

1. SDK 연동 가이드

[연동 가이드 문서]

[ThinkingEngine 데이터 연동 가이드 문서](#)

[연동 가이드 영상]



Ⓜ 유니티 데이터 연동 가이드 영상.mp4

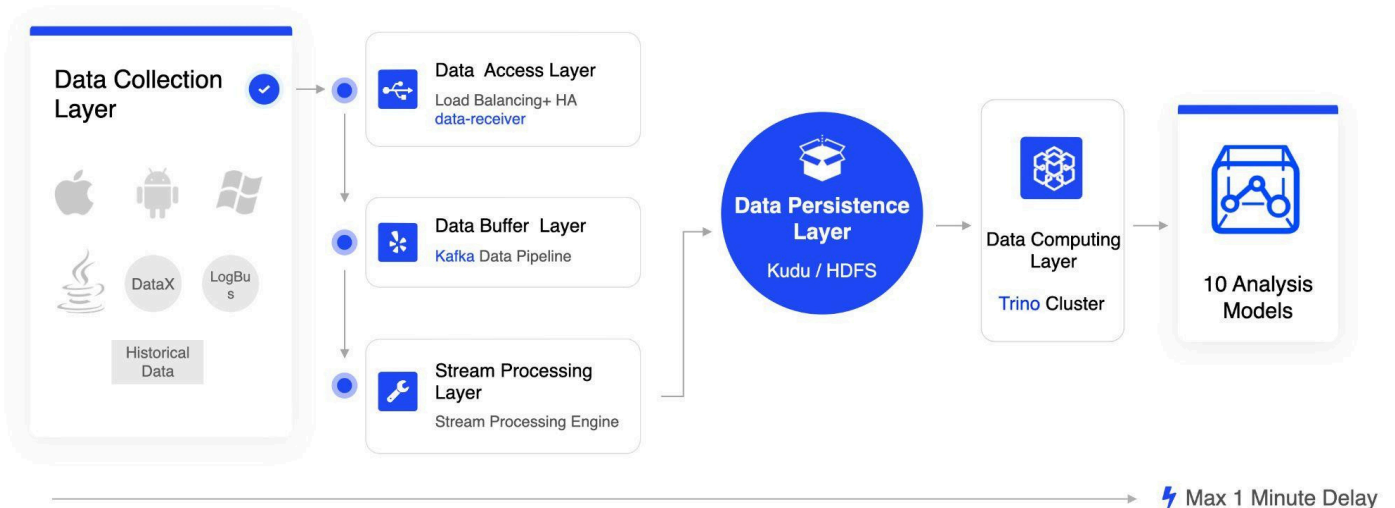
[트래킹 정책 템플릿]

Ⓜ TE 트래킹 정책 템플릿 (게임 - 방치형)

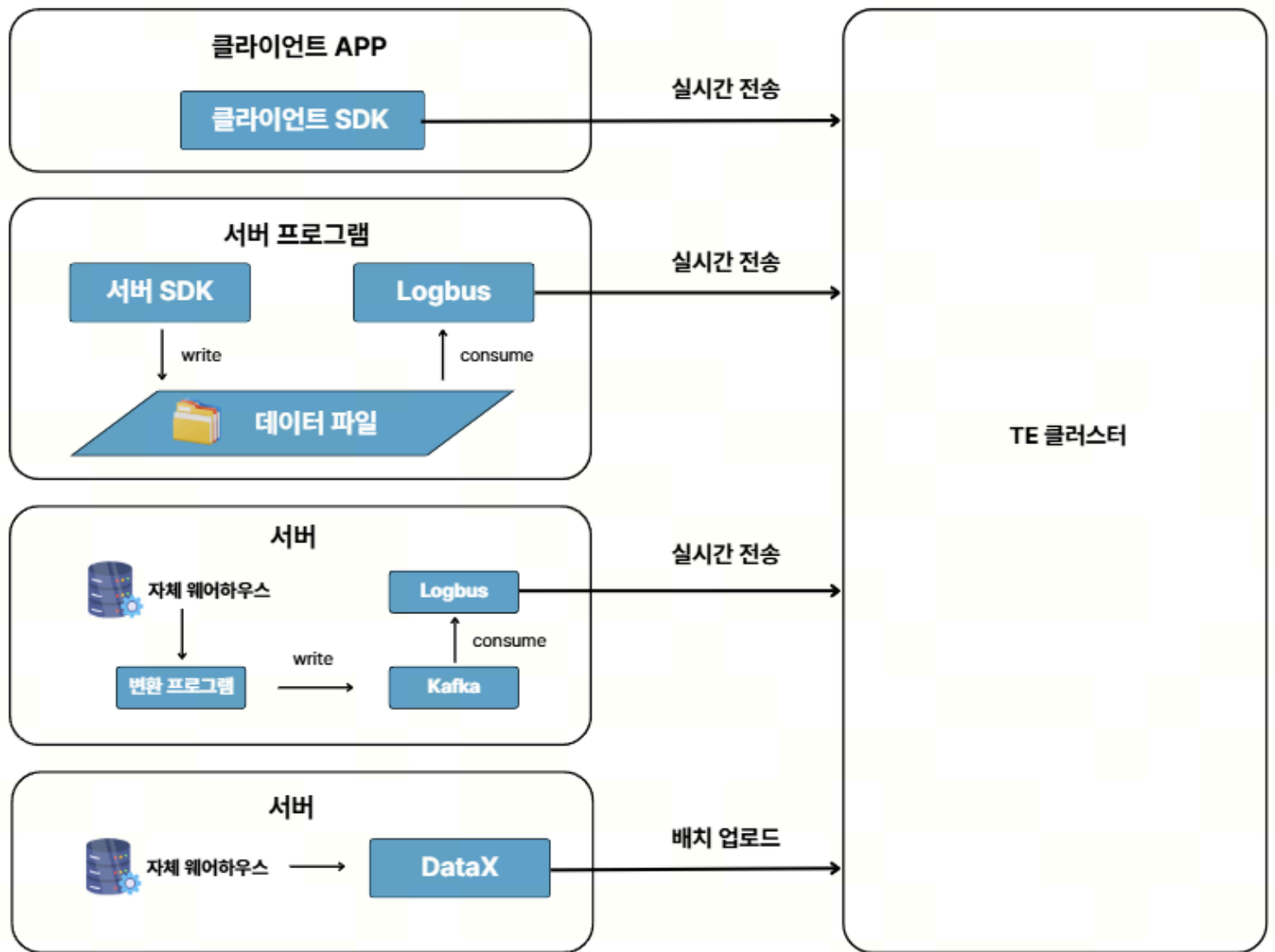
Ⓜ TE 트래킹 정책 템플릿 (웹 - 일반)

데이터 흐름 (DataFlow)

Technical Architecture



SDK 연동 구조



1. 클라이언트 SDK

지원 환경:

- JavaScript, Android, iOS, React Native, Flutter, Unity, Cocos Creator 등

주요 역할:

- 앱(웹/모바일/게임)에서 사용자의 행동(예: 클릭, 페이지 이동, 구매 등)과 디바이스 정보를 자동 또는 수동으로 수집합니다.
- 수집된 데이터는 Thinking Engine(TE) 수집 서버로 실시간 전송됩니다.

