Bugshot: Real-time Error Monitoring System

Final Project Report

Course: Object Oriented Programming (Prof. Bong-Soo Sohn) **Team Members**: 김덕환 (20200477), 정은재 (20226495) **Project**: #4

(a) Project Summary

Project Title

Bugshot - Real-time JavaScript Error Monitoring System

Team Members

Name	Student ID	Role
김덕환	20200477	Team Leader, Backend Development
정은재	20226495	Backend Development, Documentation

Brief Description

Bugshot is a backend API server that monitors JavaScript errors from web applications in real-time. The system provides:

- **Error Collection**: SDK sends error data to the backend via REST API
- **Intelligent Grouping**: SHA-256 hash-based error deduplication
- **Priority Calculation**: Automatic severity scoring based on page importance and occurrence frequency
- **Multi-channel Notifications**: Discord, Slack, Email, Telegram, Kakao Work integration
- **Session Replay Storage**: Cloudflare R2 integration for user session recording

Live Demo: https://bugshot-api.log8.kr/swagger-ui/index.html

(b) How to Compile and Execute

But I reccomend you to visit my frontend-website(https://bugshot.log8.kr) and backend-website(https://bugshot-api.log8.kr/swagger-ui/index.html)

System Requirements

Component	Requirement
Java	JDK 21 or higher
Build Tool	Gradle 8.x (wrapper included)
Database	MySQL 8.0+

Component	Requirement
Cache	Redis 6.0+ (optional)
OS	Windows 10+, macOS, Linux

How to Compile

Windows: gradlew.bat build -x test

Linux/Mac: ./gradlew build -x test

Note: -x test flag skips tests as they require MySQL connection.

How to Execute

Option 1: Use Live Demo (Recommended) Visit https://bugshot-api.log8.kr/swagger-ui/index.html to test the API without any setup.

Option 2: Run JAR File java -jar build/libs/bugshot-0.0.1-SNAPSHOT.jar

Option 3: Run with Gradle ./gradlew bootRun

Verify Execution

- Swagger UI: http://localhost:8081/swagger-ui.html
- Health Check: http://localhost:8081/actuator/health

(c) Description on Functionality

API Endpoints

1. Authentication (AuthController)

Method	Endpoint	Description
POST	/api/auth/oauth	OAuth login (GitHub, Google)
GET	/api/auth/me	Get current user info
GET	/api/auth/usage	Get usage statistics

2. Project Management (ProjectController)

Method	Endpoint	Description
GET	/api/projects	List all projects
POST	/api/projects	Create new project
GET	/api/projects/{id}	Get project detail
PUT	/api/projects/{id}	Update project

Method	Endpoint	Description
DELETE	/api/projects/{id}	Delete project
POST	/api/projects/{id}/regenerate-key	Regenerate API key

3. Error Management (ErrorController)

Method	Endpoint	Description
GET	/api/errors	List errors (with filters)
GET	/api/errors/{id}	Get error detail
PUT	/api/errors/{id}/resolve	Mark as resolved
PUT	/api/errors/{id}/ignore	Mark as ignored
PUT	/api/errors/{id}/reopen	Reopen error

4. Error Ingestion (IngestController) - Core Feature

Method	Endpoint	Description
POST	/api/ingest	Receive error from SDK
GET	/api/ingest/health	Health check

5. Dashboard (DashboardController)

Method	Endpoint	Description
GET	/api/dashboard/stats	Get statistics (Redis cached)
GET	/api/dashboard/trends	Get error trends

6. Webhook Configuration (WebhookController)

Method	Endpoint	Description
GET	/api/webhooks	List webhooks
POST	/api/webhooks	Create webhook
PUT	/api/webhooks/{id}	Update webhook
DELETE	/api/webhooks/{id}	Delete webhook
POST	/api/webhooks/{id}/test	Test webhook

7. Session Replay (SessionReplayController)

Method	Endpoint	Description

Method	Endpoint	Description
GET	/api/replays/{errorId}	Get session replay
GET	/api/replays/{errorId}/download-url	Get download URL

Key Features

1. **Error Deduplication**: Groups identical errors using SHA-256 hash of (errorType + filePath + lineNumber)
2. **Priority Calculation**: Calculates severity based on:
   - Page importance (checkout: 10x, login: 8x, homepage: 5x)
   - Error type weight (TypeError: 2.5x, NetworkError: 1.5x)
   - Occurrence frequency (logarithmic scale)
   - Recency boost (within 1 hour: 2x)
3. **Rate Limiting**: Protects API from abuse
   - API key based: 100 requests/minute
   - IP based: 20 requests/minute
4. **Multi-channel Notifications**: Supports 6 notification channels with Strategy Pattern

---

(d) How You Implemented (Important Implementation Issues)

1. Error Deduplication Algorithm

The system uses SHA-256 hash to group identical errors:

```
// Error.java
public static String calculateErrorHash(String errorType, String filePath, Integer
lineNumber) {
    String input = errorType + "|" + filePath + "|" + lineNumber;
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] hash = md.digest(input.getBytes(StandardCharsets.UTF_8));
    // Convert to hex string
}
```

2. Priority Calculation

Automatic priority scoring based on multiple factors:

```
// Error.java
public void calculatePriority(String url) {
    double pageWeight = determinePageWeight(url);      // URL-based weight
    double errorTypeWeight = determineErrorTypeWeight(); // Error type weight
    double occurrenceFactor = Math.log10(occurrenceCount + 1) + 1;
    double recencyBoost = calculateRecencyBoost();     // Time-based boost

    this.priorityScore = baseScore * pageWeight * errorTypeWeight *
occurrenceFactor * recencyBoost;
}
```

3. Event-Driven Architecture (Observer Pattern)

```
// ErrorService.java - Publisher
eventPublisher.publishEvent(new ErrorIngestedEvent(project, error, occurrence));

// PriorityCalculationListener.java - Observer 1
@Async @EventListener
public void handleErrorIngested(ErrorIngestedEvent event) {
    event.getError().calculatePriority(event.getContextUrl());
}

// NotificationListener.java - Observer 2
@Async @EventListener
public void handleErrorIngested(ErrorIngestedEvent event) {
    notificationService.notifyError(event.getProject(), event.getError());
}
```

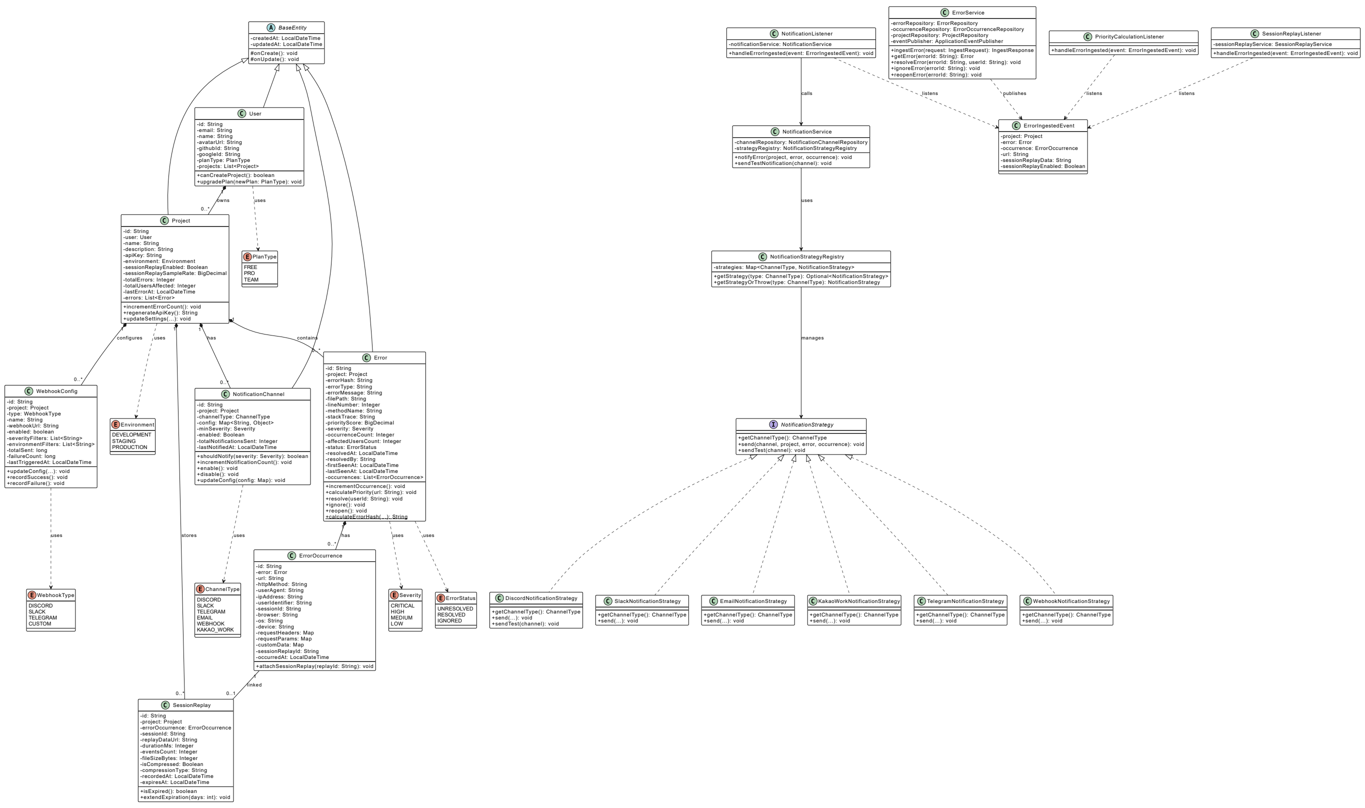
4. Strategy Pattern for Notifications

```
// NotificationStrategy.java - Interface
public interface NotificationStrategy {
    ChannelType getChannelType();
    void send(NotificationChannel channel, Project project, Error error,
ErrorOccurrence occurrence);
}

// 6 Implementations: Discord, Slack, Email, Telegram, Webhook, KakaoWork
@Component
public class DiscordNotificationStrategy implements NotificationStrategy { ... }
@Component
public class SlackNotificationStrategy implements NotificationStrategy { ... }
```

