

# 사물인터넷 서비스의 신뢰성 강화를 위한 오류 추적 시스템 설계

임호성<sup>1</sup>, 최창원<sup>2\*</sup>

<sup>1</sup>(주)네이버 연구원, <sup>2</sup>한신대학교 컴퓨터공학부 교수

## A Design on Error Tracking System for Enhanced-Reliable IoT Service

Ho-Seung Lim<sup>1</sup>, Chang-Won Choi<sup>2\*</sup>

<sup>1</sup>Researcher, Division of System Development, NAVER Corporation

<sup>2</sup>Professor, Division of Computer Engineering, Hanshin University

**요약** 본 논문에서는 사물인터넷 서비스 개발 과정에서 발생할 수 있는 오류에 대한 분석 및 추적, 알림 시스템을 설계하였다. 오류에 대한 세부적인 부가정보(OS, Browser, Device)들을 체계적으로 분석하고 오류가 발생한 상황을 개발자가 인지하도록 하여 시스템의 신뢰성을 강화하였다. 특히 서비스 내에서 오류가 발생한 경우 개발자가 이를 인지하기 어려운 상황을 고려하여 다양한 알림 매체(Email, Slack, SMS)를 통해서 인지할 수 있도록 하였다. 설계 및 개발 과정에서 사물인터넷 환경의 특성 상 여러 언어(Typescript, Swift, Android)에 대해 수용할 수 있도록 공용 인터페이스를 고안했으며, 이 인터페이스를 활용하여 모바일, 웹, 데스크톱 어플리케이션 등 여러 사물인터넷 시스템에서 발생한 오류를 수용해서 분석할 수 있다. 또한 각각의 오류 이슈를 통해 집합으로 표현함으로써 개발자가 서비스 개발 시 어떤 이슈가 있는지 쉽게 파악할 수 있으며 대시보드를 통해서 발생한 오류들을 인지할 수 있는 시각화 기능을 제공하였다.

**주제어** : 사물인터넷 시스템, 디바이스, 플랫폼, 오류 추적 기능

**Abstract** In this paper, an error tracking platform is designed for enhanced-reliable IOT system. The platform is designed to enhance reliability of IOT system by analysing additional informations(OS, Browser, Device) and by notifying error detection to developers. Especially, in the case of an error in the service which it is difficult for developers to recognize it, The platform also supports notification services through various communication media(Email, Slack, SMS). The common interface is designed to accommodate many languages(typescript, Swift, and Android) in the development process, and the interface allows users to analyze errors that occur on various platforms, including mobile/web/desktop applications. By presenting each error in groups through issues, developers can easily identify issues in the service. The visualizing function is included to recognize various error type by dashboard.

**Key Words** : IOT System, IOT Device, IOT Platform, Error Tracking

### 1. 서론

사물인터넷은 컴퓨터화된 사물들이 연결된 네트워크

를 말한다. 따라서 사물인터넷 시스템 역시 기존의 인터넷 시스템과 같은 구조를 갖고 있다. 즉, C-P-N-D로 콘텐츠, 플랫폼, 네트워크, 디바이스로 구성된다. 차이가 있

\*교신저자 : 최창원(wonya1003@gmail.com)

접수일 2020년 8월 10일 수정일 2020년 9월 2일 심사완료일 2020년 9월 16일

다면, 디바이스가 컴퓨터가 아니라 사물인터넷 디바이스라는 것이며 콘텐츠는 콘텐츠를 포함한 서비스로 대체된다. 따라서 사물인터넷의 구조는 S-P-N-D라고 할 수 있다[1,2]. 이러한 환경에서 서비스나 플랫폼을 개발하고 운영하는 동안 다양한 오류가 발생하였다고 피드백을 줄 때가 있다. 이 경우 개발자는 어느 때에 정확히 어떤 오류가 발생한 건지 알기 힘들다는 것이다. 또한, 어떠한 프로젝트를 진행 하고 수많은 테스트를 거쳐서 제품을 만들어서 배포를 시작하면 많은 사용자들이 사용하면서 실행은 되지만 각자의 환경에서 예기치 못한 오류들(시스템 환경, 404 Error, port 설정, null point등)이 발생하여 개발자가 일일이 확인하기 힘든 상황들이 종종 발생한다[4].

