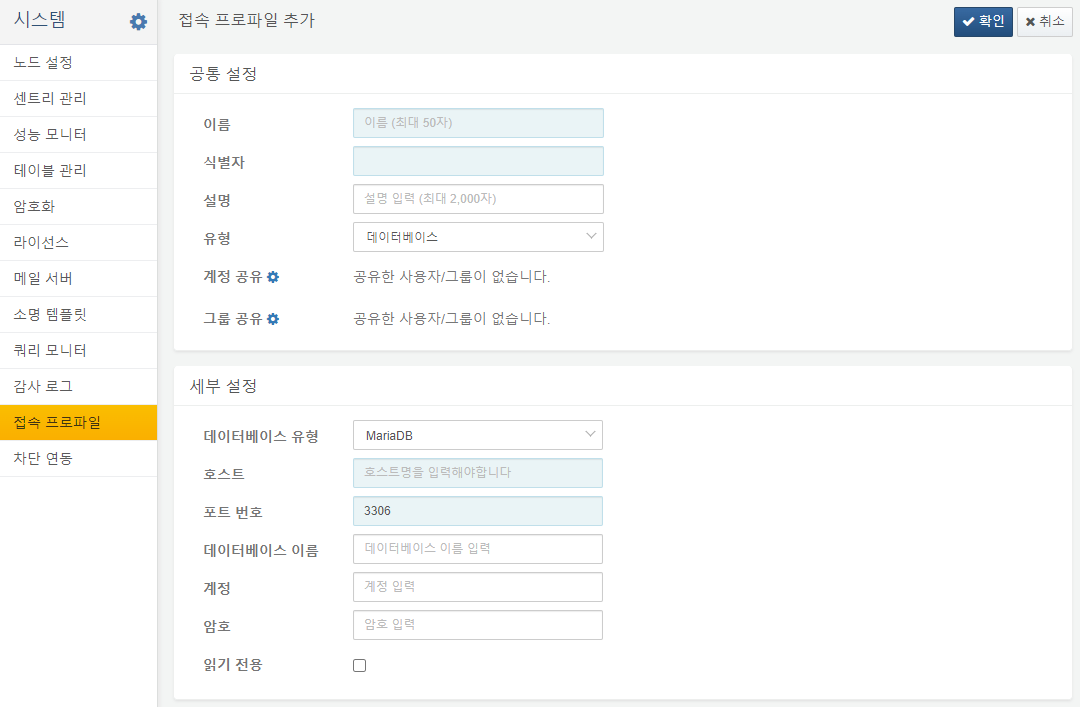
### 접속 프로파일 구현

이 절에서는 [SampleConnectProfileFactory.java](https://github.com/logpresso/logpresso-app-examples/blob/main/logpresso-sample-app/src/main/java/com/logpresso/sonar/sample/impl/SampleConnectProfileFactory.java) 코드를 설명합니다.

#### 접속 프로파일 개요

로그프레소의 쿼리 명령어나 수집기는 외부 시스템과 통신하는 경우가 많습니다. 예를 들어, [dbquery](https://docs.logpresso.com/ko/query/dbquery-command) 명령어는 외부 데이터베이스에서 SQL을 실행한 결과를 조회합니다. 외부 데이터베이스에 접속하려면 JDBC 접속 문자열, 계정, 암호가 필수적입니다. 따라서 dbquery 쿼리 명령어는 접속 프로파일을 PROFILE 매개변수로 전달받습니다. dbquery 쿼리 명령어를 실행하려면 아래와 같이 데이터베이스 접속 프로파일을 미리 설정해두어야 합니다.



한편, 접속 프로파일이 설정되었다고 하더라도 아무나 데이터베이스에 SQL 쿼리를 실행하도록 허용해서는 안 됩니다. 계정 또는 계정 그룹에 접속 프로파일의 사용 권한을 부여해야만 해당 계정에서 접속 프로파일을 이용하여 외부 시스템과 통신할 수 있습니다.

데이터베이스 뿐 아니라 외부 시스템은 대부분 인증을 요구합니다. 앱 예제는 로그프레소 소나의 REST API를 호출하는데, REST API를 호출하려면 Authorization 헤더에 API 키를 전달해야 합니다.

외부 시스템에서 요구하는 인증 방식에 따라 설정 항목이 모두 다르기 때문에, 앱은 전용 ConnectProfileFactory를 구현하여 로그프레소 플랫폼의 기능을 확장해야 합니다.

#### ConnectProfileFactory 플러그인

앱 예제에서는 SampleConnectProfileFactory를 구현하여 로그프레소 플랫폼에 등록합니다.

@Component(name = "sample-connect-profile-factory")@Instantiatepublic class SampleConnectProfileFactory extends AbstractConnectProfileFactory { @Requires private ConnectProfileFactoryRegistry connectProfileFactoryRegistry; @Validate public void start() { connectProfileFactoryRegistry.addFactory(this); } @Invalidate public void stop() { if (connectProfileFactoryRegistry != null) connectProfileFactoryRegistry.removeFactory(this); }}

위의 코드는 AbstractConnectProfileFactory를 상속한 SampleConnectProfileFactory 컴포넌트를 정의합니다. 컴포넌트가 시작할 때 ConnectProfileFactoryRegistry OSGi 서비스에 addFactory()를 호출하여 자기 자신을 등록하고, 컴포넌트가 종료될 때 removeFactory()를 호출하여 자기 자신을 제거합니다.