(e) UML Modeling for System Design

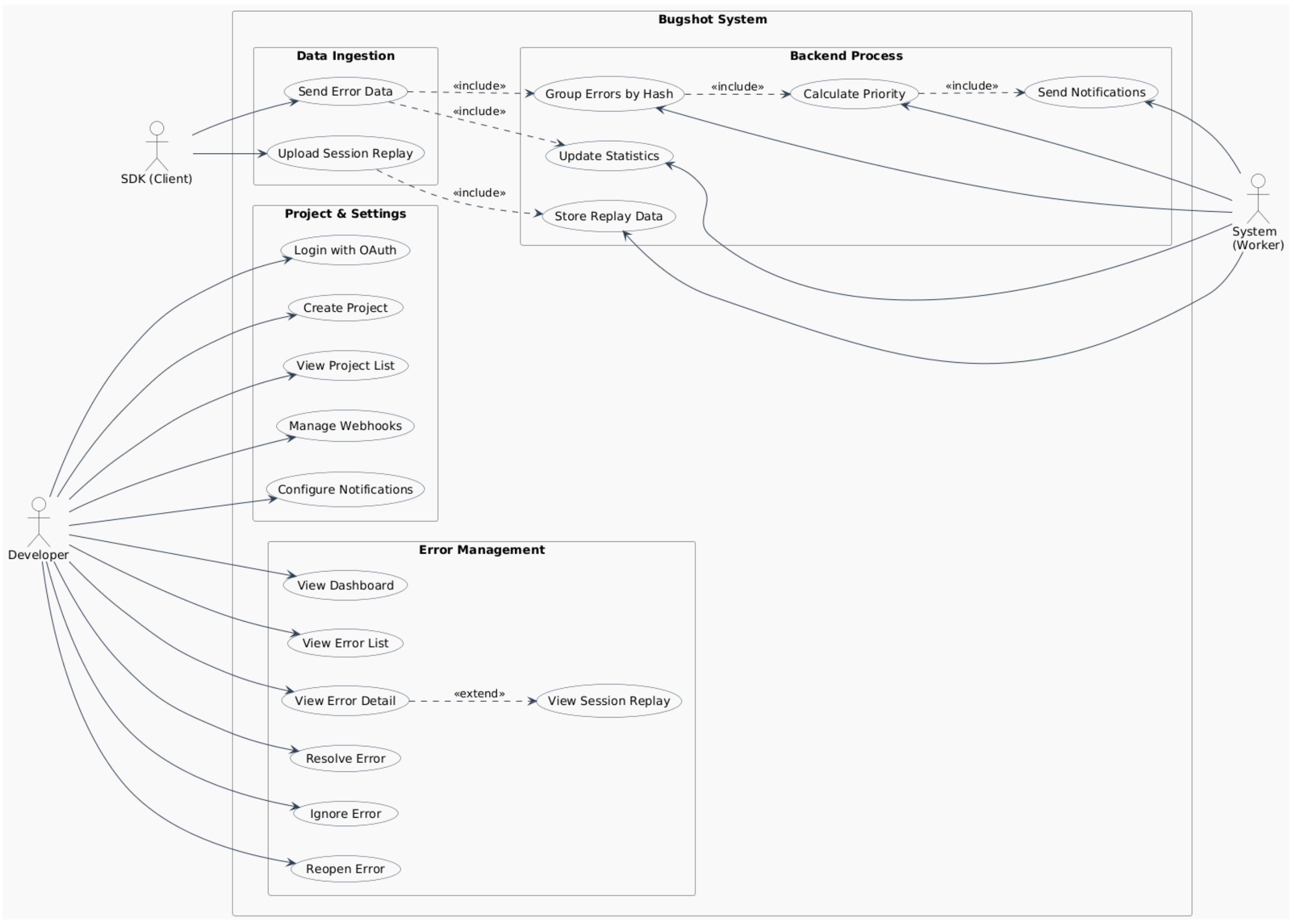
Class Diagram



Key Classes:

- BaseEntity: Abstract parent class with audit fields (createdAt, updatedAt)
- User: OAuth user entity with plan management
- Project: API key management, error collection settings
- Error: Error grouping with priority calculation
- ErrorOccurrence: Individual error occurrence records
- NotificationChannel: Multi-channel notification configuration
- NotificationStrategy: Interface for notification implementations

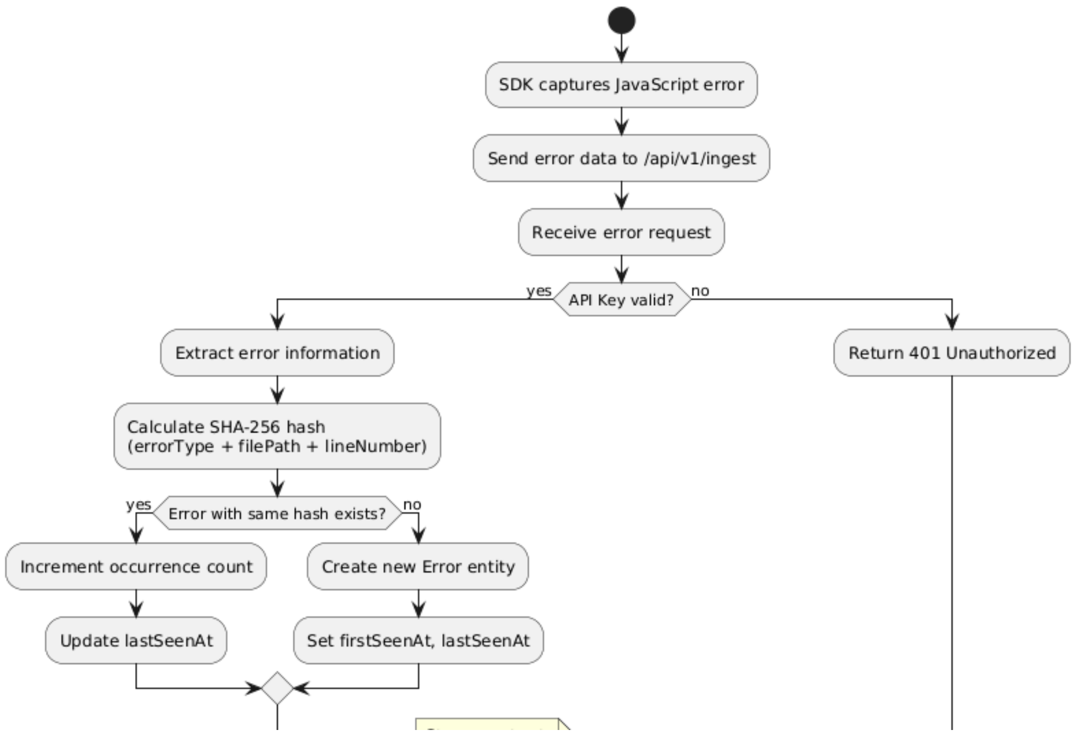
Use Case Diagram

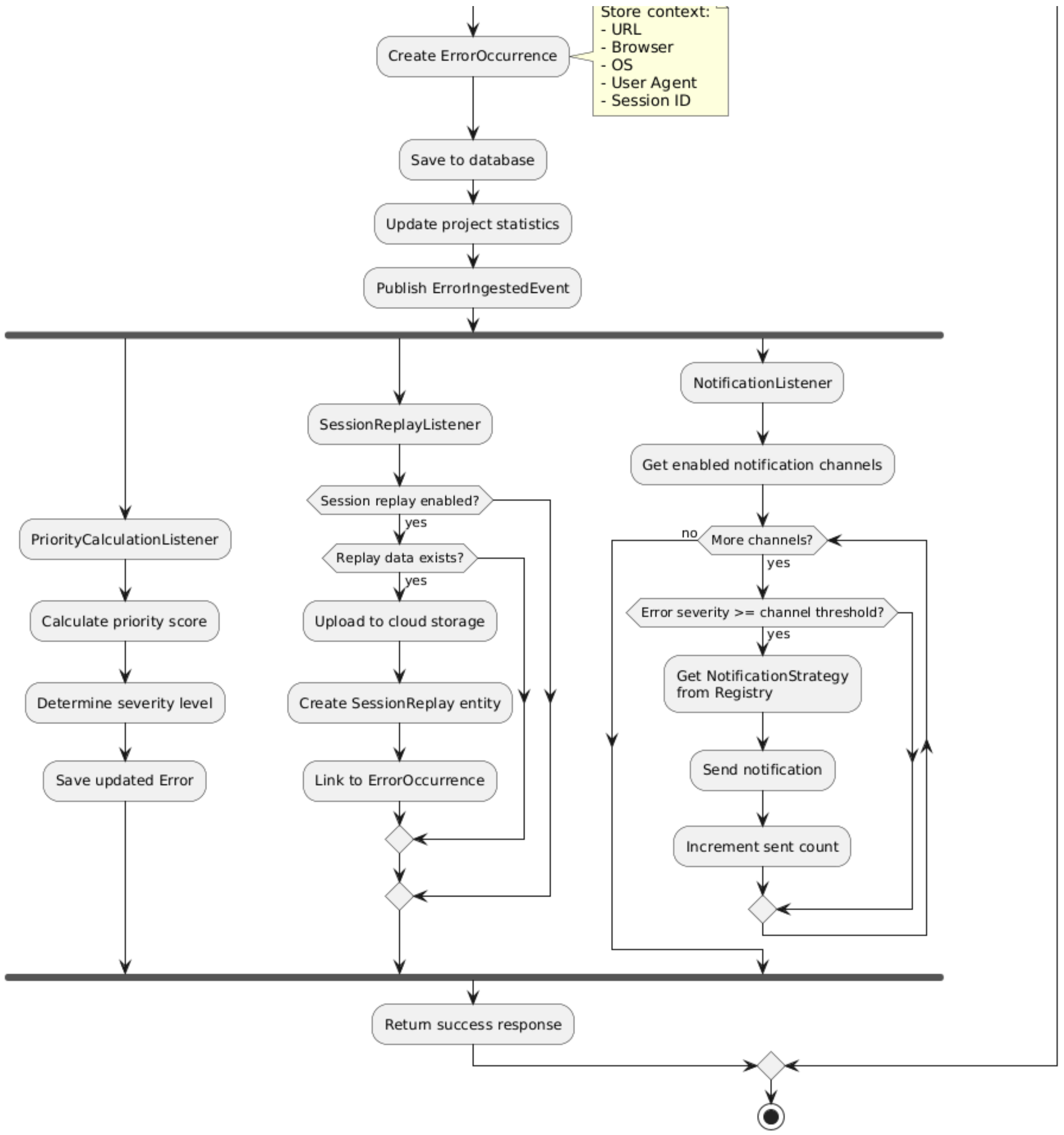


Actors:

- SDK (Client): JavaScript SDK that sends error data
- Developer: Web application developer using the dashboard
- System (Worker): Backend async processors

