

트리거 가이드

처리 우선순위 흐름

메시지 도착	
1. 그ㅌ	→ 종료 (canceled)
2. 흥사그즈흐 + 방번호	→ 시작 (새 INSERT)
3. 째흐	→ 정정 (end->start)
4. 그 + 방번호	→ 종료 (end)
5. 째 + 방번호	→ 정정 (시간수정)
6. 방번호만 (차단 신호 없음)	→ 시작 (새 INSERT)
7. 그 외 전부	→ 무시 (message)

우선순위 규칙: 위에서 아래로 먼저 매칭되는 조건이 실행됨. 예) 그ㅌ + 그 가 같이 있으면 그ㅌ(1순위)가 먼저 처리됨.

1. 시작 트리거

새로운 세션을 생성하거나, 종료된 세션을 시작 상태로 되돌리는 트리거.

#	조건	예시 메시지	처리 함수	trigger_type	is_designated
1	방번호 + 아가씨이름	103 이승기 도아	handleSessionStart()	'start'	false
2	방번호 + 수동시간	103 이승기 도아 1534	handleSessionStart()	'start'	false
3	방번호 + 째(지명)	103 이승기 도아 째	handleSessionStart()	'start'	true
4	방번호 + 째 + 수동시간	103 이승기 도아 째 1534	handleSessionStart()	'start'	true
5	흥사그즈흐 + 방번호	103 도아 흥사그즈흐	handleNewSession()	'start'	false
6	째흐 / 재진행	도아 째흐	handleResume()	'end'-'start'	-

시작 차단 - 다음 신호가 있으면 시작 트리거로 잡히지 않음:

- 그 (종료), 그 (연장), 째 (지명순번삭제), 째 (지명반순번삭제), 째 (수정)

상세 설명:

- #1~#2 일반 시작:** 방번호와 아가씨이름이 있으면 시작. 수동시간(1534 등)이 있으면 해당 시간을 start_time으로 사용, 없으면 메시지 수신 시간 사용.
- #3~#4 지명 시작:** 째 신호가 포함되면 동일하게 시작하되, DB에 is_designated=true 저장. 지명 여부는 발송기에서 메시지 내용을 다르게 보낼 때 사용.
- #5 현시간재진행:** 기존 세션과 관계없이 무조건 새 레코드를 INSERT. 현재 시간을 start_time으로 사용.
- #6 재진행:** 가장 최근 종료(trigger_type='end')된 레코드를 찾아서 is_in_progress=true, trigger_type='start'로 UPDATE. data_changed=true, end_sent_at=null로 설정하여 재발송 트리거.

이미 진행 중인 세션이 있으면 (#1~#4): 중복 시작 방지를 위해 무시됨.

2. 종료 트리거

진행 중인 세션을 종료하거나 취소하는 트리거.

#	조건	예시 메시지	처리	trigger_type	is_designated

1	₩ + 방번호 + 시간 있음	103 도아 1.5₩	UPDATE → end	'end'	false
2	₩ + 방번호 + 시간 없음	103 도아 ₩	UPDATE → end	'end'	false
3	₩ + 방번호 + 짜(지명)	103 도아 짜 1.5₩	UPDATE → end	'end'	true
4	ㄱㅌ (취소)	도아 ㄱㅌ	UPDATE → canceled	'canceled'	-

종료 차단 - 다음 신호와 함께면 ₩가 무시됨:

- ㅈ (연장), 짜ㅅㅅ (지명순번삭제), 짜ㅂㅅㅅ (지명반순번삭제)

상세 설명:

- #1 시간 있는 종료: 1.5₩ → usage_duration=1.5로 저장. 이벤트 계산(event_count)도 함께 수행.
- #2 시간 없는 종료: ₩ 만 → usage_duration=null. 나중에 시간만 재전송 가능 (정정 트리거 #4 참고).
- #3 지명 종료: 짜이 포함된 종료. is_designated=true 저장.
- #4 취소: ㄱㅌ 신호. 진행 중인 세션을 찾아 trigger_type='canceled', is_in_progress=false 로 변경. 방번호가 있으면 해당 방의 세션만, 없으면 가장 최근 세션을 취소.