본 논문에서는 사물인터넷 환경의 시스템 개발자가 어떠한 프로젝트를 진행할 때 발견되는 오류들을 다양한 형태로 인식하여 신속히 대처하고 다른 팀원들과 대시보드를 이용하여 오류에 대한 피드백을 쉽게 하여 전체 프로젝트의 완성도를 높이고 비용을 줄일 수 있는 플랫폼을 설계하였다.

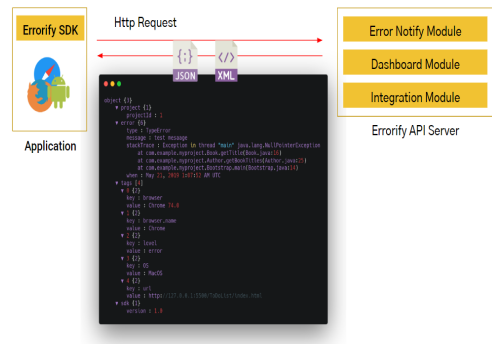
오류 추적 시스템은 다수의 솔루션들이 개발되어 운영되고 있다. SENTRY는 오픈소스를 제공하여 설치가 용이하고 모니터링이 가능하고 다양한 언어와 프레임워크를 위한 라이브러리를 지원한다는 것이 장점이지만 유료이고 최근의 오류를 따로 볼 수 없고, 검색기능이 약하며 따로 설정을 하지 않으면 이메일 알림이 범람해진다는 단점이 있다[4,5]. Track JS는 모바일에 대한 지원도 제공하고 인터페이스가 훌륭하며 실시간으로 오류보고가 가능한데 오류 보고에서 매우 유익한 스택 추적 및 철저한 세부 환경에 대한 정보를 제공해준다. 단점으로는 유료 서비스이며 오류를 분류하거나 오류들에 우선순위를 부여할 수 없고 검색 기능이 상대적으로 취약하고 오류에 대한 문서화 기능이 떨어진다.

본 논문에서 설계한 시스템은 특정오류 형태나 브라우저를 활용한 키워드 필터링을 제공하여 알림을 받을 수 있게 하여 간단한 오류로 유발되는 메일 범람을 방지하였다. 또한 전체 스택 추적뿐 아니라 오류가 발생한 시점의 기기 특성, 성능 데이터, 사용자 상황에 대한 정보를 제공한다. 그리고 가장 최근의 오류를 보여주어 여러 프로젝트를 개발할 때 설정/구성이 편리해지며 검색 기능을 추가하여 특정 시간, 특정 오류 메시지로 범위를 좁혀가면서 원하는 검색을 할 수 있게 하였다.

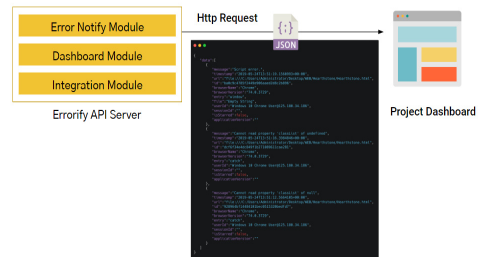
## 2. 오류 추적 플랫폼 설계

### 2.1 인터페이스 설계

인터페이스 설계에서 통신 규약 구간은 API Server와 Application 간의 통신(Fig.1), API Server와 Error Notify 매체 간의 통신 구간이다(Fig.2). 먼저, Application SDK에서 API Server로 정보를 보낼 때, API Server에서 Error Notify 매체로 정보를 보낼 때 모두 JSON 형식으로 데이터를 가공해서 보내게 된다[7,8].