Activity Diagram - Error Ingest Process

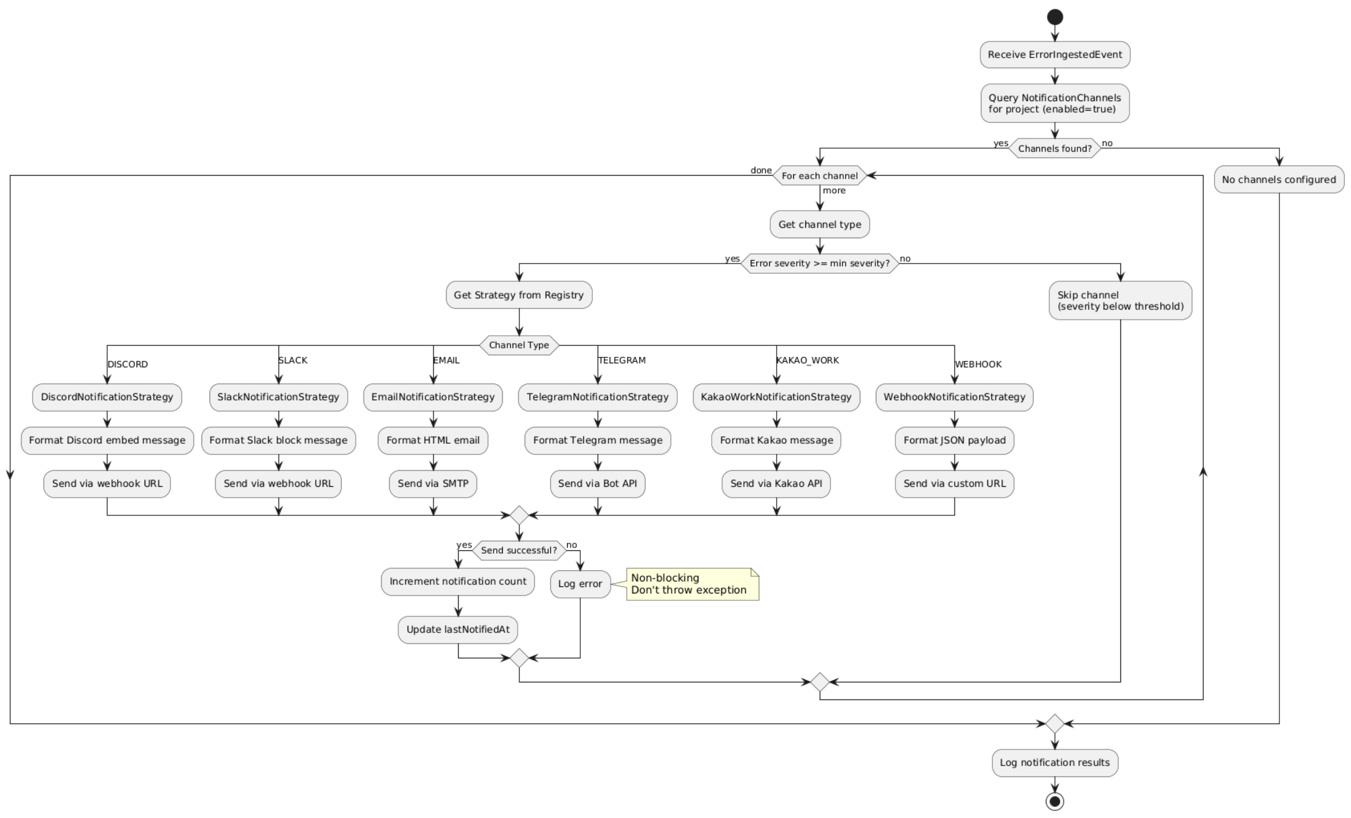

SDK captures JavaScript error
Send error data to /api/v1/ingest
Receive error request
API Key valid?
yes
no
Extract error information
Return 401 Unauthorized
Calculate SHA-256 hash
(errorType + filePath + lineNumber)
Error with same hash exists?
yes
no
Increment occurrence count
Create new Error entity
Update lastSeenAt
Set firstSeenAt, lastSeenAt



Flow:

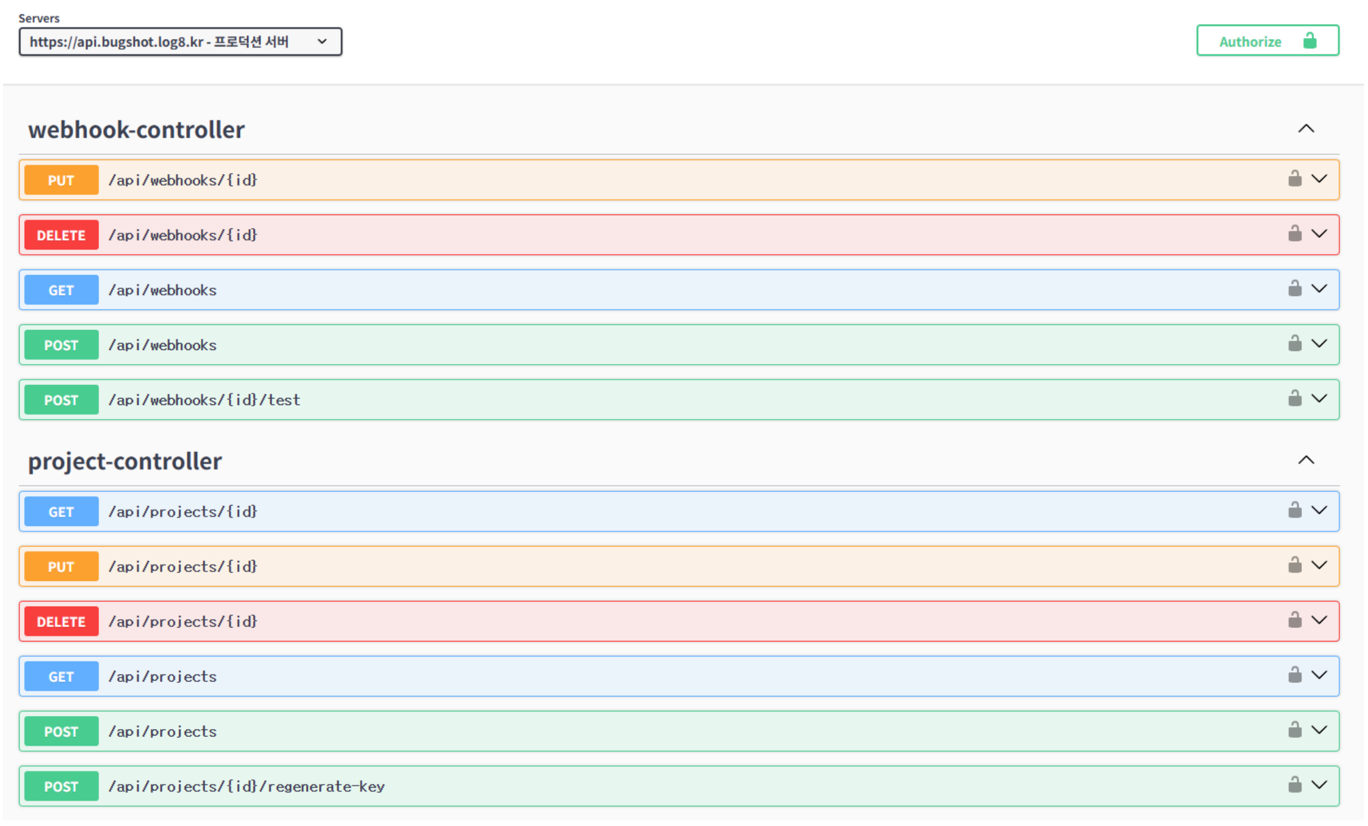
1. SDK captures JavaScript error
2. Send error data to /api/ingest
3. Validate API key
4. Calculate SHA-256 hash
5. Find or create Error entity
6. Create ErrorOccurrence
7. Publish ErrorIngestedEvent
8. Async listeners: Priority calculation, Session replay, Notifications

Activity Diagram - Notification Process


Receive ErrorIngestedEvent
Query NotificationChannels for project (enabled=true)
Channels found?
yes
no
No channels configured
For each channel
done
more
Get channel type
Error severity >= min severity?
yes
no
Get Strategy from Registry
Skip channel (severity below threshold)
Channel Type
DISCORD
SLACK
EMAIL
TELEGRAM
KAKAO_WORK
WEBHOOK
DiscordNotificationStrategy
SlackNotificationStrategy
EmailNotificationStrategy
TelegramNotificationStrategy
KakaoWorkNotificationStrategy
WebhookNotificationStrategy
Format Discord embed message
Format Slack block message
Format HTML email
Format Telegram message
Format Kakao message
Format JSON payload
Send via webhook URL
Send via webhook URL
Send via SMTP
Send via Bot API
Send via Kakao API
Send via custom URL
Send successful?
yes
no
Increment notification count
Log error
Non-blocking Don't throw exception
Update lastNotifiedAt
Log notification results

