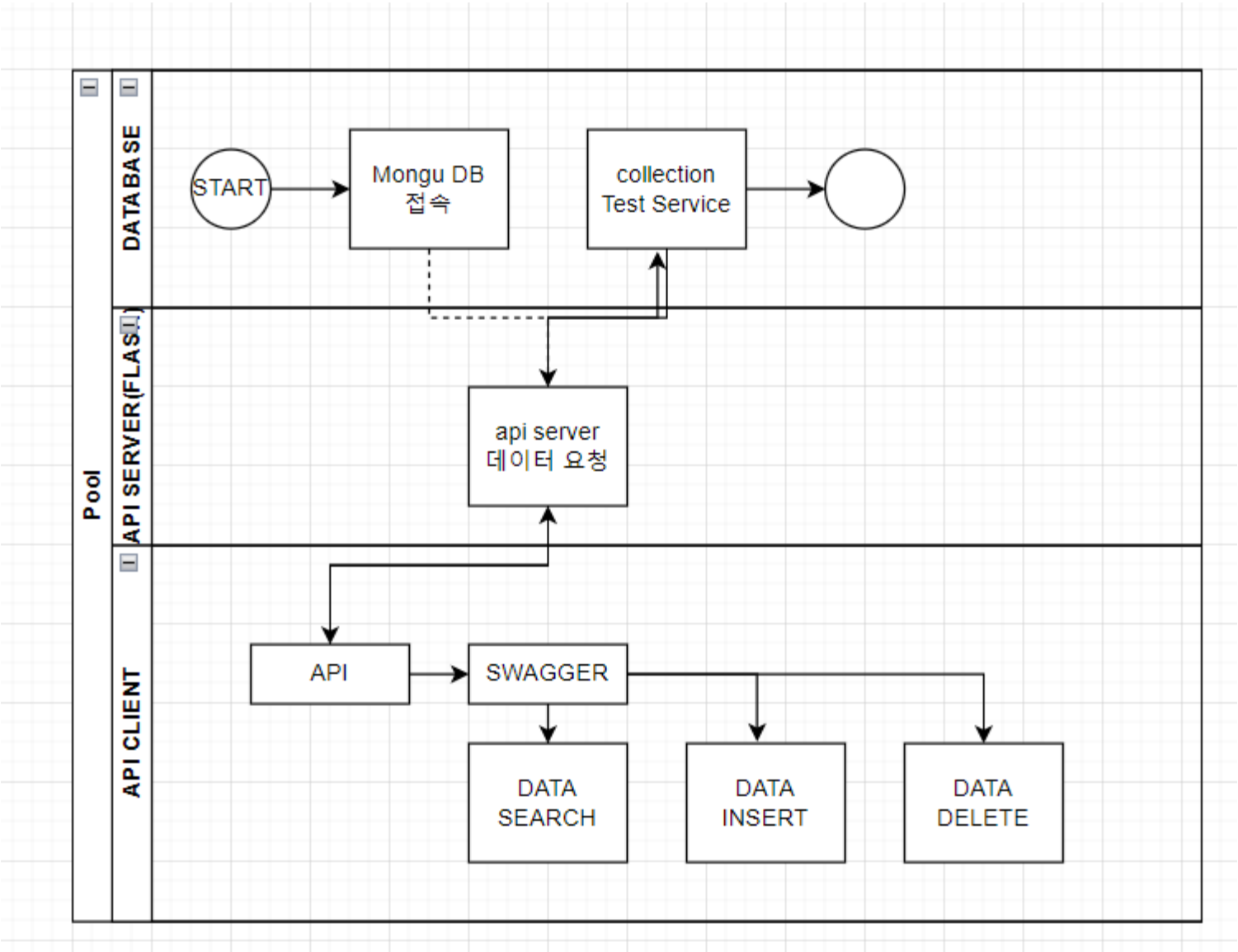


20240909 서비스 구성도
: API TEST



1. DATA SEART

GET

/user/{name} 특정 유저 조회

Parameters

Cancel

Name

Description

name

string

(path)

*

required

이윤서

이름 입력

Execute

Clear

Responses

Response content type

application/json

Curl

```
curl -X 'GET' \
'http://127.0.0.1:5000/user/%EC%9D%B4%EC%9C%A4%EC%84%9C' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:5000/user/%EC%9D%B4%EC%9C%A4%EC%84%9C
```

