapping ให้ inbox กลางลิงก์กลับหน้าเอกสารได้
7	ทดสอบเดิน flow ครบทุก route บน environment ทดสอบ
8	ส่งความเสี่ยง/ข้อค้างให้ทีมเจ้าของ library และเจ้าของโครงการตัดสิน

10. Acceptance Criteria

- SBPGI ไม่สร้างตาราง workflow ของตัวเองเลย — ใช้ engine 13 ตารางใน schema sps_store
- route ครอบคลุมทุกปุ่มบนหน้าเอกสาร และทุก route มี seq ที่ไม่ชนกัน
- วงเงินอนุมัติอยู่ที่เดียว (ยังไม่ตัดสินว่า condition_json หรือ common_code — ห้ามเก็บสองที่)
- [conditional] ถ้าเลือกทางเลือก A ของ workflow_part_display — การแสดงผลรายส่วนอ่านจาก display[] ของ getPermissionEvents ได้จริง
- ความเสี่ยงเรื่อง workflow_transaction ไม่มี PK/index ถูกยื่นเป็นเรื่องต่อทีมเจ้าของ library แล้ว
- ชื่อ function ที่จะใช้จริงถูกยืนยันกับทีมเจ้าของ library ก่อนเขียนโค้ด

11. Developer Test Checklist

No	Test
1	initializeWorkflow ซ้ำด้วย referenceId เดิมต้องไม่เกิด transaction ที่สอง
2	ยอดชดเชย 99,999 ต้องจบที่ GM · 100,000 ต้องวิ่งต่อ AVP (เกณฑ์เดียว · มติ 2026-08-18)
3	ผู้ใช้ที่ไม่ใช่ current_approver ต้องไม่ได้ปุ่มใด ๆ จาก getPermissionEvents
4	[conditional · เฉพาะทางเลือก A ของ workflow_part_display] display[] ของ state 01 ต้องเปิด WRITE เฉพาะส่วนที่ section 01 แก้ได้
5	getPendingFlowByUser ต้องคืน url_main ที่เปิดกลับหน้าเอกสารได้จริง
6	เดิน flow จนจบแล้ว workflow_history มีครบทุกขั้น