

# Разработка и реализация модели для обработки автоматически сгенерированных отчётов об ошибках

Смирнов Игорь Андреевич

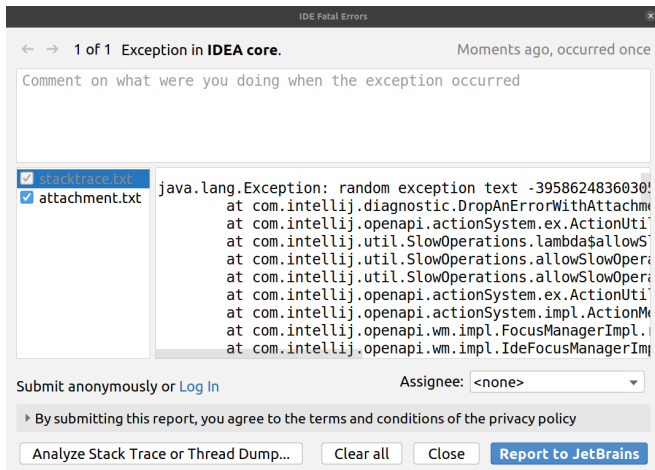
научный руководитель: Васильев Роман Алексеевич (JetBrains)

Санкт-Петербургская школа  
физико-математических и компьютерных наук

НИУ ВШЭ – Санкт-Петербург

11 июня 2021 г.

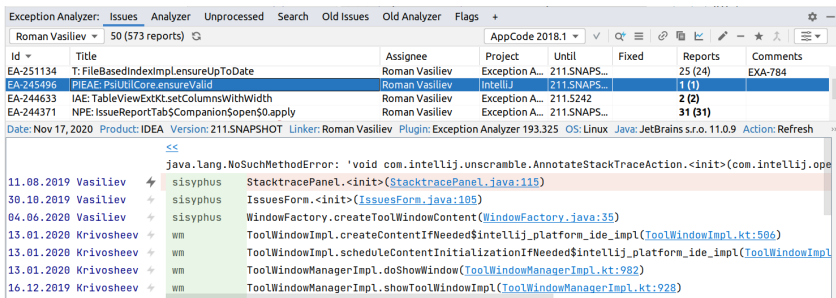
# Введение: автоматически сгенерированные отчёты об ошибках



**Рис.:** Отправка отчёта в IntelliJ IDEA

# Введение: Exception Analyzer

- Приём и обработка отчётов из проектов JetBrains
- Отчётов много  $\Rightarrow$  нужна автоматическая обработка



Id	Title	Assignee	Project	Until	Fixed	Reports	Comments
EA-251134	T: FileBasedIndexImpl.ensureUpToDate	Roman Vasiliev	Exception A...	211.SNAPS...		25 (24)	EXA-784
EA-245496	PIEAE: PsiUtilCore.ensureValid	Roman Vasiliev	IntelliJ	211.SNAPS...		1 (1)	
EA-244633	IAE: TableViewExtKt.setColumnsWithWidth	Roman Vasiliev	Exception A...	211.5242		2 (2)	
EA-244371	NPE: IssueReportTab\$Companion\$open\$0.apply	Roman Vasiliev	Exception A...	211.SNAPS...		31 (31)	

Date: Nov 17, 2020 Product: IDEA Version: 211.SNAPSHOT Linker: Roman Vasiliev Plugin: Exception Analyzer 193.325 OS: Linux Java: JetBrains s.r.o. 11.0.9 Action: Refresh

```
java.lang.NoSuchMethodError: 'void com.intellij.unscramble.AnnotateStackTraceAction.<init>(com.intellij.o...
11.08.2019 Vasiliev  sisyphus StacktracePanel.<init>(StacktracePanel.java:115)
30.10.2019 Vasiliev  sisyphus IssuesForm.<init>(IssuesForm.java:105)
04.06.2020 Vasiliev  sisyphus WindowFactory.createToolWindowContent(WindowFactory.java:35)
13.01.2020 Krivosheev wm ToolWindowImpl.createContentIfNeeded$intellij_platform_ide_impl(ToolWindowImpl.kt:506)
13.01.2020 Krivosheev wm ToolWindowImpl.scheduleContentInitializationIfNeeded$intellij_platform_ide_impl(ToolWindowImpl
13.01.2020 Krivosheev wm ToolWindowManagerImpl.doShowWindow(ToolWindowManagerImpl.kt:982)
16.12.2019 Krivosheev wm ToolWindowManagerImpl.showToolWindowImpl(ToolWindowManagerImpl.kt:928)
```