Server response

Code

Details

결과

200

Response body

```
{
  "age": 100,
  "city": "지구",
  "name": "이윤서"
}
```

Download

Response headers

```
connection: close
content-length: 75
content-type: application/json
date: Mon, 09 Sep 2024 23:35:16 GMT
server: Werkzeug/3.0.4 Python/3.8.19
```

Responses

Code

Description

200

Success

2. DATA INSERT

POST

/user

유저 등록

Parameters

Cancel

Name	Description
<div><div>payload</div><div>object</div><div><div>required</div></div><div><div>(body)</div></div></div>	<div><div>Edit Value</div><div>Model</div></div> <div><div>1) 추가할 데이터 입력</div></div> <div><div>Cancel</div></div> <div><div>Parameter content type</div><div>application/json</div></div> <div><div>2) 실행</div></div> <div><div>Execute</div><div>Clear</div></div>

```
{
  "age": 40,
  "city": "Busan",
  "name": "Lee"
},
{
  "age": 25,
  "city": "NY",
  "name": "James"
},
{
  "age": 30,
  "city": "Soeul",
  "name": "Park"
},
{
  "age": 100,
  "city": "지구",
  "name": "이윤서"
},
{
  "age": 20,
  "city": "아아",
  "name": "아아"
}
```

3. DATA DELETE

```
[
  {
    "age": 40,
    "city": "Busan",
    "name": "Lee"
  },
  {
    "age": 25,
    "city": "NY",
    "name": "James"
  },
  {
    "age": 30,
    "city": "Soeul",
    "name": "Park"
  },
  {
    "age": 100,
    "city": "지구",
    "name": "이윤서"
  },
  {
    "age": 5,
    "city": "대구",
    "name": "아아"
  }
]
```



DELETE /user 유저 삭제

Parameters

Name	Description
name string (query)	The user name <input type="text" value="아아"/>

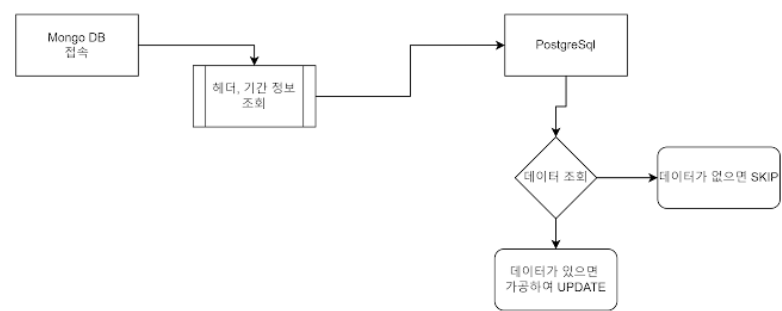
삭제할 데이터 입력

Execute



```
[
  {
    "age": 40,
    "city": "Busan",
    "name": "Lee"
  },
  {
    "age": 25,
    "city": "NY",
    "name": "James"
  },
  {
    "age": 30,
    "city": "Soeul",
    "name": "Park"
  },
  {
    "age": 100,
    "city": "지구",
    "name": "이윤서"
  }
]
```


20240910 서비스 구성도 : ASYNC TEST



DB DATA MERGE

두 개의 DATABASE 가 있습니다.

MONGODB 에는 수록되어야할 컬럼 헤더 정보 PostgreSQL DB 에는 데이터가 있으며,

Mongo DB의 정보를 토대로 PostgreSQL Database 에서 데이터를 조회하여, Mongo Db 의 Document 를 완성 하는 Job

최종 결과물

```
_id: ObjectId('663b33c2c501cd59b6864917')
  rId: ''
  maker: 'ENGEL'
  MECHCD: 'M1005'
  TimeStamp: 2024-05-07T14:59:08.000+00:00
  CURRENT_HOT_RUNNER7 : 0
  CURRENT_SCREW_POS : 126.346
  CURRENT_HOT_RUNNER8 : 0
  CURRENT_HOT_RUNNER9 : 0
  CURRENT_HOT_RUNNER3 : 0
  CURRENT_HOT_RUNNER4 : 0
  CURRENT_HOT_RUNNER5 : 0
  CURRENT_HOT_RUNNER6 : 0
  CURRENT_HOT_RUNNER10 : 0
  CURRENT_HOT_RUNNER11 : 0
  CURRENT_HOT_RUNNER12 : 0
```

