### hdfs

HDFS를 탐색하거나, 쿼리 결과를 파일로 전송할 수 있습니다.

#### 문법

파일 목록 조회

hdfs PROFILE {ls|lsr} PATH

파일 내용 읽기

hdfs PROFILE cat [format=csv|json|sequence] [limit=INT] [offset=INT] PATH

특정한 레코드의 필드 값을 텍스트, CSV, JSON 형식으로 파일 전송

hdfs PROFILE put [format=csv|json] [limit=INT] [offset=INT] [partition=t] PATH

특정한 레코드의 필드 값을 HDFS 시퀀스 형식으로 파일 전송

hdfs PROFILE put format=sequence [compression\_type=block|record] [fields=FIELD,...] [key\_type=HDFS\_TYPE] [key\_field=KEY\_FIELD] [value\_type=HDFS\_TYPE] [value\_field=VALUE\_FIELD] PATH

필수 매개변수

**PROFILE**

HDFS 연결에 사용할 접속 프로파일

프로파일은 웹 콘솔에서 구성할 수 있습니다. ENT-3.10.2009.0. SNR-3.1.2008.0 배포 버전부터 HDFS 프로파일이 접속 프로파일에 통합되었습니다.

**{ls|lsr}**

ls 명령어는 PATH로 지정된 경로의 파일 목록을 보여줍니다. lsr은 PATH로 지정한 경로의 디렉터리 뿐만 아니라 하위의 모든 디렉터리를 탐색합니다.

**cat**

HDFS 파일 시스템에 있는 텍스트 파일, CSV 파일, JSON 파일, HDFS 시퀀스 파일 내용을 조회합니다. format 옵션으로 지정된 파일 형식에 맞춰 파싱합니다.

* 텍스트 기반 파일일 때, 파일 내용을 line 필드에 출력합니다.
* CSV 형식일 때, 첫 줄을 필드 행으로 인식합니다. 데이터에 개행 문자가 있더라도 CSV 규칙에 맞으면 여러 개의 줄을 하나의 필드 값으로 인식합니다.
* JSON 형식일 때, 파일을 행 단위로 읽어서 파싱합니다.
* HDFS 시퀀스 형식일 때, HDFS의 Writable 구현을 로그프레소 타입(Java의 데이터 타입)으로 변환하여 레코드 단위로 읽어옵니다. 변환 형식은 로그프레소의 HDFS 데이터 변환 타입을 참조하십시오.
* 키 필드 이름은 key로 지정됩니다. 키는 원본 타입에 관계없이 문자열로 변환합니다.
* 값 필드 이름은 값의 타입에 따라 다른 필드를 사용합니다.
* MapWritable 타입일 때: 내부의 키-값 매핑이 반환되는 행의 필드로 반환됩니다. 하둡에 내장된 Writable 구현을 로그프레소 타입으로 변환해 출력합니다.
* MapWritable 타입이 아닐 때: value 필드에 출력합니다.

**put**

fields 옵션으로 지정된 필드의 값들을 HDFS 파일 시스템에 파일로 전송합니다. 파일은 PATH로 지정된 경로에 생성됩니다.

* 텍스트 형식일 때 fields 옵션으로 지정한 필드 값을 전송합니다. 필드를 지정하지 않으면 line 필드의 값을 전송합니다.
* CSV 형식일 때 첫번째 줄은 fields 옵션을 필드 이름 목록을 기록합니다. 필드에 값이 없으면 빈 문자열로 표시합니다. fields 옵션을 지정하지 않으면 line 필드의 값을 전송합니다.
* JSON 형식일 때, fields 옵션으로 지정한 필드 값만 전송합니다. fields 옵션을 생략하면 모든 필드 값을 전송합니다.
* HDFS 시퀀스 형식일 때, 다음과 같은 경우가 아니라면 시퀀스 형식으로 전송합니다. 변환 형식은 로그프레소의 HDFS 데이터 변환 타입을 참조하십시오.
* 전송할 레코드의 키나 값 중에 하나라도 비어 있는 행은 전송하지 않습니다.
* 값과 타입이 일치하지 않는 경우,
* string 타입은 문자열로 변환합니다.
* int, long, float, double과 같은 숫자 타입은 0으로 변환합니다.
* bool 타입은 false로 변환합니다.
* 정밀도를 손상하지 않고 변환할 수 있으면 변환 후 출력합니다 (예: long 타입이 지정되었지만 int 값이 들어오면 long으로 변환 후 출력).

**PATH**

파일의 절대 경로

선택 매개변수

**compression\_type=block|record**

압축 형식. 이 옵션이 없으면 압축하지 않습니다. record는 레코드 단위 압축, block은 블록 단위 압축을 의미합니다.

**format=csv|json|sequence**

조회하거나 전송할 파일 형식. 텍스트 파일을 조회하거나 전송할 때에는 이 옵션을 생략합니다.

**key\_type=HDFS\_TYPE**

로그프레소의 HDFS 데이터 변환 타입에서 HDFS 타입에 정의된 타입을 지정합니다.

**key\_field=KEY\_FIELD**

키 필드 이름. 설정하지 않으면 1부터 시작하는 LongWritable 카운터를 사용합니다.

**limit=INT**

파일을 읽어올 때 출력할 행의 개수(기본값: 무제한).

