

Deus 서버를 소개합니다

<https://github.com/5d-jh>



주은 연습장
Deus team

File / Component



Page



페이지 88

unwrap 버그

이미지 추출

데이터 싱크 버그

잡제기 버그

페이지 83

페이지 82

페이지 81

페이지 80

• 오프라인 결제 홈 / 편의점

페이지 78

페이지 76

페이지 77

페이지 74

페이지 75

페이지 73

페이지 72

Layers

Scroll

> Frame 2/4

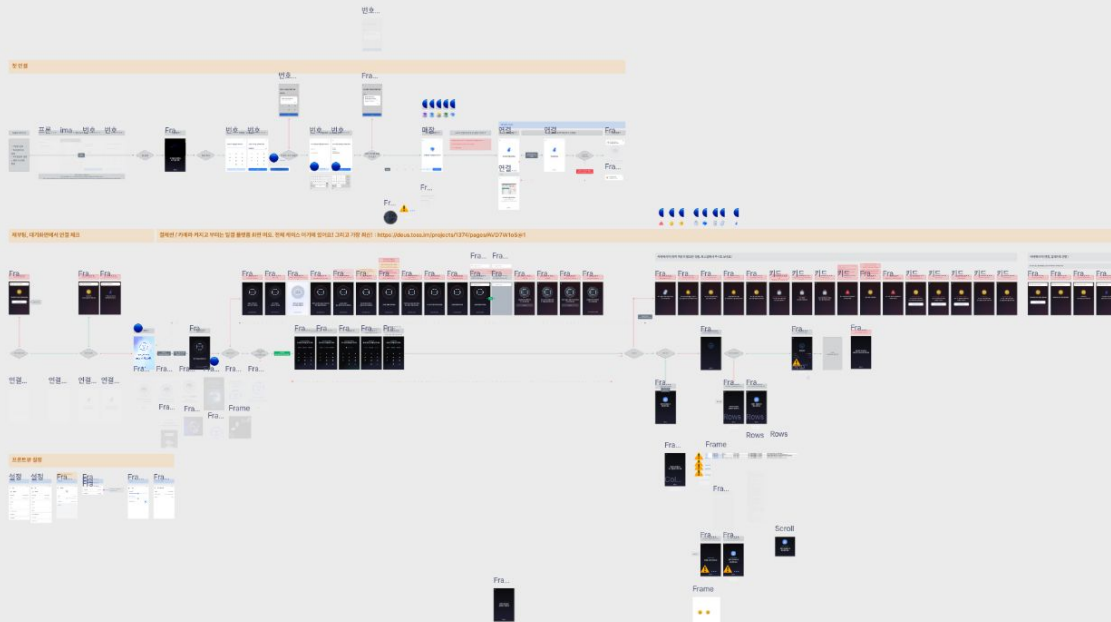
> Frame

> Frame

AnimateText [공지원종료]



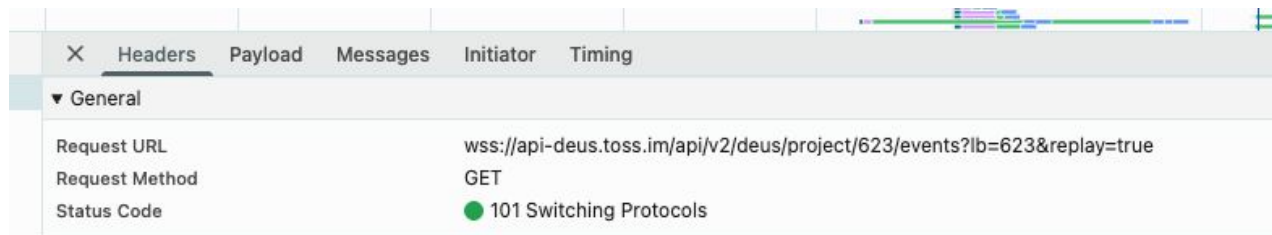
Test Description



왜 Figma를 쓰지 않나요?

웹소켓 서버

Sticky Session



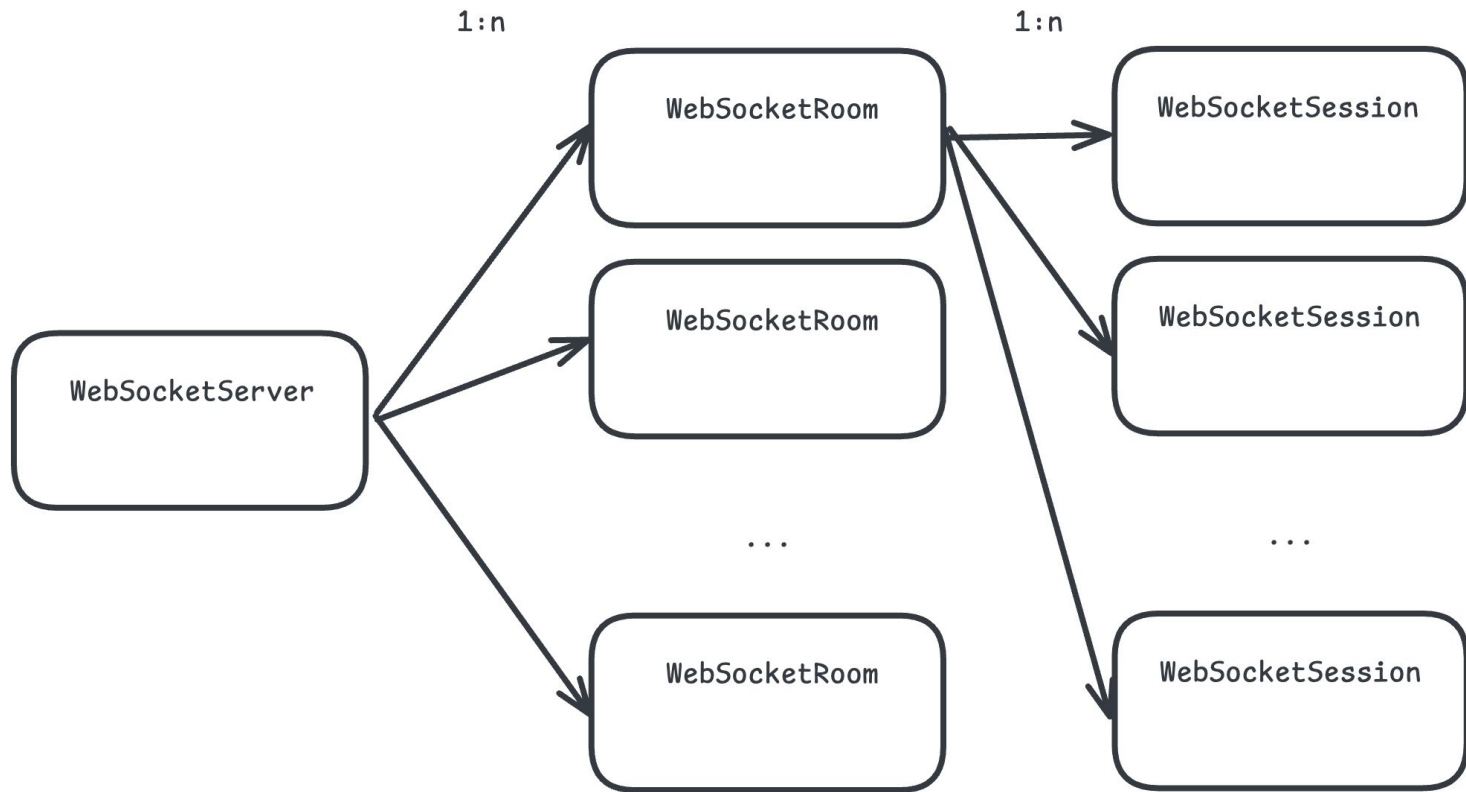
The screenshot shows a network tool interface with a tabbed header containing 'X', 'Headers', 'Payload', 'Messages', 'Initiator', and 'Timing'. The 'Headers' tab is selected. Below the tabs is a section labeled 'General' with a dropdown arrow. The 'General' section displays the following information:

Request URL	wss://api-deus.toss.im/api/v2/deus/project/623/events?lb=623&replay=true
Request Method	GET
Status Code	● 101 Switching Protocols

ConsistentHashLB

Consistent Hash-based load balancing can be used to provide soft session affinity based on HTTP headers, cookies or other properties. The affinity to a particular destination host may be lost when one or more hosts are added/removed from the destination service.