@Component, @Instantiate, @Requires, @Validate, @Invalidate는 모두 iPOJO 어노테이션입니다. 각 어노테이션에 대한 설명은 아래와 같습니다:

* @Component: 이 클래스가 iPOJO 컴포넌트임을 선언합니다. iPOJO 컴포넌트는 유일한 이름을 지정해야 합니다.
* @Instantiate: iPOJO 컴포넌트 인스턴스를 생성하도록 지시합니다. 만약 @Instantiate를 선언하지 않으면, 앱 번들을 시작하더라도 컴포넌트 인스턴스가 생성되지 않습니다.
* @Requires: 서비스 인터페이스 의존성을 선언합니다. 별도의 할당(assignment) 구문이 없지만, 컴포넌트가 시작될 때 iPOJO 프레임워크가 자동으로 ConnectProfileFactoryRegistry 인터페이스를 제공하는 OSGi 서비스 객체를 주입할 것입니다. 이를 injected POJO(Plain Old Java Object)라고 표현하는데, 여기에서 iPOJO라는 프레임워크 이름이 나오게 되었습니다.
* @Validate: 컴포넌트가 시작될 때 호출될 콜백을 선언합니다. @Requires로 선언된 모든 의존성이 충족되어야만 컴포넌트가 시작됩니다.
* @Invalidate: 컴포넌트가 중지될 때 호출될 콜백을 선언합니다. @Requires로 선언된 의존성 중 하나라도 충족되지 않으면 컴포넌트가 중지됩니다. OSGi 번들을 중지하거나 삭제하면 서비스가 제거되면서 이 서비스에 의존하던 컴포넌트들도 중지됩니다.

#### ConnectProfileFactory 구현

**식별자**

getType()은 ConnectProfileFactoryRegistry에 등록된 각 ConnectProfileFactory 객체를 구분하는데 필요한 식별자를 반환합니다. 일반적으로 알파벳 소문자와 하이픈을 사용하여 정의합니다.

public String getType() { return "sample";}

**표시 이름**

getDisplayName()은 사용자 인터페이스에 표시할 접속 프로파일 유형의 이름을 반환합니다. 로케일에 따라 적합한 문자열을 반환해야 합니다.

public String getDisplayName(Locale locale) { if (locale != null && locale.equals(Locale.KOREAN)) return "샘플"; return "Sample";}

**설명**

getDescription()은 사용자 인터페이스에 표시할 접속 프로파일 유형에 대한 설명을 반환합니다. 로케일에 따라 적합한 문자열을 반환해야 합니다.

public String getDescription(Locale locale) { if (locale != null && locale.equals(Locale.KOREAN)) return "샘플 접속 설정을 관리합니다."; return "Manage connection properties of sample app.";}

**보호 설정 항목 목록**

getProtectedConfigKeys()는 사용자 인터페이스에 표시해서는 안 되는 설정 항목의 식별자 목록을 반환합니다. 예를 들어, 암호나 API 키는 관리자라 하더라도 화면에서 조회할 수 없어야 합니다. 보호되는 설정 항목은 재설정만 가능하고 조회가 불가능하며, 로그프레소 플랫폼 내에서도 암호화되어 보관됩니다.

public Set<String> getProtectedConfigKeys() { return Set.of("api\_key");}

**설정 명세 목록**

getConfigOptions()은 접속 프로파일의 설정 양식(Form)을 정의합니다. 접속 프로파일의 설정 명세인데 LoggerConfigOption 이름의 타입을 사용하는 것이 다소 어색하겠지만, 수집기의 설정 양식을 정의할 때에도 동일한 타입을 사용하여 동적으로 설정 양식을 정의합니다.

public List<LoggerConfigOption> getConfigOptions() { LoggerConfigOption endpoint = new MutableStringConfigType("endpoint", t("Endpoint", "엔드포인트"), t("REST API URL", "REST API 주소"), true); LoggerConfigOption apiKey = new MutableStringConfigType("api\_key", t("API key", "API 키"), t("GUID format", "GUID 형식"), true); return Arrays.asList(endpoint, apiKey);}

아래 3개 타입을 사용할 수 있습니다:

* MutableStringConfigType: 변경 가능한 문자열 설정 항목
* MutableIntegerConfigType: 변경 가능한 32비트 정수 설정 항목
* MutableBooleanConfigType: 변경 가능한 불린 설정 항목

리스트로 반환한 설정 항목의 순서대로 화면에 표시됩니다.

**설정 요약**

접속 프로파일 목록에서 빠르게 설정 세부사항을 확인할 수 있도록 핵심적인 설정만 요약하여 반환합니다. 로케일에 따라 적절한 문자열을 반환해야 합니다.

public String describeConfigs(ConnectProfile profile, Locale locale) { String endpoint = profile.getString("endpoint"); if (locale != null && locale.equals(Locale.KOREAN)) return "엔드포인트: " + endpoint; return "Endpoint: " + endpoint;}