Рис.: Плагин Exception Analyzer

## Исключения:

- Трассировка стека
- Исключения-причины (caused by) и подавленные исключения

## Сбои виртуальных машин (JBR):

- Трассировка стека
- Значения регистров
- Отображения из регистров в память
- Состояния потоков

## Зависания пользовательского интерфейса:

- Дерево зависания
- Кадру стека соответствует время

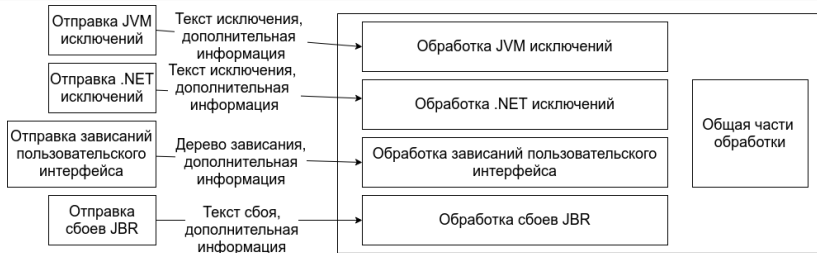
# Введение: обработка отчётов в Exception Analyzer

Часть сервера	Сбои	Исключения
Сравнение отчётов	Медленное	Быстрое
Группировка отчётов	Медленная	Быстрая
Поиск похожих отчётов	Упрощённый	Полный
Индексирование отчётов	Нет	Есть
Поиск по отчётам	Неполный	Полный

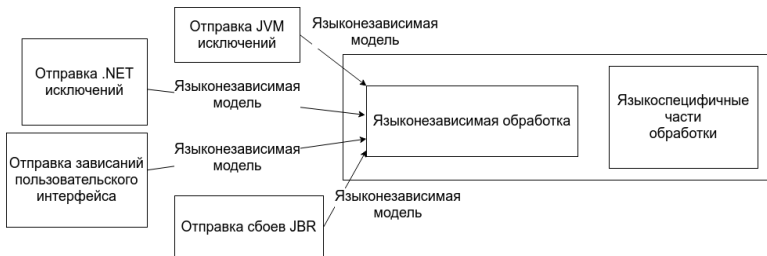
- Сейчас отчёты хранятся как текст трассировки стека с дополнительной информация об окружении. Поддерживаются только JVM-исключения, сбои JBR, .NET исключения, зависания пользовательского интерфейса
- Для добавлении нового типа отчётов приходится с нуля писать код по работе с ними

- В Exception Analyzer нет общей модели для работы с отчётами
- Многие алгоритмы по работе с отчётами языконезависимы
- Хочется сделать модель, подходящую для разных видов отчётов, для их отправки и обработки
- Формирование модели отчёта на стороне клиента позволит добавлять новые языки, не затрагивая сервер, и позволит пользователям самим управлять сравнением отчётов
- На стороне клиента можно получать больше информации об окружении (версии библиотек, источники кадров стека)

# Мотивация



**Рис.:** Отправка отчёта сейчас



**Рис.:** Отправка отчёта в языконезависимой модели

## Sentry<sup>1</sup>:

- Есть модель для исключений
- JSON-файл с покадровой информацией об исключении и вспомогательной информацией
- Библиотеки-клиенты для популярных языков программирования (более 30)

## Другие системы<sup>2</sup>:


- Имеют похожую на Sentry модель отчёта
- Модели подходят для описания исключений

## Но:

- Модели поддерживают только исключения
- В моделях недостаточно данных для алгоритмов Exception Analyzer (подсистемы, связь с исходным кодом, связь со вложениями)

---

<sup>1</sup><https://github.com/getsentry/sentry>

<sup>2</sup><https://github.com/airbrake>, <https://github.com/MindscapeHQ> 



**Цель работы:** разработать модель для языконезависимой обработки отчётов и внедрить её в Exception Analyzer.

**Задачи:**

- Разработать модель, в рамках которой возможно реализовать всю текущую функциональность Exception Analyzer.
- Реализовать на сервере обработку отчётов в языконезависимом формате.
- Реализовать библиотеку-клиент для JVM-языков, с использованием которой возможна отправка отчётов в языконезависимом формате из IntelliJ IDEA.