[Fig. 1] System Interface(1)



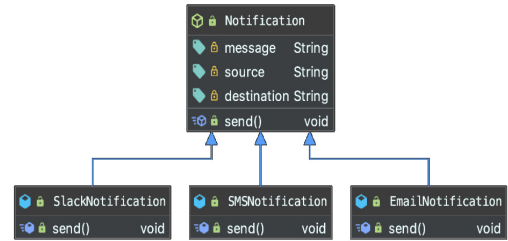
[Fig. 2] System Interface(2)

### 2.2 DB 설계

DB는 효율적인 오류 추적이 가능하도록 <Table 1>처럼 User, Comment, Tag, Team, Project, Activity, Error 등 [Fig. 3]처럼 7개의 엔티티로 구성하여 설계하였다[13].

<Table 1> Entity List

ENTITY	
USER	User Information
COMMENT	Comment Data for Team Member of Project
TAG	Tag Information of Error Types
TEAM	Team Information
PROJECT	Project Entity
ACTIVITY	User Activity Entity
ERROR	Error Type Entity

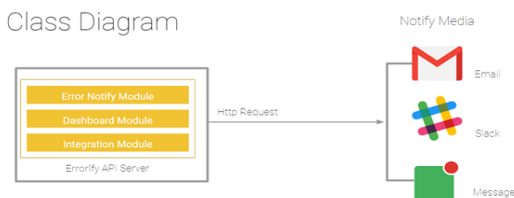


[Fig. 5] Class Diagram(2)



[Fig. 3] DB Structure

시스템 설계에 사용하는 클래스는 Email, Slack, SMS, Notification 등 4 개의 클래스로 정의하였다. Notification 클래스는 다른 Notify Media가 추가되더라도 이전 모듈에 영향을 주지 않고 쉽게 추가할 수 있도록 클래스를 상속받은 서브 클래스로 설계하였다. [Fig.4]와 [Fig.5]는 각 클래스들의 상관 관계를 다이어그램으로 나타냈다.



[Fig. 4] Class Diagram(1)

### 2.3 SDK

[Fig. 6]은 Application SDK에서 API Server로 데이터를 보낼 때 설계한 JSON 형식이다. project 칼럼에서 project ID를 가공해서 전달하고, error 칼럼에서 type, line, message, stackTrace, when을 전달한다. user 칼럼에서는 userIp, userName을, tag 칼럼에서는 key, value 형식으로 환경에 대한 정보를 전달하게 된다. 대시보드에서는 서버 쪽에 복수개의 요청을 할 수 있으며 그림에서는 대시보드에서 오류 정보만을 요청하여 오류 정보들을 모아 배열로 데이터를 보낸다. 이렇게 여러 객체에 대한 정보를 JSON형식으로 필요한 데이터만 뽑아내어 가공한 후 전달한다[11].

```

object {3}
  project {1}
    projectId : 1
  error {7}
    type : TypeError,
    line : 7,
    message : test message,
    stackTrace : Exception in thread "main" java.lang.NullPointerException
      at com.example.myproject.Book.getTitle(Book.java:16)
      at com.example.myproject.Author.getBookTitles(Author.java:25)
      at com.example.myproject.Bootstrap.main(Bootstrap.java:14),
    when : May 21, 2019 1:07:52 AM UTC
  user {2}
    userIp : 203.252.17.149,
    userName : hosu Lee
  tags {4}
    0 {2}
      key : browser,
      value : Chrome 74.0
    1 {2}
      key : browser.name,
      value : Chrome
    2 {2}
      key : level,
      value : error
    3 {2}
      key : OS,
      value : MacOS
    4 {2}
      key : url,
      value : http://127.0.0.1:5500/ToDoList/index.html
  sdk {1}
    version : 1.0
    
```

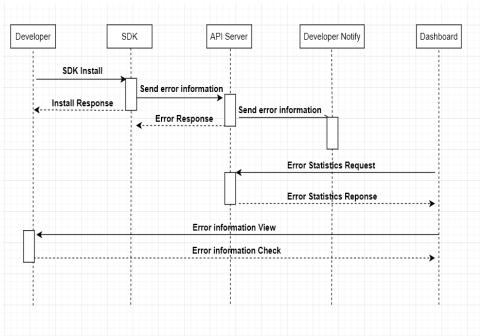
[Fig. 6] SDK JSON Data

### 2.4 시스템 동작 시퀀스

제한한 시스템의 동작 시퀀스는 [Fig. 7]처럼 개발자가 Application SDK를 설치하고 개발 작업 도중 오류

가 발생하게 되면 오류 정보들이 API Server로 전달되고, 그 오류 정보들을 Error Notify Media로 가공해서 전달하게 된다. 모든 객체간의 상호작용은 http Request, http Response 형식으로 동작되도록 설계하였다.

