

# LogMate

---

## System Test Result



Version	1.0
Authors	강찬욱, 양승원, 정주연
Date	2025.10.31

## 1. 개요 (Overview)

문서 목적	LogMate 시스템의 테스트 수행 결과를 기록하고 SRS 요구사항 충족 여부를 검증한다.
테스트 대상	<ul style="list-style-type: none"><li>- Agent</li><li>- Streaming Server</li><li>- API Server</li><li>- Frontend Server</li><li>- AI Server</li></ul>
테스트 수행자	강찬욱 / Agent, Streaming Server 단위 테스트 양승원 / Frontend, AI Server 단위 테스트 정주연 / API Server 단위 테스트  System, Acceptance Test는 모두 진행.
참조 문서	LogMate_SRS(v2.0) LogMate_STP(v1.0)

## 2. 테스트 환경 (Test Environment)

### 1. Unit Test

구분	항목	상세 내용
Agent	테스트 도구	JUnit 5, Mockito, WireMock

	테스트 환경	Local 환경 (개발자 PC)
	운영체제	Windows 11
Streaming Server	테스트 도구	JUnit 5, Mockito, Spring Boot Reactor Test, Testcontainers
	테스트 환경	Local Docker 기반 테스트 컨테이너 환경
	운영체제	Ubuntu 22.04 LTS (Docker Desktop)
API Server	테스트 도구	JUnit 5, Spring Boot Test, MockMvc
	테스트 환경	Local Spring Boot 환경
AI Server	테스트 도구	PyTest
	테스트 환경	Local Python 3.10 환경

## 2. System Test

항목	상세 내용
테스트 환경	Netlify + AWS 통합 환경 (실제 배포 버전 기준)
테스트 도구	Postman, OpenSearch DashBoard, Kafka UI
Frontend 환경	Netlify
Backend 연결 서버	API Server (AWS EC2), Streaming Server (AWS EC2), AI Server (AWS EC2)

### 3. Acceptance Test

항목	상세 내용
테스트 환경	Netlify + AWS 통합 환경 (실제 배포 버전 기준)
사용자 접근 환경	브라우저: Chrome
Frontend 환경	Netlify
Backend 연결 서버	API Server (AWS EC2), Streaming Server (AWS EC2), AI Server (AWS EC2)
테스트 수행 위치	브라우저에서 실제 UI 조작을 통한 테스트

### 3. Unit Test 결과

#### 3.1 Agent Unit Test 결과

TC ID	테스트 항목	테스트 데이터	절차 요약	예상 결과	실제 결과	결과	관련 FR / 비고
AG-UT-001	올바른 인자 파싱 성공	--agentId=12345, --email=test@example.com, --password=securePassword	① extract(args) 실행 ② 인자 파싱 후 AgentArguments 생성	agentId="12345", email="test@example.com", password="securePassword" 정상 파싱	예상과 동일한 값으로 객체 생성됨	P	FR-1.1.1 명령행 인자 파싱 기능
AG-UT-002	필수 인자(agentId) 누락	--email, --password만 포함	① extract(args) 실행 ② 누락된 인자 검증 수행	IllegalArgumentException 발생 메시지: "Missing required argument: --agentId="	예외 발생 및 메시지 일치 확인	P	FR-1.1.1
AG-UT-003	필수 인자(email) 누락	--agentId, --password만 포함	① 인자 파싱 ② email 누락 검증	IllegalArgumentException 발생 "Missing required argument: --email="	예외 발생 및 메시지 일치	P	FR-1.1.1
AG-UT-004	필수 인자(password) 누락	--agentId, --email만 포함	① 인자 파싱 ② password 누락 검증	IllegalArgumentException 발생 "Missing required argument: --password="	예외 발생 및 메시지 일치	P	FR-1.1.1
AG-UT-005	불필요한 인자 무시	위 정상 인자 + extraArgument	① extract(args) 실행 ② prefix 없는 인자 무시	extraArgument는 무시되고 나머지 정상 파싱	예상대로 동작, 추가 인자 무시됨	P	FR-1.1.1
AG-UT-006	중복 인자 발생 시 예외 처리	--agentId 두 번 포함 (12345, 54321)	① 중복 인자 파싱 시 충돌 검증	IllegalArgumentException 발생 "Duplicate argument: agentId"	예외 발생 및 메시지 일치	P	FR-1.1.1
AG-UT-007	인자 값이 비어 있는 경우 예외 처리	--agentId=	① 파싱 시 값이 비어 있음을 검증	IllegalArgumentException 발생 "Missing required argument: --agentId="	예외 발생 및 메시지 일치	P	FR-1.1.1