# Разработка модели: представление ошибок

```
"errors": [  
  {  
    // тип, метка и имя -- идентификатор  
    "type": "jvm.exception.trace", "label": "main", "name": "some.Exception",  
    "attachment": "stacktrace.txt",  
    "structure": {  
      "type": "sequence", // тип структуры. Последовательность, дерево...  
      "elements": [  
        { // только самая важная информация об ошибке: сравнение и навигация  
          "type": "jvm.frame",  
          // идентификатор (для сравнения)  
          "id": "com.intellij.openapi.diff.impl.dir.DirDiffTableModel.lambda$reloadModel$0",  
          "name": "DirDiffTableModel.lambda$reloadModel$0", // человекочитаемое имя  
          "textReference": { "start": 2, "end": 3 } // ссылка на вложение  
        },  
        ...  
      ]  
    }  
  },  
  { // ошибок может быть несколько  
    "type": "jvm.exception.trace", "label": "caused1", "name": "java.lang.NullPointerException",  
    "attachment": "stacktrace.txt",  
    "structure": {  
      "type": "sequence"
```



# Разработка модели: символы, окружение, метайнформация

```
"meta": { "protocol": "0.0.1_jvm", "reporter": "42" }, // метайнформация
"properties": { "attachments": ["stacktrace.txt", "message.txt"] },
"environment": [ // окружение: ОС, среда выполнения, библиотеки, плагины...
{
    "type": "IDE",
    "id": "IntelliJ IDEA 212.2345",
    "name": "IntelliJ IDEA", "version": "212.2345"
}, ...
],
"symbols": [ // символы: связь с исходным кодом
{
    "type": "jvm.frame",
    "elementId": "com.intellij.openapi.SomeClass.someMethod", // ссылка на элемент ошибки
    "environmentItemId": "IntelliJ IDEA 212.2345", // ссылка на элемент окружения
    "fileName": "SomeClass.java",
    "line": 251,
    "subsystems": [ // подсистемы
        "com.intellij.openapi.SomeClass", "com.intellij.openapi", "com.intellij", "com"
    ]
},
...
]
```

# Работа с моделью на сервере: сбои JBR

- Перевод из старого представления в новую модель
- Адаптер поверх модели
- Языконезависимая реализация алгоритмов, используемых для сравнения исключений
- Вся имеющаяся функциональность сбоев JBR была поддержана с использованием модели и протестирована. Время обработки отличается незначительно

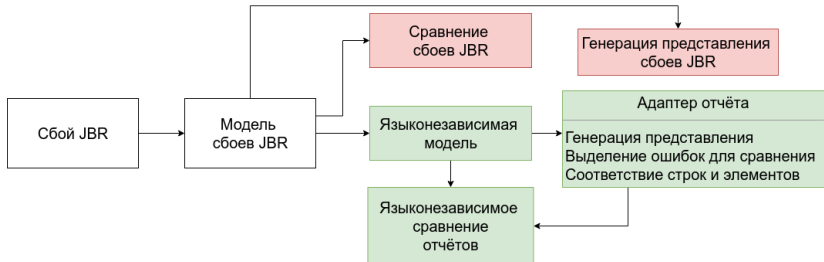


Рис.: использование модели для обработки сбоев JBR

# Работа с моделью на сервере: улучшение поддержки сбоев JBR

- Переиспользование инфраструктуры индексирования JVM исключений для сбоев
- Построение бинарного представления ошибки
- Использование индекса и бинарного представления для ускорения сравнения (ускорение в 4-8 раз относительно языкозависимой реализации,  $\approx 1200-1800$  сравнений в секунду)
- Языконезависимые алгоритмы визуализации сравнения и выделения важных элементов

# Работа с моделью на сервере: исключения и зависания

- Обобщены все алгоритмы, используемые для обработки исключений и зависаний (сравнения, кластеризация, удаление рекурсии, ...)
- Языконезависимые алгоритмы сравнения протестированы на 10 000 отчётах из базы данных (JVM, .NET, JBR), на JVM и .NET исключениях результаты совпадают с прошлой реализацией

