2014.12.11

1. 설치
   1. rJava

install.packages(“rJava”)

* 1. RLogpresso

install.packages(<pathtopackage>, repos = NULL, type="source")

R CMD install <pathtopackage>

1. API
   1. RLogpresso.version()

JVM, 아라크네 로그DB 클라이언트, RLogpresso 라이브러리의 버전을 반환합니다.

**Return**

JVM: Java Virtual Machine의 버전

araqne\_logdb\_client: 의존하는 아라크네 로그DB 클라이언트 버전

RLogpresso: RLogpresso 라이브러리의 버전

* 1. RLogpresso.create(output=TRUE)

데이터베이스 조회에 필요한 R 객체를 반환합니다.

반환된 객체를 이용하여 2.c~2.q의 함수를 이용하여 데이터베이스를 조회합니다.

**Argument**

output: 일부 함수에서 사용자의 편의를 위한 메세지의 출력 여부

**Return**

데이터베이스와의 통신을 위한 객체

* 1. $connect(**host**, port=8888, **loginName**, password=””, connectTimeout=0, readTimeout=10000)

데이터베이스에 접속합니다.

**Argument**

host: 도메인 혹은 IP 주소

port: 포트 번호

loginName: 데이터베이스의 계정

password: 로그인 계정의 암호

connectTimeout: 접속 타임아웃 (밀리초)

readTimeout: 읽기 타임아웃 (밀리초)

* 1. $disconnect()

데이터베이스의 접속을 종료합니다.

* 1. $query(**query**, interval=1)

쿼리를 요청하여 결과값을 얻습니다.

**Argument**

query: 요청할 쿼리 문자열

interval: 쿼리 종료 여부 확인 간격 (초)

**Return**

요청한 쿼리의 결과값

* 1. $createQuery(**query**)

쿼리를 생성합니다. 생성 직후엔 대기 상태이며, $startQuery를 통해 실행하여야 합니다.

**Argument**

query: 요청할 쿼리 문자열

**Return**

생성된 쿼리의 ID

* 1. $startQuery(**id**)

생성되어있는 쿼리를 실행합니다.

**Argument**

id: 쿼리 ID

* 1. $stopQuery(**id**)

실행중인 쿼리를 정지합니다.

**Argument**

id: 쿼리 ID

* 1. $removeQuery(**id**)

쿼리를 삭제합니다.

**Argument**

id: 쿼리 ID

* 1. $fetch(**id**, **offset**, **limit**)

실행중인 쿼리 또는 실행이 완료된 쿼리의 결과값을 요청합니다.

**Argument**

id: 쿼리 ID

offset: 건너 뛸 결과 행 개수

limit: 가져올 최대 행 개수

**Return**

요청한 분량의 쿼리 결과값

* 1. $queries()

현재 생성되어있는 쿼리 목록을 조회합니다.

**Return**

id: 쿼리 ID

query\_string: 원본 쿼리 문자열

status: 쿼리 동작 상태 (Waiting, Running, Stopped, Cancelled, Ended)

loaded\_count: 쿼리 결과 적재 건수

is\_background: 백그라운드 실행 여부

start\_time: 쿼리 시작 시각

finish\_time: 쿼리 종료 시각

* 1. $queryStatus(**id**)

쿼리의 상세 상태를 조회합니다.

**Argument**

id: 쿼리 ID

**Return**

id: 쿼리 ID

query\_string: 원본 쿼리 문자열

status: 쿼리 동작 상태 (Waiting, Running, Stopped, Cancelled, Ended)

loaded\_count: 쿼리 결과 적재 건수

is\_background: 백그라운드 실행 여부

start\_time: 쿼리 시작 시각

finish\_time: 쿼리 종료 시각

elapsed: 쿼리 시작 후 경과 시간 (밀리초)

commands: 커맨드별 동작 상태(status)와 데이터 처리 건수(push\_count)

error\_code: 에러 코드

error\_detail: 에러 상세 내용

* 1. $logFileTypes()

사용 가능한 로그 저장소의 파일 타입 목록을 조회합니다.