AG-UT-008	PullerConfig YAML 파일 정상 로드	valid-puller-config.yml 내용 예시: pullURL: url intervalSec: 30	YamlConfigLoader에 파일명 전달. loadPullerConfig() 호출	PullerConfig 객체가 정상 생성되고 pullURL="url", intervalSec=30으로 맵핑됨	PullerConfig가 정상 생성되었고 필드 값 일치 확인	P	FR-1.1.3
AG-UT-009	PullerConfig 파일이 존재하지 않을 때 예외 처리	non-existent-config.yml	존재하지 않는 파일명으로 로더 생성 loadPullerConfig() 실행	YAMLEException 발생 (SnakeYAML 내부 IOException 포장)	YAMLEException이 발생하여 테스트 통과	P	FR-1.1.3
AG-UT-010	PullerConfig YAML 형식 오류 시 처리	invalid-puller-config.yml (잘못된 구조의 YAML 예: {invalid: })	잘못된 YAML 파일명 전달. loadPullerConfig() 실행.	ConstructorException 발생 (SnakeYAML 파싱 실패)	ConstructorException 발생하여 테스트 통과	P	FR-1.1.3
AG-UT-011	AgentConfig, PullerConfig, PipelineConfig 유효성 검증	- AgentConfig: null, 공백 agentId / accessToken / eTag - PullerConfig: null, 잘못된 URL, intervalSec < 10	① ConfigValidator.validate()에 각각의 잘못된 설정 전달 ② IllegalArgument Exception 발생 여부 확인	각 케이스별로 IllegalArgumentException 발생 예외 메시지는 항목별 명세와 일치	모든 잘못된 입력에서 IllegalArgumentException 정상 발생	P	FR-1.1.4
AG-UT-012	Agent 인증 성공	AgentArguments → ID=testAgentId, Email=testEmail@example.com, Password=testPassword AuthClient.login() → Optional(LoginResponse: "mockAccessToken")	① AgentAuthenticator 실행 ② AuthClient.login() Mock 호출 ③ 응답이 존재할 경우 AgentConfigHolder에 토큰 저장	AuthClient.login()이 1회 호출되고 AgentConfigHolder에 "mockAccessToken" 저장됨	AuthClient.login() 정상 호출 및 액세스 토큰 저장 확인됨	P	FR-1.1.5

AG-UT-013	Agent 인증 실패 (잘못된 자격)	AgentArguments → ID=testAgentId, Email=wrongEmail@example.com, Password=wrong Password AuthClient.login() → Optional.empty()	AgentAuthenticator 실행 잘못된 자격으로 로그인 시도 응답 없음 → 예외 발생	RuntimeException 발생, 초기화 중단, 오류 로그 출력	RuntimeException 발생 및 AuthClient.login() 1회 호출 확인됨	P	FR-1.1.5 Agent 인증 실패 시 오류 처리
AG-UT-014	Agent 인증 실패 (응답 없음)	AgentArguments → ID=testAgentId, Email=testEmail@example.com, Password=testPassword AuthClient.login() → Optional.empty()	AgentAuthenticator 실행 응답 Optional 비어 있음 예외 발생	RuntimeException 발생	RuntimeException 정상 발생 확인됨	P	FR-1.1.5 응답 비정상 시 예외 처리 검증
AG-UT-015	멀티라인 비활성화 시 Raw 라인 그대로 반환	isEnabled = false, rawLines = ["line1","line2"]	① MultilineProcessor 생성 ② process() 호출	입력과 동일한 배열 반환 (["line1","line2"])	반환값이 입력과 동일하며 LogParser 호출 없음	P	FR-1.4.2
AG-UT-016	모든 라인이 정상 포맷일 때 수정 없이 반환	isEnabled = true, 모든 라인 isFormatCorrect = true	① 정상 라인 두 개 입력 ② process() 호출	입력과 동일한 배열 반환	예상과 일치	P	FR-1.4.1 FR-1.4.3
AG-UT-017	모든 라인이 비정상일 때 병합 및 버퍼 플러시	isFormatCorrect = false, maxLines = 3, rawLines = ["line1","line2","line3","line4"]	① process() 호출 ② 3줄씩 병합 후 플러시	[MERGED]\nline1\nline2\nline3, [MERGED]\nline4 반환	병합 출력 예상과 일치	P	FR-1.4.1 FR-1.4.3 FR-1.4.4
AG-UT-018	정상 라인 등장 시 버퍼 플러시 후 정상 라인 추가	line1,line2 = false, line3 = true, maxLines = 3	① process() 호출 ② 병합 후 정상 라인 별도 출력	[MERGED]\nline1\nline2, line3 반환	결과 예상과 일치	P	FR-1.4.5
AG-UT-019	입력 종료 시 남은 버퍼 자동 플러시	isFormatCorrect = false, rawLines = ["line1","line2"], maxLines = 5	① process() 호출 ② 남은 버퍼 자동 출력	[MERGED]\nline1\nline2 반환	예상 결과와 일치	P	FR-1.4.5