실시간 편집



실시간 편집

```
function (wss: WebSocketServer) {  
  wss.on('room', (room) => {  
    // every new room(project)  
    room.on('session', (session) => {  
      // every new session(browser tab)  
      session.on('message', (message) => {  
        // every new message  
      })  
    })  
  })  
}
```

실시간 편집

```
function Announcer(wss: WebSocketServer) {  
  wss.on('room', (room) => {  
    room.on('session', (session) => {  
      room.broadcast(  
        `please welcome ${session.userName}!`  
      );  
      session.on('message', (message) => {  
      });  
    });  
  });  
}
```

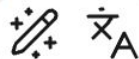

실시간 편집

```
function MessageSaver(wss: WebSocketServer) {  
  wss.on('room', (room) => {  
    room.on('session', (session) => {  
      session.on('message', (message) => {  
        repository.insert({  
          message,  
          user: session.user,  
          projectName: room.project.name,  
        })  
      });  
    });  
  });  
}
```

실시간 편집

How to detect and close broken connections?

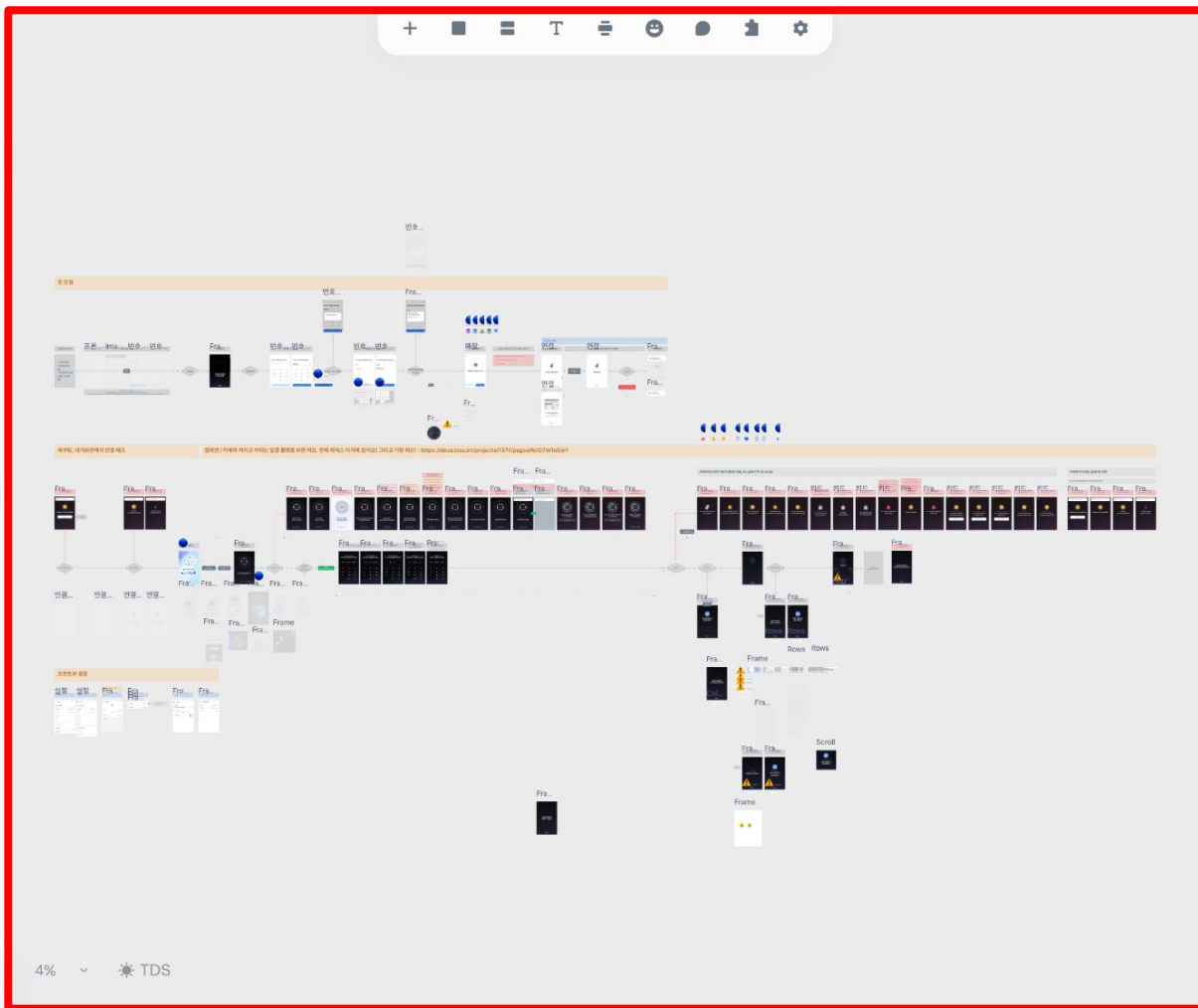
Sometimes, the link between the server and the client can be interrupted in a way that keeps both the server and the client unaware of the broken state of the connection (e.g. when pulling the cord).



In these cases, ping messages can be used as a means to verify that the remote endpoint is still responsive.

```
import { WebSocketServer } from 'ws';  
  
function heartbeat() {
```

데이터 편집



거대한 JSON

JSONPatch

Internet Engineering Task Force (IETF)
Request for Comments: 6902
Category: Standards Track
ISSN: 2070-1721

P. Bryan, Ed.
Salesforce.com
M. Nottingham, Ed.
Akamai
April 2013

JavaScript Object Notation (JSON) Patch

Abstract

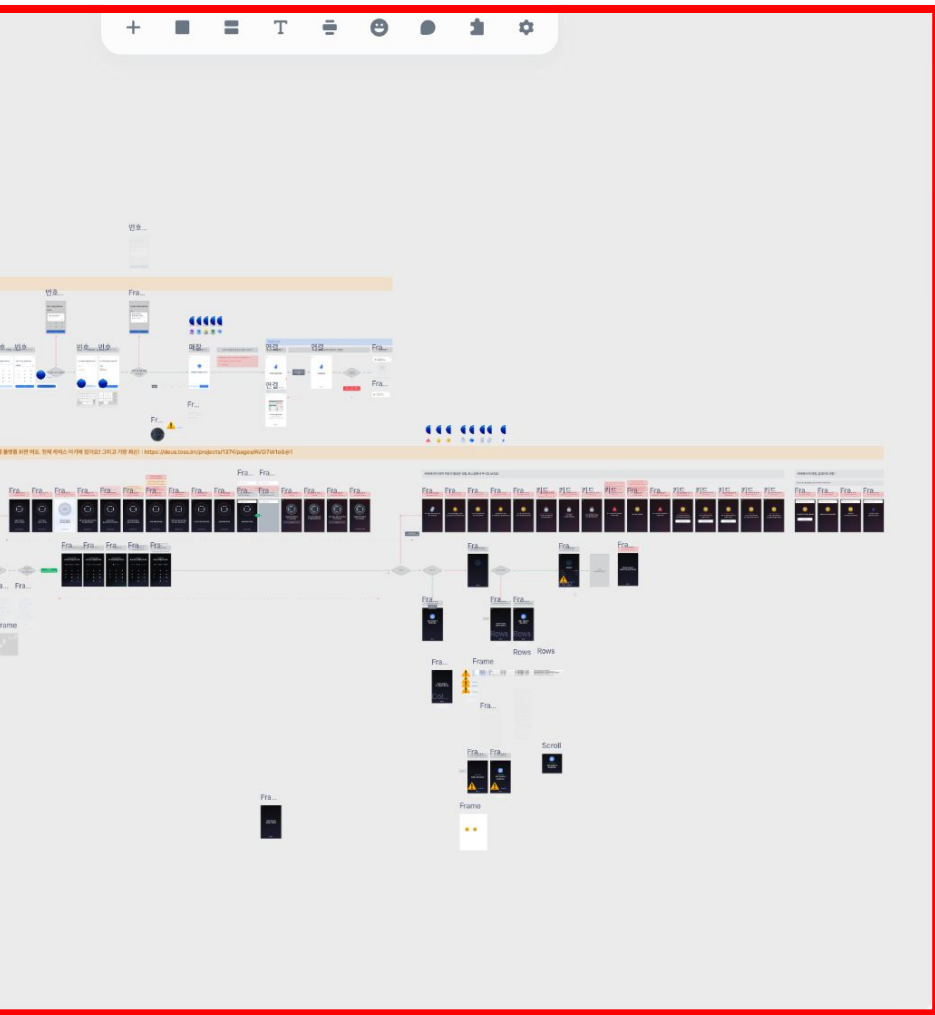
JSON Patch defines a JSON document structure for expressing a sequence of operations to apply to a JavaScript Object Notation (JSON) document; it is suitable for use with the HTTP PATCH method. The "application/json-patch+json" media type is used to identify such patch documents.