Клиент-библиотека для отправки отчётов о необработанных исключениях

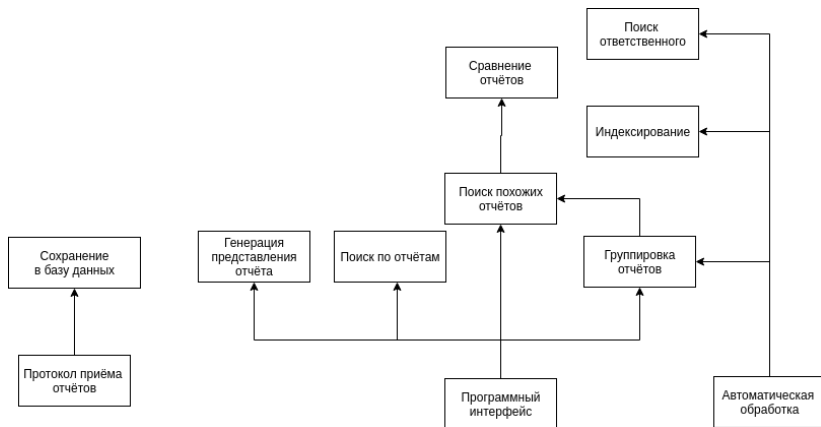
- Общая часть (Kotlin Multiplatform)
  - Будет переиспользована для JavaScript исключений
  - Преобразование разобранной трассировки в вид модели
  - Установка опциональных полей
  - Отправка отчёта
- Языкозависимая часть
  - Разбор исключения
  - Выделение дополнительной информации (проект, плагин, свойства системы)

Размер отчётов увеличился в 2-4 раза.

- Разработана модель отчётов для Exception Analyzer, позволяющая поддержать  $\geq 30$  языков.
- Обработка отчётов на сервере реализована в языконезависимом стиле.
- Улучшена обработка сбоев JBR. Их сравнение ускорилось в 4-8 раз.
- Реализована библиотека-клиент для исключений из JVM-языков. Размер отчётов увеличился в 2-4 раза.
- Отправка отчётов в новой модели будет внедрена в одну из ближайших версий IntelliJ IDEA.

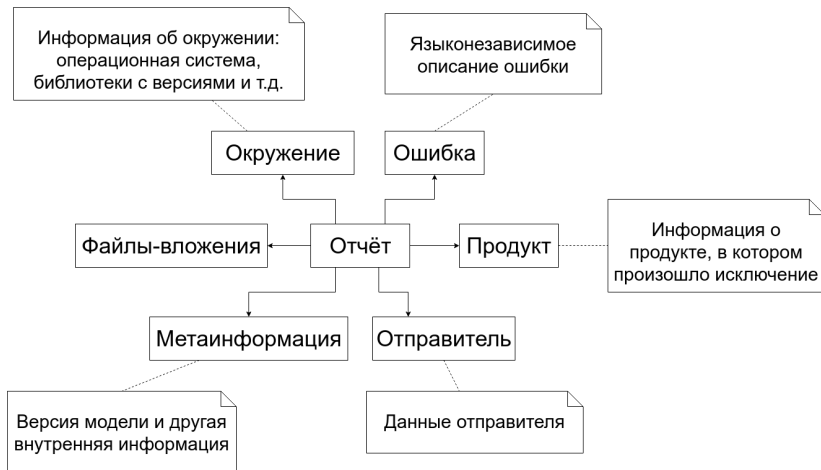


# Обработка отчётов в Exception Analyzer



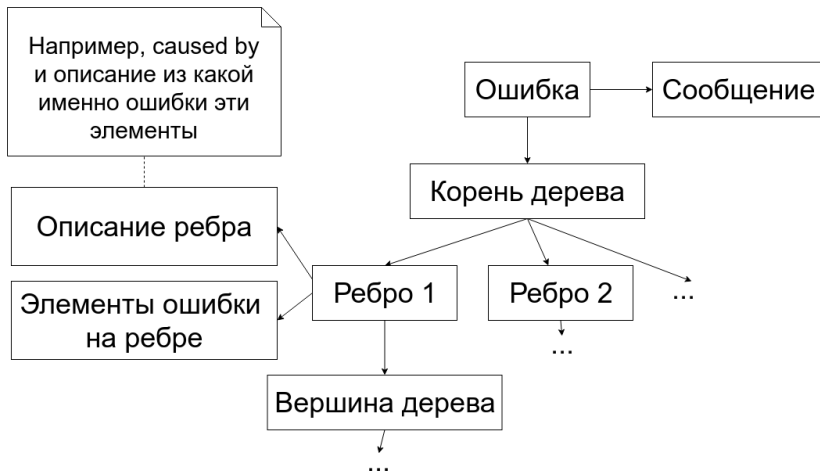
**Рис.:** Основные компоненты Exception Analyzer, связанные с обработкой отчётов