AG-UT-020	Z(UTC) 타임존 로그 파싱	"2025-09-21T10:15:30Z INFO ..."	① parse() 호출 ② UTC 그대로 처리	timestamp=2025-09-21T10:15:30Z, 필드 모두 정상 파싱	모든 필드 값 및 UTC 일치	P	FR-1.5.2
AG-UT-021	+09:00 오프셋 포함 로그 파싱	"2025-09-21T19:15:30+09:00 INFO ..."	① parse() 호출 ② UTC 변환 확인	timestamp=10:15:30Z (변환), 필드 정상	변환 값 일치	P	FR-1.5.2
AG-UT-022	밀리초 포함 오프셋 로그 파싱	"2025-09-21T19:15:30.987+09:00 INFO ..."	① 밀리초 제거 후 UTC 변환 확인	timestamp=10:15:30Z, 필드 정상	변환 정상	P	FR-1.5.2
AG-UT-023	타임존 누락 시 Asia/Seoul 적용	"2025-09-21T19:15:30 INFO ..."	① 로컬 타임존 해석 후 UTC 변환	timestamp=10:15:30Z	변환 일치	P	FR-1.5.2
AG-UT-024	빈 라인 입력 시	" "	① 파싱 실패 ② 현재 UTC 시간 사용	isFormatCorrect=false, timestamp≈now()	결과 일치	P	FR-1.5.1 FR-1.5.3 FR-1.5.4
AG-UT-025	여분의 공백 포함 로그 파싱	"2025-09-21T10:15:30Z INFO 123 --- ..."	① 토큰 간 공백 정규화	필드 정상 파싱	결과 일치	P	FR-1.5.1 FR-1.5.3 FR-1.5.4
AG-UT-026	잘못된 타임스탬프 처리	"InvalidTime INFO 123 ..."	① 타임스탬프 파싱 실패 시 fallback	isFormatCorrect=false, timestamp≈now()	결과 일치	P	FR-1.5.1 FR-1.5.3 FR-1.5.4
AG-UT-027	필드 누락 로그 처리	"2025-09-21T10:15:30Z INFO --- :"	① 필드 검증 실패 시 원문 유지	isFormatCorrect=false, message=원문	결과 일치	P	FR-1.5.1 FR-1.5.3 FR-1.5.4
AG-UT-028	타임스탬프만 존재	"2025-09-21T10:15:30Z"	① 필드 파싱 불가 → 실패 처리	isFormatCorrect=false, message=원문	결과 일치	P	FR-1.5.1 FR-1.5.3 FR-1.5.4
AG-UT-029	완전히 잘못된 로그 포맷	"SOMETHING INVALID LOG"	① 파싱 실패 시 현재 UTC 시간 사용	isFormatCorrect=false, timestamp≈now()	결과 일치	P	FR-1.5.1 FR-1.5.3 FR-1.5.4
AG-UT-030	허용된 로그 레벨 및 키워드 모두 만족	allowedLevels=["INFO"], requiredKeywords=["error"], message="This is an error message"	① 필터 생성 후 로그 전달 ② 조건 일치 확인	accept(log)=true	로그 허용됨	P	FR-1.6.3

AG-UT-031	로그 레벨 불일치로 필터 차단	allowedLevels=["WARN"], log.level="INFO"	① 필터 생성 후 로그 전달 ② 레벨 불일치 검증	accept(log)=false	로그 차단됨	P	FR-1.6.3
AG-UT-032	필수 키워드 누락으로 필터 차단	requiredKeywords = ["critical"], message="Normal message"	① 필터 생성 ② 메시지에 키워드 부재 확인	accept(log)=false	로그 차단됨	P	FR-1.6.4
AG-UT-033	필터링 설정이 비어있는 경우 모두 허용	allowedLevels=[], requiredKeywords = [], allowedMethods=[ ]	① 필터 생성 후 로그 전달 ② 모든 조건 비활성화	accept(log)=true	로그 허용됨	P	FR-1.6.3 FR-1.6.4
AG-UT-034	허용된 HTTP 메서드 매칭 시 로그 허용	allowedMethods=[ "GET"], log.method="GET "	① 필터 생성 후 로그 전달 ② 메서드 일치 확인	accept(log)=true	로그 허용됨	P	FR-1.6.5
AG-UT-035	메서드 불일치로 로그 차단	allowedMethods=[ "POST"], log.method="GET "	① 필터 생성 후 로그 전달 ② 불일치 확인	accept(log)=false	로그 차단됨	P	FR-1.6.5
AG-UT-036	메서드 필터링 비활성화 시 모두 허용	allowedMethods=[ ], log.method="DELETE"	① 필터 생성 후 로그 전달 ② 메서드 필터 없음	accept(log)=true	로그 허용됨	P	FR-1.6.5
AG-UT-037	파일 미존재 시 자동 생성	삭제된 파일 상태에서 save() 실행	① 폴더·파일 존재 여부 확인 ② 저장 후 로드	파일 자동 생성 + 1개 로그 로드됨	예상 일치	P	FR-1.8.1
AG-UT-038	로그 전송 성공 및 응답 처리	로그 1건 (INFO log1), 서버응답 200 OK	① Mock 서버 200 응답 설정 ② export(logs) 호출	서버가 200 응답일 때 성공 처리, 실패 로그 없음	전송 성공 및 실패 로그 0건 확인	P	FR-1.7.1 / FR-1.7.7 (로그 전송 및 응답 코드 처리)
AG-UT-039	배치 단위 전송 확인	로그 3건 (log1, log2, log3), 배치 크기 100 B	① PayloadBatcher Spy 생성 ② export() 호출	로그가 배치 단위로 분할 전송됨 (split() 호출 확인)	split() 메서드 정상 호출 확인	P	FR-1.7.2 (배치 단위 처리)
AG-UT-040	전송 재시도 정책 검증	첫 2회 500 응답 후 3번째 200 응답 시나리오	① Mock 서버 시나리오 설정	최대 3회 이내 재시도 후 성공, 실패 로그 없음	2회 시도 후 200 응답으로 성공 확인	P	FR-1.7.3