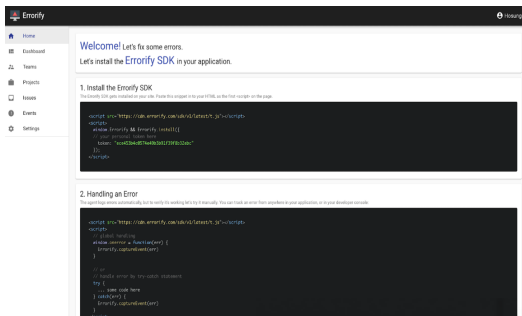
SDK 설치 중 발생한 오류들은 API 서버에 전달되어 다른 개발자(프로젝트 팀원들)들이 오류의 발생 여부를 인식하게 되고 이 상황을 대시보드 등을 통해 오류의 유형이나 통계 정보 등을 확인하여 다시 API 서버에 전송하여 오류 처리에 대한 피드백을 제공한다. 오류 발생에 대한 상황은 개발자(다른 개발자 포함), APT 서버 및 대시보드 전반에 걸쳐 통지되어 일관된 오류 추적을 가능하게 하였다.



[Fig. 7] Sequence Diagram

### 3. 오류 추적 시스템 동작

사용자 인터페이스는 원활한 UI를 구현하기 위해 storybook.js를 사용하였고 디자인 측면을 고려하여 Material-UI를 사용하였다. 사용자 가입은 github과 연동하여 로그인을 할 수 있게 하였고 저장된 아이디로 로그인할 수 있다. [Fig. 8]은 설계된 시스템의 메인 페이지이다.



[Fig. 8] System Main Page

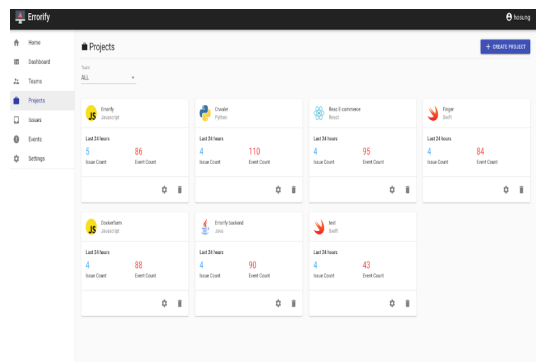
메인 페이지에선 좌측에 시스템 메인 메뉴를 배치하였고 SDK 설치 방법과 error handling 방법에 대한 페이지를 중앙에 배치하였다.

오류에 대한 전체적인 통계를 출력하도록 대시보드 기능([Fig. 9])을 제공하였으며 오류를 시간별로 보여주는 그래프(상단)와 오류에 대한 환경(IE, Chrome, Firefox 등)에 대해 출력하는 파이 차트(하단), 그리고 최근에 발생한 오류에 대해 테이블 형식으로 출력한다.

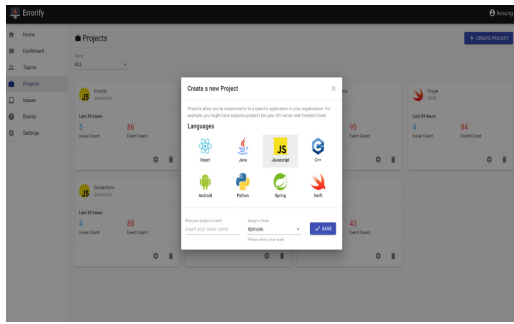


[Fig. 9] Dashboard

프로젝트 페이지는 [Fig. 10]처럼 개발자가 현재 수행하고 있는 프로젝트들의 정보를 출력하도록 설계하였다. 또한 각 프로젝트별로 최근 발생한 오류들의 정보와 전체 프로젝트의 오류 정보를 종합적으로 출력하게 하여 효율적인 프로젝트 수행을 할 수 있도록 하였다([Fig. 11]).