# Модель 1



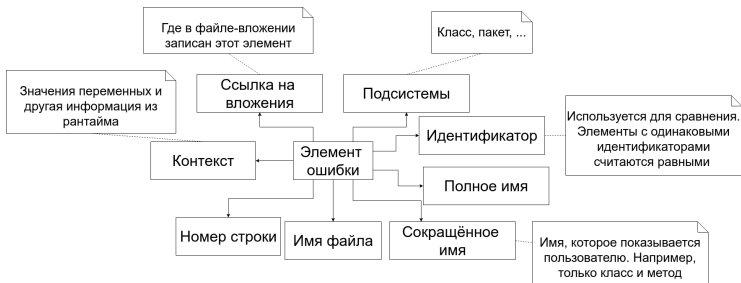
**Рис.:** Первая версия модели

# Модель 1: ошибка



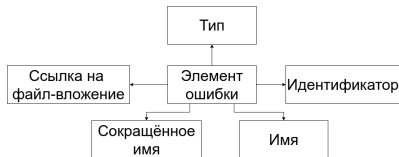
**Рис.:** Ошибка в первой версии модели

# Модель 1: элемент

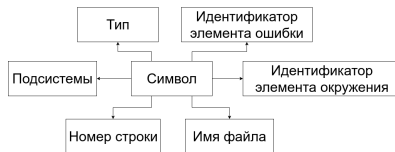


**Рис.:** Элемент ошибки в первой версии модели

# Модель 2: символ и элемент

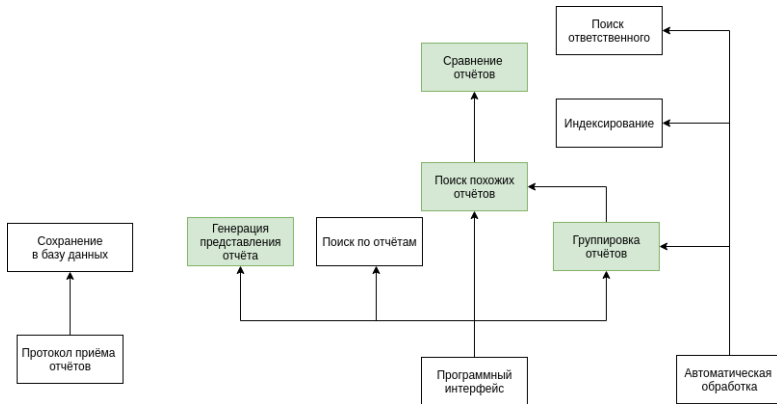


**Рис.:** Элемент ошибки



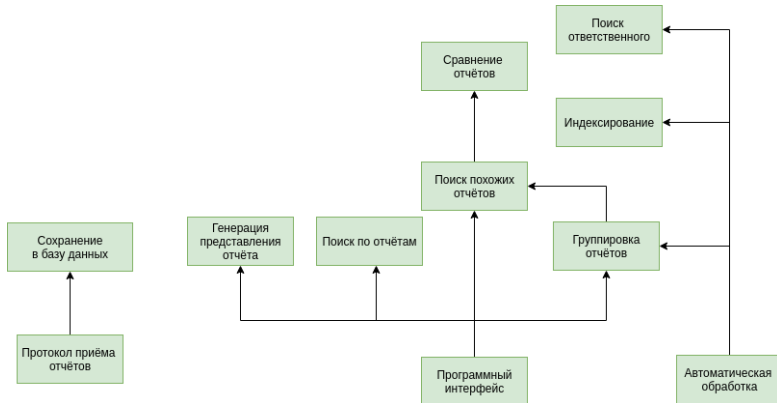
**Рис.:** Символ

# Поддержка сбоев JBR (сервер)



**Рис.:** Требуемые изменения для поддержания работы со сбоями JBR в рамках модели

## Поддержка модели для исключений и зависимостей пользовательского интерфейса (сервер)



**Рис.:** Требуемые изменения для поддержания работы с моделью на всём сервере

# Введение: сравнение отчётов в Exception Analyzer

- Используются алгоритмы на основе косинусного сходства<sup>1</sup>, TF-IDF<sup>2</sup> и взвешенного редакционного расстояния<sup>3</sup>
- Оптимизация: сравнивать не кадры стека, а их пакеты или классы (в других языках модули, подсистемы)

---

<sup>1</sup>P. Runeson et al. (2007) "Detection of duplicate defect reports using natural language processing". In: Proceedings of the 29th international conference on Software Engineering

<sup>2</sup>Lerch J, Mezini M (2013) "Finding duplicates of your yet unwritten bug report". In: CSMR 2013: 17th European conference on software maintenance and reengineering. IEEE, pp 69-78