			② maxRetryCount =3 설정 후 export() 호출				
AG-UT-041	중복 전송 방지 확인	동일 로그 1건, maxRetryCount = 3	① 서버 200 응답 설정 ② export() 호출 후 전송 횟수 검증	성공 응답 후 재전송 없음 (1회 요청)	요청 1회 전송 확인	P	FR-1.7.4
AG-UT-042	인증 헤더 포함 전송	accessToken = token-123, 로그 1건	① Mock 서버 Authorization 헤더 검증 ② export() 호출	헤더에 Authorization: Bearer token-123 포함	헤더 검증 성공	P	FR-1.7.5
AG-UT-043	GZIP 압축 전송 확인	compressEnabled = true, 로그 1건	① 압축 옵션 활성화 ② export() 호출 및 헤더 검증	요청 헤더에 Content-Encoding: gzip 포함	Mock 서버 헤더 검증 성공	P	FR-1.7.6
AG-UT-044	로그 저장 후 정상 로드	List<ParsedLogData> 2개 (message, message2)	① save() 호출 ② loadAll() 수행	2개의 로그가 저장·로드됨 message, message2 순서 유지	결과 일치	P	FR-1.8.2 FR-1.8.3
AG-UT-045	save() 호출 시 로그 추가(append)	로그 2회 저장 (message1, message2)	① 첫 저장 후 재저장 ② loadAll() 실행	2개 로그 순서대로 존재	실제 순서 유지됨	P	FR-1.8.2 FR-1.8.3
AG-UT-046	clear() 호출 시 로그 파일 비움	로그 1개 저장 후 clear() 실행	① 저장 → clear → loadAll	빈 리스트 반환 (0건)	정상 작동	P	FR-1.8.6

### 3.2 Streaming Server Unit Test 결과

TC ID	테스트 항목	테스트 데이터	절차 요약	예상 결과	실제 결과	결과 (Pass/Fail)	관련 FR / 비고
ST-UT-001	SpringBoot 로그 수신 성공	POST /api/v1/streaming/logs/springboot/{agentId}/{thNum}Body: 정상 로그 JSON	① agentId, thNum 유효값으로 요청 ② logProducer.sendLog() Mock 성공 반환 ③ 응답 확인	HTTP 200 OK로그 DTO → SpringBootParsedLog 매플 정상logProducer 호출 1회 이상 확인	예상과 일치	P	FR-2.1.1 (로그 수신)

ST-UT-002	Tomcat 로그 수신 성공	POST /api/v1/streaming/logs/tomcat/{agentId}/{thNum}Body: 정상 로그 JSON	① logProducer.sendLog() Mock 성공 ② 응답 코드 확인	HTTP 200 OK TomcatAccess ParsedLog로 매핑 성공	예상과 일치	P	FR-2.1.1
ST-UT-003	Agent ID 누락 시 검증 실패	agentId = ""	① validate() 수행 ② 예외 발생 여부 확인	IllegalArgumentExceptionException 발생 HTTP 400 Bad Request	예외 발생 확인	P	FR-2.1.3 (유효성 검사)
ST-UT-004	잘못된 쓰레드 번호 입력	thNum = "abc"	① validate() 수행 ② 예외 발생 확인	IllegalArgumentExceptionException 발생	예외 발생 확인	P	FR-2.1.3
ST-UT-005	GZIP 압축 로그 자동 해제	Header: Content-Encoding : gzipBody: Gzip 압축 JSON	① Gzip 압축된 로그 요청 전송 ② 필터 작동 여부 로그 확인	[GzipDecodingWebFilter] Detected gzip request 로그 출력 Body 자동 복원 컨트롤러까지 전달	예상과 일치	P	FR-2.1.5 (압축 해제)
ST-UT-006	압축 해제 실패 (손상된 gzip 데이터)	gzip header만 존재	① 잘못된 gzip 바디 전송 ② 예외 발생 확인	IllegalArgumentExceptionException 발생 (Failed to decompress gzip body)	예외 발생	P	FR-2.1.5
ST-UT-007	압축 해제 후 Content-Type 변경 확인	Header: Content-Encoding : gzip	① 요청 수행 후 Header 확인	Content-Type이 application/json으로 변경 Content-Encoding 제거됨	일치 확인	P	FR-2.1.5
ST-UT-008	로그 파서 매핑 검증	LogType = SPRING_BOOT	① JSON 로그 요청 ② ParsedLogMapper.toEntity() 호출 확인	DTO → SpringBootParsedLog 정상 변환	일치	P	FR-2.1.4
ST-UT-009	로그 전송 실패 시 에러 처리	logProducer.sendLog() 에러 throw	① onErrorResume() 로직 검증	에러 로그 출력 후 스트림 정상 종료 (Mono.empty())	예상과 일치	P	FR-2.1.1
ST-UT-010	압축 크기 제한 초과	20MB gzip body	① maxBytes 초과 gzip 전송 ② 예외 확인	IllegalArgumentExceptionException ("Compressed body too large") 발생	예상과 일치	P	FR-2.1.5