**offset=INT**

파일을 읽어올 때 건너뛸 행 개수(기본값: 0).

**partition=t**

t로 설정하면 매크로를 이용해 디렉터리 경로를 지정할 수 있습니다. 매크로를 이용하면 디렉터리나 파일 이름을 시간에 따라 변경합니다.

**fields=FIELD,...**

put 명령으로 HDFS로 전송할 필드 목록. 구분자로 쉼표(,)를 사용합니다.

**value\_type=HDFS\_TYPE**

로그프레소의 HDFS 데이터 변환 타입에서 HDFS 타입에 정의된 타입을 지정합니다.

**value\_field=VALUE\_FIELD**

값 필드 이름. 설정하지 않으면 전체 필드를 하나의 MapWritable로 전송합니다.

#### 설명

cat으로 파일을 읽어들일 때 PATH에 와일드카드(\*)를 사용하면 특정 문자열 패턴을 포함한 모든 파일을 한 번에 조회할 수 있습니다.

put으로 파일을 전송할 때 partition 옵션을 t로 설정하면 매크로를 이용해 시간에 따라 디렉터리 및 파일 경로를 변경하도록 경로를 지정할 수 있습니다. 파티션 옵션을 지정하고 경로에 매크로를 사용하지 않으면 쿼리가 실패합니다.

사용할 수 있는 매크로는 {logtime:FMT}과 {now:FMT}가 있습니다.

매크로는 중괄호 쌍({ })으로 감싸 입력합니다. 입력 예시는 사용 예 7번을 참조하십시오.

로그프레소는 Java 표준 데이터 타입과 IP 주소와 같이 로그프레소에서 정의한 데이터 타입을 사용합니다. HDFS에서 데이터를 가져오거나 전송할 때 HDFS 데이터 타입에 맞춰 변환 작업을 수행합니다. 타입별 데이터 변환은 다음 표를 참조하십시오.

**로그프레소와 HDFS 데이터 변환 타입**

|  |  |  |
| --- | --- | --- |
| 로그프레소 타입 | HDFS 타입 | 설명 |
| string | Text | 문자열 |
| null | NullWritable | 널 (null) |
| bool | BooleanWritable | 불리언 (참/거짓) |
| int | IntWritable, VIntWritable | 4바이트(32비트) 정수 |
| long | LongWritable, VLongWritable | 8바이트(64비트) 정수 |
| float | FloatWritable | 단정도 실수(single precision) |
| double | DoubleWritable | 배정도 실수(double precision) |

#### 사용 예

vm 이름의 프로파일로 접속하여 루트 경로 파일 목록을 조회

hdfs vm ls /

출력하는 필드는 다음과 같습니다.

* type(문자열): 디렉터리인 경우 dir, 파일인 경우 file
* name (문자열): 파일 이름
* path (문자열): 파일의 절대 경로
* replication (정수): 복제본 수, 디렉터리인 경우 0
* file\_size (정수): 파일 크기, 디렉터리인 경우 0
* block\_size (정수): 블럭 크기, 디렉터리인 경우 0
* modified\_at (날짜): 마지막 수정 시각
* permission (문자열): 권한 설정
* owner (문자열): 소유자
* group (문자열): 소유그룹

vm 프로파일로 접속하여 /tmp/LICENSE.txt 파일의 첫 행을 건너뛰고 5개 행을 조회

hdfs vm cat offset=1 limit=5 /tmp/LICENSE.txt

vm 프로파일로 접속하여 /tmp/malware.csv 파일에서 3개 행을 조회

hdfs vm cat format=csv limit=3 /tmp/malware.csv

vm 프로파일로 접속하여 /tmp/iis.json 파일에서 1개 행을 조회

hdfs vm cat format=json limit=1 /tmp/iis.json

vm 프로파일로 접속하여 /tmp/classloading.seq 파일에서 2개 레코드를 조회

hdfs vm cat format=sequence limit=2 /tmp/classloading.seq

JMX 클래스 로딩 로그 중 UnloadedClassCount와 LoadedClassCount만 /tmp/class.txt 경로에 출력

table classloading | hdfs vm put fields=UnloadedClassCount,LoadedClassCount /tmp/class.txt

sys\_cpu\_logs 로그를 /tmp 밑의 날짜별 디렉터리에 출력

table sys\_cpu\_logs | eval line=concat("idle: ", idle, ", kernel: ", kernel, ", user: ", user) | hdfs vm put partition=t /tmp/{logtime:yyyyMMdd}/cpu.txt

JMX 클래스로딩 로그 중 LoadedClassCount, UnloadedClassCount, TotalLoadedClassCount 출력

table classloading | hdfs vm put format=csv fields=LoadedClassCount,UnloadedClassCount,TotalLoadedClassCount /tmp/classloading.csv

JMX 클래스로딩 로그를 JSON 파일로 출력

table classloading | hdfs vm put format=json /tmp/classloading.json

JMX 클래스로딩 로그 전체를 HDFS 시퀀스 파일로 출력

table classloading | hdfs vm put format=sequence /tmp/classloading.seq

JMX 클래스로딩 로그 중 LoadedClassCount 값을 출력

table classloading | hdfs vm put format=sequence value\_type=long value\_field=LoadedClassCount /tmp/classloading.seq