장점:

- 앱에 SDK만 설치하면 별도의 서버 개발 없이 손쉽게 데이터 수집이 가능합니다.
- 다양한 플랫폼(웹, 모바일, 게임 등)에서 바로 사용할 수 있어 연동이 간편합니다.
- 자동 추적 기능(예: 페이지 진입/이탈 등)도 지원되어 개발 부담을 줄여줍니다.

예시:

- 웹에서는 JavaScript SDK, 모바일 앱은 Android/iOS SDK, 게임은 Unity/Cocos Creator SDK를 사용하면 됩니다.

2. 서버 SDK + LogBus2 (데이터 수집 도구)

지원 환경:

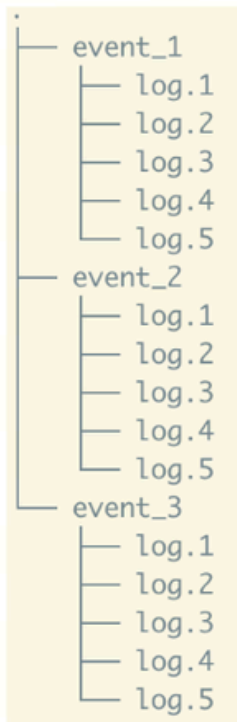
- Java, Python, Node.js 등 서버 사이드 언어

서버 SDK 역할:

- 서버에서 발생하는 이벤트(예: 결제, 인증, 실시간 유저 수, 서버 연동 등)나 클라이언트에서 수집이 어려운 추가 유저 정보를 수집합니다.
- 수집된 데이터를 TE 수집 서버로 전송합니다.

LogBus2 역할:

- 대량의 과거 데이터, 배치 데이터, 로그 파일 등 기존 형태의 데이터를 전송하기에 최적화된 도구입니다.
- 파일 또는 Kafka 등에서 데이터를 읽어 TE 서버로 업로드합니다.
- 실시간 모니터링, 자동 업로드, 효율적인 메모리 사용 등 대규모 데이터 처리에 강점이 있습니다.



```
# 创建文件夹
self._folder_path = "../data/"
if not os.path.exists(self._folder_path):
    os.makedirs(self._folder_path)
self.file_path = self._folder_path + "/" + date_bar + ".log"

res = self.bq.query(sql).result()
num = 0
for row in res:
    num += 1
    template = {
        "properties": {
            "distinct_id": row['user_id']
            "account_id": row['u_gid']
            "time": datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            "type": "user_setOnce"
            # template['type'] = "user_set"
            "properties['install_source'] = row["install_source"] if row["install_source"] is not None else ""
        }
    }
    with open(self.file_path, 'a+', encoding='utf-8') as fw:
        fw.write(json.dumps(template))
        fw.write("\n")
print(num)
```

언제 사용하나요?

- 서버에서만 알 수 있는 데이터(예: 백엔드 연동, 보안상 클라이언트에 노출되면 안 되는 정보 등)를 수집할 때

- 대량의 과거 로그, 배치 데이터 등 비실시간 데이터도 Thinking Engine에 연동하고 싶을 때
- 클라이언트 SDK만으로 데이터 요구사항을 충족하기 어려울 때

참고:

- LogBus2는 실시간 및 대용량 데이터 업로드에 최적화되어 있어, 파일 모니터링이나 Kafka 연동 등 다양한 데이터 소스를 지원합니다.

3. [RestfulAPI](#)

주요 역할:

- SDK 설치 없이, 직접 HTTP API 방식으로 데이터를 TE 서버로 전송할 수 있습니다.
- 서버/클라이언트 어디서든 HTTP 요청을 통해 데이터를 전송할 수 있으므로, 별도의 SDK 연동이 어렵거나 불필요한 환경에서 유용합니다.

사용 예시:

- 사내 시스템, 외부 서비스, 서버리스 환경 등 SDK 설치가 불가능한 곳에서 데이터 전송
- 특정 트리거(예: 배치 작업, 외부 이벤트 등)에서 TE로 데이터 전송이 필요한 경우

장점:

- 언어나 플랫폼에 구애받지 않고 HTTP로 데이터 전송 가능
- 유연하게 커스텀 데이터 연동 가능

개발 시 참고사항:

- 전송 데이터는 반드시 TE의 데이터 규칙(필드명, 형식 등)을 따라야 하며, 유저 식별 규칙도 정확히 지켜야 합니다.
- 데이터 전송 전, TE 데이터 모델(유저/이벤트 테이블)과 명명 규칙(이벤트명, 속성명 등)을 반드시 확인하세요.

참고:

- RESTful API는 공식 연동 가이드의 데이터 전송 규칙 및 샘플을 반드시 확인해야 하며, 데이터의 정확성과 일관성을 위해 사전 설계가 중요합니다.

2. 트래킹 정책 이해하기

데이터 트래킹 정책은 유저의 행동 데이터를 체계적으로 수집, 전송, 분석하기 위해 설계된 표준화된 규칙과 프로세스를 의미합니다. 이를 통해 데이터의 일관성을 확보하고, 분석 목표에 부합하는 인사이트를 도출하며, 비즈니스 의사결정을 지원합니다.

다른 회사에서는 이를 **[이벤트 택소노미 설계]**라고도 부르며, 주요 목표는 다음과 같습니다:

- 데이터 수집의 일관성 확보.
- 분석 목표에 적합한 데이터 정의.
- 엔지니어링 및 분석 팀 간의 명확한 소통 지원.