JSONPatch

```
import { applyPatch } from 'fast-json-patch';

const result = applyPatch({ a: 1, b: 1 }, [
  { path: '/c', op: 'add', value: 1 }
  { path: '/a', op: 'remove' }
]);

// result ===
{
  "b": 1
  "c": 1
}
```



- 대부분 상황에서 디자이너 혼자 작업
- 각 트리의 노드가 **Unique ID** 소유
- 간혹 삭제 액션으로 고아 노드 발생 가능성
- ➔ FE에서 고아 노드에 대한 **Fallback** 처리
- 서버에서는 메시지가 들어온 시간순으로 정렬해서 먼저 들어온 것부터 적용
- 사내 오픈 이래 치명적인 정합성 이슈 X

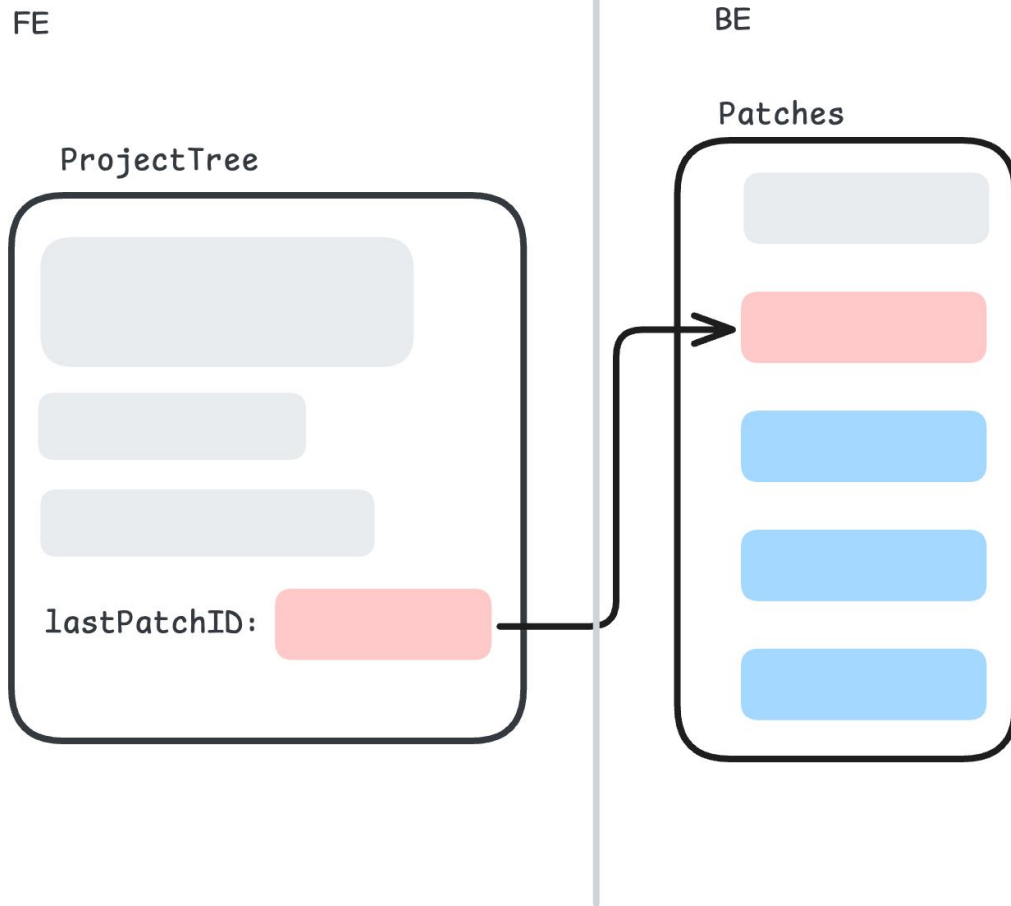
프로젝트 편집하기

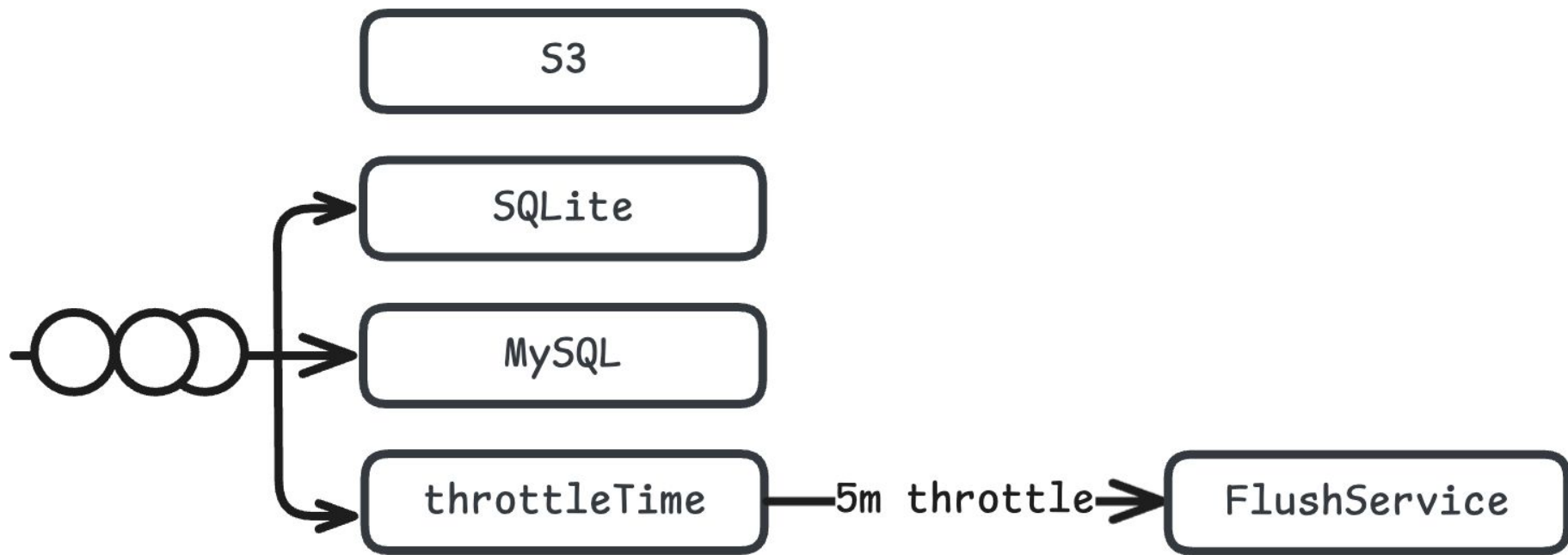
- ProjectTree: 프로젝트 전체 구조를 담고 있는 JSON 기반 트리
- ProjectPatch: Tree 일부를 수정하는 메시지

BufferedPatch

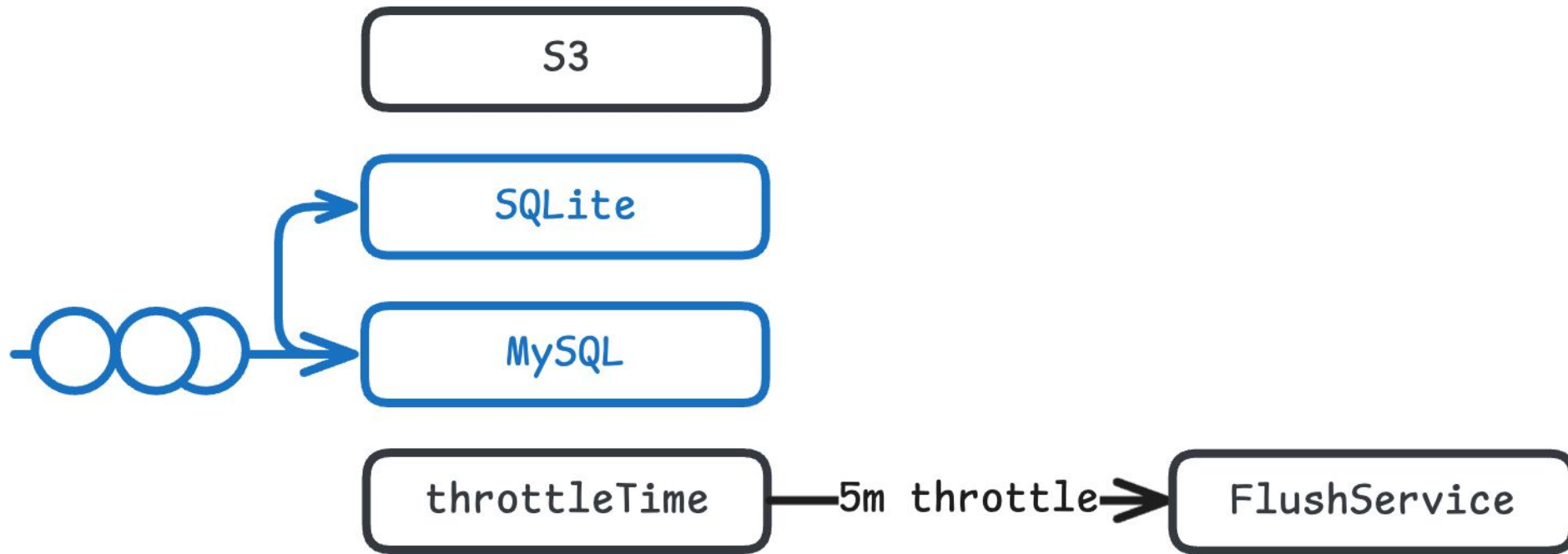
- lru-cache 라이브러리를 사용해서 **ProjectTree**들을 메모리에 로드, **JSONPatch**를 실시간으로 적용
- 하지만 **ProjectTree**가 점점 커지고, 프로젝트 갯수가 늘어남
➡ 빈번한 **tree dispose**와 **load** 발생, 성능 저하
- **JSONPatch**를 **ProjectTree**에 바로 적용시키지 않도록 수정
- 패치를 잠시 쌓아두고, 일정 시간이 지난 후 **Tree**에 **Patch**를 적용함 (**Flush**)
- **Flush** 하기 전 다른 유저가 접속하면, **BufferedPatch**와 **Tree**를 함께 내려줌