진행 중인 세션이 없을 때 ₩가 오면 (#1~#3):

- 시간이 있으면 → 정정 트리거 #4로 처리 (이미 종료된 레코드의 usage_duration 업데이트)
- 시간이 없으면 → 무시 (INSERT 하지 않음)

3. 정정 트리거

이미 존재하는 세션의 정보를 수정하는 트리거.

#	조건	예시 메시지	처리	data_changed
1	ㅈㅈ 단독 + 시간패턴	ㅈㅈ 103 도아 1534	start_time UPDATE	true
2	ㅈㅈ + ₩ (기존 start → end)	ㅈㅈ 103 도아 1534 1.5₩	수정+종료	false
3	ㅈㅈ + ₩ (기존 end → end)	ㅈㅈ 103 도아 1534 1.5₩	수정+종료	true
4	이미 종료 + 시간 재전송	103 도아 3₩	usage_duration만 UPDATE	설정 안 함
5	ㅈㅈㅎ / 재진행	도아 ㅈㅈㅎ	end→start 되돌림	true

상세 설명:

- #1 시간 수정 (ㅈㅈ 단독): ₩ 없이 ㅈㅈ만 있을 때. 시간패턴(1534 등)을 추출하여 start_time만 변경. data_changed=true로 재발송 트리거. 시간패턴이 없으면 무시됨.
- #2 수정+종료 (trigger_type 변경): ㅈㅈ + ₩ 조합. 기존 레코드가 trigger_type='start' (진행 중)일 때 → trigger_type='end'로 변경됨. trigger_type 자체가 바뀌었으므로 발송기가 새로운 트리거로 자동 감지함 → data_changed=false.
- #3 수정+종료 (trigger_type 동일): ㅈㅈ + ₩ 조합. 기존 레코드가 이미 trigger_type='end' (종료 상태)일 때 → trigger_type='end'로 동일함. trigger_type이 안 바뀌었으므로 발송기가 변경을 감지할 수 없음 → data_changed=true로 재발송 알림.
- #4 시간 재전송: 먼저 103 도아 ₩ (시간 없이 종료)한 후, 나중에 103 도아 3₩로 시간만 보냈을 때. 진행 중인 세션이 없으므로 가장 최근 trigger_type='end' 레코드를 찾아 usage_duration과 event_count만 업데이트. data_changed 설정 안 함 → 발송기가 usage_duration IS NOT NULL + trigger_type='end' 조건으로 직접 감지.

- #5 재진행 (zzz): 가장 최근 종료 레코드를 시작 상태로 되돌림. `trigger_type='end'→'start'`, `is_in_progress=true`, `end_time=null`, `usage_duration=null`, `end_sent_at=null`. `data_changed=true`로 재발송 트리거.

멀티라인 zzz (메시지 첫 줄에 zzz):

메시지 첫 줄이 zzz로 시작하면, 메시지 내 모든 아가씨에게 정정이 적용됨.

zzz3人구	← 첫 줄에 zzz → 전체 정정
907 가을 조아 3ㅠ 제로 2.5ㅠ 초롱 1ㅠ	← 모든 아가씨: zzz + ㅠ
205 태산 이선 2.5 달래 2 빙희 1.5ㅠ	← 달래: zzz + 2ㅠ (정정+종료)
910 이태곤 에이 3ㅠ 정현 2ㅠ	← 에이: zzz + 3ㅠ, 정현: zzz + 2ㅠ

- 첫 줄이 zzz로 시작하면 → ㅠ이 있는 아가씨에게만 `isCorrection=true` 전파
- ㅠ이 없는 아가씨는 정정 적용 안 됨 (새 시작 등 정상 처리)
- 각 아가씨의 ㅠ/시간 등은 개별 줄에서 파싱
- 결과: ㅠ이 있는 아가씨는 zzz + 해당시간ㅠ 와 동일하게 처리