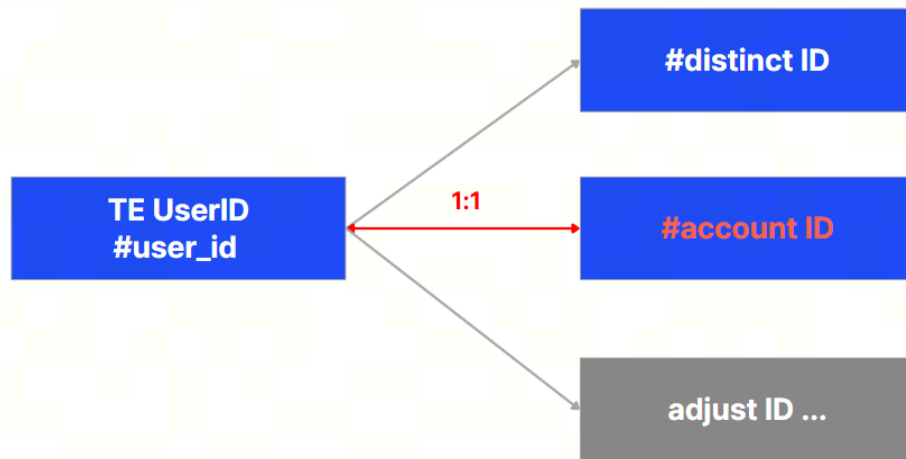
데이터 트래킹 정책 구성 요소

1. 유저 식별 체계

유저의 계정, 행동, 상태를 명확히 식별하기 위해 ID 체계를 설정합니다.

- #account_id: 유저 계정 기반 식별.
- #distinct_id: 임시 유저 식별.

- **#user_id**: 시스템에서 자동 생성된 통합 ID.



2. 이벤트 및 이벤트 속성

유저의 행동을 이벤트로 정의하고, 이를 세부적으로 나타내는 속성을 설계합니다.

- **주요 이벤트 유형:**
 - **핵심 활동:** 결제, 회원가입.
 - **일반 활동:** 클릭, 화면 전환.
 - **자동 수집 이벤트:** ta_app_install, ta_app_start, ta_app_end, ta_app_crash
- **속성 예시:**
 - purchase_amount: 결제 금액.
 - item_id: 아이템 ID
 - battle_duration: 전투 시간

3. 유저 속성

유저의 상태와 행동 이력을 설명하는 속성.

- **고정값:** 변하지 않는 데이터(가입 날짜, 유입 채널).
- **최신값:** 마지막 상태(최종 로그인 시간, 마지막 구매 아이템).
- **누적값:** 과거 행동 누적 데이터(누적 결제 금액, 총 로그인 횟수).

4. 공통 이벤트 속성

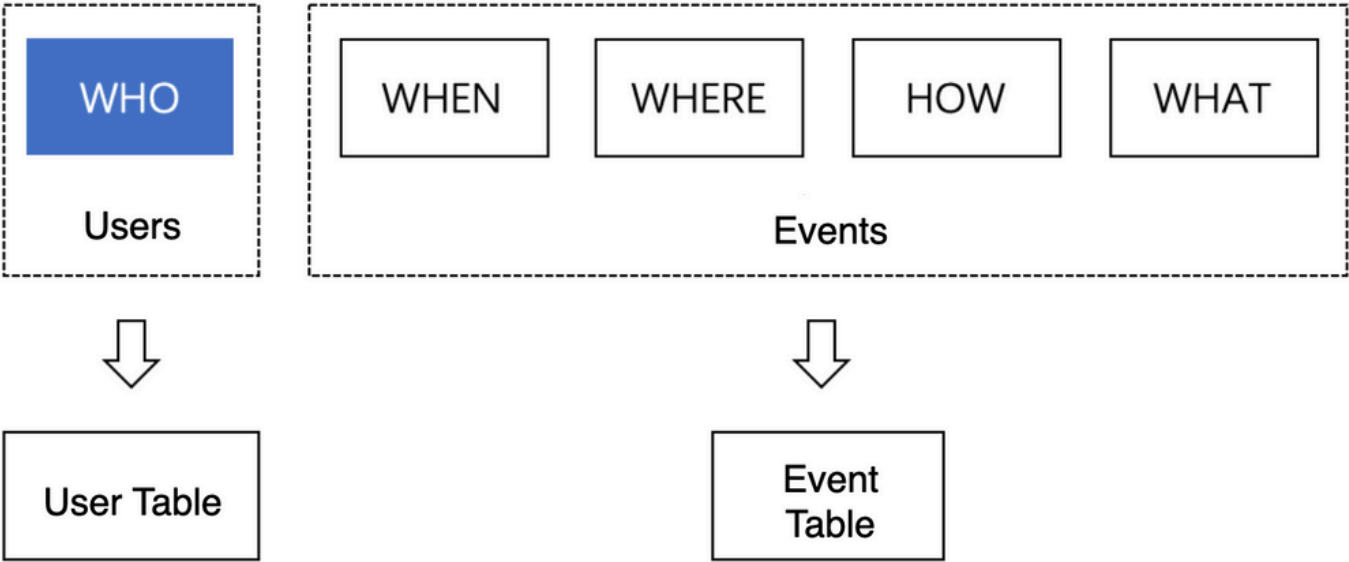
모든 이벤트에 공통적으로 포함되는 속성으로, 분석의 기반이 됩니다.

- **유저 상태:** VIP 등급, 유저 레벨.
- **결제 관련:** 구매 금액, 구매 아이템 ID.
- **환경 속성:** 디바이스 OS, 네트워크 상태.

3. 데이터 구조

데이터 규칙 가이드를 필수로 읽으시길 바랍니다.

Thinking Engine는 크게 유저 데이터와 이벤트 데이터가 저장되는 2개의 테이블 구조로 구성됩니다.



[유저 테이블]

역할 : 유저의 상태, 특징, 속성 등을 기록하는 데 사용됩니다. 예를 들어, 유저의 누적 충전 금액, 현재 레벨, 첫 로그인 시간 등과 같은 정보를 저장합니다. 이러한 정보는 이벤트 테이블에서 모든 충전 이벤트를 합산하여 얻을 수 있지만, 사용자 테이블에 직접 저장하면 계산을 반복하지 않아도 되므로 효율적입니다.