(f) Execution Results

1. API Documentation (Swagger UI)



webhook-controller

PUT /api/webhooks/{id}

DELETE /api/webhooks/{id}

GET /api/webhooks

POST /api/webhooks

POST /api/webhooks/{id}/test

project-controller

GET /api/projects/{id}

PUT /api/projects/{id}

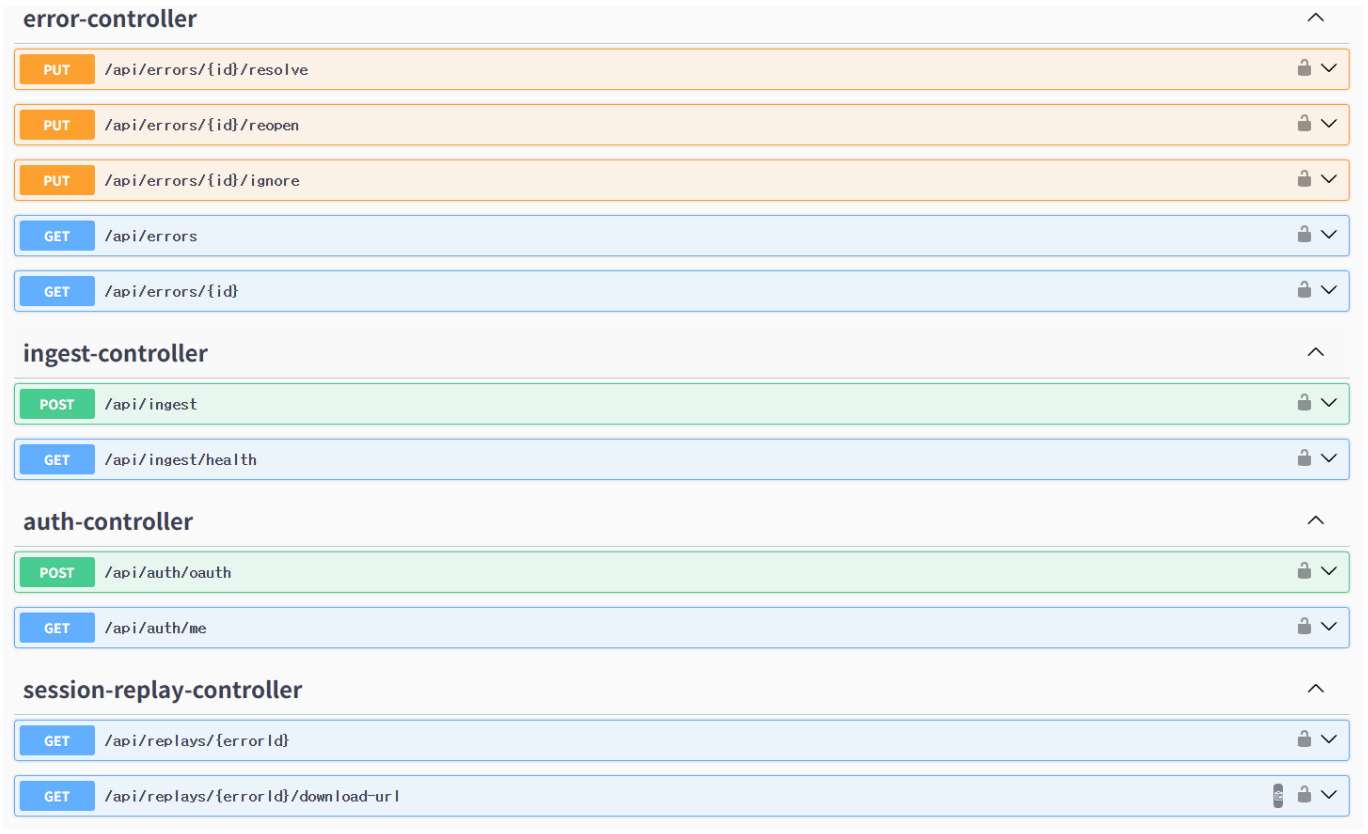
DELETE /api/projects/{id}

GET /api/projects

POST /api/projects

POST /api/projects/{id}/regenerate-key

Figure 1: Webhook and Project API endpoints in Swagger UI


error-controller

PUT /api/errors/{id}/resolve

PUT /api/errors/{id}/reopen

PUT /api/errors/{id}/ignore

GET /api/errors

GET /api/errors/{id}

ingest-controller

POST /api/ingest

GET /api/ingest/health

auth-controller

POST /api/auth/oauth

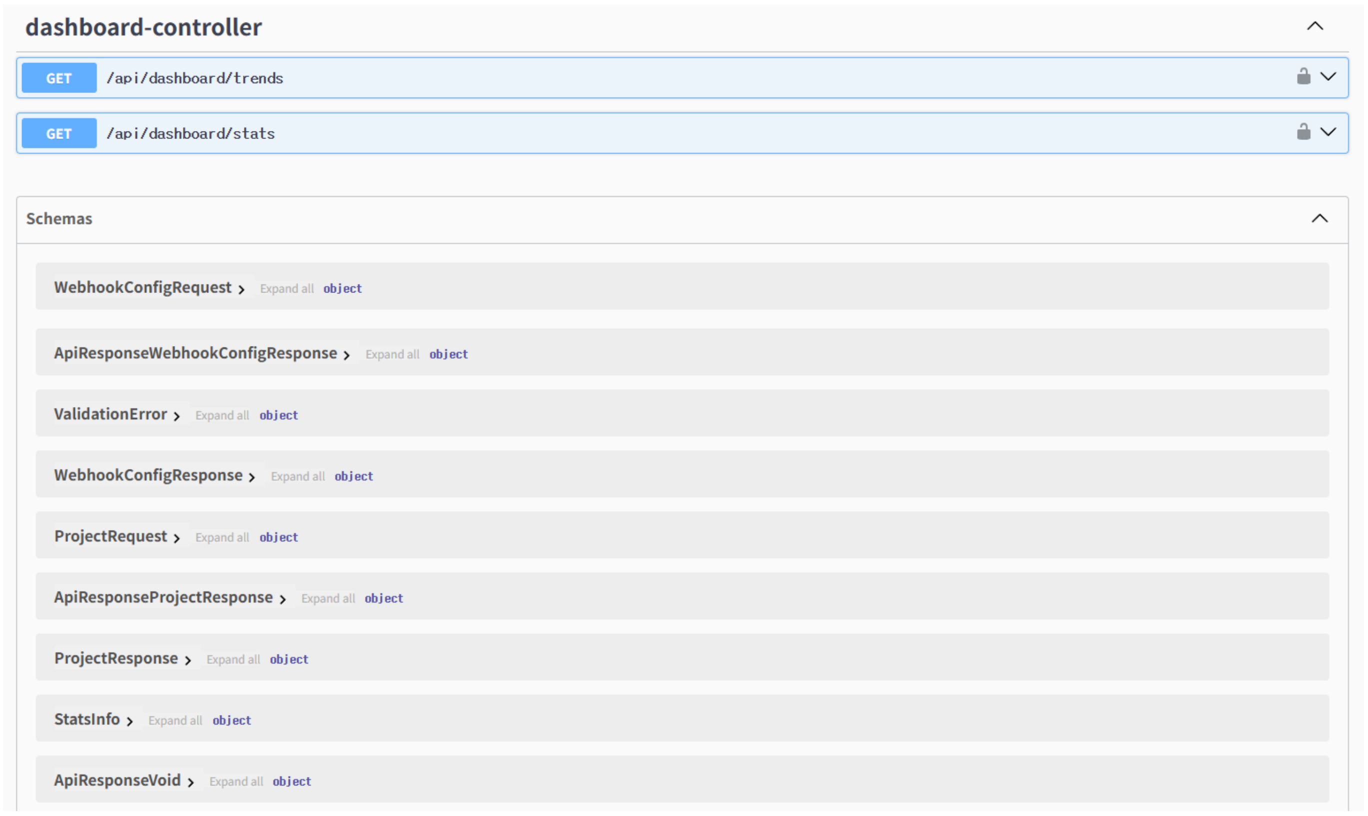
GET /api/auth/me

session-replay-controller

GET /api/replays/{errorId}

GET /api/replays/{errorId}/download-url

Figure 2: Error, Ingest, Auth, and Session Replay API endpoints


dashboard-controller

GET /api/dashboard/trends

GET /api/dashboard/stats

Schemas

WebhookConfigRequest › Expand all object

ApiResponseWebhookConfigResponse › Expand all object

ValidationError › Expand all object

WebhookConfigResponse › Expand all object

ProjectRequest › Expand all object

ApiResponseProjectResponse › Expand all object

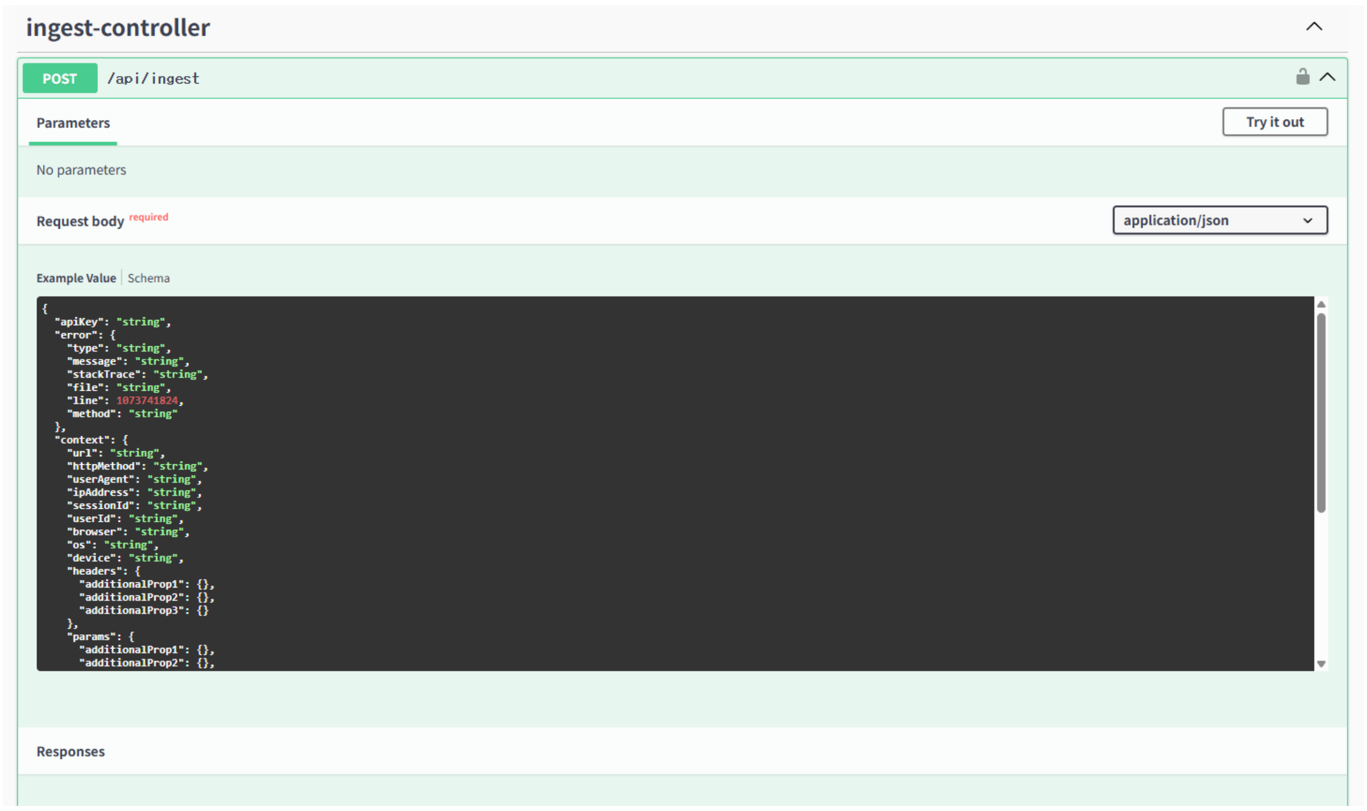
ProjectResponse › Expand all object

StatsInfo › Expand all object

ApiResponseVoid › Expand all object

Figure 3: Dashboard API and data schemas

2. Error Ingest API (Core Feature)


ingest-controller

POST /api/ingest

Parameters

Try it out

No parameters

Request body required

application/json

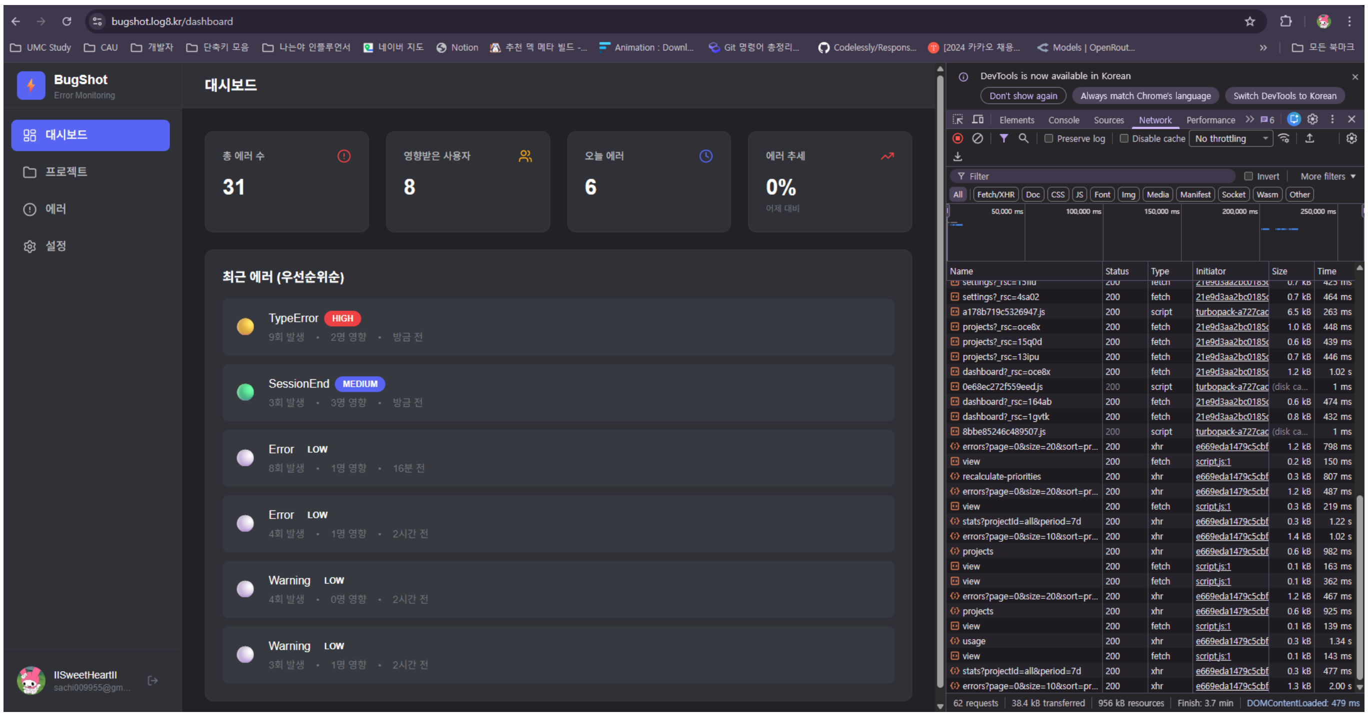
Example Value | Schema

```
{
  "apiKey": "string",
  "error": {
    "type": "string",
    "message": "string",
    "stackTrace": "string",
    "file": "string",
    "line": 1073741824,
    "method": "string"
  },
  "context": {
    "url": "string",
    "httpMethod": "string",
    "userAgent": "string",
    "ipAddress": "string",
    "sessionId": "string",
    "userId": "string",
    "browser": "string",
    "os": "string",
    "device": "string",
    "headers": {
      "additionalProp1": {},
      "additionalProp2": {},
      "additionalProp3": {}
    },
    "params": {
      "additionalProp1": {},
      "additionalProp2": {},
```