ST-UT-011	정상 입력 검증 통과	agentId=123e4567-e89b-12d3-a456-426614174000t hNum=1234	① validate() 호출	예외 없이 통과	일치	P	FR-2.1.3 유효성 검사
ST-UT-012	agentId = null	agentId=null, thNum=1234	① validate() 호출 ② 예외 발생 확인	InvalidRequestException 발생	일치	P	FR-2.1.3
ST-UT-013	agentId 공백 문자열	" ", "1234"	① validate() 호출	InvalidRequestException 발생	일치	P	FR-2.1.3
ST-UT-014	agentId 형식 불일치	"invalid-agent-id", "1234"	① validate() 호출	InvalidRequestException 발생	일치	P	FR-2.1.3
ST-UT-015	threadNumber = null	agentId 유효, thNum=null	① validate() 호출	InvalidRequestException 발생	일치	P	FR-2.1.3
ST-UT-016	threadNumber 공백 문자열	"123e4567...", "	① validate() 호출	InvalidRequestException 발생	일치	P	FR-2.1.3
ST-UT-017	threadNumber 비수자 문자열	"123e4567...", "abc"	① validate() 호출	InvalidRequestException 발생	일치	P	FR-2.1.3
ST-UT-018	threadNumber 범위 초과 (음수)	"123e4567...", "-1"	① validate() 호출	InvalidRequestException 발생	일치	P	FR-2.1.3
ST-UT-019	threadNumber 범위 초과 (65536 이상)	"123e4567...", "65536"	① validate() 호출	InvalidRequestException 발생	일치	P	FR-2.1.3

### 3.3 API Server Unit Test 결과

TC ID	테스트 항목	테스트 데이터	절차 요약	예상 결과	실제 결과	결과 (Pass / Fail)	관련 FR / 비고
AP-UT-001	로그인 성공/실패	{ "email": "test@example.com", "password": "test1234!!" }	정상/오류 자격으로 로그인 요청	성공 시 200 및 JWT 발급, 실패 시 오류 메시지	예상과 일치	P	FR-3.1.1
AP-UT-002	회원가입 중복 이메일	GET api/users/check-email?email=test@example.com	이미 등록된 이메일로 가입 요청	중복 오류 반환	예상과 일치	P	FR-3.1.2
AP-UT-003	로그아웃 처리	POST /api/users/logout	토근 폐기 요청	블랙리스트/세션 종료 처리	예상과 일치	P	FR-3.1.3

AP-UT-004	팀 목록 조회	GET /api/teams	로그인 사용자의 팀 목록 조회	본인 소속 팀만 반환	예상과 일치	P	FR-3.2.1
AP-UT-005	대시보드 목록/상태 조회	GET /api/folders/{folderId}/dashboards	특정 팀/폴더 대시보드 조회	목록 + 상태 필드 포함 반환	예상과 일치	P	FR-3.2.2
AP-UT-006	대시보드 생성/수정	POST /api/folders/{folderId}/dashboards PUT /api/folders/{folderId}/dashboards/{dashboardId} "name": "Error Dashboard", "logPath": "/var/log/app/error.log"	설정 정보로 생성/수정 요청	DB 반영 및 값 일치	예상과 일치	P	FR-3.2.3
AP-UT-007	고급 설정 등록/조회 및 eTag 발급, 갱신	"agentId": null, "puller": { "intervalSec": 5 }, "logPipelineConfigs": [ //고급 설정 정보들 ]	파서/필터/멀티 라인 설정 등록	저장 및 eTag 갱신, 조회 일치	예상과 일치	P	FR-3.2.4, FR-3.2.5
AP-UT-008	Agent Pulling 지원	GET api/config?agentId=2043f57e-a432-4584-8108-3cd26683f220&etag=b3976bc9-83c4-4dea-8b5e-8c95cbf79d99	agentId + etag(same/changed) 요청	변경 시 JSON, 동일 시 304	예상과 일치	P	FR-3.2.6
AP-UT-009	대시보드 삭제	DELETE /api/folders/{folderId}/dashboards/{dashboardId}	삭제 요청 수행	삭제 정책에 맞게 처리	예상과 일치	P	FR-3.2.7
AP-UT-010	팀 정보 수정	"name": "팀 수정 이름", "description": "팀 수정 설명", "members": [ { "email": "test1234@example.com", "role": "VIEWER" }, { "email": "test12345@exa	팀 이름/설명 변경 요청	필드 업데이트	예상과 일치	P	FR-3.3.1

