### 쿼리 명령어 파서 구현

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

#### 쿼리 명령어 파서 개요

로그프레소의 쿼리는 파이프로 구분된 쿼리 명령어의 조합입니다. 로그프레소 쿼리 엔진은 쿼리 문자열을 파이프 문자로 분할한 다음, 공백으로 구분되는 첫번째 토큰을 쿼리 명령어의 이름으로 인식합니다. 쿼리 엔진은 쿼리 명령어의 이름으로 QueryCommandParser 객체를 검색하여 쿼리 명령어 파싱을 요청합니다.

쿼리 명령어 파서는 QueryCommandParser 인터페이스를 구현합니다:

public interface QueryCommandParser { String getCommandName(); QueryCommand parse(QueryContext context, String commandString); Map<String, QueryErrorMessage> getErrorMessages(); QueryCommandHelp getCommandHelp(); QueryParserService getQueryParserService(); void setQueryParserService(QueryParserService queryParserService);}

각 메소드는 아래의 기능을 구현합니다:

* getCommandName(): 로그프레소 플랫폼에서 유일한 쿼리 명령어 이름을 반환합니다. 쿼리 명령어 이름은 쿼리 창에서 자동으로 문법 강조됩니다.
* parse(): 쿼리 명령어 구문을 파싱하여 QueryCommand 객체를 반환합니다.
* getErrorMessages(): 쿼리 파싱 중에 발생하는 오류 코드에 대한 지역화된 메시지를 제공합니다.
* getCommandHelp(): 쿼리 명령어에 대한 설명, 옵션, 출력 필드 세부사항을 제공합니다. 이 메소드는 기본 구현이 제공됩니다.
* getQueryParserService(): 쿼리 파서 서비스 객체를 반환합니다. 파서에서 서브 쿼리 등 재귀적으로 파싱을 요청해야 하는 경우에 사용합니다.
* setQueryParserService(): 쿼리 파서 서비스 객체를 설정합니다. 단위 테스트 외에는 이 메소드를 직접 호출하면 안 됩니다.

#### 접속 프로파일 쿼리 명령어 파서

앱에서 확장하는 쿼리 명령어는 대부분 외부 시스템과 상호작용하게 되므로 접속 프로파일을 사용합니다. 접속 프로파일을 사용하는 쿼리 명령어 파서는 ConnectProfileQueryCommandParser 클래스를 상속하여 간단하게 구현할 수 있습니다. SampleQueryCommandParser는 앱에서 구현하는 여러 개의 쿼리 명령어 파서가 상속할 기본 구현을 제공합니다.

public abstract class SampleQueryCommandParser extends ConnectProfileQueryCommandParser { protected static final String ERR\_SERVICE\_UNAVAILABLE = "204000"; protected static final String ERR\_PROFILE\_REQUIRED = "204001"; protected static final String ERR\_NAME\_REQUIRED = "204002"; public SampleQueryCommandParser() { super("sample", ERR\_SERVICE\_UNAVAILABLE, ERR\_PROFILE\_REQUIRED); }}

생성자의 매개변수는 아래와 같습니다:

* profileType: 쿼리 명령어가 사용하는 접속 프로파일 유형의 식별자입니다. 즉, ConnectProfileFactory의 getType() 반환 값과 일치해야 합니다.
* emptyProfileErrorCode: 현재 세션에 허용된 접속 프로파일이 하나도 없는 경우 발생시킬 오류 코드입니다.
* missingProfileErrorCode: 쿼리 명령어의 profile 매개변수가 필수인데 profile 값이 지정되지 않은 경우에 발생시킬 오류 코드입니다.

protected List<String> getSupportedOptions() { return new ArrayList<String>(getCommandHelp().getOptions().keySet());}

쿼리 명령어는 COMMAND-NAME KEY1=VALUE1 KEY2=VALUE2의 형태로 정의됩니다. getSupportedOptions()은 사용 가능한 매개변수의 목록을 반환합니다. 이 코드는 도움말에 정의된 모든 매개변수를 반환하도록 구현되어 있습니다.

protected abstract QueryCommand parse(QueryContext context, SampleParams params);

SampleQueryCommandParser를 상속하는 쿼리 명령어 파서는 이미 파싱되어 SampleParams로 전달되는 매개변수를 이용할 수 있습니다. 각 쿼리 명령어 파서는 필수 매개변수 검사 등 추가적인 검증을 수행합니다.

protected ConnectProfileParams parseParams(QueryContext context, Map<String, String> opts) { SampleParams params = new SampleParams(); params.setName(opts.get("name")); return params;}protected QueryCommand parse(QueryContext context, ConnectProfileParams params, String commandString) { return parse(context, (SampleParams) params);}

* parseParams(): 모든 확장 쿼리 명령어의 매개변수를 일괄적으로 파싱합니다. 개별 쿼리 명령어마다 매개변수를 파싱할 수도 있지만, 한 지점에 매개변수 파싱을 모아두면 공통 매개변수 파싱이 더 수월합니다.
* parse(): ConnectProfileQueryCommandParser 클래스는 쿼리 명령어 구문에서 키=값 형식의 매개변수 패턴을 파싱하여 맵을 구성한 후 parseParams()를 호출하고, 그 결과로 반환되는 ConnectProfileParams 객체를 parse() 메소드에 다시 전달합니다. 이미 파싱된 쿼리 명령어의 매개변수를 활용하기만 하면 됩니다.