Responses

Figure 4: POST /api/ingest - SDK sends error data with this endpoint

3. Dashboard



BugShot
Error Monitoring

대시보드
프로젝트
에러
설정

대시보드

총 에러 수 31
영향받은 사용자 8
오늘 에러 6
에러 추세 0% 어제 대비

최근 에러 (우선순위순)

TypeError HIGH
9회 발생 • 2명 영향 • 방금 전

SessionEnd MEDIUM
3회 발생 • 3명 영향 • 방금 전

Error LOW
8회 발생 • 1명 영향 • 16분 전

Error LOW
4회 발생 • 1명 영향 • 2시간 전

Warning LOW
4회 발생 • 0명 영향 • 2시간 전

Warning LOW
3회 발생 • 1명 영향 • 2시간 전

IISweetHeartII
sachi009955@gm...

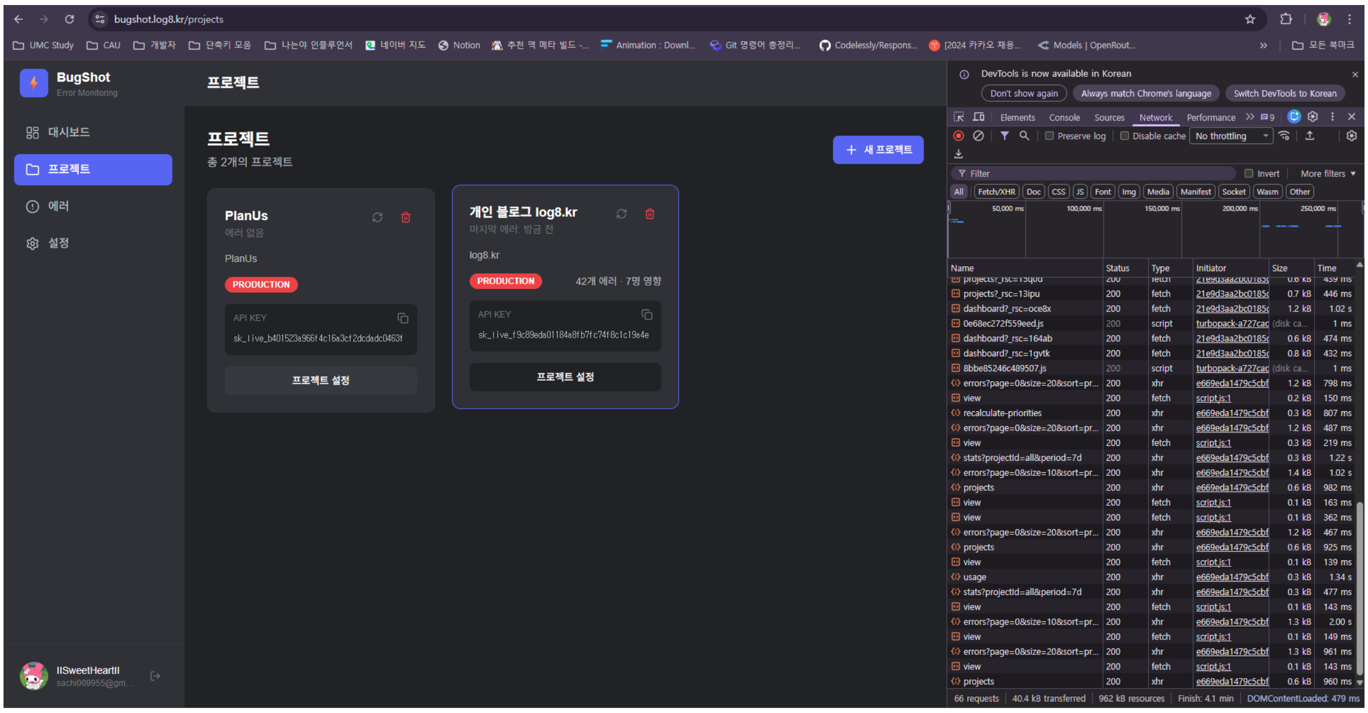
DevTools is now available in Korean
Don't show again | Always match Chrome's language | Switch DevTools to Korean

Name	Status	Type	Initiator	Size	Time
settings?_rsc=4sa02	200	fetch	21e9d3aa2bc0185c	0.7 kB	464 ms
a178b719c5326947.js	200	script	turbopack-a727cac	6.5 kB	263 ms
projects?_rsc=oce8x	200	fetch	21e9d3aa2bc0185c	1.0 kB	448 ms
projects?_rsc=15q0d	200	fetch	21e9d3aa2bc0185c	0.6 kB	439 ms
projects?_rsc=13ipu	200	fetch	21e9d3aa2bc0185c	0.7 kB	446 ms
dashboard?_rsc=oce8x	200	fetch	21e9d3aa2bc0185c	1.2 kB	1.02 s
0e68ec272f559eed.js	200	script	turbopack-a727cac	(disk ca...	1 ms
dashboard?_rsc=164ab	200	fetch	21e9d3aa2bc0185c	0.6 kB	474 ms
dashboard?_rsc=1gvtk	200	fetch	21e9d3aa2bc0185c	0.8 kB	432 ms
8bbe85246c489507.js	200	script	turbopack-a727cac	(disk ca...	1 ms
errors?page=0&size=20&sort=pr...	200	xhr	e669eda1479c5cbf	1.2 kB	798 ms
view	200	fetch	script.js:1	0.2 kB	150 ms
recalculate-priorities	200	xhr	e669eda1479c5cbf	0.3 kB	807 ms
errors?page=0&size=20&sort=pr...	200	xhr	e669eda1479c5cbf	1.2 kB	487 ms
view	200	fetch	script.js:1	0.3 kB	219 ms
stats?projectId=all&period=7d	200	xhr	e669eda1479c5cbf	0.3 kB	1.22 s
errors?page=0&size=10&sort=pr...	200	xhr	e669eda1479c5cbf	1.4 kB	1.02 s
projects	200	xhr	e669eda1479c5cbf	0.6 kB	982 ms
view	200	fetch	script.js:1	0.1 kB	163 ms
view	200	fetch	script.js:1	0.1 kB	362 ms
errors?page=0&size=20&sort=pr...	200	xhr	e669eda1479c5cbf	1.2 kB	467 ms
projects	200	xhr	e669eda1479c5cbf	0.6 kB	925 ms
view	200	fetch	script.js:1	0.1 kB	139 ms
usage	200	xhr	e669eda1479c5cbf	0.3 kB	1.34 s
view	200	fetch	script.js:1	0.1 kB	143 ms
stats?projectId=all&period=7d	200	xhr	e669eda1479c5cbf	0.3 kB	477 ms
errors?page=0&size=10&sort=pr...	200	xhr	e669eda1479c5cbf	1.3 kB	2.00 s

62 requests | 38.4 kB transferred | 956 kB resources | Finish: 3.7 min | DOMContentLoaded: 479 ms

Figure 5: Dashboard showing error statistics and priority-sorted errors

4. Project Management



프로젝트

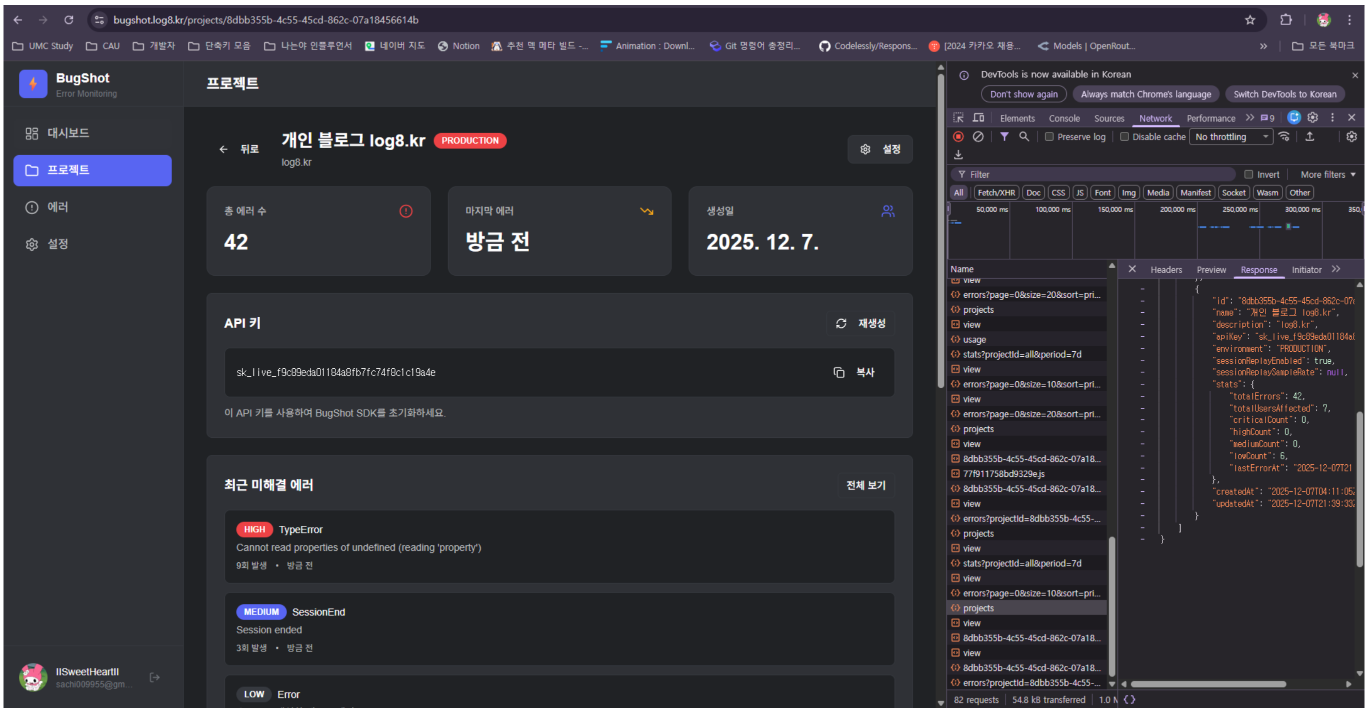
총 2개의 프로젝트

PlanUs
에러 없음
PlanUs
PRODUCTION
API KEY
sk_live_b401523a966f4c16a3cf2dcdadc0463f
프로젝트 설정

