

1단계: 프로젝트 개요

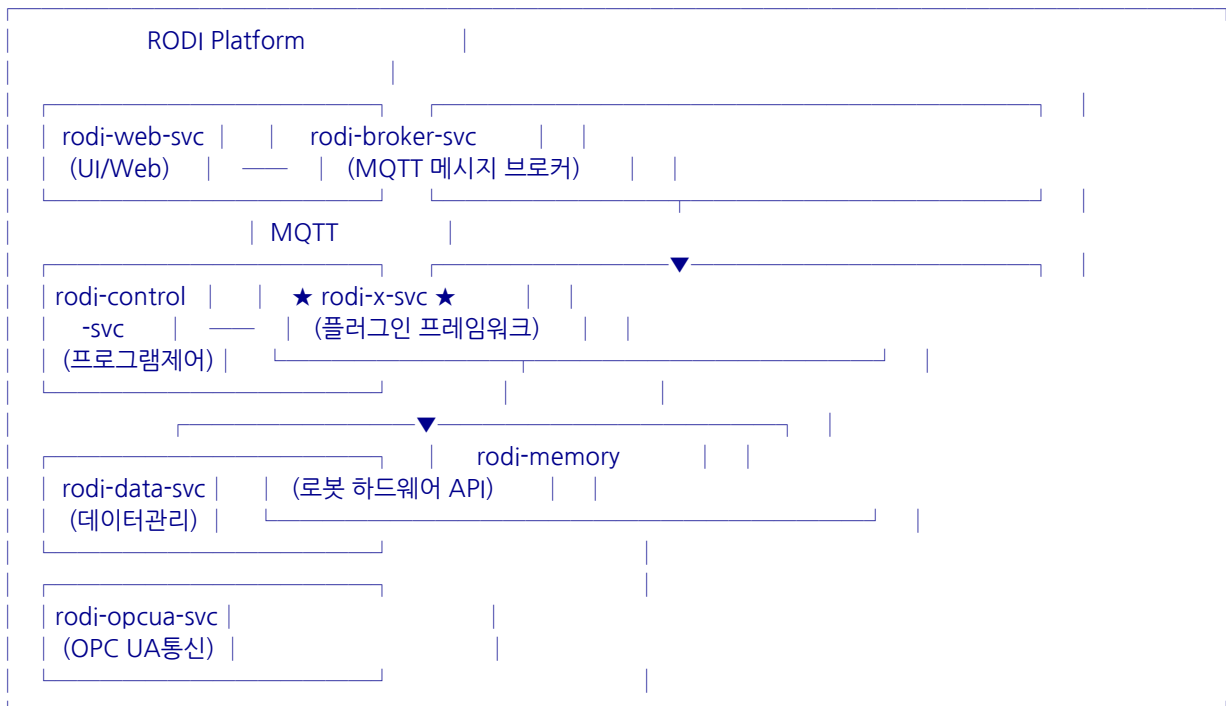
이 서비스는 무엇인가?

rodi-x-svc (Robot Operating System Interactive eXtensions)는 RODI 로봇 제어 플랫폼의 플러그인 실행 및 확장 관리 서비스입니다.

쉽게 말하면, 로봇 프로그래밍 UI에서 사용하는 커스텀 명령어, 확장 페이지, 위젯, 백그라운드 서비스를 플러그인 형태로 로드하고 실행하는 역할을 합니다.

전체 시스템에서의 위치

RODI 플랫폼은 여러 마이크로서비스로 구성됩니다:



rodi-x-svc의 핵심 역할

1. 플러그인 로딩: .asar 아카이브 형태의 플러그인을 추출하고 로드
2. 서비스 실행: 플러그인이 제공하는 4가지 서비스 타입 (프로그램노드, 확장, 위젯, 데몬) 실행
3. 이벤트 중계: MQTT를 통해 UI(web-svc)와 플러그인 사이의 이벤트를 중계
4. 스크립트 생성: 플러그인 코드를 로봇 실행용 스크립트로 변환
5. 데이터 관리: 플러그인별 데이터 모델의 영속화 및 동기화

기술 스택

| 항목 | 기술 | 비고 |

|-----|-----|-----|

| 런타임 | Node.js | JavaScript (순수 JS, TypeScript 없음) |

| 메시징 | MQTT (v2.1.3) | rodi-broker-svc를 통한 Pub/Sub |

| 로깅 | Winston (v2.4.4) | 파일/콘솔/데이터 3채널 |

| 프레임워크 | Express (v4.14.0) | 웹 서버 (제한적 사용) |

| DOM | JSDOM (v12.0.0) | 플러그인 HTML 렌더링 |

| 아카이브 | asar/asar-fs | Electron 아카이브 포맷 |

| 유틸리티 | lodash, underscore, q | 레거시 패턴 |

| IPC | node-ipc (v9.1.1) | 프로세스 간 통신 |

> 참고: 의존성 버전이 전반적으로 오래되어 있습니다 (2016~2018년 수준). devDependencies가 없고 테스트 프레임워크도 미설정 상태입니다.

핵심 개념 용어

이 프로젝트를 이해하려면 다음 용어를 먼저 알아야 합니다:

| 용어 | 의미 |

|-----|-----|

| Assembly | 플러그인 패키지 단위. **.asar** 아카이브 또는 JS 디렉토리 |

| Activator | 플러그인 진입점. **start()** 메서드를 가진 모듈 |

| ServiceContext | 플러그인이 서비스를 등록할 때 사용하는 컨텍스트 객체 |

| Contribution | 플러그인이 제공하는 기능 단위 (페이지 핸들러 + 데이터 모델) |

| PageHandler | JSDOM 기반 HTML 페이지 관리자 |

| PresentationComposite | Contribution + PageHandler + DataModel + RodiAPI를 묶은 복합 객체 |

| rodix_api | 전역으로 노출되는 플러그인 SDK (서비스 타입 정의) |

| TGOS | 프로젝트 전체의 루트 경로 및 설정을 관리하는 시스템명 |

| ClinkAPI | 로봇 하드웨어와 통신하는 저수준 API (DLL 기반) |

설정 파일 위치

| 파일 | 경로 (서비스 기준) | 역할 |

|-----|-----|-----|

| `framework.json` | `config/framework.json` | MQTT 연결, 이벤트 토픽, 로깅 설정 |

| `runtime.json` | `config/runtime.json` | Python/Java 런타임 경로 |

| `config.json` | `../../../config.json` (TGOS 루트) | 저장소 경로, 로봇 환경(INTIME/시뮬레이션), 로봇 스펙 |

| `package.json` | `package.json` | npm 의존성, 서비스 메타데이터 |

다음 단계

프로젝트의 역할과 위치를 파악했으면, [2단계: 디렉토리 구조](#)에서 실제 파일 배치를 살펴봅니다.