		mple.com" , "role": "ADMIN"} ]					
AP-UT-011	권한변경(ADMIN만)	"name": "팀 수정 이름", "description": "팀 수정 설명", "members": [ { "email": "test1234@example.com", "role": "VIEWER" }, { "email": "test12345@example.com", "role": "ADMIN" } ]	ADMIN 아닌 사용자가 권한 변경 요청	거부 처리	예상과 일치	P	FR-3.3.2
AP-UT-012	팀 생성/삭제시 권한 검증	"name": "팀 생성", "description": "테스트용 팀", "members": [ { "email": "test1234@example.com", "role": "ADMIN" }, { "email": "test12345@example.com" } ]	팀 생성/삭제 API 호출	권한 검증 및 정상 처리	예상과 일치	P	FR-3.3.4, FR-3.3.5
AP-UT-013	프로필 조회/수정/탈퇴	{     "currentPassword": "test11111!",     "newEmail": "juyeonNewNewNew@email.com",     "newPassword": "test1111!" }	마이페이지 관련 API 호출	정상 조회/수정, 탈퇴 처리	예상과 일치	P	FR-3.4.1, FR-3.4.2, FR-3.4.4
AP-UT-014	Webhook 등록/테스트/전송 실패 기록	<pre>POST/api/webhooks {   "type": "SLACK",   "url": "https://hooks.slack.com/services/XXXXX/XXXXX/XXX" }</pre> <pre>POST /api/webhooks/test?url=https://xxxx/xxx</pre> <pre>Post /api/webhooks/t</pre>	URL 저장, 테스트 전송, 실패 기록 확인	각 단계 결과가 DB/로그에 반영	예상과 일치	P	FR-3.6.1 ~FR-3.6.3

		rigger?message= 트리거 발생					
--	--	---------------------------	--	--	--	--	--

### 3.4 AI Server Unit Test 결과

TC ID	테스트 항목	테스트 데이터	절차 요약	예상 결과	실제 결과	결과 (Pass/Fail)	관련 FR / 비고
AI-UT-001	모델 로드 성공	Pkl파일 기반 모델 로드	초기화 호출	인스턴스 생성 및 가중치 로드 OK	일치	Pass	FR-5.1.1
AI-UT-002	모델 로드 실패	Pkl 파일이 없을 경우 로드 실패	초기화 호출(오류 조건)	예외(또는 오류 코드)·오류 로그 기록	일치	Pass	FR-5.1.1
AI-UT-003	전처리 유효성	올바른 형식에 맞는 로그 전송	extract_features 호출	피처 형식에 맞는 로그 형식을 수신	일치	Pass	FR-5.1.1
AI-UT-004	전처리 입력 오류 처리	올바르지 않은 형식에 맞는 전송	extract_features 호출(유효하지 않은 입력)	명시적 Validation 실패 반환(올바른 형태가 아닌 메시지 출력)	일치	Pass	FR-5.1.1
AI-UT-005	스코어 범위 보장	임의의 웹 로그 데이터로 전송	compute_score_for_log 호출	모델의 출력 값( $0.0 \leq score \leq 1.0$ ) 을 0~100사이의 점수로 반환	일치	Pass	FR-5.1.1
AI-UT-006	임계값 판정	임의의 정상/비정상 로그 전송	is_anomaly(score, thr) 검증	75점 이상의 점수는 anomaly로 판단	일치	Pass	FR-5.1.1

### 3.5 Frontend Unit Test 결과

TC ID	테스트 항목	테스트 데이터	절차 요약	예상 결과	실제 결과	결과 (Pass/Fail)	관련 FR / 비고
FE-UT-001	로그인 품 유효성 & API 호출	- 올바른 이메일/비밀번호 - 잘못된 이메일/비밀번호	유효/무효 입력 후 로그인 호출	정상입력 또는 에러 표시	일치	Pass	FR-4.1.1