개인 블로그 log8.kr
마지막 에러: 방금 전
log8.kr
PRODUCTION 42개 에러 · 7명 영향
API KEY
sk_live_f9c89eda01184a8fb7fc74f8c1c19a4e
프로젝트 설정

```
Name                              Status  Type    Initiator           Size          Time
projects?_rsc=13ipu               200     fetch   21e9d3aa2bc0185c    0.7 kB        446 ms
dashboard?_rsc=oce8x              200     fetch   21e9d3aa2bc0185c    1.2 kB        1.02 s
0e68ec272f559eed.js               200     script  turbopack-a727cac   (disk ca...   1 ms
dashboard?_rsc=164ab              200     fetch   21e9d3aa2bc0185c    0.6 kB        474 ms
dashboard?_rsc=1gvtk              200     fetch   21e9d3aa2bc0185c    0.8 kB        432 ms
8bbe85246c489507.js               200     script  turbopack-a727cac   (disk ca...   1 ms
errors?page=0&size=20&sort=pr...  200     xhr     e669eda1479c5cbf    1.2 kB        798 ms
view                              200     fetch   script.js:1         0.2 kB        150 ms
recalculate-priorities            200     xhr     e669eda1479c5cbf    0.3 kB        807 ms
errors?page=0&size=20&sort=pr...  200     xhr     e669eda1479c5cbf    1.2 kB        487 ms
view                              200     fetch   script.js:1         0.3 kB        219 ms
stats?projectId=all&period=7d     200     xhr     e669eda1479c5cbf    0.3 kB        1.22 s
errors?page=0&size=10&sort=pr...  200     xhr     e669eda1479c5cbf    1.4 kB        1.02 s
projects                          200     xhr     e669eda1479c5cbf    0.6 kB        982 ms
view                              200     fetch   script.js:1         0.1 kB        163 ms
view                              200     fetch   script.js:1         0.1 kB        362 ms
errors?page=0&size=20&sort=pr...  200     xhr     e669eda1479c5cbf    1.2 kB        467 ms
projects                          200     xhr     e669eda1479c5cbf    0.6 kB        925 ms
view                              200     fetch   script.js:1         0.1 kB        139 ms
usage                             200     xhr     e669eda1479c5cbf    0.3 kB        1.34 s
stats?projectId=all&period=7d     200     xhr     e669eda1479c5cbf    0.3 kB        477 ms
view                              200     fetch   script.js:1         0.1 kB        143 ms
errors?page=0&size=10&sort=pr...  200     xhr     e669eda1479c5cbf    1.3 kB        2.00 s
view                              200     fetch   script.js:1         0.1 kB        149 ms
errors?page=0&size=20&sort=pr...  200     xhr     e669eda1479c5cbf    1.3 kB        961 ms
view                              200     fetch   script.js:1         0.1 kB        143 ms
projects                          200     xhr     e669eda1479c5cbf    0.6 kB        960 ms
66 requests | 40.4 kB transferred | 962 kB resources | Finish: 4.1 min | DOMContentLoaded: 479 ms
```

Figure 6: Project list with API keys for SDK integration



개인 블로그 log8.kr PRODUCTION
log8.kr

총 에러 수 42 | 마지막 에러 방금 전 | 생성일 2025. 12. 7.

API 키
sk_live_f9c89eda01184a8fb7fc74f8c1c19a4e
이 API 키를 사용하여 BugShot SDK를 초기화하세요.

최근 미해결 에러

HIGH TypeError
Cannot read properties of undefined (reading 'property')
9회 발생 · 방금 전

MEDIUM SessionEnd
Session ended
3회 발생 · 방금 전

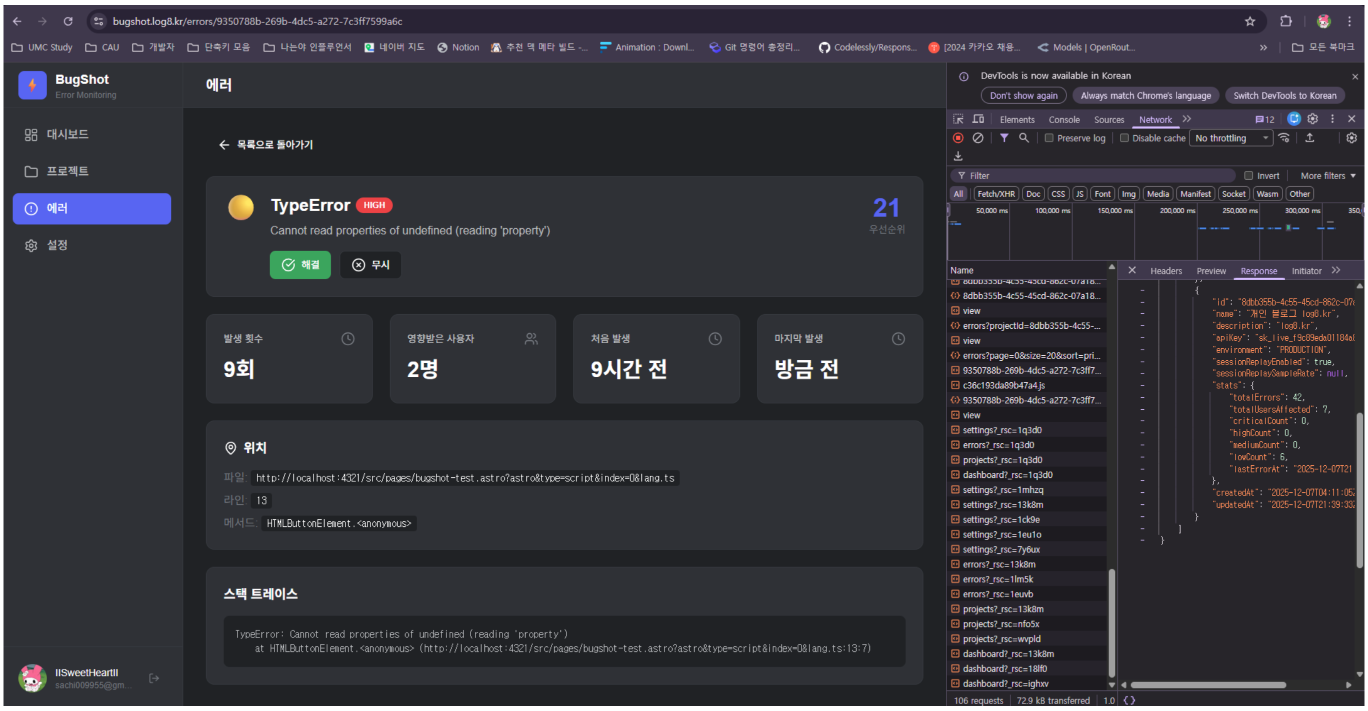
LOW Error

```
Name
errors?page=0&size=20&sort=pri...
projects
view
usage
stats?projectId=all&period=7d
view
errors?page=0&size=10&sort=pri...
view
errors?page=0&size=20&sort=pri...
projects
view
8dbb355b-4c55-45cd-862c-07a18...
77f911758bd9329e.js
8dbb355b-4c55-45cd-862c-07a18...
view
errors?projectId=8dbb355b-4c55-...
projects
view
stats?projectId=all&period=7d
view
errors?page=0&size=10&sort=pri...
projects
view
8dbb355b-4c55-45cd-862c-07a18...
view
8dbb355b-4c55-45cd-862c-07a18...
errors?projectId=8dbb355b-4c55-...
82 requests | 54.8 kB transferred | 1.0 M
```

```
      {
        "id": "8dbb355b-4c55-45cd-862c-07a
        "name": "개인 블로그 log8.kr",
        "description": "log8.kr",
        "apiKey": "sk_live_f9c89eda01184a8
        "environment": "PRODUCTION",
        "sessionReplayEnabled": true,
        "sessionReplaySampleRate": null,
        "stats": {
          "totalErrors": 42,
          "totalUsersAffected": 7,
          "criticalCount": 0,
          "highCount": 0,
          "mediumCount": 0,
          "lowCount": 6,
          "lastErrorAt": "2025-12-07T21
        },
        "createdAt": "2025-12-07T04:11:05
        "updatedAt": "2025-12-07T21:39:33
      }
    ]
  }
```

Figure 7: Project detail showing errors, API key, recent unresolved errors

5. Error Detail


bugshot.log8.kr/errors/9350788b-269b-4dc5-a272-7c3ff7599a6c

BugShot
Error Monitoring

에러

← 목록으로 돌아가기

TypeError HIGH
Cannot read properties of undefined (reading 'property')
21
우선순위

해결 무시

발생 횟수	영향받은 사용자	처음 발생	마지막 발생
9회	2명	9시간 전	방금 전

위치
파일: `http://localhost:4321/src/pages/bugshot-test.astro?astro&type=script&index=0&lang.ts`
라인: 13
메서드: `HTMLButtonElement.<anonymous>`

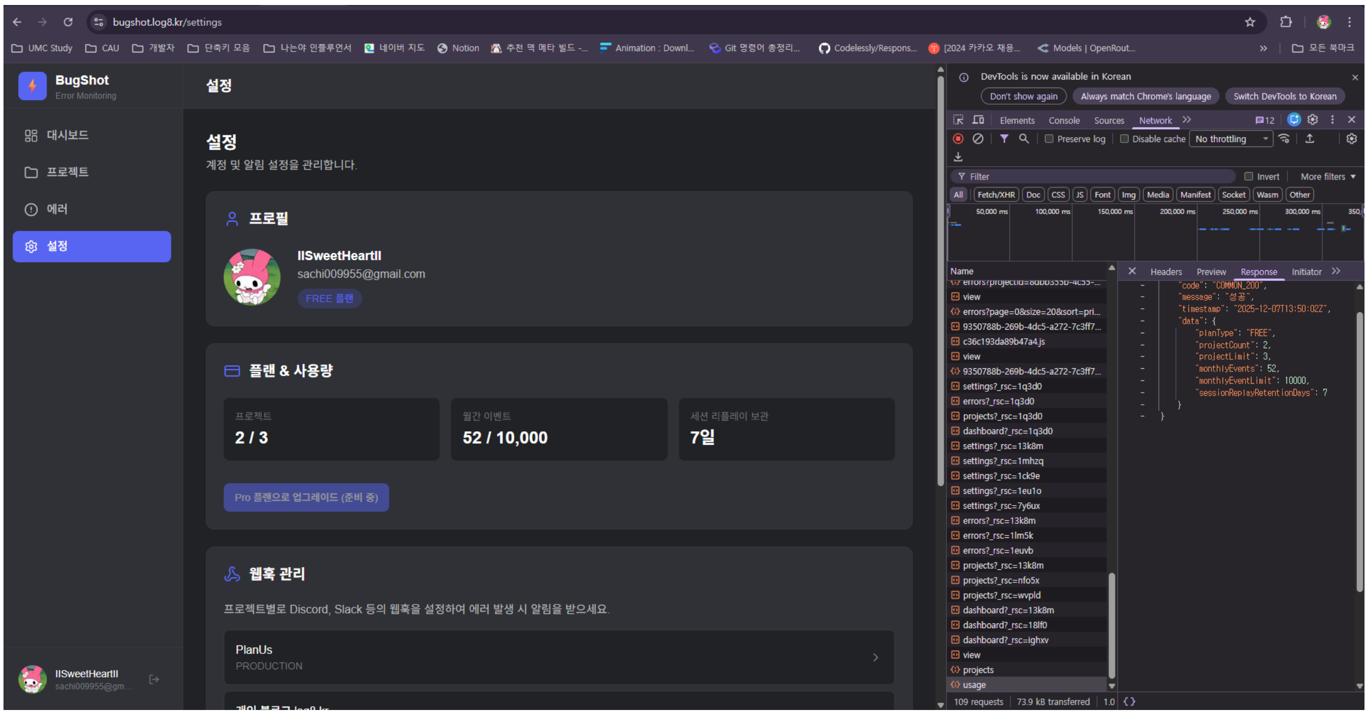
스택 트레이스
```
TypeError: Cannot read properties of undefined (reading 'property')
    at HTMLButtonElement.<anonymous> (http://localhost:4321/src/pages/bugshot-test.astro?astro&type=script&index=0&lang.ts:13:7)
```

```
{
    "id": "8dbb355b-4c55-45cd-862c-07a
    "name": "개인 블로그 log8.kr",
    "description": "log8.kr",
    "apiKey": "sk_live_f9c89eda01184a
    "environment": "PRODUCTION",
    "sessionReplayEnabled": true,
    "sessionReplaySampleRate": null,
    "stats": {
        "totalErrors": 42,
        "totalUsersAffected": 7,
        "criticalCount": 0,
        "highCount": 0,
        "mediumCount": 0,
        "lowCount": 6,
        "lastErrorAt": "2025-12-07T21
    },
    "createdAt": "2025-12-07T04:11:05
    "updatedAt": "2025-12-07T21:39:33
  }
 ]
}
```