파란색 부분이
BufferedPatch

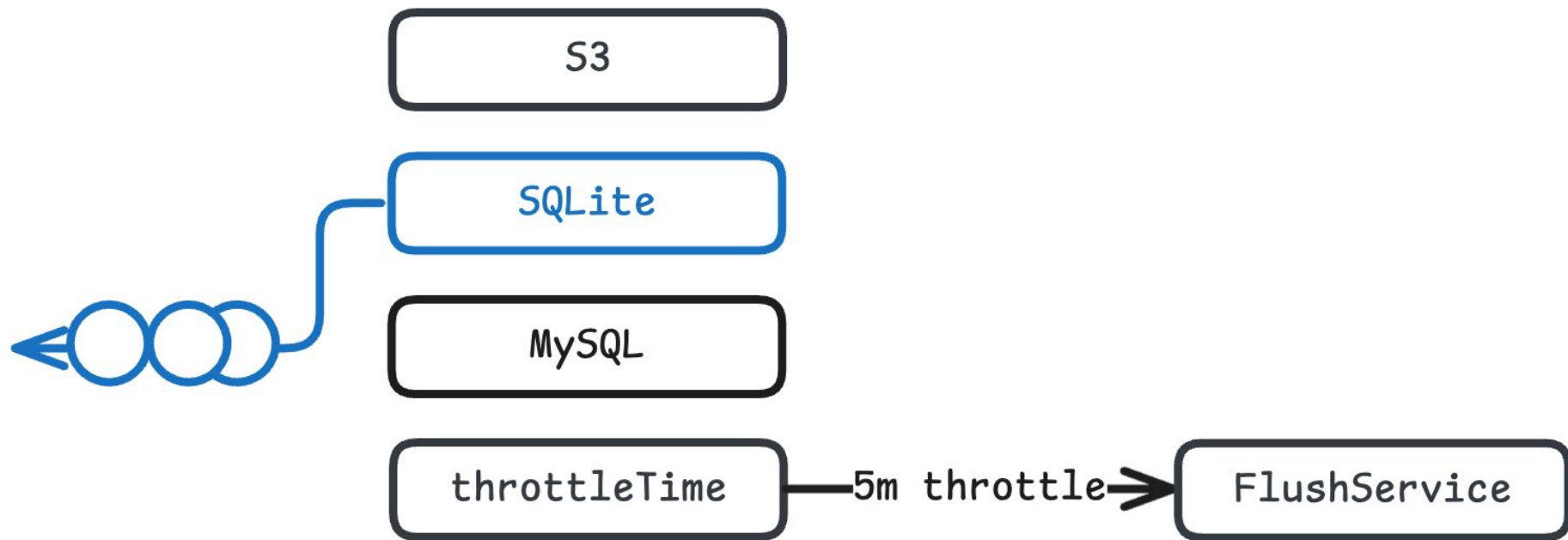




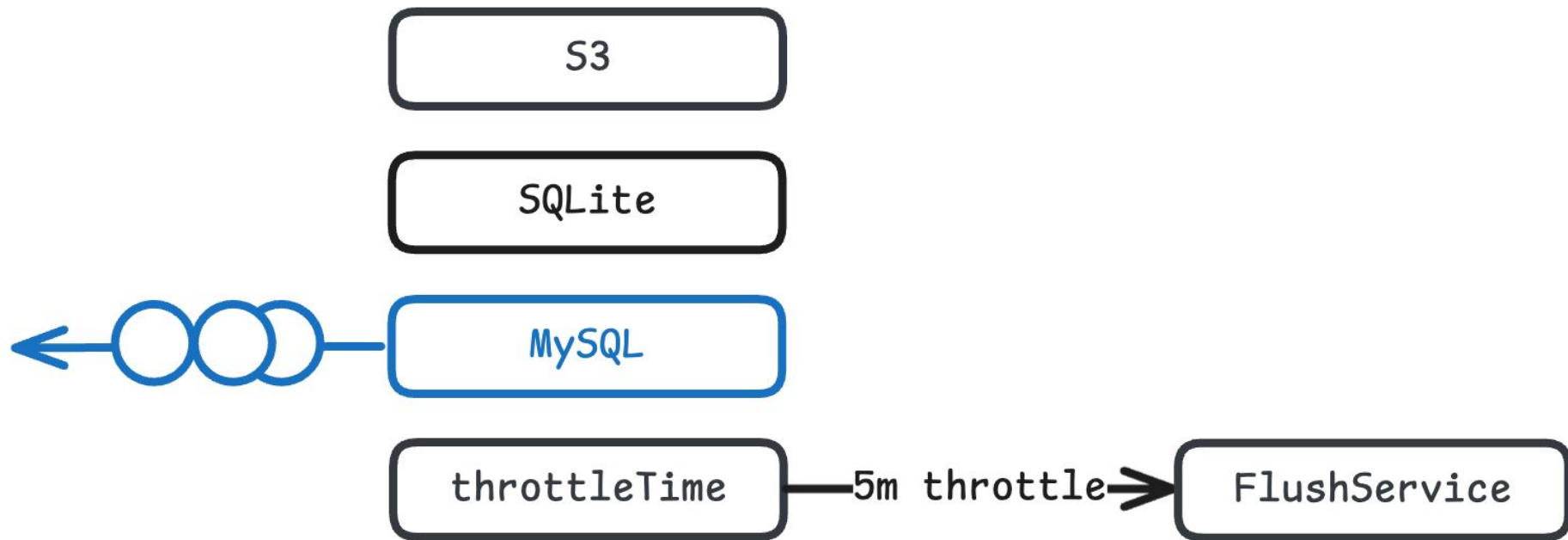
Patch 저장하기

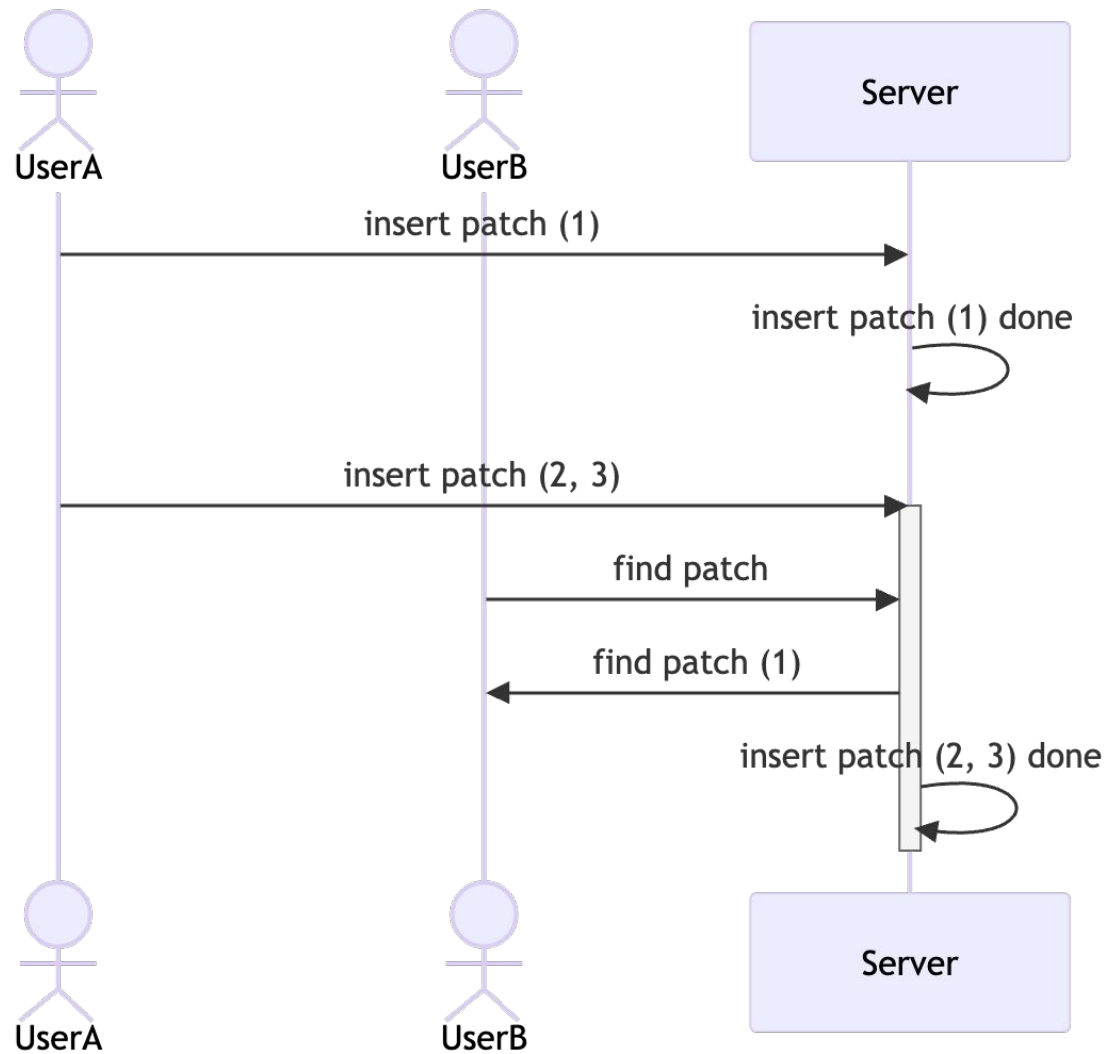


BufferedPatch 재생하기 웹소켓 접속



Patch 불러오기 기타 작업





SQLite

- 서버는 메시지를 받으면 온갖 **async listener** 들을 실행 함
- 루프나 **I/O** 지연으로 데이터 정합성에 문제가 발생할 수 있음
- 트랜잭션 **Lock**을 사용하지 않으면서 정합성 이슈를 최대한 방어하는 방향

➡ 패치를 가져오고 삽입하는 과정을 동기적으로 처리하는 설계으로 이어짐

SQLite

- Dynamic Linking을 통해 프로세스에 라이브러리로 연결된 DB
- IPC, 즉 프로세스 밖으로 통신하지 않음
- Async I/O 없음

better-sqlite3

11.10.0 • Public • Published a month ago

 [Readme](#)

 [Code](#)

Beta

 2 Dependencies

 1,698 Dependents

 139 Versions

better-sqlite3

The fastest and simplest library for SQLite in Node.js.

- Full transaction support
- High performance, efficiency, and safety
- Easy-to-use synchronous API (*better concurrency than an asynchronous API... yes, you read that correctly*)

Install

```
> npm i better-sqlite3
```



Repository

 github.com/WiseLibs/better-sqlite3

Usage

```
const db = require('better-sqlite3')('foobar.db', options);

const row = db.prepare('SELECT * FROM users WHERE id = ?').get(userId);
console.log(row.firstName, row.lastName, row.email);
```

Seeding

상황

- 서버가 처음 배포되면 **SQLite**는 완전히 백지 상태
- **MySQL**에서 각 프로젝트 별 **lastPatchID**를 기록

구현방법

- 이를 통해 어떤 패치를 마지막으로 트리에 적용했는지 확인
- 소켓 접속 시 **Fastify preHandler hook**에서 **MySQL->SQLite** 작업 수행

롤백 저널 vs. Write Ahead Log

롤백 저널