No.	#user_id :	#account_id :	#distinct_id :	#active_time :	#reg_time :	#server_time :	#update_time :
1	1245459230375292928	5805cf22-e7bf-402f-8aca-3e3456f30fcc	73e1881b-1c9f-45ec-916f-1330b027c1bd	2025-04-23 18:21:04.000	2025-04-23 18:21:52.000	2025-05-18 12:12:00.189	2025-04-25 22:29:09.00
2	1245459230928936960	7c463677-8af4-416e-8e7b-ce0dc4d7960b	6e8b7909-77ea-4aad-bb85-92349ee96c97	2025-02-10 06:39:14.000	2025-02-10 06:39:42.000	2025-05-18 12:12:00.189	2025-02-15 04:13:00.00
3	1245459231239311360	366e6a32-ff2a-4813-a39e-94f351de2f2b	80fe3823-53e0-40c5-b7f9-bfab6af83460	2025-03-09 13:54:02.000	2025-03-09 13:54:31.000	2025-05-18 12:12:00.189	2025-03-09 14:08:32.00
4	1245459231897821184	c564f3e0-c47b-477a-9ec1-d6941edad558	db37f52e-4cdd-4921-98fe-2c5d6ff96001	2025-05-11 18:03:22.000	2025-05-11 18:03:46.000	2025-05-18 12:12:00.189	2025-05-11 21:47:03.00
5	1245459231910404096	0b60e06e-50e2-4579-81c2-c4a74ed13a86	ac2778b1-79e7-416f-99a8-ff827b239b9c	2025-02-19 23:08:09.000	2025-02-19 23:08:42.000	2025-05-18 12:12:00.189	2025-02-19 23:27:38.00
6	1245459231922991104	9f4e69fe-d03f-44c6-a006-2bb636d5241f	5c0f9980-d357-4d61-825c-b5b426dd2d7e	2025-03-20 21:31:11.000	2025-03-20 21:31:46.000	2025-05-18 12:12:00.189	2025-03-20 21:44:29.00
7	1245459232531156992	5c2c1d9f-5a7f-4c6c-8f0a-5026b621b17d	c30b24a2-c446-4757-a3d3-54b1d009bd2e	2025-03-07 22:16:31.000	2025-03-07 22:16:56.000	2025-05-18 12:12:00.189	2025-03-08 10:27:04.00
8	1245459233068027904	a6c0a9b6-7a07-4691-a3a9-ebbe1e53398f	0dd42482-eb2a-46f9-9b8c-1933d3ad3bce	2025-04-02 13:51:16.000	2025-04-02 13:52:10.000	2025-05-18 12:12:00.189	2025-04-02 13:58:37.00
9	1245459283252883480	0612f97d-109d-4be1-829f-bf789f71af31	95957643-377e-45f0-bcbc-9bbc38bf76bc	2025-05-21 08:21:57.000	2025-05-21 08:22:50.000	2025-05-18 12:12:00.189	2025-05-21 08:29:11.00

속성 제한 : 사용자 속성은 최대 500개까지 가능합니다.

[이벤트 테이블]

역할 : 유저 행동을 추상화한 데이터를 저장합니다. 유저 행동은 "누가", "언제", "어디서", "무엇을", "어떻게" 했는지, 그리고 그 결과는 어떠한지로 정의되며, 이러한 정보를 하나의 데이터 테이블에 기록합니다. 예를 들어, 441번 유저가 3월 14일에 신

용카드로 3000원을 충전했다는 정보가 이벤트 테이블에 저장될 수 있습니다.

No.	\$part_event :	\$part_date :	#user_id :	#event_name :	#event_time :	#account_id :	#distinct_id :	#server_time :
9	leave_guild	2025-05-27	1245467651824689152	leave_guild	2025-05-27 10:56:07.000	9e00c941-c3d5-435e-9a6b-fb208dc92ee7	bed71656-4bfc-4d86-9ab3-0975311069de	2025-06-17 12:03:08.257
8	leave_guild	2025-05-27	1245460209715908632	leave_guild	2025-05-27 01:23:49.000	51bcbcd9-3e4a-4055-addd-c2a97eb946f8	c1a74286-bdc2-42bb-a162-7868237b8daf	2025-06-17 12:03:08.257
10	leave_guild	2025-05-27	1245464195596558382	leave_guild	2025-05-27 01:02:14.000	2db8db4b-61df-4ca3-a20f-8852050cd702	b577d77a-f081-47b8-bba1-503f80f11433	2025-06-17 12:03:08.257
7	cost	2025-05-27	1245461967049596932	cost	2025-05-27 00:00:00.000	(null)	system	2025-06-17 12:02:52.601
6	cost	2025-05-27	1245461967049596932	cost	2025-05-27 00:00:00.000	(null)	system	2025-06-17 12:02:52.601
5	cost	2025-05-27	1245461967049596932	cost	2025-05-27 00:00:00.000	(null)	system	2025-06-17 12:02:52.601
4	cost	2025-05-27	1245461967049596932	cost	2025-05-27 00:00:00.000	(null)	system	2025-06-17 12:02:52.601
3	cost	2025-05-27	1245461967049596932	cost	2025-05-27 00:00:00.000	(null)	system	2025-06-17 12:02:52.601
2	cost	2025-05-27	1245461967049596932	cost	2025-05-27 00:00:00.000	(null)	system	2025-06-17 12:02:52.601
1	cost	2025-05-27	1245461967049596932	cost	2025-05-27 00:00:00.000	(null)	system	2025-06-17 12:02:52.601

속성 제한 : 이벤트 속성은 최대 1500개, 이벤트는 최대 1000개까지 가능합니다

데이터 모델의 활용

데이터 수집 : 데이터 수집 계획을 수립할 때, 이러한 데이터 모델을 고려하여 어떤 유저 행동 데이터를 수집할지 결정합니다. 예를 들어, 유저 결제 상황을 분석하려면 유저 결제 행동 데이터를 수집해야 합니다.

데이터 분석 : 분석 모듈에서는 이러한 데이터 모델을 기반으로 다양한 유형의 분석을 수행합니다. 예를 들어, 이벤트 분석 모델을 사용하여 일정 기간 동안 유저가 특정 행동을 수행한 집계 지표나 지표의 변화 추세를 계산할 수 있습니다.

4. 예시 코드 (unity)

다음 코드에서 1번과 2번(초기화와 자동 수집 이벤트 활성화)은 반드시 이어서 작성하고, 전체적으로는 순서대로 코드를 구성할 것을 권장합니다.