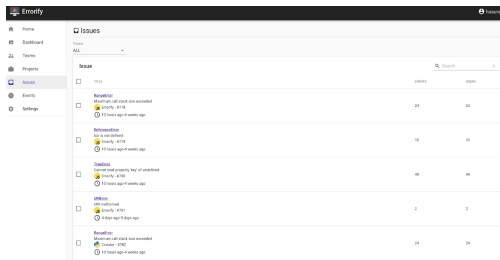


[Fig. 10] Project Page

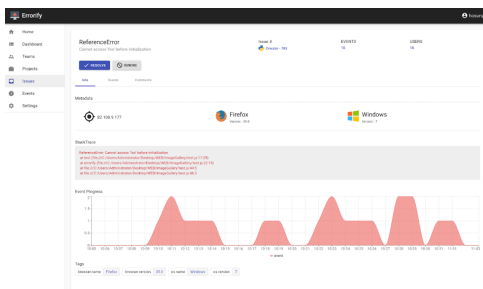


[Fig. 11] Project Team Modal

[Fig. 12]의 이슈 페이지는 발생한 오류들을 분류하여 오류의 유형대로 그룹 단위로 출력하도록 하였다. 이 기능은 개발자와 팀원들이 발생하는 오류의 유형을 쉽게 인지할 수 있고 오류의 발생이 시스템 전반에 영향을 주는 부분들을 체계적으로 인식하여 적절한 대처가 가능하도록 한다[15].



[Fig. 12] Issue Page



[Fig. 13] Issue Page(Detailed)

오류에 대한 수정/보완 작업을 수행하기 위해서 각 이슈별로 상세 페이지를 [Fig. 13]처럼 제공하였다. 이 기능은 발생한 오류가 어떤 환경(browser, os, language)에서 발생한 것인지를 시각화하여 아이콘 형태로 출력하는 부분과 error capture 내용, 발생한 error에 대한 시

각적 그래프로 구성하였다.

이벤트 페이지는 개발자가 진행하고 있는 프로젝트에 대해 모든 오류들을 출력한다. [Fig. 14]처럼 어떤 프로젝트에서 발생한 오류인지 구별해주고, 해당 기기의 IP 주소와 발생 시간을 출력한다. 또한 특정 키워드로 검색할 수 있는 기능을 제공하여 오류에 대한 정확한 정보를 제공한다.

Name	Project	IP	Time
Application call stack size exceeded	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid code point in string	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid code point in URI	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid code point in path	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid array length	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid argument must be between 0 and 100	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid argument must be between 0 and 100	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid argument must be between 1 and 100	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid argument must be between 1 and 100	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid argument must be between 0 and 26	Emory	194.27.247.10	2019-11-18 04:00:01
Invalid event value	Emory	194.27.247.10	2019-11-18 04:00:01
Application call stack size exceeded	Emory	194.27.247.10	2019-11-18 04:00:01

[Fig. 14] Event Page

## 4. 결론

본 논문에서는 사물인터넷 서비스를 개발하고 운영하면서 발생하는 오류들의 다양한 정보들을(OS, Browser, Device)를 분석하여 오류가 발생하는 상황을 개발자가 그대로 재현할 수 있는 기능을 제공해주고, 오류 발생 시 개발자가 바로 인식할 수 있도록 여러 알림매체(Email, Slack, SMS)를 통해서 알림 서비스를 제공하는 오류 추적 시스템을 설계하고 개발하였다. 제안된 시스템은 서비스를 개발하고 운영하는데 있어 필연적으로 발생할 수밖에 없는 오류를 하나의 관리대상으로 정의하여 분석, 알림 서비스를 개발자들에게 제공함으로써 시스템의 신뢰성을 강화할 수 있다.