- 트랜잭션이 시작되면 원본 **DB** 파일 중 일부를 페이지 단위로 **journal** 파일로 백업
- 커밋 시 **journal** 파일을 원본 **DB**와 합침

WAL

- 데이터 변경 요청을 **WAL** 파일 끝에 기록
- 커밋 시 **WAL** 파일에 커밋 완료 기록
- 적절한 체크포인트 타이밍이 오면 **DB** 파일에 변경 사항을 적용
- 읽기 작업 시 **WAL** 파일에서 검색한 다음 없으면 **DB** 파일 검색

성능 테스트

```
console.time('took');  
sqlite.exec('begin;');  
for (const { id, project, patch, createTime } of patches) {  
  repository.add({  
    id,  
    projectID: project.id,  
    patch,  
    regTs: formatDate(createTime!, 'yyyy-MM-dd HH:mm:ss'),  
  });  
  totalCount++;  
}  
sqlite.exec('commit;');  
console.timeEnd('took');
```

수동 커밋

```
console.time('took');  
  
for (const { id, project, patch, createTime } of patches) {  
  repository.add({  
    id,  
    projectID: project.id,  
    patch,  
    regTs: formatDate(createTime!, 'yyyy-MM-dd HH:mm:ss'),  
  });  
  totalCount++;  
}  
  
console.timeEnd('took');
```

자동 커밋

성능 테스트 (실제 로직 기반)

순위	트랜잭션 모드	커밋 유형	걸린 시간
2	WAL	수동	47.855ms
3	WAL	자동	48.677ms
1	롤백 저널	수동	40.192ms
4	롤백 저널	자동	52.99ms

성능 테스트 (격리된 테스트 환경, n=50,000)

순위	트랜잭션 모드	커밋 유형	걸린 시간
2	WAL	수동	78.942ms
3	WAL	자동	780.816ms
1	롤백 저널	수동	67.493ms
4	롤백 저널	자동	30.635s

약 447배

편차의 원인

begin

commit



```
console.time('took');  
  
for (const { id, project, patch, createTime } of patches) {  
  repository.add({  
    id,  
    projectID: project.id,  
    patch,  
    regTs: formatDate(createTime!, 'yyyy-MM-dd HH:mm:ss'),  
  });  
  totalCount++;  
}  
  
console.timeEnd('took');
```