SDK 다운로드 및 코드 복사 등 : [여기서](#)

순서 : SDK 초기화 - 모든 자동 수집 이벤트 활성화 - 유저 로그인 설정 - 공통 이벤트 속성 설정 - 이벤트 전송 - 유저 속성 설정

```
using ThinkingData.Analytics;

if (유저가 개인정보 보호 정책에 동의함)
{
    // 1. SDK 초기화
    TDAalytics.Init("APPID", "SERVER");

    // 2. 모든 자동 수집 이벤트 활성화 (몇개만 보낼려면 자동 이벤트 수집 가이드 참고)
    ThinkingAnalyticsAPI.EnableAutoTrack(TDAutoTrackEventType.All);

    // 3. 유저가 로그인한 경우, 계정 ID를 설정하며 고유 식별자로 사용
    TDAalytics.Login("TA");

    // 4. 공통 이벤트 속성 설정 (모든 이벤트에 자동 포함)
    Dictionary<string, object> superProperties = new Dictionary<string, object>();
    superProperties["channel"] = "ta"; // String
    superProperties["age"] = 1; // Number
    superProperties["isSuccess"] = true; // Boolean
    superProperties["birthday"] = DateTime.Now; // DateTime
    superProperties["object"] = new Dictionary<string, object>() { { "key", "value" } }; // Object
    superProperties["object_arr"] = new List<object>() { new Dictionary<string, object>() { { "key", "value" } } }; // Object List
    superProperties["arr"] = new List<object>() { "value" }; // Array
    TDAalytics.SetSuperProperties(superProperties); // 공통 이벤트 속성 설정

    // 5. 이벤트 전송 (상품 구매)
    Dictionary<string, object> properties = new Dictionary<string, object>() { { "product_name", "상품 이름" } };
    TDAalytics.Track("product_buy", properties);

    // 5.1 예시 - 이벤트 전송 (랭킹) - Object Array 활용
    Dictionary<string, object> rank_info = new Dictionary<string, object>()
    {
        { "id", "123051" }, // String
        { "rank", 1 }, // Number
        { "score", 85 } // Number
    };

    List<object> rank_list = new List<object> { rank_info };

    Dictionary<string, object> eventProperties = new Dictionary<string, object>()
    {
        { "product_name", "아이템 이름" }, // String
        { "rank_info", rank_list } // Object Array (객체 리스트)
    };

    TDAalytics.Track("product_buy_with_rank", eventProperties);
}

// 6. 유저 속성 설정
TDAalytics.UserSet(new Dictionary<string, object>() { { "user_name", "TA" } });
```

주의 사항

1. 맨 처음 전송된 데이터를 기준으로 TE 시스템은 해당 데이터의 유형을 판별합니다.

(예: 처음 item_info 이벤트는 list 유형으로 전송 → 다음에 코드를 수정해 object array로 전송했을 경우 오류로 판단해 NULL값으로 저장됨.) * 이미 실수로 전송된 경우에는 싱킹데이터 매니저에게 문의

2. 싱킹데이터에서 유형을 변경할 수 있지만, 변경 전 데이터는 NULL로 처리됩니다.

- 처음부터 올바른 속성 유형으로 전송해야 데이터 손실을 막을 수 있습니다.

3. 이벤트 이름이 공백이 포함된 경우 수집되지 않습니다.

(예: 잘못된 형태 item info 올바른 형태 → item_info)

4. Array(Object) 인데 그 안에 item_id 와 item_num을 안에 담지않고 따로 보내시면 안됩니다.

5. 결제 속성에 결제 금액의 경우 꼭 number(int) 값으로 보내야합니다. 왜냐하면 글로벌 서비스의 경우 통화(환율 변환) 기능을 사용해야하는데 string으로 보내면 이용할 수 없습니다.

gacha_item_id	가차 아이템 ID	number	
cost_item_id	지불 아이템 ID	number	
cost_item_price	지불 아이템 금액	number	
reward_item_info	보상 아이템 정보	array(object)	값 예시: "reward_item_info": [{"item_id":"1","item_num":100}, {"item_id":"2","item_num":200}]
reward_item_info.item_id	보상 아이템 ID	string	
reward_item_info.item_num	보상 아이템 수량	number	
is_retry	한번 더 여부	boolean	한번더 진행(True:0), 한번더 진행 안함(FALSE:0)

// 1. 개별 보상 아이템 정보 생성

```
Dictionary<string, object> reward_item = new Dictionary<string, object>()
```

```
{
```

```
    { "item_id", "1" },
```

```
object    { "item_num", 100 }
```

```
group    };
```

// 2. 객체 배열로 만들

```
예 List<object> reward_item_info = new List<object> { reward_item };
```

```
시 // 3. 이벤트 속성에 추가
```

```
코 Dictionary<string, object> eventProperties = new Dictionary<string, object>()
```

```
드 {
```

```
    { "reward_item_info", reward_item_info }
```

```
};
```

// 4. 이벤트 전송

```
TDAnalytics.Track("reward_event", eventProperties);
```

5. 시스템 제한

프리셋 속성 및 시스템 필드

프리셋 속성

- 프리셋 속성은 TE에서 생성하거나 수집한 속성을 의미합니다.
- 모든 프리셋 속성은 "#"로 시작하며, 이벤트 속성으로만 사용됩니다.
- 이러한 속성들은 고정된 이름과 의미를 가집니다.

시스템 필드

- 시스템 필드는 데이터 구조에서 사용되는 필드(예: `#account_id`, `#event_time`) 또는 데이터베이스에서 특수한 목적으로 사용되는 필드입니다.
- 시스템 필드는 분석 모델에서 직접 사용되지 않거나 간접적으로만 사용됩니다.
- 시스템 필드는 이벤트 속성이나 유저 속성으로 사용할 수 없으며, 데이터 전송 시 포함해서는 안 됩니다.

주의사항

1. "#"로 시작하는 속성의 제한

- ([링크에서 확인](#)) 프리셋 속성을 제외하고, "#"로 시작하는 속성은 모두 불법 필드로 간주되어 데이터베이스에 저장되지 않습니다.
- 따라서, 커스텀 속성을 "#"로 시작하지 않도록 설정하세요.

2. 시스템 필드 사용 제한

- 시스템 필드는 이벤트 속성이나 유저 속성으로 사용할 수 없습니다.

3. 프리셋 속성 사용 권장 사항

- `#ip`를 제외한 모든 프리셋 속성은 직접 사용하지 않는 것이 좋습니다.
- 클라이언트 SDK와 기타 데이터 전송 방식을 동시에 사용하는 경우, 여러 데이터 소스 간 속성을 일관되게 유지해야 할 때, ThinkingData 팀의 도움을 받아 설정하세요.

데이터 규칙

데이터를 전송할 때는 일정한 규칙을 따라야 합니다:

- **명명 규칙:** 이벤트명이나 속성명에는 문자, 숫자, 밑줄(_)만 포함할 수 있으며, 최대 50자까지 가능합니다.