**Return**

log\_file\_type: 로그 파일 타입 이름

* 1. $createTable(**name**, **logFileType**)

새 테이블을 생성합니다.

**Argument**

name: 생성할 테이블의 이름

logFileType: 생성할 테이블의 파일 타입

* 1. $dropTable(**name**)

테이블을 삭제합니다.

**Argument**

name: 삭제할 테이블의 이름

* 1. $tables()

테이블의 목록을 조회합니다.

**Return**

table\_name: 테이블 이름

* 1. $tableInfo(**name**)

테이블의 상세 정보를 조회합니다.

**Argument**

name: 조회할 테이블의 이름

**Return**

table\_name: 테이블 이름

schema: 테이블 스키마

metadata: 테이블 메타데이터

1. 사용 예

|  |
| --- |
| > RLogpresso.version()  JVM araqne\_logdb\_client RLogpresso  "1.8.0\_25" "1.0.4" "1.0.0"  > client <- RLogpresso.create()  > client$connect("localhost", , "araqne")  **connected to localhost:8888 as araqne**  > client$tables()  table\_name  1 sys\_mem\_logs  2 sys\_cpu\_logs  3 test  4 sys\_table\_trends  5 sys\_disk\_logs  6 sys\_query\_logs  7 sys\_logger\_trends  8 sys\_audit\_logs  9 araqne\_query\_logs  10 sys\_gc\_logs  > client$tableInfo("test")  $table\_name  [1] "test"  $schema  name type length  1 finfo1 char 8  2 finfo2 integer 0  3 finfo3 char 256  $metadata  name value  1 meta4 10101010  2 meta3 qwerqewr  3 meta2 asdfasf  4 meta1 12341234  > data <- client$query("table sys\_mem\_logs")  > summary(data)  heap\_used bf1\_count iv\_count  Min. : 50593920 Min. :0 Min. :0  1st Qu.:106055904 1st Qu.:0 1st Qu.:0  Median :169110912 Median :0 Median :0  **(이하 생략)**  > |

|  |
| --- |
| > attach(RLogpresso.create())  > connect("localhost", , "araqne")  **connected to localhost:8888 as araqne**  > createQuery("table sys\_cpu\_logs | search idle > 50 | limit 10")  **created query 4**  [1] 4  > queries()  id query\_string status  1 4 table sys\_cpu\_logs | search idle > 50 | limit 10 Waiting  loaded\_count is\_background start\_time finish\_time  1 0 FALSE  > startQuery(4)  **started query 4**  > queries()  id query\_string status  1 4 table sys\_cpu\_logs | search idle > 50 | limit 10 Ended  loaded\_count is\_background start\_time  1 10 FALSE 2014-12-10 18:13:04 2014-12-10  finish\_time  1 18:13:04  > queryStatus(4)  $id  [1] 4  $query\_string  [1] "table sys\_cpu\_logs | search idle > 50 | limit 10"  $status  [1] "Ended"  $loaded\_count  [1] 10  $is\_background  [1] FALSE  $start\_time  [1] "2014-12-10 18:13:04"  $finish\_time  [1] "2014-12-10 18:13:04"  $elapsed  [1] 16  $commands  command status push\_count  1 table sys\_cpu\_logs End 21  2 search (idle > 50) End 19  3 limit 10 End 10  $error\_code  NULL  $error\_detail  NULL  > fetch(4, 3, 5)  \_table idle kernel \_id user \_time  1 sys\_cpu\_logs 69.23 25.64 1170 5.12 Wed Dec 10 18:12:48 JST 2014  2 sys\_cpu\_logs 68.42 23.68 1169 7.89 Wed Dec 10 18:12:43 JST 2014  3 sys\_cpu\_logs 74.35 20.51 1168 5.12 Wed Dec 10 18:12:38 JST 2014  4 sys\_cpu\_logs 72.97 21.62 1166 5.4 Wed Dec 10 18:12:28 JST 2014  5 sys\_cpu\_logs 75.67 18.91 1165 5.4 Wed Dec 10 18:12:23 JST 2014  > |