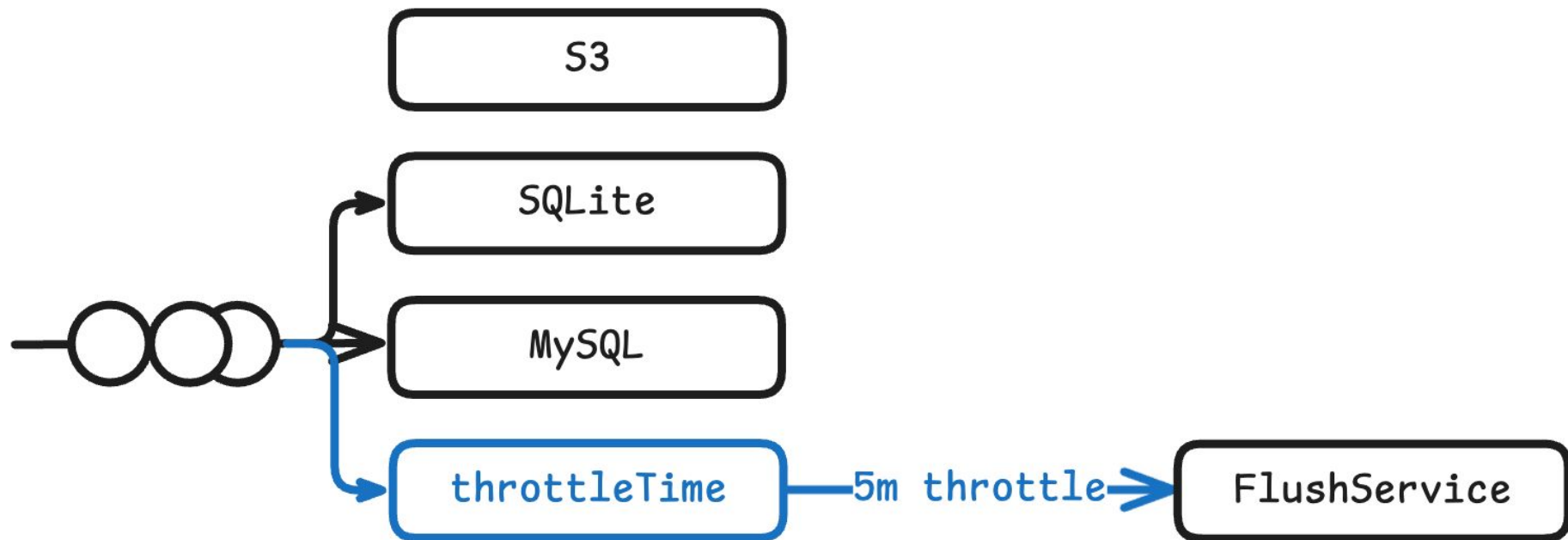
자동 커밋

Deus에 적용하기

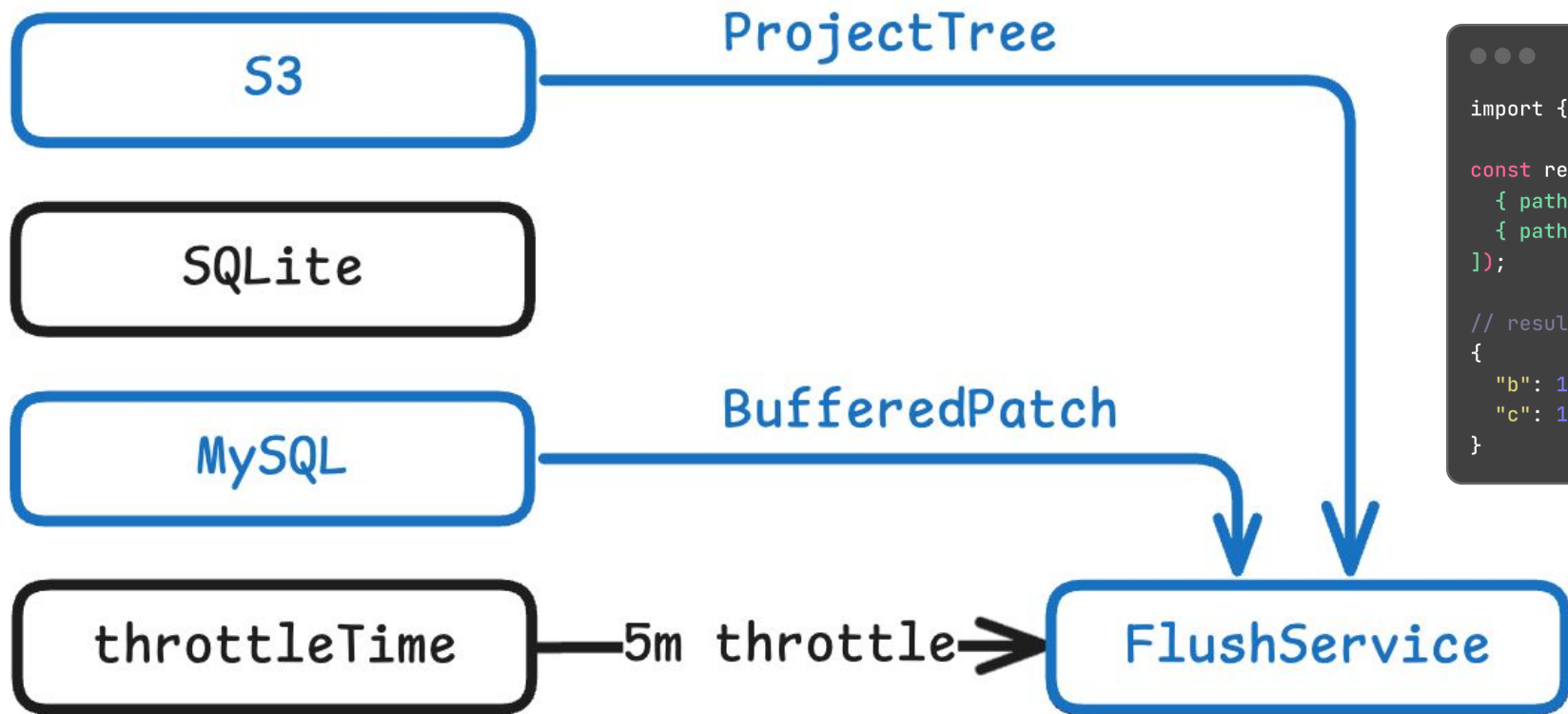
- 롤백 저널은 트랜잭션 구간을 잘 지정하면 빛을 받함
- 그러나 **seeding** 과정을 거치고 나면 데이터가 산발적으로 들어옴
- 트랜잭션 구간을 잘 잡기 어려움

➡ WAL mode + 자동 트랜잭션

Flush



Flush

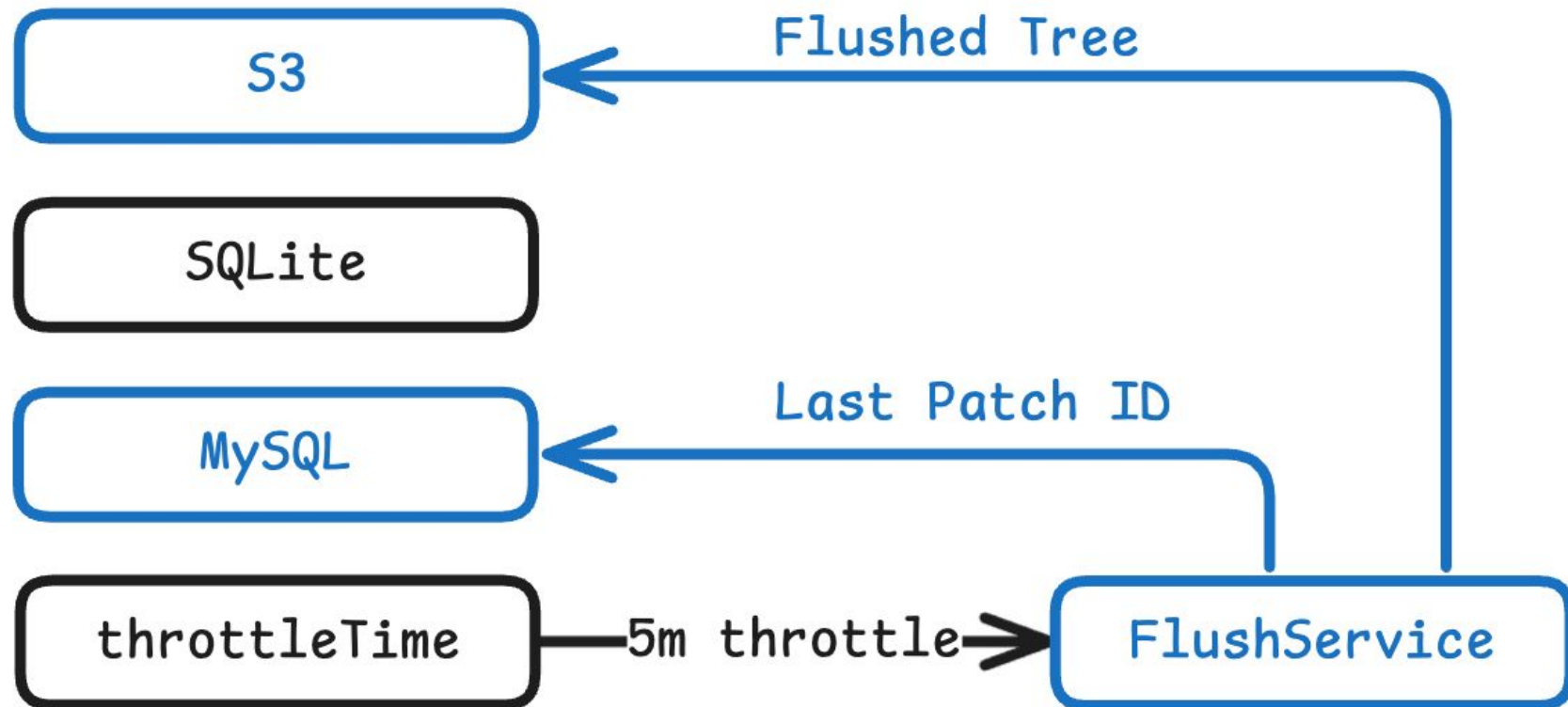


```
import { applyPatch } from 'immer';

const result = applyPatch(
  { path: '/c', op: 'push' },
  { path: '/a', op: 'push' }
);

// result ===
{
  "b": 1
  "c": 1
}
```