- **속성 값의 데이터 타입:** 숫자, 문자열, 시간, 불리언, 리스트, 객체 등 다양한 타입을 지원하며, 각 타입마다 제한 사항이 있습니다.

주의

- **ThinkingData**에 저장되는 데이터는 게스트 ID나 계정 ID 중 하나를 반드시 포함해야 합니다.
- 클라이언트 SDK는 기본적으로 게스트 ID를 생성하며, 로그인 시 계정 ID를 설정하여 두 ID를 모두 전합니다.
- 속성 값의 데이터 유형은 최초 전송 시 정의되며, 이후 변경 불가능합니다. (ThinkingData 팀에 변경 요청해야함)

데이터 유형

TE 데이터 유형	값 예시	값 설명	데이터 유형
Number	123,1.23	데이터 범위는 -9E15부터 9E15까지입니다.	Number
String	"ABC","상해"	문자열의 기본 최대 크기는 2KB입니다.	String
Time	"2019-01-01 00:00:00","2019-01-01 00:00:00.000"	"yyyy-MM-dd HH:mm:ss.SSS" 또는 "yyyy-MM-dd HH:mm:ss" 형식을 사용할 수 있으며, 날짜를 나타내려면 "yyyy-MM-dd 00:00:00"을 사용할 수 있습니다.	String
boolean	true, false	-	Boolean
List	["a","1","true"]	리스트의 모든 요소는 문자열 유형으로 변환되며, 최대 500개의 요소를 포함할 수 있습니다.	Array(String)
Object	{ "hero_name": "Liu Bei", "hero_level": 22, "hero_equipment": ["Twin Swords of Justice", "Red Hare"], "hero_support_status": false }	객체 내부의 각 하위 속성(Key)은 자체 데이터 유형을 가지며, 값 설명은 위의 해당 유형 속성을 참고하십시오. 객체 내에는 최대 100개의 하위 속성을 포함할 수 있습니다.	Object
Object Group	{ "hero_info": [{ "hero_name": "Liu Bei", "hero_level": 22, "hero_equipment": ["Twin Swords of Justice", "Red Hare"], }] }	객체 그룹 내의 각 하위 속성(Key)은 자체 데이터 유형을 가지며, 값 설명은 위의 해당 유형 속성을 참고하십시오. 객체 그룹 내에는 최대 500개의 객체를 포함할 수 있습니다.	Array(Object)

	<pre>"hero_support_status": false }, { "hero_name": "Liu Bei", "hero_level": 22, "hero_equipment": ["Twin Swords of Justice", "Red Hare"], "hero_support_status": false }] }</pre>	
--	--	--

단일 프로젝트 메타데이터 제한

종류	상한(최대 한도)
이벤트	500개
이벤트 속성	1000개
유저 속성	500개
차원 테이블 속성	3000개
	하나의 차원 테이블 속성에서 업로드할 수 있는 데이터 양: 기본 크기는 200MB이며, 최대 2GB까지 설정할 수 있습니다. 행 수에는 제한이 없습니다.
가상 속성	3000개
가상 이벤트 내 이벤트 개수 제한	20개

전송 데이터 크기 및 길이 제한

종류	상한(최대 한도)
단일 JSON	2KB
account_id/distinct_id 길이 제한	128자 (문자)

시스템 기능 (단일 프로젝트 내 제한)

종류	상한(최대 한도)
이벤트 분석의 분석 지표 개수	50개
퍼널 분석의 단계 개수	30개
경로 분석에서 선택 가능한 이벤트 개수	30개
퍼널 분석에서 그룹화 항목의 값 개수	퍼널 차트에서 최대 4개의 그룹화 항목만 표시 가능
각 모델에서 설정할 수 있는 지표 필터링 조건/전역 필터링 조건 개수	100개 초과
유저 태그	최대 200개 생성 가능
	커스텀 조건 태그당 최대 20개의 태그 값 설정 가능
결과 코호트 개수	각 분석가 및 상위 권한 구성원은 최대 50개 생성 가능하며, 구성원별 제한은 독립적으로 계산됨
조건 코호트, ID 코호트	각각 30개
	유저 ID 코호트 파일 업로드 최대 크기는 3GB
경고(알림) 관리 개수	최대 50개 작업
SQL 조회 시 다운로드 가능한 데이터 양	100만 건
리포트 우측 상단 '데이터 다운로드'에서 다운로드 가능한 데이터 양	50만 건

데이터 권한 (단일 프로젝트 내 제한)

종류	상한(최대 한도)
구성원 그룹	50개
커스텀 역할 생성 개수	30개

Open API

종류	상한(최대 한도)
커스텀 조회 API	querySql은 동기 조회이며, 단일 조회에서 반환할 수 있는 데이터 최대량은 5억 건입니다.
	submit-sql은 비동기 조회이며, 결과를 페이지 단위로 반환합니다. 기본적으로 한 페이지당 1만 행이 반환되며, 전체 조회 데이터에는 제한이 없습니다.

6. 데이터 전송 확인

“ 주의 사항

- 1. 데이터를 전송할때는 라이브 환경에 바로 전송하지 마시고, 꼭 테스트 프로젝트를 생성해서 테스트 프로젝트에서 확인 후, 라이브 환경으로 전송하시기를 추천 드립니다. (프로젝트 생성은 [가이드](#)를 참조바랍니다.)
- 2. 테스트 프로젝트와 라이브 프로젝트 동시에 데이터 전송하는 작업은 [다중 인스턴스 가이드](#)를 참조바랍니다. (각 연동 가이드에 다중 인스턴스 가이드 참조, 현재 링크는 유니티)

”

1. 데이터 수신

이 페이지의 데이터는 약 10분 간격으로 업데이트됩니다.