오류 수집에는 다양한 언어(Typescript, Swift, Android)로 개발된 시스템에서 발생하는 오류들을 모두 수용할 수 있도록 공용 인터페이스를 설계하고 활용하였다. 이 인터페이스에 맞게 오류 정보를 설계된 시스템에 전달하면 이를 분석하고 개발자의 요구에 맞도록 가공할 수 있다. 분석된 데이터는 오류에 종류에 따라 이슈별로 묶여지고 개발자는 모든 오류를 일일이 확인할 필요 없이 이슈에 대해서 확인하고 다른 개발자들과 처리할 수 있다.

설계된 시스템은 기존의 솔루션과 비교하였을 때 다음과 같은 기능들이 개선되었다.

- 1) 특정 오류 유형 등을 키워드로 필터링해서 사용자 정의로 알림을 받을 수 있게 하여 작은 오류들로 인해 대량의 메일 알림을 방지하였다.
- 2) 전체 스택 추적뿐 아니라 오류가 발생한 시점의 기기 특성, 성능 데이터, 사용자 상황을 포함시켜 오류에 대한 관리가 효과적으로 수행될 수 있다.
- 3) 이슈 관리 기능을 추가하여 gitHub나 Trello와 연동하여 체계적인 오류 관리를 가능케 하였다.
- 4) 오류에 대한 시간적인 정보를 제공하여 다수의 프로젝트 작업 시 설정 / 구성이 편리해졌다.
- 5) 오류 로그를 텍스트 형식으로 형식화하였다.

향후에 VCS(Version Control System)과 연동하거나 PMS(Project Management System)과 연동하는 기능이 추가된다면 특정 버전에서 발생한 오류인지를 추적하는 기능과 운영 중에 발생한 오류들을 그룹화해서 PMS상에 이슈로 등록해주는 기능 등을 추가할 수 있어 시스템의 활용도가 크게 증가될 것이다.

## REFERENCES

- [1] H.Kim, Internet of Things(Concepts, Implementation and Business), Hongrung Publishing, 2018.
- [2] C.Lee and C.Choi, "A Simple Cost Analysis of Host ID-LOC Separating protocol using SDN Features," JKIoT, Vol.2, No.4, pp.41-47, 2016.
- [3] K.Lee, "A Design on Learning Model using Triz on Project-based Learning in IOT," JKIoT, Vol.5, No.3, pp.29-35, 2019.
- [4] S.Lee, "An Analysis of Software Development Process Based on Software Engineering in IOT Environment," JKIoT, Vol.6. No.1, pp.25-31, 2020.
- [5] A.Montazerolghaem and M.Yaghmaee, "Load-balanced and QoS-aware Software-defined Internet of Things," IEEE Internet of Things Journal, DOI 10.1109/JIOT.2020.2967081, 2020.
- [6] <https://www.opennetworking.org>
- [7] JavaScript(<https://ko.wikipedia.org/wiki/JavaScript>)
- [8] TypeScript(<https://ko.wikipedia.org/wiki/TypeScript>)
- [9] React.js(<https://ko.wikipedia.org/wiki/React.js> )
- [10] Node.js(<https://ko.wikipedia.org/wiki/Node.js>)
- [11] JSX(<https://ko.wikipedia.org/wiki/React.js>)
- [12] GraphQL(<https://ko.wikipedia.org/wiki/GraphQL>)
- [13] MariaDB(<https://ko.wikipedia.org/wiki/MariaDB>)
- [14] Docker(<https://ko.wikipedia.org/wiki/Docker>)
- [15] Git(<https://ko.wikipedia.org/wiki/git>)

임 호 성(Ho-Sung Lim)

[정회원]



- 2020년 2월 : 한신대학교 컴퓨터 공학과 졸업(학사)
- 2020년 3월 ~ 현재 : ㈜네이버

<관심분야>

사물 인터넷, 소프트웨어 공학

최 창 원(Chang-Won Choi)

[중신회원]



- 1990년 2월 : 고려대학교 전산과 학과 졸업(학사)
- 1992년 2월 : 고려대학교 전산과 학과 졸업(석사)
- 1995년 8월 : 고려대학교 전산과 학과 졸업(박사)
- 1996년 ~ 현재 : 한신대학교 컴퓨터공학부 교수

<관심분야>

유무선 네트워크, 시스템 분석, 사물 인터넷