106 requests | 72.9 kB transferred

Figure 8: Error detail - TypeError with HIGH priority, occurrences, affected users, stack trace

6. Webhook Configuration


bugshot.log8.kr/settings

BugShot
Error Monitoring

설정

설정
계정 및 알림 설정을 관리합니다.

프로필
IISweetHeartII
sachi009955@gmail.com
FREE 플랜

플랜 & 사용량

프로젝트	월간 이벤트	세션 리플레이 보관
2 / 3	52 / 10,000	7일

Pro 플랜으로 업그레이드 (준비 중)

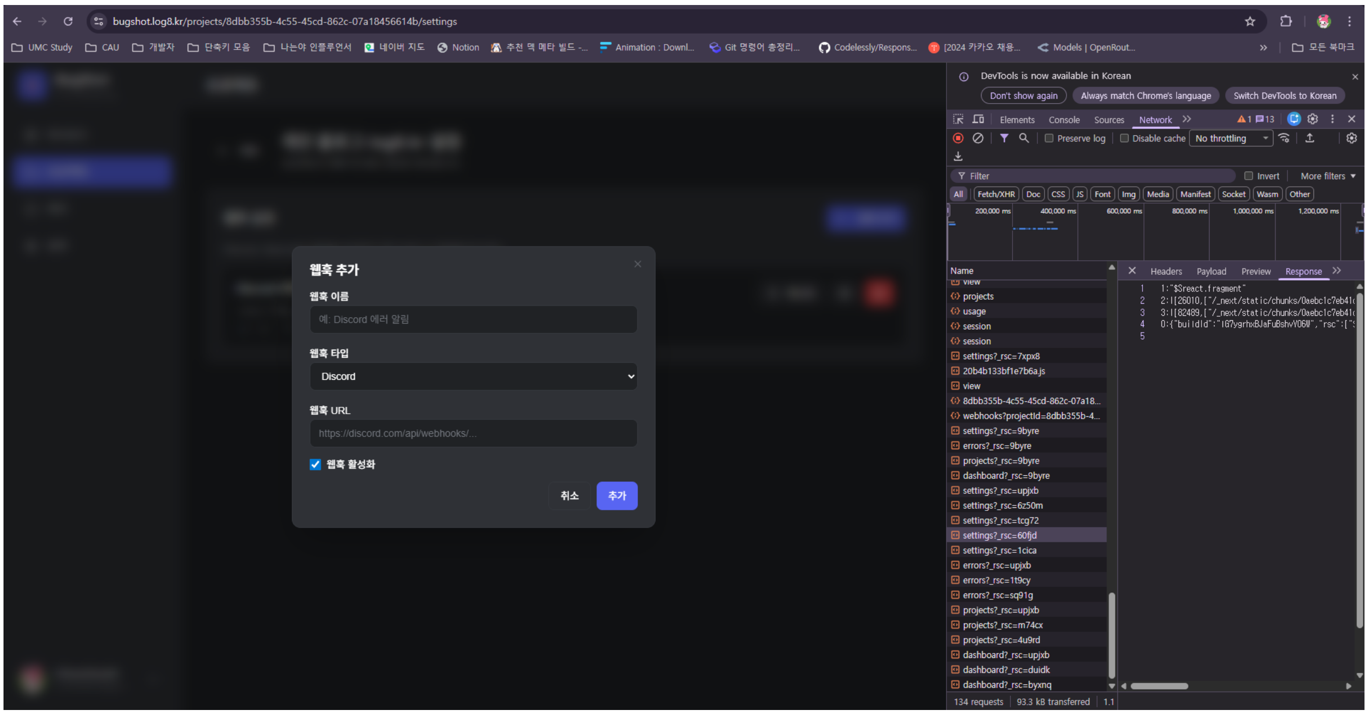
웹훅 관리
프로젝트별로 Discord, Slack 등의 웹훅을 설정하여 에러 발생 시 알림을 받으세요.

PlanUs
PRODUCTION

```
    "code": "COMMON_200",
    "message": "성공",
    "timestamp": "2025-12-07T13:50:02Z",
    "data": {
        "planType": "FREE",
        "projectCount": 2,
        "projectLimit": 3,
        "monthlyEvents": 52,
        "monthlyEventLimit": 10000,
        "sessionReplayRetentionDays": 7
    }
}
```

109 requests | 73.9 kB transferred

Figure 9: User settings - plan, usage statistics, webhook management


bugshot.log8.kr/projects/8dbb355b-4c55-45cd-862c-07a18456614b/settings

웹훅 추가

웹훅 이름

예: Discord 에러 알림

웹훅 타입

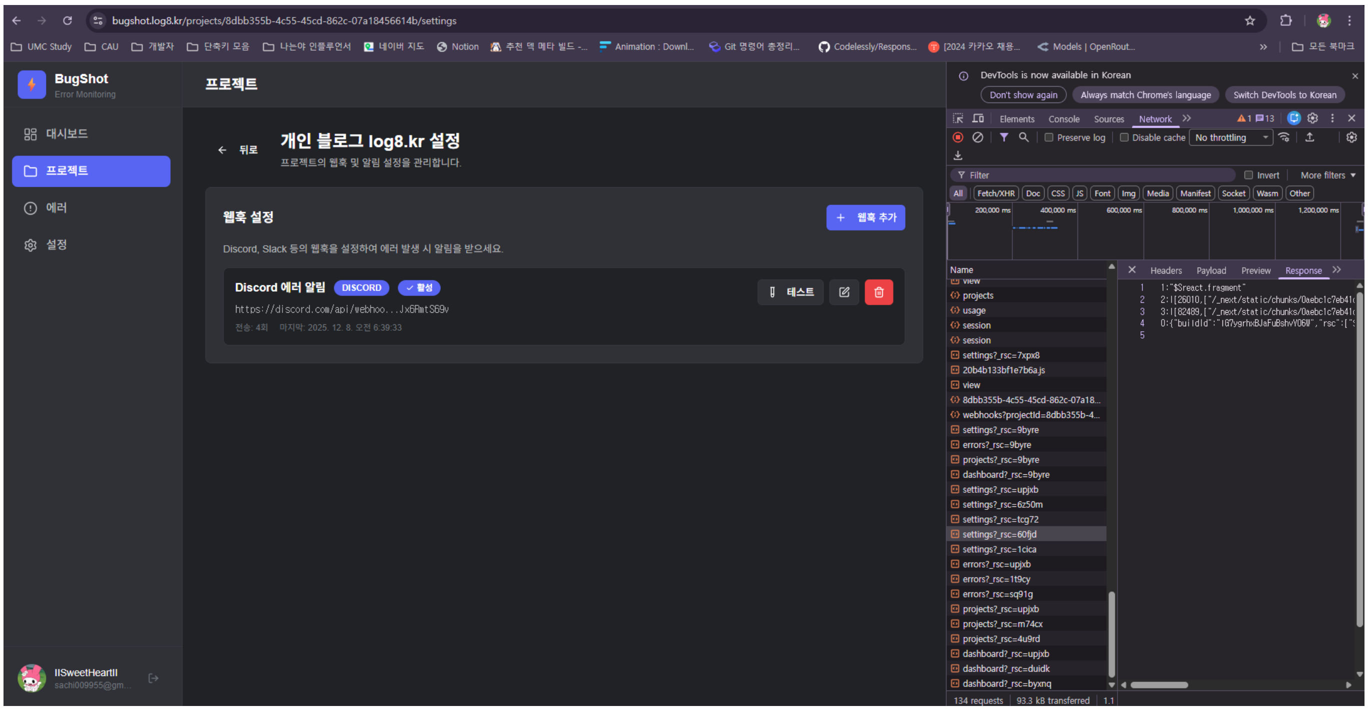
Discord

웹훅 URL

https://discord.com/api/webhooks/...

[✓] 웹훅 활성화

Figure 10: Adding Discord webhook


bugshot.log8.kr/projects/8dbb355b-4c55-45cd-862c-07a18456614b/settings

개인 블로그 log8.kr 설정

프로젝트의 웹훅 및 알림 설정을 관리합니다.

웹훅 설정

Discord, Slack 등의 웹훅을 설정하여 에러 발생 시 알림을 받으세요.