대시보드

분석

유저

운영

데이터

데이터

데이터 관리

메타 이벤트

이벤트 속성

유저 속성

데이터 테이블

지표

통화

트래킹 관리

트래킹 정책

데이터 검증

데이터 수신

실시간 데이터

디버그 모드

데이터 연동

연동 가이드

서드파티 통합

데이터 수신

수신 차단

수집 시간

2024-12-13 00:00

>

2024-12-13 15:50

🗄

🔄

🔍 검색

구간 데이터

수신 1,212 개; 저장 1,212 개; 오류 1 개; 실패 0 개

수신 = 저장된 데이터 + 저장 실패 데이터. 저장된 데이터에는 오류가 발생한 데이터도 포함됩니다

<input type="checkbox"/>	데이터 이름	별칭	수신	저장	오류	실패	작업
<input type="checkbox"/>	유저 속성	유저 속성	0	0	0	0	🕒
<input type="checkbox"/>	click_event	클릭(전체)	200	200	1	0	🕒
<input type="checkbox"/>	demo_form_submit_success 차단됨	-	0	0	0	0	🕒
<input type="checkbox"/>	engaged_session	-	44	44	0	0	🕒
<input type="checkbox"/>	first_interaction 차단됨	첫 상호작용	0	0	0	0	🕒
<input type="checkbox"/>	form_submit_success	폼 제출 성공	1	1	0	0	🕒
<input type="checkbox"/>	image_click	클릭(이미지)	0	0	0	0	🕒
<input type="checkbox"/>	link_click	클릭(링크)	0	0	0	0	🕒
<input type="checkbox"/>	new_session	세션 시작	0	0	0	0	🕒
<input type="checkbox"/>	page_exit	페이지 종료	65	65	0	0	🕒
<input type="checkbox"/>

총 20 개의 기록

1

100 / 페이지

항목	설명
수신	Receiver가 수신한 총 데이터 수집량 (저장된 데이터 건수가 아닌, 수집된 원본 데이터 개수 기준)
저장	처리 후 이벤트 및 유저 테이블에 저장되거나 업데이트된 데이터 수집량 (정상 저장 + 이상 데이터 포함)
오류	이벤트 및 유저 테이블에 저장되었지만, 속성 수준의 오류(예: 잘못된 속성명, 속성 타입 불일치 등)가 발생한 데이터 → 오류 속성은 NULL 처리됨
실패	필수 필드 누락(#event_name 없음), 제한 범위를 초과한 #event_time 값 등으로 인해 저장되지 못한 데이터 수집량

1.1 오류 정보 조회

"오류" 또는 "실패" 항목에 마우스를 올리면 [보기] 버튼이 나타납니다.

대시보드

분석

유저

데이터

운영

New

데이터

데이터 관리

이벤트

이벤트 속성

유저 속성

데이터 테이블

지표

통화

트래킹 관리

트래킹 정책

데이터 검증

데이터 수신

실시간 데이터

디버그 모드

데이터 연동

연동 가이드

서드파티 통합

데이터 수신

수신 차단

수집 시간 2025-02-02 00:00 > 2025-02-02 17:50

구간 데이터 : 수신 49.56만 개 ; 저장 49.56만 개 ; 오류 5.72만 개 ; 실패 0 개 수신 = 저장된 데이터 + 저장 실패 데이터. 저장된 데이터에는 오류가 발생한 데이터도 포함됩니다

<input type="checkbox"/>	데이터 이름	별칭	수신	저장	오류	실패	작업
<input type="checkbox"/>	유저 속성	유저 속성	0	0	0	0	🔍
<input type="checkbox"/>	battle_fail	전투 실패	2,012	2,012	961	0	🔍
<input type="checkbox"/>	battle_start	전투 시작	42,159	42,159	18,563	0	🔍
<input type="checkbox"/>	battle_win	전투 승리	181,430	181,430	37,753 보기	0	🔍
<input type="checkbox"/>			57,073	57,073	0	0	🔍
<input type="checkbox"/>			286	286	0	0	🔍
<input type="checkbox"/>			1,317	1,317	0	0	🔍
<input type="checkbox"/>			519	519	0	0	🔍
<input type="checkbox"/>			3,047	3,047	1	0	🔍
<input type="checkbox"/>			3,124	3,124	0	0	🔍

총 23 개의 기록

1 2 3 10 / 페이지 이동하기 페이지

오류 데이터 상세 페이지에서는 해당 이벤트의 오류 데이터에 대한 상세 정보를 확인할 수 있습니다.

데이터 수신 / 오류

battle_win

시작 시간 2025-02-02 00:00

종료 시간 2025-02-02 17:50

오류 37,753 개

이벤트 테이블 속성값이 올바르지 않습니다 ①

속성 [maxbattlepower] 값이 올바르지 않습니다.

{\"#account_id\":\"USER_65fbb196a4bce2b802dcaaa_Taiwan\",\"#time\":\"2025-02-01 22:59:19.009\",\"#distinct_id\":\"999690b0-75d9-4f87-a087-034d33bd8fb5\",\"#uuid\":\"b7e801f2-6e06-4e1a-af6d-3add5447163c\",\"#type\":\"track\",\"#event_name\":\"battle_win\",\"properties\":{\"#lib_version\":\"3.0.7\",\"MaxBattlePower\":\"1.84810354818315E+19\",\"StageIndex\":39,\"Difficulty\":0,\"#os\":\"Android\",\"#zone_offset\":8,\"BattleType\":\"HuntMode\",\"#ram\":\"1.3/3.9\",\"#data_source\":\"Native_SDK\",\"#screen_height\":540,\"#device_model\":\"2210132C\",\"#system_language\":\"zh\",\"#network_type\":\"WiFi\",\"#lib\":\"Unity\",\"#device_type\":\"Tablet\",\"#disk\":\"19.9/28.8\",\"AdKey\":\"\",\"#carrier\":\"SmarTone Mobile Communications Limited\",\"#device_id\":\"f9a30288fe7bb0bb\",\"#bundle_id\":\"com.Cryptect.Mercenary\",\"ChapterIndex\":3,\"#screen_width\":960,\"#install_time\":\"2024-10-13 23:43:46.631\",\"IsSweep\":false,\"#simulator\":true,\"IsFirstClear\":true,\"#fps\":19,\"MercenaryGroupLevel\":54687,\"ScenarioIndex\":4,\"#manufacturer\":\"Xiaomi\",\"RewardItemInfo\":{\"#Type\":\"Goods\",\"Count\":798511243,\"Key\":\"Gold\"},{\"#Type\":\"Goods\",\"Count\":6992704556,\"Key\":\"Exp\"},{\"#Type\":\"Goods\",\"Count\":3300,\"Key\":\"ReinforceStone\"},{\"#Type\":\"Goods\",\"Count\":145,\"Key\":\"CreatureEssence\"},{\"#Type\":\"Goods\",\"Count\":40,\"Key\":\"EvolutionEssence\"},{\"#Type\":\"Goods\",\"Count\":1,\"Key\":\"ManaStone\"},{\"#Type\":\"Goods\",\"Count\":3,\"Key\":\"TacticsManual\"}},\"#os_version\":\"9\",\"#app_version\":\"1.9.0\"}}

실패 데이터 상세 페이지에서는 해당 이벤트의 저장 실패 원인을 확인할 수 있습니다.