Flush



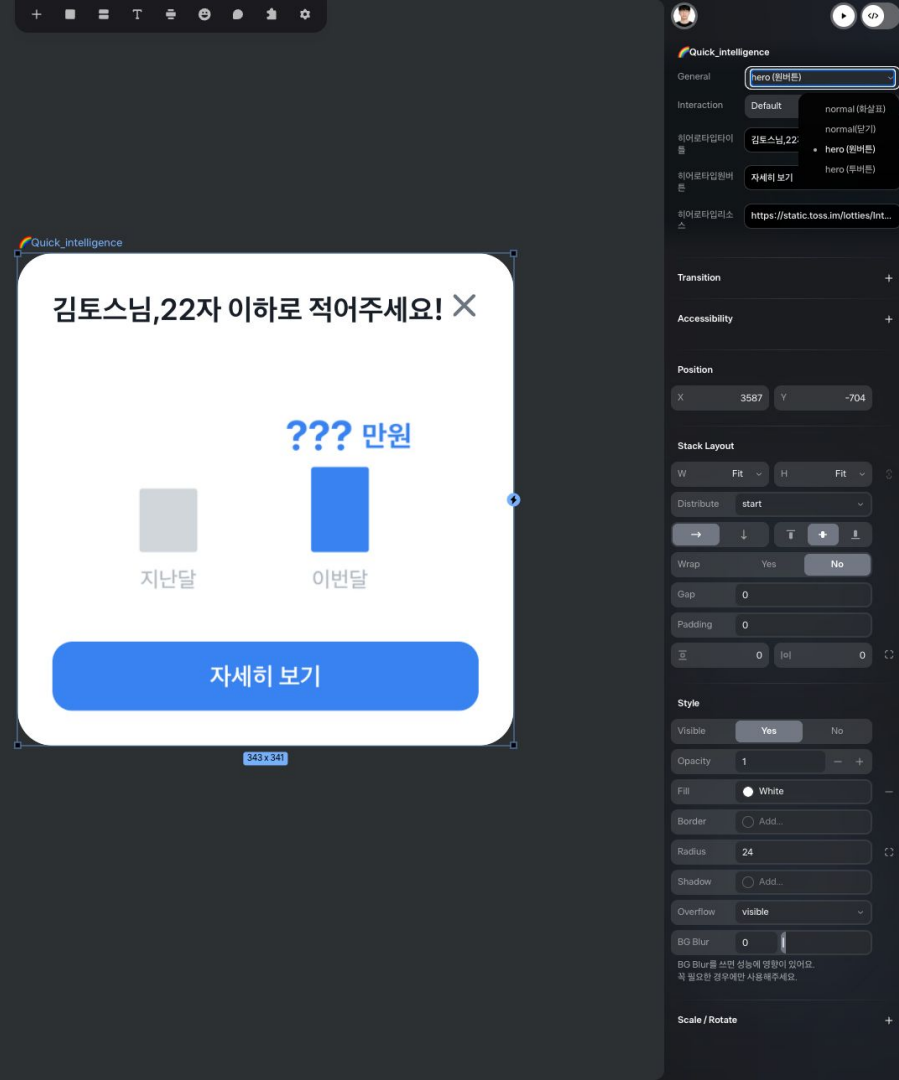


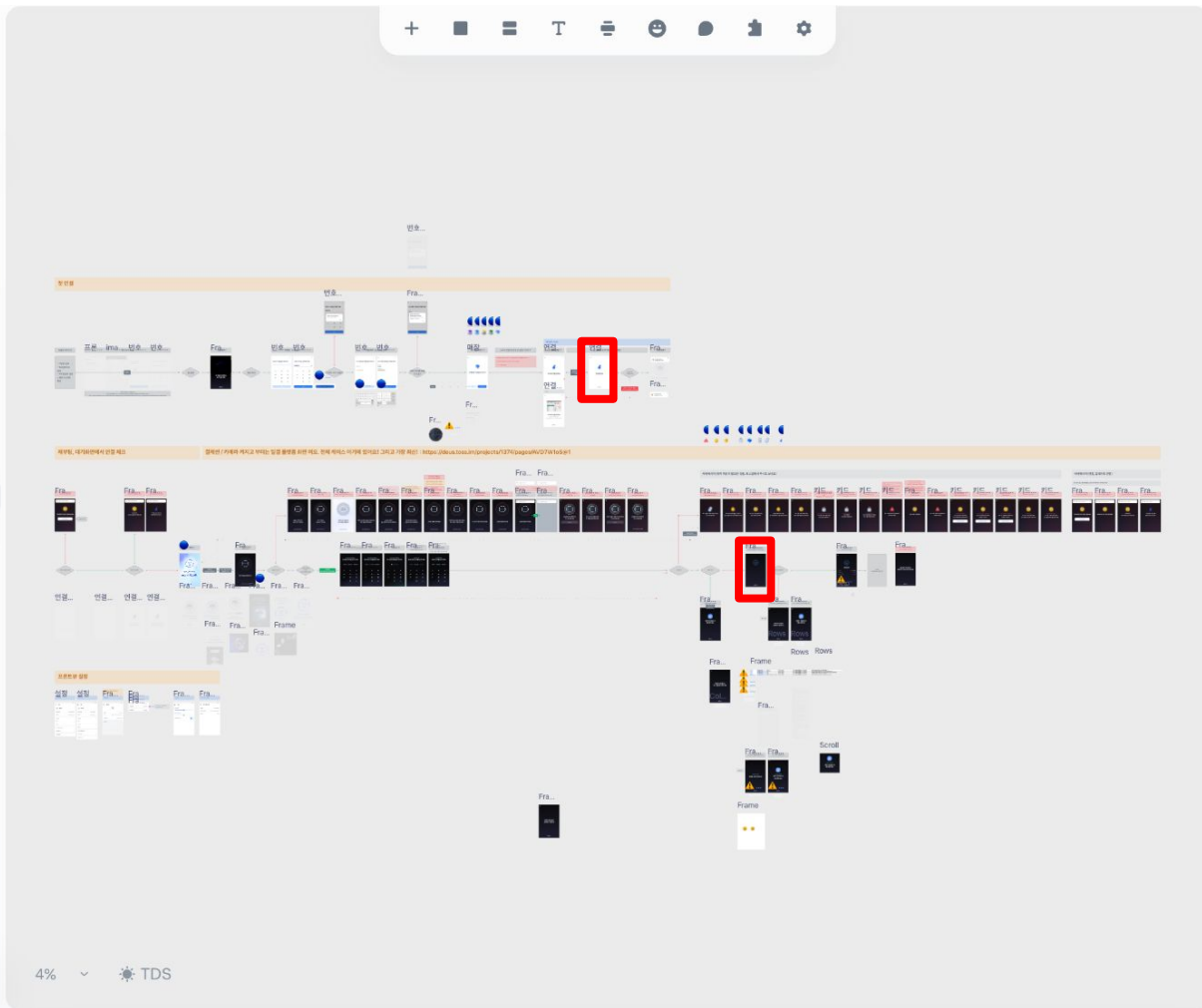
Node.js Chapter 채용

검색

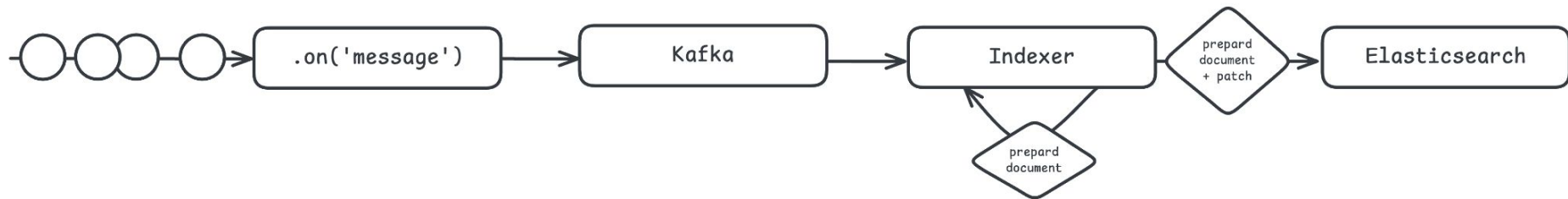
디자인 리소스의

- 버튼 색상
- 버튼 갯수
- 리소스 URL
- 문구
- ...등등등





검색



Visualize

- Kibana가 지원하는 강력한 시각화 기능을 통해, 컴포넌트 속성 분포를 손쉽게 파악하고 싶다
- Nested Property를 집계하려 했으나, Kibana에서 지원하지 않음

```
{
  "group" : "fans",
  "user" : [ 1
    {
      "first" : "John",
      "last" : "Smith"
    },
    {
      "first" : "Alice",
      "last" : "White"
    }
  ]
}
```



Nested Properties

Visualize

- Kibana가 지원하는 강력한 시각화 기능을 통해, 컴포넌트 속성 분포를 손쉽게 파악하고 싶다
- Nested Property를 집계하려 했으나, Kibana에서 지원하지 않음

Nested field support in Visualize #58175

Open

Listed in [#60126](#)



Bargs opened on Feb 21, 2020

Contributor

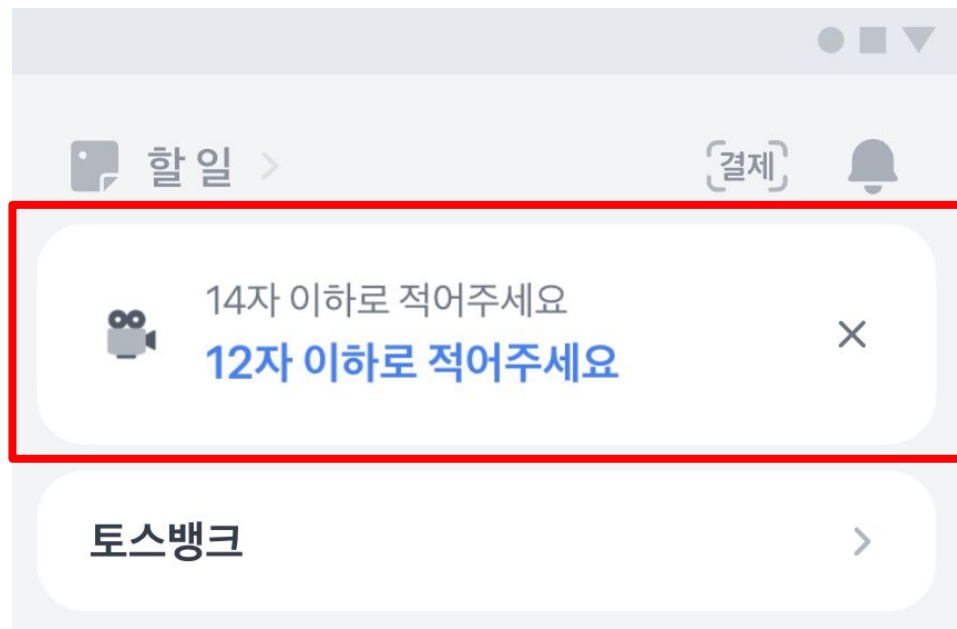


Part of [#1084](#)