예시:

zzz 907 가을 조아 3ㅠ 제로 2.5ㅠ 수린 연정 여리 2개씩 ㅠ
205 태산 달래

- 조아, 제로, 수린 등: ㅠ 있음 → 정정+종료 (zzz+ㅠ)
- 달래: ㅠ 없음 → 정정 적용 안 됨 → 일반 시작 트리거

4. 무시됨 (else → type: 'message')

어떤 핸들러에도 잡히지 않고 `message_logs`에만 저장되는 경우:

#	조건	예시 메시지	무시 이유
1	○자 (연장)	103 도아 ○자	별도 핸들러 없음, 시작/종료 모두 차단
2	ㅈㅁㅅㅅ (지명순번삭제)	103 도아 ㅈㅁㅅㅅ	별도 핸들러 없음, 시작/종료 모두 차단
3	ㅈㅁㅂㅅㅅ (지명반순번삭제)	103 도아 ㅈㅁㅂㅅㅅ	별도 핸들러 없음, 시작/종료 모두 차단
4	ㅎㅅㄱㅈㅈㅎ + 방번호 없음	도아 ㅎㅅㄱㅈㅈㅎ	방번호 필수인데 없음
5	ㅠ + 방번호 없음	도아 ㅠ	방번호 필수인데 없음
6	zzz + 방번호 없음	zzz 도아 1534	방번호 필수인데 없음
7	zzz + 방번호 + 시간패턴 없음	zzz 103 도아	수정할 시간이 없음
8	아가씨 이름만	도아	방번호도 신호도 없음

참고: 무시된 메시지도 `message_logs` 테이블에는 항상 저장됨.

5. `data_changed` 플래그 정리

발송기가 변경사항을 감지하는 방법은 2가지:

- `trigger_type` 변경 → 자동 감지 (`data_changed` 불필요)
- `data_changed = true` → 수동 알림 (`trigger_type`이 안 바뀔 때 사용)

상황	<code>data_changed</code>	감지 방법

일반 시작	설정 안 함	trigger_type='start'로 감지
일반 종료 (시간 있음)	설정 안 함	trigger_type='end' + usage_duration으로 감지
일반 종료 (시간 없음)	설정 안 함	trigger_type='end'로 감지
취소 (ㅌㅌ)	설정 안 함	trigger_type='canceled'로 감지
ㅈㅈ 시간수정	true	trigger_type 안 바뀌므로 플래그 필요
ㅈㅈ+ㄲ (start→end)	false	trigger_type 변경으로 자동 감지
ㅈㅈ+ㄲ (end→end)	true	trigger_type 안 바뀌므로 플래그 필요
이미 종료 + 시간 재전송	설정 안 함	usage_duration + trigger_type='end'로 감지
ㅈㅈㅎ 재진행	true	end→start 변경이지만 end_sent_at 초기화 필요

신호(Signal) 목록

코드	이름	설명	트리거 영향
ㄱ	END	세션 종료	종료 트리거
ㅈㅈ	CORRECTION	수정 (방번호/시간 변경)	정정 트리거
ㅈㅈㅎ	RESUME	재진행 (종료→시작 되돌림)	정정 트리거
ㅎㅅㅈㅈㅎ	NEW_SESSION	현시간재진행 (새 세션)	시작 트리거
ㅈㅁ	DESIGNATED	지명	시작/종료에 is_designated=true 부여
ㄱㅌ	CANCEL	취소	종료 트리거 (canceled)
ㅇㅈ	EXTENSION	연장	무시 (시작/종료 차단)
ㅈㅁㅅㅅ	DESIGNATED_FEE	지명순번삭제	무시 (시작/종료 차단)
ㅈㅁㅂㅅㅅ	DESIGNATED_HALF_FEE	지명반순번삭제	무시 (시작/종료 차단)