1의 데이터에 존재하는 API 항목을 기준으로
Postgresql 에서 조회하여
최종 결과물 형태로 데이터 완성

1. 원본 Mongo DB Data

```
>
  _id: ObjectId("66dfa2027a0da3689607833e")
  rId: "66dfa2027a0da3689607833c"
  maker: "LSM"
  MECHCD: "1001"
  API: Object
    fromDateTime: 2024-09-10T01:32:59.900+00:00
    toDateTime: 2024-09-10T01:33:42.000+00:00
    fsString: "PD2004_CURRENT_BACKPRESS", "PD2004_CURRENT_SCREW_POS", "PD2004_CURRENT_..."
```

2. Postgresql data 형태 , table

id	received time	log barcode	current_backpress	current_inject_press	current_screw_pos	current_screw_gpm	current_screw_gpd	cur
1	1,717,168,200,000	PD0001	34.799	221.762	0	0.242		[NULL]
2	1,717,168,200,194	PD1001	14	30.90	-1	0		[NULL]
3	1,717,168,200,419	PD0004	3	112.5	80	9		[NULL]
4	1,717,168,200,812	PD1005	5.5	67	0	0.5		[NULL]
5	1,717,168,200,824	PD0005	8.2	68.2	33	5.0	0	
6	1,717,168,200,890	PD0118	2.815	12.057	0	0		[NULL]
7	1,717,168,201,189	PD0006	5	54.7	40	4.9		[NULL]
8	1,717,168,201,170	PD0002	36.3	125.6	0	40		[NULL]
9	1,717,168,201,178	PD0006	0.7	79.2	0	0.1	38	
10	1,717,168,201,242	PD0001	13	30.90	0	0.05		[NULL]
11	1,717,168,201,829	PD0004	3	120	80	7.8		[NULL]
12	1,717,168,201,831	PD1005	5.3	67	0	0.1		[NULL]
13	1,717,168,201,930	PD0005	8.1	74.4	34	4.7	0	
14	1,717,168,202,080	PD0118	-0.256	175.344	0	0		[NULL]
15	1,717,168,202,408	PD1001	13	30.90	-1	0.01		[NULL]
16	1,717,168,202,601	PD0006	4.9	68.2	40	8.4		[NULL]
17	1,717,168,202,822	PD0002	74.2	85.2	0	31.4		[NULL]
18	1,717,168,202,642	PD0006	0.8	79.2	0	0	38	
19	1,717,168,203,060	PD0118	-0.269	175.344	0	0.012		[NULL]
20	1,717,168,203,189	PD1005	5.1	67	0	0.7		[NULL]
21	1,717,168,203,167	PD0004	0.2	124.9	16	0.3		[NULL]

1. MongoDB 접속
2. API (추가할 필드 항목) 있는 거 조회
3. PostgreSQL 접속 및 데이터 조회
 1. 데이터 있음 : MongoDB
MongoDB Document Update
 2. 데이터가 없음 :
Pass

파일(F) 편집(E) 탐색(N) 검색(A) SQL 편집기 데이터베이스(D) 윈도우(W) 도움말(H)

Database Navigator x 프로젝트

Enter a part of object name here

- > DBEaver Sample Database (SQLite)
- > INTERONE - 192.168.160.199:15432
 - > Databases
 - > INTERONE
 - > Schemas
 - > public
 - > temp
 - > Event triggers
 - > Extensions
 - > Storage
 - > System Info
 - > Roles
 - > Administer
 - > System Info
 - > postgres - localhost:5432