Similar to querying, aggregating on nested fields requires users to make some choices depending on what info seek. Imagine doing a terms aggregation on a nested field with a simple "count" metric. For each bucket, do you want to count the number of

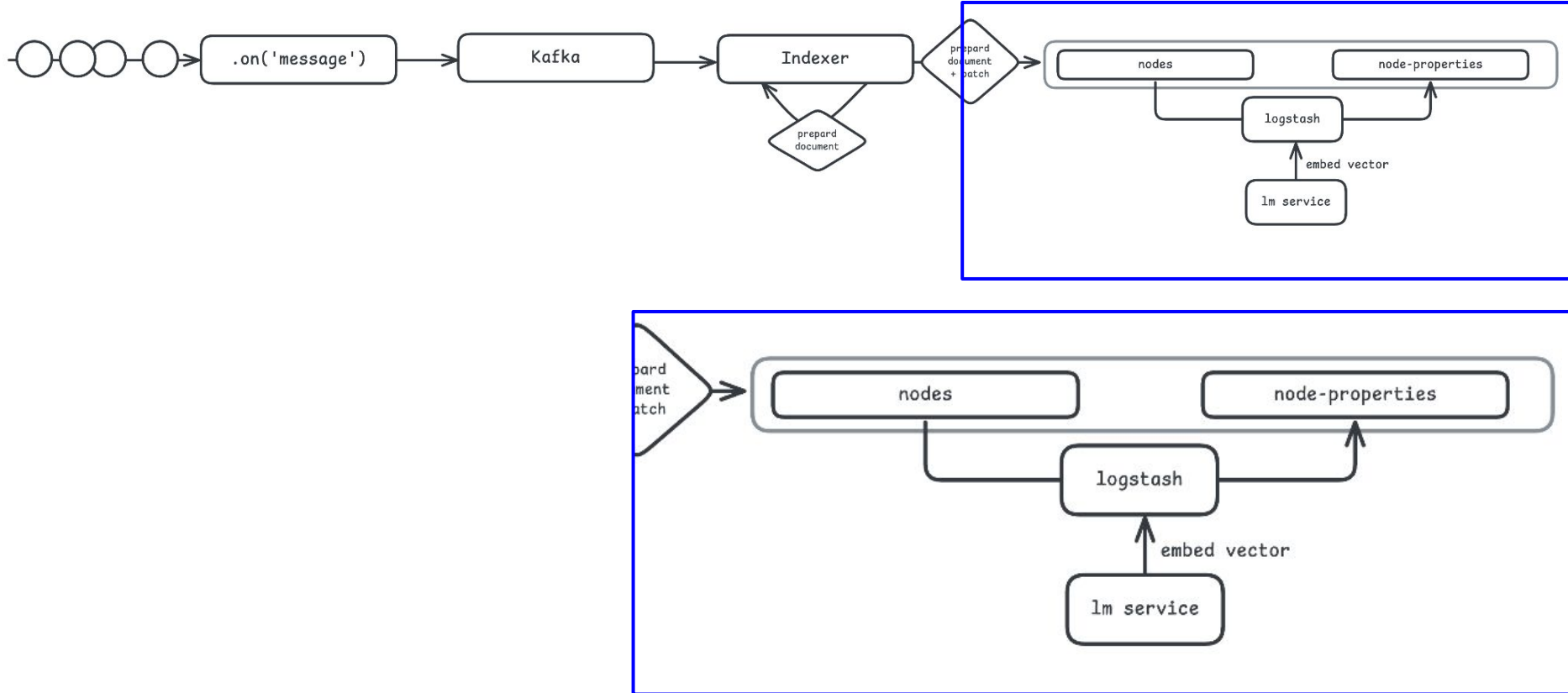
유사어 검색

- 기획자/디자이너들은 “인텔리전스”의 문구를 잘 쓰고 싶음
- 그러기 위해서 비슷한 내용의 인텔리 문구들을 찾아서, 좋은 문구를 참고 유도

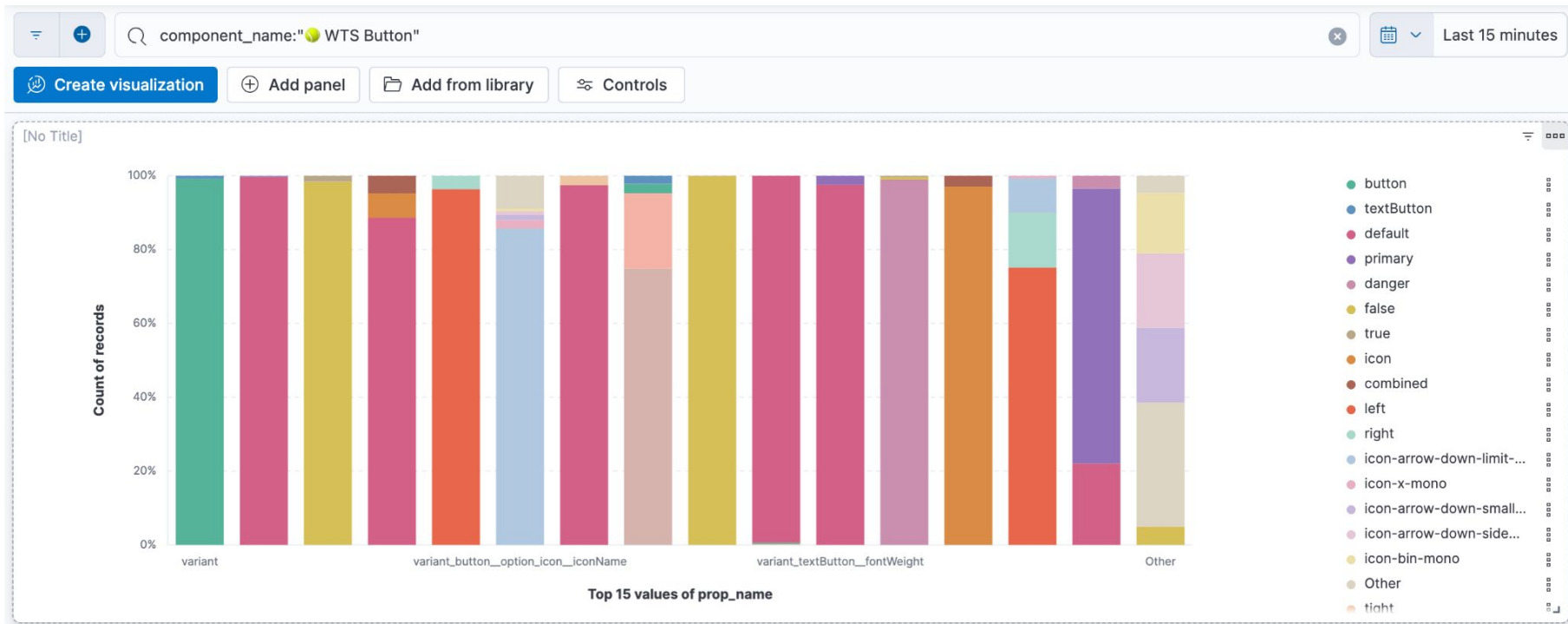


← 인텔리전스

Visualization



Kibana Visualize



@huggingface/transformers TS

3.5.2 • Public • Published 21 days ago

 Readme

 Code Beta

 4 Dependencies

 93 Dependents

 44 Versions



Transformers.js

npm v3.5.2 downloads 91k/week jsdelivr 279k/week license Apache-2.0 website online

State-of-the-art Machine Learning for the Web

Run 🤖 Transformers directly in your browser, with no need for a server!

Install

```
> npm i @huggingface/transformers
```

Repository

🔗 github.com/huggingface/transformers.js

Homepage

🔗 [github.com/huggingface/transformers...](https://github.com/huggingface/transformers.js)



Node.js Chapter 채용