public Map<String, QueryErrorMessage> getErrorMessages() { Map<String, QueryErrorMessage> errors = new HashMap<>(); errors.put(ERR\_SERVICE\_UNAVAILABLE, newMsg("No available sample profile found.", "사용 가능한 샘플 프로파일이 없습니다.")); errors.put(ERR\_PROFILE\_REQUIRED, newMsg("Specify valid sample profile.", "샘플 프로파일 이름을 입력해주세요.")); errors.put(ERR\_NAME\_REQUIRED, newMsg("Specify name option in the sample-create-subnet-group command.", "sample-create-subnet-group 명령어에 name 옵션을 지정하세요.")); return errors;}

getErrorMessage()는 에러 코드와 지역화된 메시지의 쌍을 반환합니다. 쿼리 명령어 파서는 QueryParseException 예외 객체를 던질 때 에러 코드를 지정합니다. 로그프레소 쿼리 엔진은 에러 코드를 다시 지역화된 메시지로 변환하는 과정에서 이 메소드의 반환 값을 참조합니다.

#### 쿼리 명령어 파서 구현

SampleSubnetGroupsCommandParser 클래스는 위에서 정의한 SampleQueryCommandParser 클래스를 상속합니다.

public class SampleSubnetGroupsCommandParser extends SampleQueryCommandParser { public String getCommandName() { return "sample-subnet-groups"; } protected QueryCommand parse(QueryContext context, SampleParams params) { return new SampleSubnetGroupsCommand(params); }}

이 파서는 sample-subnet-groups라는 이름의 쿼리 명령어를 정의하는데, 별도의 매개변수가 없으므로 parse()는 별다른 과정 없이 바로 SampleSubnetGroupsCommand 객체를 생성하여 반환하게 됩니다.

이 파서에서 주목할 부분은 생성자의 메타데이터입니다.

setDescription(Locale.ENGLISH, "Get subnet groups from the Logpresso server.");setOutput("guid", ValueType.STRING, Locale.ENGLISH, "GUID", "");

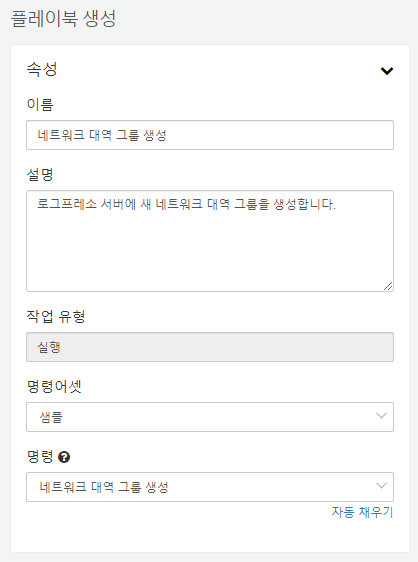
* setDescrpition(): 쿼리 도움말에 표시할 로케일별 명령어 설명을 설정합니다.
* setOutput(): 쿼리 도움말에 표시할 출력 필드의 이름, 타입, 설명을 설정합니다.

SampleCreateSubnetGroupCommandParser 클래스에서는 다른 메타데이터도 확인할 수 있습니다.

setDisplayGroup(Locale.KOREAN, "샘플");setDisplayName(Locale.KOREAN, "네트워크 대역 그룹 생성");setDescription(Locale.KOREAN, "로그프레소 서버에 새 네트워크 대역 그룹을 생성합니다.");setOption("name", REQUIRED, Locale.KOREAN, "이름", "새 네트워크 대역 그룹 이름");

* setDisplayGroup(): 쿼리 명령어 그룹. 플레이북의 명령어셋 항목에 매핑됩니다.
* setDisplayName(): 쿼리 명령어 표시 이름. 플레이북의 명령 항목에 매핑됩니다.
* setOption(): 명령어 매개변수를 정의합니다. Ctrl+Space 입력 시 매개변수 도움말과 자동 완성에 활용됩니다.

예를 들어, 플레이북 디자이너의 실행 작업을 생성할 때 샘플 명령어셋을 선택하면, 아래와 같이 네트워크 대역 그룹 생성 항목을 확인할 수 있습니다.



쿼리 명령어 파싱 중 예외는 아래와 같이 발생시킵니다. create-subnet-group 명령어는 name 매개변수가 필수인데, 이 값을 지정하지 않은 경우 QueryParseException 예외를 발생시켜서 파싱이 실패하도록 합니다.

protected QueryCommand parse(QueryContext context, SampleParams params) { if (params.getName() == null) throw new QueryParseException(ERR\_NAME\_REQUIRED, -1); return new SampleCreateSubnetGroupCommand(params);}

다음 절에서는 이렇게 구현된 쿼리 명령어 파서를 등록하는 부분을 알아보겠습니다.