FE-UT-002	회원가입 품	유효/무효 입력값 (이메일, 비밀번호 등)	필수 필드 검증 후 가입 호출	성공 메시지/다음 화면 이동	일치	Pass	FR-4.1.2
FE-UT-003	로그아웃	로그인된 상태	로그아웃 버튼 클릭	토큰 삭제 및 비로그인 상태 전환	일치	Pass	FR-4.1.3
FE-UT-004	팀 선택 및 데이터 로드	팀 설정창에 존재하는 데이터	팀 선택 UI에서 팀 변경	선택 팀의 데이터 로드/표시	일치	Pass	FR-4.2.1
FE-UT-005	대시보드 상태 표시	로그 수집중인지 아닌지	목록 API 결과 렌더링	상태 배지/필드 표시	일치	Pass	FR-4.2.2
FE-UT-006	대시보드 추가/수정 품	대시보드 추가/수정 버튼 클릭	입력값 바인딩/저장 호출	성공/실패 메시지 처리	일치	Pass	FR-4.2.3
FE-UT-007	대시보드 설정 전송	대시보드 설정 저장 후 조회	이름/경로/전 송주소 등 입력	API 성공 시 저장됨	일치	Pass	FR-4.2.4
FE-UT-008	파싱 룰 미리보기	대시보드 고급설정 창 입력	샘플 로그 수신 후 미리보기	구조화 결과 렌더링	일치	Pass	FR-4.2.6
FE-UT-009	필터 조건 등록	대시보드 고급설정 창 입력	키워드/레벨 입력 후 저장	서버에 조건 저장 요청	일치	Pass	FR-4.2.7
FE-UT-010	권한 UI 제약	팀 페이지 팀원 설정 변경	비ADMIN 사용자의 권한 버튼	비활성/미노출 처리	일치	Pass	FR-4.3.2
FE-UT-011	마이페이지 조회/변경	마이페이지 클릭/올바른 패스워드 입력 후 변경	프로필 조회/수정 UI	결과 메시지 및 상태 반영	일치	Pass	FR-4.4.1~ FR-4.4.3
FE-UT-012	알림 목록/읽음/ 스크롤	모든 페이지 서칭(팀, 개인, 대시보 드 페이지 클릭)	무한 스크롤 및 읽음 처리	최신순 표시/다음 페이지 로드	일치	Pass	FR-4.5.1~ FR-4.5.4
FE-UT-013	외부 알림 설정 UI	내 설정 페이지에서 웹훅 등록/조회	Webhook URL 입력/저장	저장됨 표시	일치	Pass	FR-4.6.1

FE-UT-014	실시간 로그 수신/필터	대시보드에서 들어오는 로그 조회	WebSocket 연결/필터 전달	수신 로그 렌더 및 필터 적용	일치	Pass	FR-4.7.1, FR-4.7.2
-----------	-----------------	----------------------	--------------------------	------------------------	----	------	-----------------------

#### 4. System Test 결과

TC ID	테스트 항목	테스트 환경 / 데이터	절차 요약	예상 결과	실제 결과	결과	관련 FR
ST-001	Agent ↔ API Server 인증 통신	Agent 실행, API Server 가동	Agent가 실행 시 API 서버 /api/v1/agents/auth에 ID/PW로 인증 요청	HTTP 200 응답, AccessToken 수신	AccessToken 정상 발급 및 로그 기록 확인	P	FR-1.1.5, FR-3.2.6
ST-002	Agent ↔ API Server 설정 Pull	API 서버에 Dashboard Config 존재	Agent 주기적 Pull 요청 수행	eTag 비교 후 설정 변경 시 JSON 설정 수신	변경 시 설정 반영 로그 출력	P	FR-1.2.1~ 1.2.4
ST-003	Agent ↔ Streaming Server 로그 전송	Agent tailing 활성	Agent가 수집 로그를 /api/v1/streaming/logs/streamingboot/{agentId}로 전송	Streaming 서버 로그 수신 및 ACK(200 OK) 반환	수신 로그 개수와 일치 확인	P	FR-1.3.1, FR-2.1.1
ST-004	Streaming Server ↔ Kafka Messaging	Kafka broker 연결	Streaming 서버가 수신 로그를 log-stream 토픽에 produce	Kafka consumer에서 동일 메시지 수신 확인	Kafka UI 및 로그에서 확인됨	P	FR-2.2.1

ST-005	Streaming Server ↔ AI Server 로그 분석 요청	AI 서버 REST API /api/v1/analyze 활성	Streaming 서버가 로그 batch를 AI 서버로 POST	AI 서버로부터 anomaly score 응답 수신	JSON 응답(score, label) 정상	P	FR-2.3.1, FR-5.1.1
ST-006	Streaming Server ↔ OpenSearch 로그 저장	OpenSearch 인덱스 생성됨	Streaming 서버가 Kafka에서 consume한 로그를 OpenSearch에 저장	log-yyyy.MM .dd 인덱스에 로그 적재	Kibana 검색 시 데이터 확인됨	P	FR-2.3.2
ST-007	Streaming Server ↔ Frontend 실시간 로그 스트리밍	WebSocket 연결	Frontend가 /ws/logs 구독 요청	Streaming 서버가 새 로그 발생 시 실시간 push	Frontend 로그 창에 실시간 표시	P	FR-2.3.3, FR-4.7.1
ST-008	Frontend ↔ API Server 사용자 인증	로그인 UI에서 POST /api/v1/auth/ login	이메일/비밀번호 전송	JWT 토큰 및 사용자 정보 반환	로그인 성공 후 대시보드 이동	P	FR-3.1.1, FR-4.1.1
ST-009	Frontend ↔ API Server 설정 조회	로그인된 상태	GET /api/v1/dashboard/config 호출	JSON Config 응답 수신	대시보드 설정 값 일치	P	FR-3.2.5, FR-4.2.4
ST-010	API Server ↔ Frontend Webhook 연동	Slack Webhook 등록	API 서버가 알림 이벤트 발생 시 Slack POST 전송	Slack 메시지 수신 확인	Slack에 정상 알림 수신	P	FR-3.6.1, FR-3.6.3
ST-011	Frontend ↔ Streaming Server 로그 검색	로그 저장 완료 상태	/api/v1/streaming/logs/se arch?agentId=... 호출	검색 결과 JSON 리스트 반환	UI 결과 일치	P	FR-2.4.1