<sup>3</sup>Vasiliev R. et al. (2020) "TraceSim: A Method for Calculating Stack Trace Similarity". In: MaLTesQuE 2020: Proceedings of the 4th ACM SIGSOFT International Workshop on Machine-Learning Techniques for Software-Quality Evaluation



# Введение: исключения в разных языках программирования

- Сообщение об ошибке
- Трассировка стека
  - Для каждого кадра стека указываются:
    - Имя функции/метода, класса, файла
    - Номер строки в файле
    - ...
- Исключения-причины (caused by) и подавленные исключения

```
ru.spbhse.exceptions.test.WrappedException: java.lang.IndexOutOfBoundsException:
    index = 20 is greater than collection size (9)
    at ru.spbhse.exceptions.test.MainKt.bar(Main.kt:13)
    at ru.spbhse.exceptions.test.MainKt.foo(Main.kt:18)
    at ru.spbhse.exceptions.test.MainKt.main(Main.kt:23)
    at ru.spbhse.exceptions.test.MainKt.main(Main.kt)
Caused by: java.lang.IndexOutOfBoundsException: index = 20 is greater than collection
    size (9)
    at ru.spbhse.exceptions.test.FooBar.fooBar(FooBar.kt:6)
    at ru.spbhse.exceptions.test.MainKt.baz(Main.kt:6)
    at ru.spbhse.exceptions.test.MainKt.bar(Main.kt:11)
    ... 3 more
```

**Листинг 1:** Пример исключения в языке Kotlin

# Введение: отчёты об ошибках

- Отправляемые вручную
  - Используются системы для отслеживания проблем (YouTrack, Jira, Bugzilla, ...)
  - Пользователь должен сам описать проблему и приложить нужные файлы
  - Часто пользователи предоставляют недостаточно информации<sup>1</sup>
- Автоматически сгенерированные
  - Требуют минимальных действий от пользователя
  - Позволяют получать больше отчётов
  - Отчёты о необработанных исключениях, сбоях виртуальных машин, зависаниях пользовательского интерфейса, ...

---

<sup>1</sup>N. Bettenburg, S. Just, A. Schröter, C. Weiss, R. Premraj, and T. Zimmermann (2008) "What makes a good bug report?" in Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, 2008, pp. 308–318.

# Sentry 1

```
"sentry.interfaces.Exception": {
  "exc_omitted": null,
  "values": [
    {
      "mechanism": null,
      "module": "io.sentry.example",
      "raw_stacktrace": null,
      "stacktrace": {
        "frames": [
          {
            "abs_path": "Thread.java",
            "filename": "Thread.java",
            "function": "run",
            "in_app": false,
            "lineno": 748,
            "module": "java.lang.Thread"
          },
          ...
        ],
        "frames_omitted": null,
        "registers": null
      },
      "thread_id": null,
      "type": "ApiException",
      "value": "Authentication failed, token expired!"
    }
  ]
}
```

# Sentry 2

```
"platform": "java",
"release": "3.0.2",
"sdk": {
  "client_ip": "172.56.7.13",
  "name": "sentry-java",
  "version": "1.4.0-3ded0"
},
"sentry.interfaces.Breadcrumbs": {
  "values": [
    {
      "level": "debug",
      "message": "Querying for user.",
      "timestamp": 1501799139,
      "type": "default"
    },
    ...
  ]
}
```

# Sentry 3

```
"platform": "java",
"release": "3.0.2",
"sdk": {
  "client_ip": "172.56.7.13",
  "name": "sentry-java",
  "version": "1.4.0-3ded0"
},
"sentry.interfaces.Breadcrumbs": {
  "values": [
    {
      "level": "debug",
      "message": "Querying for user.",
      "timestamp": 1501799139,
      "type": "default"
    },
    ...
  ]
}
```

# Sentry 4

```
"sentry.interfaces.Http": {
  "data": "{\"logged_in\":[\"1\"]}\",
  "env": {
    "AUTH_TYPE": null,
    "LOCAL_ADDR": "0:0:0:0:0:0:0:1",
    "LOCAL_NAME": "localhost"
  },
  "headers": [
    [
      "Accept",
      "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*"
    ],
    ...
  ],
  "method": "GET",
  "query_string": "logged_in=1",
  "url": "http://localhost:8080/"
},
"sentry.interfaces.Message": {
  "message": "Authentication failed, token expired!"
},
```