2. 실시간 데이터

최신 저장된 이벤트 데이터 1000개만 표시됩니다.

JSON 데이터의 각 행은 형식을 표시하고 복사할 수 있습니다.

대시보드 분석 유저 운영 데이터

데이터

실시간 데이터 오류 데이터

최근 저장된 1000개의 데이터를 표시 데이터 다운로드

업로드 시간

데이터 이름

표본 데이터

2024-12-13 15:51:29

ta_page_show

{\"#time\":\"2024-12-13 15:51:17.173\",\"#distinct_id\":\"193bea314080-08d86d7bb51e6d-26011851-2073600-193bea31409b5f\",\"#uid\":\"72eb0ecc-c829-790-e166-9184af35b910\",\"#type\":\"track\",\"#event_name\":\"ta_page_show\",\"properties\":{\"#lib_version\":\"2.0.3\",\"#referrer_host\":\"www.thinkingdata.kr\",\"#zone_offset\":9,\"#os\":\"Windows\",\"#device_id\":\"193bea314080-08d86d7bb51e6d-26011851-2073600-193bea31409b5f\",\"#data_source\":\"JavaScript\",\"#screen_height\":1080,\"#referrer\":\"https://www.thinkingdata.kr/blog\",\"#screen_width\":1920,\"#browser\":\"chrome\",\"#system_language\":\"ko-KR\",\"#ua\":\"mozilla/5.0 (windows nt 10.0; win64; x64; applewebkit/537.36 (khtml, like gecko) chrome/131.0.0.0 safari/537.36; #lib\":\"js\",\"#browser_version\":\"131.0.0.0\",\"#title\":\"게임 데이터 관리, 왜 해야 하는대? 왜 필요한대?\",\"#utm\":{\"#\":\"url\":\"https://www.thinkingdata.kr/blog/data-manage\",\"#url_path\":\"/blog/data-manage\"}}}

2024-12-13 15:51:08

ta_page_hide

{\"#time\":\"2024-12-13 15:50:53.223\",\"#distinct_id\":\"1920864b74a1433-0ef98b9e1aed12-26001051-2073600-1920864b74b1872\",\"#uid\":\"90285fac-2a12-4ae7-a76e-bcc188ed4ac1\",\"#type\":\"track\",\"#event_name\":\"ta_page_hide\",\"properties\":{\"#lib_version\":\"2.0.3\",\"#referrer_host\":\"docs-v2.thinkingdata.kr\",\"#zone_offset\":9,\"#os\":\"Windows\",\"#duration\":3.799,\"#device_id\":\"1920864b74a1433-0ef98b9e1aed12-26001051-2073600-1920864b74b1872\",\"#data_source\":\"JavaScript\",\"#screen_height\":1080,\"#referrer\":\"https://docs-v2.thinkingdata.kr/\",\"#screen_width\":1920,\"#browser\":\"chrome\",\"#system_language\":\"ko\",\"#ua\":\"mozilla/5.0 (windows nt 10.0; win64; x64; applewebkit/537.36 (khtml, like gecko) chrome/131.0.0.0 safari/537.36; #lib\":\"js\",\"#browser_version\":\"131.0.0.0\",\"#title\":\"Thinkingdata | 글로벌 게임 빅데이터 분석 솔루션\",\"#utm\":{\"#\":\"url\":\"https://www.thinkingdata.kr/\",\"#url_path\":\"/\"}}}

2024-12-13 15:50:49.424

{\"#time\":\"2024-12-13 15:50:49.424\",\"#distinct_id\":\"1920864b74a1433-0ef98b9e1aed12-26001051-2073600-1920864b74b1872\",\"#uid\":\"6a6521e2-57b9-c5da-ad68-ec15e328296\",\"#type\":\"track\",\"#event_name\":\"ta_page_show\",\"properties\":{\"#lib_version\":\"2.0.3\",\"#referrer_host\":\"docs-v2.thinkingdata.kr\",\"#zone_offset\":9,\"#os\":\"Windows\",\"#duration\":3.799,\"#device_id\":\"1920864b74a1433-0ef98b9e1aed12-26001051-2073600-1920864b74b1872\",\"#data_source\":\"JavaScript\",\"#screen_height\":1080,\"#referrer\":\"https://docs-v2.thinkingdata.kr/\",\"#screen_width\":1920,\"#browser\":\"chrome\",\"#system_language\":\"ko\",\"#ua\":\"mozilla/5.0 (windows nt 10.0; win64; x64; applewebkit/537.36 (khtml, like gecko) chrome/131.0.0.0 safari/537.36; #lib\":\"js\",\"#browser_version\":\"131.0.0.0\",\"#title\":\"Thinkingdata | 글로벌 게임 빅데이터 분석 솔루션\",\"#utm\":{\"#\":\"url\":\"https://www.thinkingdata.kr/\",\"#url_path\":\"/\"}}}

총 1000 개의 기록

1 2 3 4 5 ... 10 100 / 페이지 이동하기 페이지

2.1 오류 데이터 조회

- 10분 단위로 각 오류 유형별 데이터 건수를 집계하여 표시합니다.
- 속성 수준의 오류에 대해서는 구체적인 오류 원인과 오류 샘플을 제공합니다.
- 최대 1,000건의 최근 오류 데이터를 조회할 수 있습니다.

데이터

데이터 관리

이벤트

이벤트 속성

유저 속성

데이터 테이블

지표

통화

트래킹 관리

트래킹 정책

데이터 검증

데이터 수신

실시간 데이터

디버그 모드

데이터 연동

연동 가이드

서드파티 통합

대시보드

분석

유저

운영

데이터

실시간 데이터

오류 데이터

오류 원인별로 최근 1000개의 통계 데이터를 표시하며, 10분마다 통계 결과를 자동으로 새로 고침 합니다. [데이터 다운로드](#)

날짜	수집 시간	오류	처리	오류 유형	표본 데이터
2025-01-24	17:20-17:30	148	저장, 오류 속성 삭제	이벤트 속성 유형이 올바르지 않습니다	ⓘ • 속성 [contentId] 형식이 맞지 않습니다. 올바른 형식: [number]
2025-01-24	17:10-17:20	7,947	저장, 오류 속성 삭제	이벤트 속성 유형이 올바르지 않습니다	ⓘ • 속성 [contentId] 형식이 맞지 않습니다. 올바른 형식: [number]