

명세서

API 명세서

1. API 개요

본 API는 사용자 관리 및 데이터 스트리밍 파이프라인 구축을 위한 기능을 제공합니다. 사용자는 회원가입, 로그인을 통해 서비스를 이용할 수 있으며, NiFi를 통해 데이터 스트리밍 파이프라인을 생성, 관리, 실행할 수 있습니다.

2. API 엔드포인트

API 이름	HTTP 메서드	엔드포인트	요청 파라미터	응답	상태 코드
회원가입	POST	/signup	- username : 사용자 이름 (string, 필수) - password : 비밀번호 (string, 필수)	- message : 성공/실패 메시지 (string) - user_id : 생성된 사용자 ID (integer)	201 (성공) 400 (잘못된 청)
로그인	POST	/login	- username : 사용자 이름 (string, 필수) - password : 비밀번호 (string, 필수)	- access_token : JWT access token (string) - refresh_token : JWT refresh token (string)	200 (성공) 401 (인증 실패)
스트리밍 데이터 수집	POST	/streaming/start	- user_id : 사용자 ID (integer, 필수) - topic : Kafka 토픽 이름 (string, 필수)	- message : 성공/실패 메시지 (string) - process_group_id : 생성된 프로세스 그룹 ID (string)	201 (성공) 400 (잘못된 청)
스트리밍 데이터 수집 종료	DELETE	/streaming/stop/{process_group_id}	- process_group_id : 프로세스 그룹 ID (string, 필수)	- message : 성공/실패 메시지 (string)	200 (성공) 404 (프로세스 그룹 없음)
NiFi 접속 여부	GET	/nifi/status		- status : NiFi 상태 (string)	200 (성공)
NiFi 토큰 발행	POST	/nifi/token		- token : NiFi 토큰 (string)	200 (성공)
NiFi 토큰 만료일	GET	/nifi/token/expiry		- expiry : 토큰 만료일 (datetime)	200 (성공)
NiFi client_id 발급	POST	/nifi/client_id		- client_id : NiFi client_id (string)	200 (성공)
전체 유저 프로세스 그룹 조회	GET	/nifi/process-groups	- user_id : 사용자 ID (integer, 필수)	- process_groups : 프로세스 그룹 목록 (array)	200 (성공)
유저 프로세스 그룹 삭제	DELETE	/nifi/process-groups/{process_group_id}	- process_group_id : 프로세스 그룹 ID (string, 필수)	- message : 성공/실패 메시지 (string)	200 (성공) 404 (프로세스 그룹 없음)
유저 프로세스 그룹 생성	POST	/nifi/process-groups	- user_id : 사용자 ID (integer, 필수) - name : 프로세스 그룹 이름 (string, 필수)	- message : 성공/실패 메시지 (string) - process_group_id : 생성된 프로세스 그룹 ID (string)	201 (성공) 400 (잘못된 청)

ConsumeKafka 생성	POST	/nifi/processors/consumekafka	- - <code>process_group_id</code> : 프로세스 그룹 ID (string, 필수) - <code>topic</code> : Kafka 토픽 이름 (string, 필수)	- <code>message</code> : 성공/실패 메시지 (string) - <code>processor_id</code> : 생성된 프로세서 ID (string)	201 (성공) < 400 (잘못된 청)
PutMongo 생성	POST	/nifi/processors/putmongo	- - <code>process_group_id</code> : 프로세스 그룹 ID (string, 필수) - <code>mongodb_url</code> : MongoDB 연결 URL (string, 필수) - <code>database</code> : MongoDB 데이터베이스 이름 (string, 필수) - <code>collection</code> : MongoDB 컬렉션 이름 (string, 필수)	- <code>message</code> : 성공/실패 메시지 (string) - <code>processor_id</code> : 생성된 프로세서 ID (string)	201 (성공) < 400 (잘못된 청)
PutMongo 수정	PUT	/nifi/processors/putmongo/{processor_id}	- <code>processor_id</code> : 프로세서 ID (string, 필수) - <code>mongodb_url</code> : MongoDB 연결 URL (string, 선택) - <code>database</code> : MongoDB 데이터베이스 이름 (string, 선택) - <code>collection</code> : MongoDB 컬렉션 이름 (string, 선택)	- <code>message</code> : 성공/실패 메시지 (string)	200 (성공) < 404 (프로세.음)
유저 프로세스 그룹 안의 프로세서 목록	GET	/nifi/process-groups/{process_group_id}/processors	- - <code>process_group_id</code> : 프로세스 그룹 ID (string, 필수)	- <code>processors</code> : 프로세서 목록 (array)	200 (성공)
프로세서들 관계 연결	POST	/nifi/connections	- - <code>process_group_id</code> : 프로세스 그룹 ID (string, 필수) - <code>source_id</code> : 연결 소스 프로세서 ID (string, 필수) - <code>destination_id</code> : 연결 대상 프로세서 ID (string, 필수)	- <code>message</code> : 성공/실패 메시지 (string) - <code>connection_id</code> : 생성된 연결 ID (string)	201 (성공) < 400 (잘못된 청)
프로세서 실행	PUT	/nifi/processors/{processor_id}/run-status	- <code>processor_id</code> : 프로세서 ID (string, 필수) - <code>status</code> : 실행 상태 (string, 필수, "RUNNING" 또는 "STOPPED")	- <code>message</code> : 성공/실패 메시지 (string)	200 (성공) < 404 (프로세.음)
유저 프로세서 그룹 안 전체 관계 조회	GET	/nifi/process-groups/{process_group_id}/connections	- - <code>process_group_id</code> : 프로세스 그룹 ID (string, 필수)	- <code>connections</code> : 연결 목록 (array)	200 (성공)
KafkaStreaming 생성	POST	/nifi/processors/kafkastreaning	- - <code>process_group_id</code> : 프로세스 그룹 ID (string, 필수) 	- <code>message</code> : 성공/실패 메시지 (string) - <code>processor_id</code> : 생	201 (성공) < 400 (잘못된 청)