ST-012	AI Server ↔ Streaming Server 이상 탐지 후 알림	비정상 로그 전송	AI 서버 anomaly score > 0.8 시 알림 생성	Streaming 서버가 알림 큐로 전달	Slack 및 UI 알림 표시	P	FR-2.3.1, FR-5.1.5
--------	---	-----------	-----------------------------------	------------------------	------------------	---	--------------------

## 5. Acceptance Test 결과

TC ID	시나리오	사전 조건	절차 요약	예상 결과	실제 결과	결과	관련 FR / 비고
AT-001	사용자 로그인	계정 등록 완료	로그인 정보 입력 → 로그인 클릭	대시보드로 이동	정상 로그인 및 화면 전환	Pass	FR-3.1.1, FR-4.1.1
AT-002	회원가입 및 팀 생성	초기화면	회원가입 → 팀 생성	새 계정 및 팀 생성 성공	정상 등록 및 로그인	Pass	FR-3.1.2, FR-3.3.4
AT-003	대시보드 추가	로그인 상태	'대시보드 추가' 클릭 → 경로 등록	대시보드 생성	UI 및 DB 반영 확인	Pass	FR-3.2.3, FR-4.2.3
AT-004	실시간 로그 표시	Agent 연결됨	'수집 시작' 클릭	로그 실시간 표시	대시보드에 로그 표시	Pass	FR-3.7.1, FR-4.7.1

AT-005	로그 필터 적용	로그 존재	'ERROR' 필터 설정	ERROR 로그만 표시	정상 필터링 확인	Pass	FR-4.7.2
AT-006	AI 이상 탐지 확인	AI 서버 작동	공격 로그 발생	이상 탐지 알림 표시	Dashboard에 이상 로그 표시	Pass	FR-5.1.5
AT-007	로그 검색	로그 저장 완료	검색어 입력 → 검색	검색 결과 표시	결과 정상 표시	Pass	FR-2.4.1
AT-008	사용자 정보 수정	로그인 상태	마이페이지 → 이메일 수정	정보 갱신 성공	수정 후 즉시 반영	Pass	FR-3.4.2
AT-009	로그 재전송 복구	네트워크 장애	로그 실패 후 복구	Fallback 재전송 성공	정상 재전송 확인	Pass	FR-1.8.5
AT-010	대시보드 설정 조회	대시보드 존재	설정 화면 접근	고급 설정 정보 표시	Etag 및 Config 확인	Pass	FR-3.2.5
AT-011	Webhook 테스트 전송	Webhook 등록 전	테스트 전송 클릭	테스트 메시지 수신	Discord 테스트 알림 수신	Pass	FR-3.6.2
AT-012	Discord 알림 수신	Webhook 등록	이상점수 70점 이상 로그 발생	Discord 알림 수신	Discord에 메시지 확인	Pass	FR-3.6.3

## 6. 테스트 결과 요약 (Result Summary)

총 테스트 케이스 수	123
성공 (Pass)	123
실패 (Fail)	0
부분 성공 (Partial)	0
성공률 (%)	100%

## 7. 결론

본 프로젝트의 테스트는 Unit Test, System Test, Acceptance Test의 세 단계로 수행되었다.

각 단위 모듈에 대한 Unit Test는 JUnit, Mockito, Reactor Test 등을 이용하여 자동화되었으며, 모든 주요 기능에 대해 정상적인 결과를 확인하였다.

Postman을 이용한 System Test에서는 실제 AWS에 배포된 서버 간 통신을 중심으로 수행하였으며, Agent–Streaming–API–AI–Frontend 간의 데이터 흐름이 정상적으로 이루어지는 것을 검증하였다.

또한 실제 사용자 환경(Netlify 배포 프론트엔드)을 통한 Acceptance Test에서도 전체 서비스 플로우(로그 수집 → 전송 → 저장 → 시각화)가 예상대로 동작함을 확인하였다.

다만 시간적 제약으로 인해 Integration Test 및 Performance Test는 수행하지 못하였다. 이는 향후 시스템의 확장성과 안정성을 검증하기 위해 반드시 추가로 진행되어야 할 과제로 남아 있다.

현재 테스트 케이스의 성공률은 100%로 기록되었으나, 테스트 항목이 비교적 상위 수준(기능 단위)으로 구성되어 있어 케이스를 보다 세분화하여 추가 검증이 필요하다.

결론적으로, 본 테스트를 통해 LogMate 시스템의 기본 동작과 서버 간 연동 기능은 안정적으로 작동함을 확인하였지만 남은 보완 과제로는 통합/성능 테스트의 추가 수행과 테스트 케이스의 체계적 확장, 자동화 수준 향상이 필요하다.