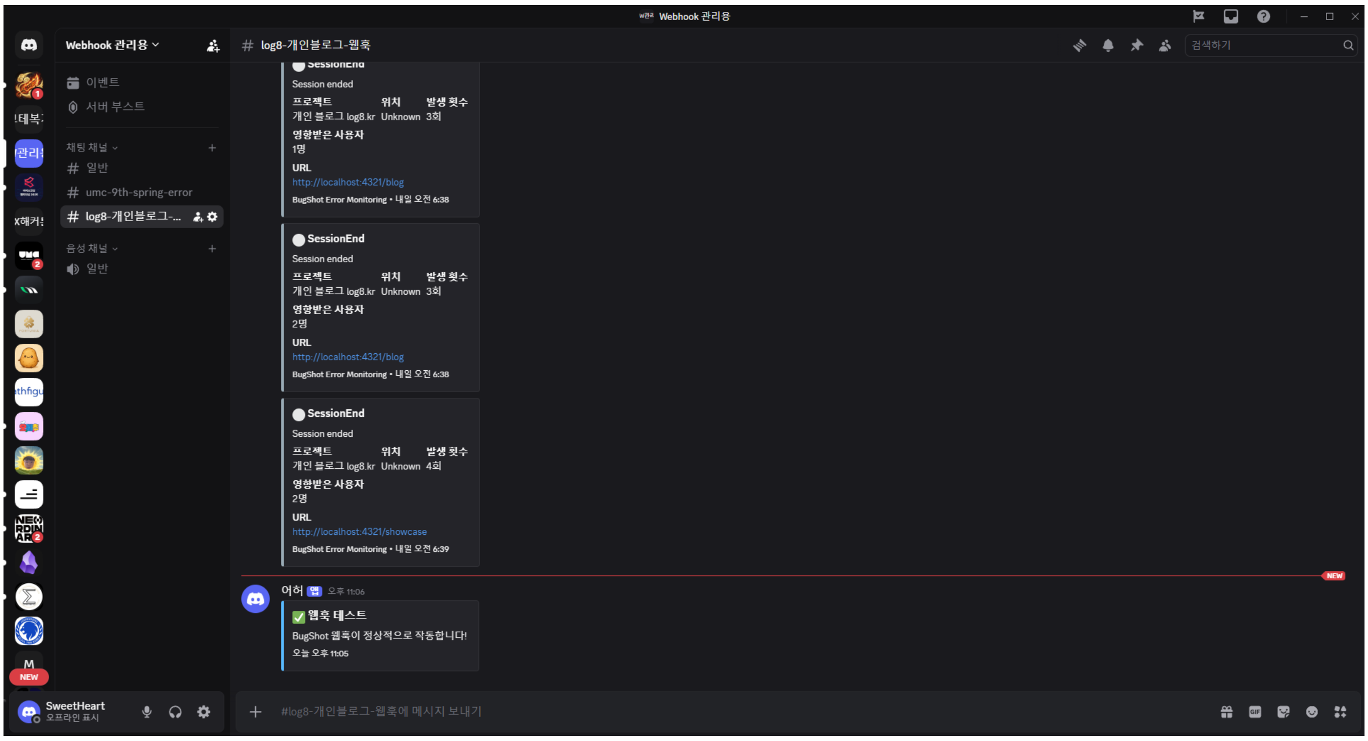
Discord 에러 알림 DISCORD ✓ 활성

https://discord.com/api/webhoo...Jx6RmtS69v

전송: 4회 마지막: 2025. 12. 8. 오전 6:39:33

Figure 11: Configured Discord webhook with test button

7. Notification Result



SessionEnd
Session ended
프로젝트 개인 블로그 log8.kr | 위치 Unknown | 발생 횟수 3회
영향받은 사용자
1명
URL
http://localhost:4321/blog
BugShot Error Monitoring • 내일 오전 6:38

SessionEnd
Session ended
프로젝트 개인 블로그 log8.kr | 위치 Unknown | 발생 횟수 3회
영향받은 사용자
2명
URL
http://localhost:4321/blog
BugShot Error Monitoring • 내일 오전 6:38

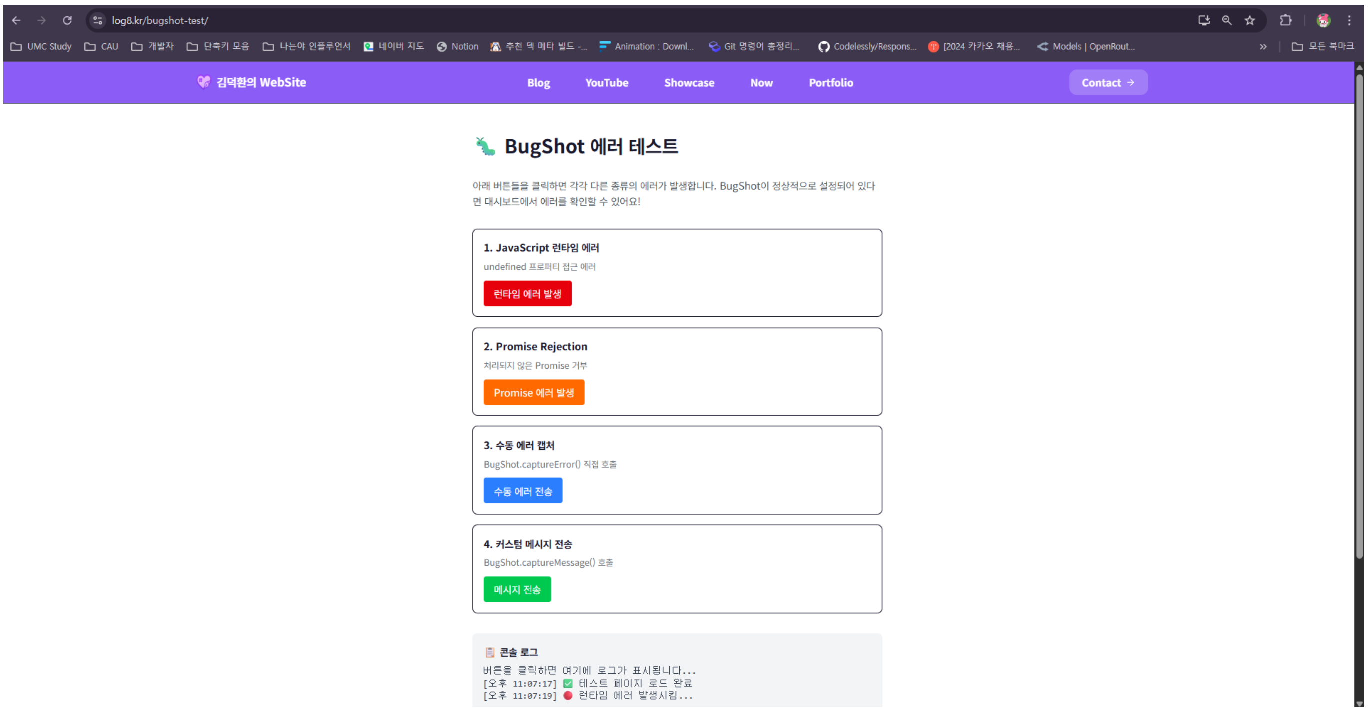
SessionEnd
Session ended
프로젝트 개인 블로그 log8.kr | 위치 Unknown | 발생 횟수 4회
영향받은 사용자
2명
URL
http://localhost:4321/showcase
BugShot Error Monitoring • 내일 오전 6:39

어허 앱 오후 11:06
✅ 웹훅 테스트
BugShot 웹훅이 정상적으로 작동합니다!
오늘 오후 11:05

#log8-개인블로그-웹훅에 메시지 보내기

Figure 12: Discord receiving error notifications from Bugshot

8. SDK Test Page



BugShot 에러 테스트

아래 버튼들을 클릭하면 각각 다른 종류의 에러가 발생합니다. BugShot이 정상적으로 설정되어 있다면 대시보드에서 에러를 확인할 수 있어요!

1. JavaScript 런타임 에러
undefined 프로퍼티 접근 에러
런타임 에러 발생

2. Promise Rejection
처리되지 않은 Promise 거부
Promise 에러 발생

3. 수동 에러 캡처
BugShot.captureError() 직접 호출
수동 에러 전송

4. 커스텀 메시지 전송
BugShot.captureMessage() 호출
메시지 전송

콘솔 로그
```
버튼을 클릭하면 여기에 로그가 표시됩니다...
[오후 11:07:17] ✅ 테스트 페이지 로드 완료
[오후 11:07:19] 🔴 런타임 에러 발생시킴...
```

Figure 13: SDK test page with buttons to trigger different error types

---

(g) Object-Oriented Concepts Applied

1. Inheritance

BaseEntity provides common audit fields to all entities:

```
@MappedSuperclass
public abstract class BaseEntity {
    @CreatedDate private LocalDateTime createdAt;
    @LastModifiedDate private LocalDateTime updatedAt;
}

// All entities inherit from BaseEntity
@Entity public class User extends BaseEntity { ... }
@Entity public class Project extends BaseEntity { ... }
@Entity public class Error extends BaseEntity { ... }
```

Inheritance Hierarchy: BaseEntity (abstract) ├── User ├── Project ├── Error ├── ErrorOccurrence ├── NotificationChannel ├── SessionReplay └── WebhookConfig

2. Polymorphism

NotificationStrategy interface with 6 implementations:

```
// Interface
public interface NotificationStrategy {
    ChannelType getChannelType();
    void send(NotificationChannel channel, Project project, Error error,
ErrorOccurrence occurrence);
}

// Implementations
@Component public class DiscordNotificationStrategy implements
NotificationStrategy { ... }
@Component public class SlackNotificationStrategy implements NotificationStrategy
{ ... }
@Component public class EmailNotificationStrategy implements NotificationStrategy
{ ... }

// Polymorphic call - Runtime dispatch
NotificationStrategy strategy = registry.getStrategy(channel.getChannelType());
strategy.send(channel, project, error, occurrence);
```

3. Encapsulation

Entity classes expose behavior through methods, not direct field access:

```
@Entity
@Getter  // Only getter, no setter
public class Error extends BaseEntity {
    private Integer occurrenceCount = 1;
    private ErrorStatus status = ErrorStatus.UNRESOLVED;
    // Business logic encapsulated in methods
    public void incrementOccurrence() {
        this.occurrenceCount++;
        this.lastSeenAt = LocalDateTime.now();
    }

    public void resolve(String userId) {
        this.status = ErrorStatus.RESOLVED;
        this.resolvedAt = LocalDateTime.now();
    }

    // Complex logic in private methods
    private double determinePageWeight(String url) { ... }
}
```

4. Abstraction

Complex logic abstracted behind simple interfaces:

```
// Simple public interface
public void notifyError(Project project, Error error, ErrorOccurrence occurrence)
{
    List<NotificationChannel> channels =
channelRepository.findByProjectIdAndEnabledTrue(project.getId());
    for (NotificationChannel channel : channels) {
        if (channel.shouldNotify(error.getSeverity())) {
            sendNotification(channel, project, error, occurrence);
        }
    }
}

// Internal complexity hidden in private methods
```

5. Design Patterns Applied

Pattern	Location	Purpose
Strategy	notification/strategy/*	Different notification channels
Observer	error/event/listener/*	Async event processing
Template Method	BaseEntity	Entity lifecycle management
Factory	Error.calculateErrorHash()	Error hash generation
Repository	*Repository interfaces	Data access abstraction
Builder	Lombok @Builder	Complex object construction

(h) Conclusion

Project Summary

Bugshot successfully demonstrates the application of Object-Oriented Programming concepts in building a real-world error monitoring system:

1. **Practical OOP Application**: Inheritance (BaseEntity), Polymorphism (NotificationStrategy), Encapsulation (Entity methods), and Abstraction (Service layer)
2. **Design Pattern Usage**: Strategy Pattern for notifications, Observer Pattern for event-driven architecture, Template Method for entity lifecycle
3. **Real-world Deployment**: The system is deployed at https://bugshot-api.log8.kr

Technical Achievements

- SHA-256 based error deduplication algorithm
- Multi-factor priority scoring system
- Event-driven architecture with Spring @Async
- Strategy pattern supporting 6 notification channels
- Rate limiting (API key + IP based)
- RESTful API with Swagger documentation

Lessons Learned

- Importance of proper abstraction for maintainability
- Value of design patterns in solving common problems
- Benefits of event-driven architecture for scalability

Appendix

Technology Stack

Category	Technology
Language	Java 21
Framework	Spring Boot 3.5
Database	MySQL 8.0, Redis 6.0
ORM	Spring Data JPA
Build	Gradle 8.x
API Docs	SpringDoc OpenAPI (Swagger)
Async	Spring @Async, @EventListener
Storage	Cloudflare R2

Links

- Service Link: https://bugshot.log8.kr
- Swagger UI: https://bugshot-api.log8.kr/swagger-ui/index.html
- GitHub: https://github.com/IISweetHeartII/bugshot

End of Report