Properties 엔티티 관계도

INTERONE Databases INTERONE Schemas temp

Name: temp Namespace ID: 16846

Comment: Owner: postgres

	테이블명	Object ID	Owner	Tablespace	Row Count Estimate	Has Row-L
Tables	api_datas	16,847	postgres	pg_default	-1	
Foreign Tables	api_datas_engel	17,152	interx	pg_default	17,510,300	
Views	api_datas_lsm	17,131	interx	pg_default	37,786,144	
Materialized Views	api_job_list	16,854	postgres	pg_default	-1	
Indexes	code_item	16,861	postgres	pg_default	23,237	
Functions	mass_if	16,865	postgres	pg_default	1,050,432	
Sequences	p_detail_mes_auto	16,873	postgres	pg_default	3,228	
Data types	p_inject_base	16,876	postgres	pg_default	-1	
Aggregate functions	p_results	25,327	interx	pg_default	6,112,654	
권한	p_results_auto	16,888	postgres	pg_default	-1	
소스	p_results_bad	16,895	postgres	pg_default	-1	
	tsdata	17,121	interx	pg_default	2,408,449	
	tsdata_lsm	25,346	interx	pg_default	0	

13 items

Project - General x

Name DataSource

 - > Bookmarks
 - > Dashboards
 - > Diagrams
 - > Scripts

Save ... Revert Refresh

테이블명: Object ID:

Tablespace: Owner:

☐ Has Row-Level Security ☐ Partitions Extra Options:

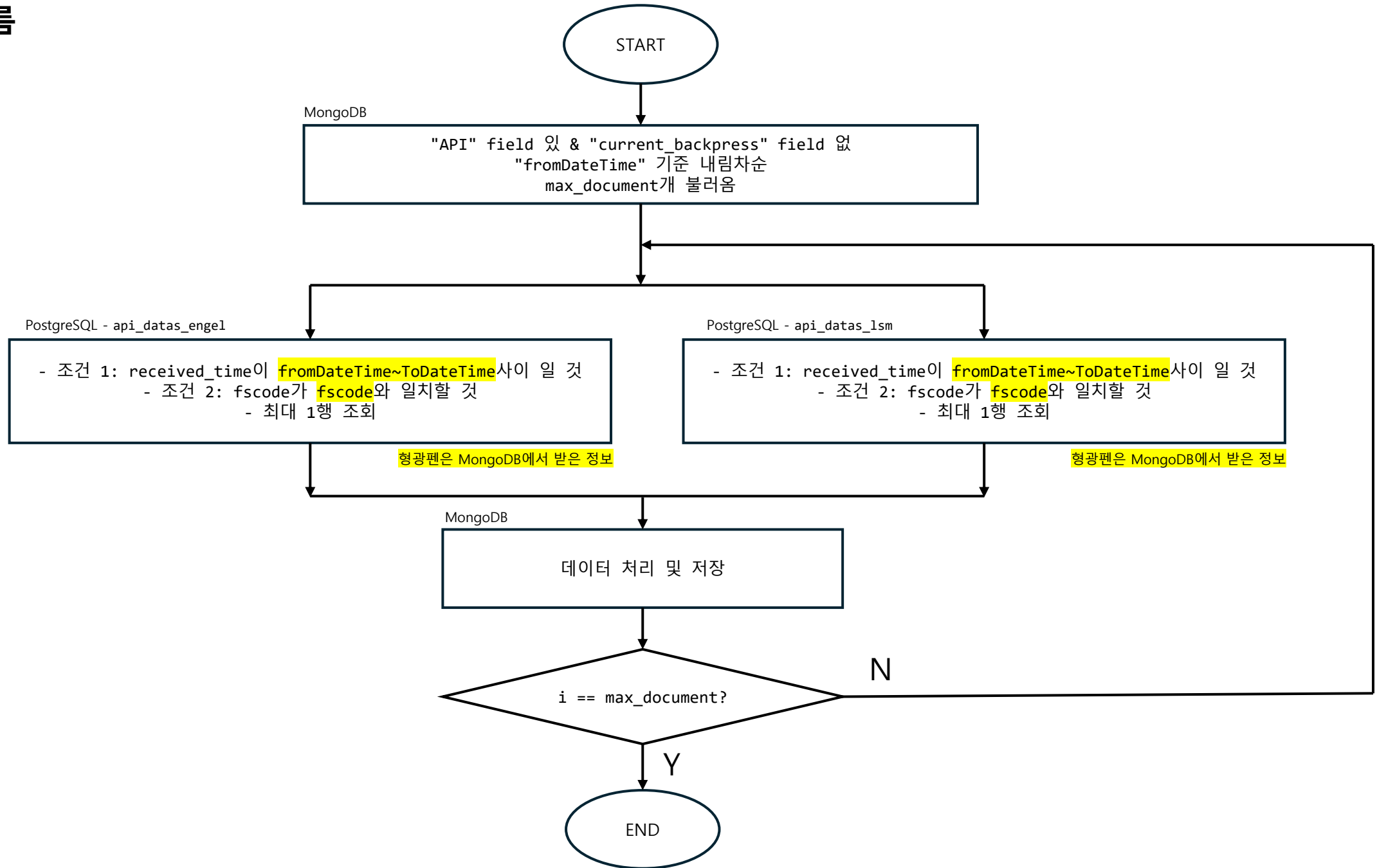
Partition by:

Comment:

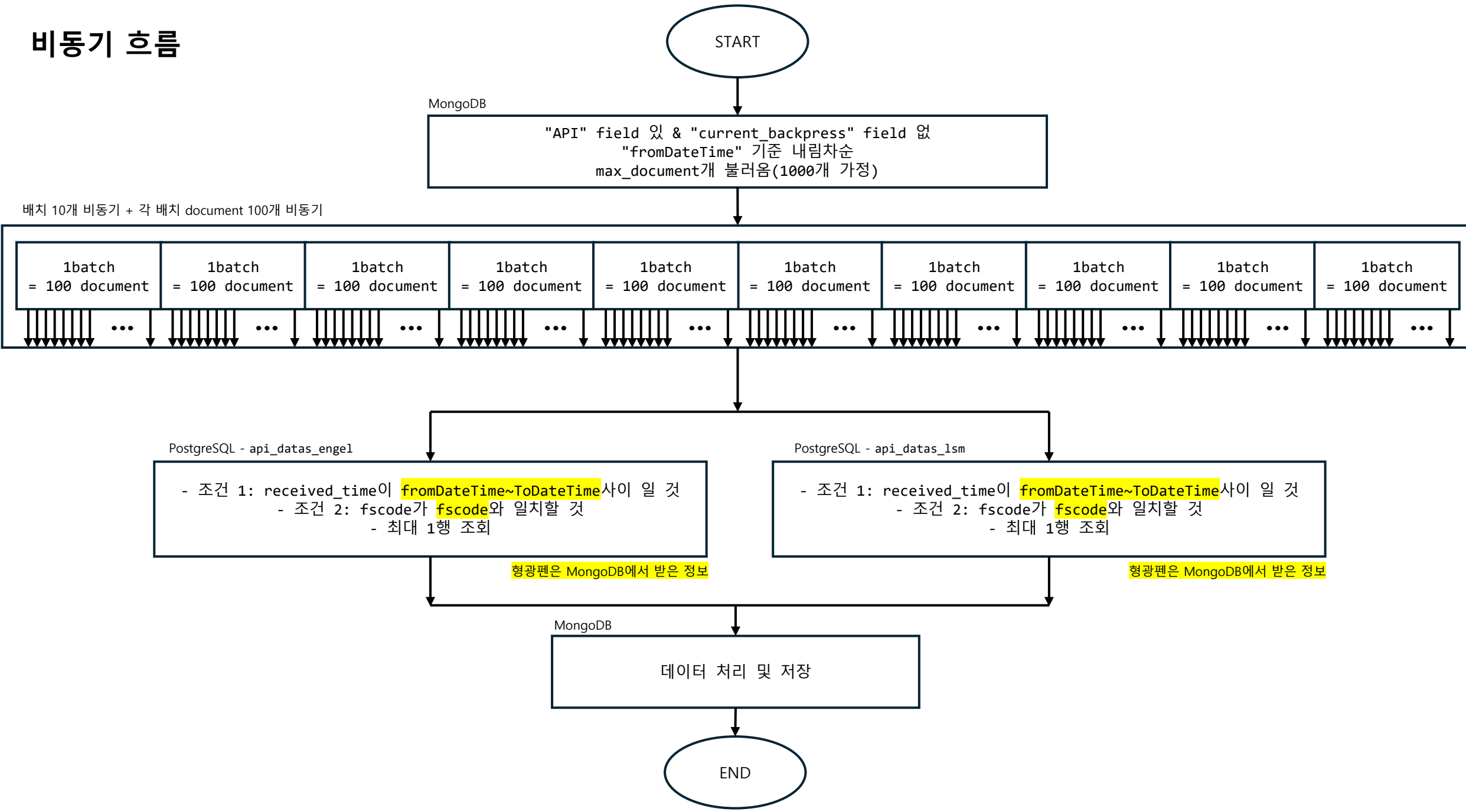
	컬럼명	#	Data type	Identity	Collation	Not Null	Default
Columns	received_time	1	int8			[v]	
Constraints	frcode	2	varchar(10)		default	[v]	
Foreign Keys	current_backpress	3	float4			[]	0
Indexes	current_inject_press	4	float4			[]	0
Dependencies	current_screw_pos	5	float4			[]	0
References	current_screw_rpm	6	float4			[]	0
Partitions	current_screw_spd	7	float4			[]	0
Triggers	current_temp_dryer	8	float4			[]	0
Rules	current_temp_controller_fixed1	9	float4			[]	0
Policies	current_temp_controller_fixed2	10	float4			[]	0
Statistics	current_temp_controller_moving1	11	float4			[]	0
권한	current_temp_controller_moving2	12	float4			[]	0
DDL	current_temp_mold_fixed1	13	float4			[]	0
Virtual	current_temp_mold_fixed2	14	float4			[]	0
	current_temp_mold_moving1	15	float4			[]	0
	current_temp_mold_moving2	16	float4			[]	0
	shot_number	17	int4			[]	0