			- topic : Kafka 토픽 이름 (string, 필수)	성된 프로세서 ID (string)	
DAG 생성	POST	/dags	- user_id : 사용자 ID (integer, 필수) - name : DAG 이름 (string, 필수) - description : DAG 설명 (string, 선택)	- message : 성공/실패 메시지 (string) - dag_id : 생성된 DAG ID (string)	201 (성공) 400 (잘못된 청)
DAG 삭제	DELETE	/dags/{dag_id}	- dag_id : DAG ID (string, 필수)	- message : 성공/실패 메시지 (string)	200 (성공) 404 (DAG 없음)
DAG 조회	GET	/dags/{dag_id}	- dag_id : DAG ID (string, 필수)	- dag : DAG 정보 (object)	200 (성공) 404 (DAG 없음)
DAG 상태 변경	PUT	/dags/{dag_id}/status	- dag_id : DAG ID (string, 필수) - status : DAG 상태 (string, 필수, "RUNNING" 또는 "PAUSED")	- message : 성공/실패 메시지 (string)	200 (성공) 404 (DAG 없음)

3. 데이터 모델

사용자:

```
{
  "id": 1,
  "username": "user1",
  "password": "hashed_password"
}
```

프로세스 그룹:

```
{
  "id": "process_group_id",
  "name": "process_group_name",
  "user_id": 1
}
```

프로세서:

```
{
  "id": "processor_id",
  "name": "processor_name",
  "type": "ConsumeKafka",
  "properties": {
    "topic": "topic_name"
  },
  "process_group_id": "process_group_id"
}
```

연결:

```
{
  "id": "connection_id",
  "source_id": "source_processor_id",
  "destination_id": "destination_processor_id",
}
```

```
"process_group_id": "process_group_id"
}
```

DAG:

```
{
  "id": "dag_id",
  "name": "dag_name",
  "description": "dag_description",
  "status": "RUNNING",
  "user_id": 1
}
```

4. 인증

API는 JWT (JSON Web Token) 기반 인증을 사용합니다. 로그인 API를 통해 발급받은 access token을 Authorization 헤더에 Bearer 토큰 형태로 포함하여 API 요청을 보내야 합니다.

5. 오류 처리

API는 표준 HTTP 상태 코드를 사용하여 오류를 표시합니다.

- 400 Bad Request: 잘못된 요청 파라미터 또는 요청 형식
- 401 Unauthorized: 인증 실패
- 404 Not Found: 요청된 리소스를 찾을 수 없음
- 500 Internal Server Error: 서버 오류

6. 사용 예제

회원가입 (curl):

```
curl -X POST -H "Content-Type: application/json" -d '{
  "username": "new_user",
  "password": "password"
}' /signup
```

로그인 (curl):

```
curl -X POST -H "Content-Type: application/json" -d '{
  "username": "user1",
  "password": "password"
}' /login
```

프로세스 그룹 생성 (curl):

```
curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer <access_token>"
-d '{
  "user_id": 1,
  "name": "new_process_group"
}' /nifi/process-groups
```

7. 추가 정보

- API는 JSON 형식으로 데이터를 주고받습니다.
- API 요청에는 Content-Type 헤더를 application/json으로 설정해야 합니다.
- API 응답은 UTF-8 인코딩을 사용합니다.