컬럼명	#	Data type	Identity	Collation	Not Null	Default
received_time	1	int8			[v]	
frcode	2	varchar(10)		default	[v]	
current_backpress	3	float4			[]	0
current_close_press	4	float4			[]	0
current_hot_runner1	5	float4			[]	0
current_hot_runner2	6	float4			[]	0
current_hot_runner3	7	float4			[]	0
current_hot_runner4	8	float4			[]	0
current_hot_runner5	9	float4			[]	0
current_hot_runner6	10	float4			[]	0
current_hot_runner7	11	float4			[]	0
current_hot_runner8	12	float4			[]	0
current_hot_runner9	13	float4			[]	0
current_hot_runner10	14	float4			[]	0
current_hot_runner11	15	float4			[]	0
current_hot_runner12	16	float4			[]	0
current_hot_runner13	17	float4			[]	0
current_hot_runner14	18	float4			[]	0
current_hot_runner15	19	float4			[]	0
current_hot_runner16	20	float4			[]	0
current_hot_runner17	21	float4			[]	0
current_hot_runner18	22	float4			[]	0
current_hot_runner19	23	float4			[]	0
current_hot_runner20	24	float4			[]	0
current_hot_runner21	25	float4			[]	0
current_hot_runner22	26	float4			[]	0
current_hot_runner23	27	float4			[]	0
current_hot_runner24	28	float4			[]	0
current_hot_runner25	29	float4			[]	0
current_hot_runner26	30	float4			[]	0
current_hot_runner27	31	float4			[]	0
current_hot_runner28	32	float4			[]	0
current_hot_runner29	33	float4			[]	0
current_hot_runner30	34	float4			[]	0
current_hot_runner31	35	float4			[]	0
current_hot_runner32	36	float4			[]	0
current_inject_press	37	float4			[]	0

동기 흐름



비동기 흐름



결과

Document 수	동기[s]	비동기/batch X[s]	비동기/batch O[s]
100	3.6	1.310	0.730 (batch_size=10)
1000	20.44	9.332	6.347 (batch_size=100)

GET /cnt

Parameters

Cancel

No parameters

Execute

Clear

Responses

Response content type application/json

Curl

curl -X 'GET' \ 'http://127.0.0.1:5555/cnt' \ -H 'accept: application/json'

Request URL

http://127.0.0.1:5555/cnt

Server response

Code Details

200

Response body

"Data Count": 494

Download

GET /cnt

Parameters

Cancel

No parameters

Execute

Clear

Responses

Response content type application/json

Curl

curl -X 'GET' \ 'http://127.0.0.1:5555/cnt' \ -H 'accept: application/json'

Request URL

http://127.0.0.1:5555/cnt

Server response

Code Details

200

Response body

"Data Count": 2246

Download

API를 통해 실시간 데이터 처리 현황 파악 가능