Introduction to Sentry

The modern error logging and aggregation platform

@mozillazg

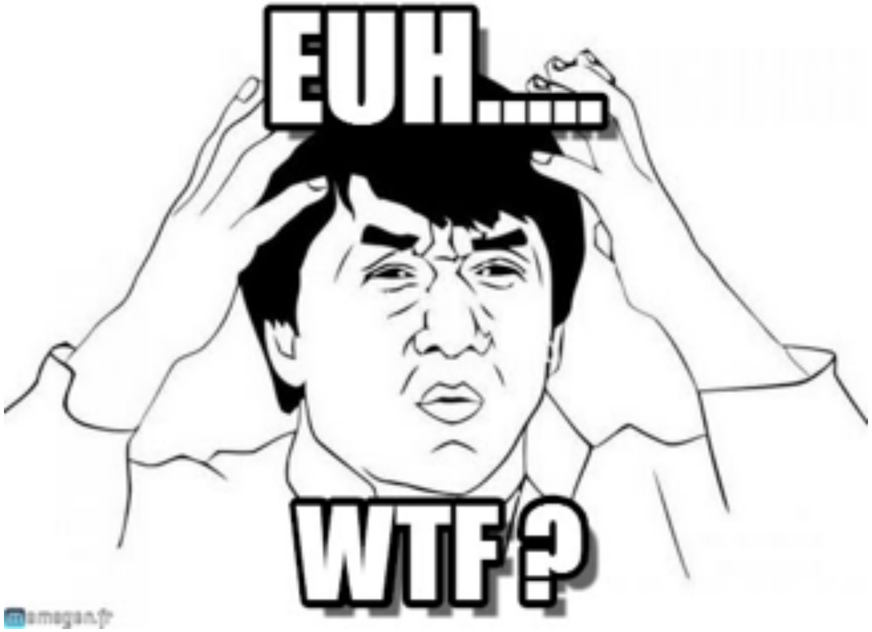
大纲

- 想解决的问题
- 为什么要用 Sentry
- Sentry 简介
- Hello World

想解决的问题

想解决的问题

- 什么？有客户说出问题了?
- 错误日志中信息这么少，怎么查?



为什么要用 Sentry

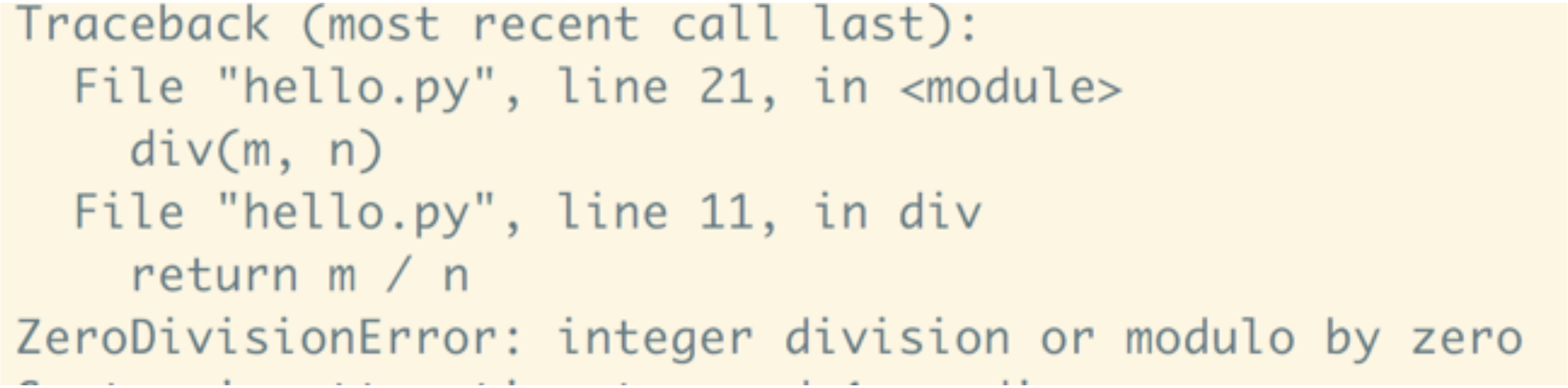
为什么要用 Sentry

- **告警通知功能**
- **包含详细的错误信息（出现异常位置的代码）**
- **包含错误堆栈中每一行代码变量的值**
- 客户端侵入性小（只需增加几行代码即可）
- 支持多种编程语言

  * 支持公司技术栈所用到的所有编程语言

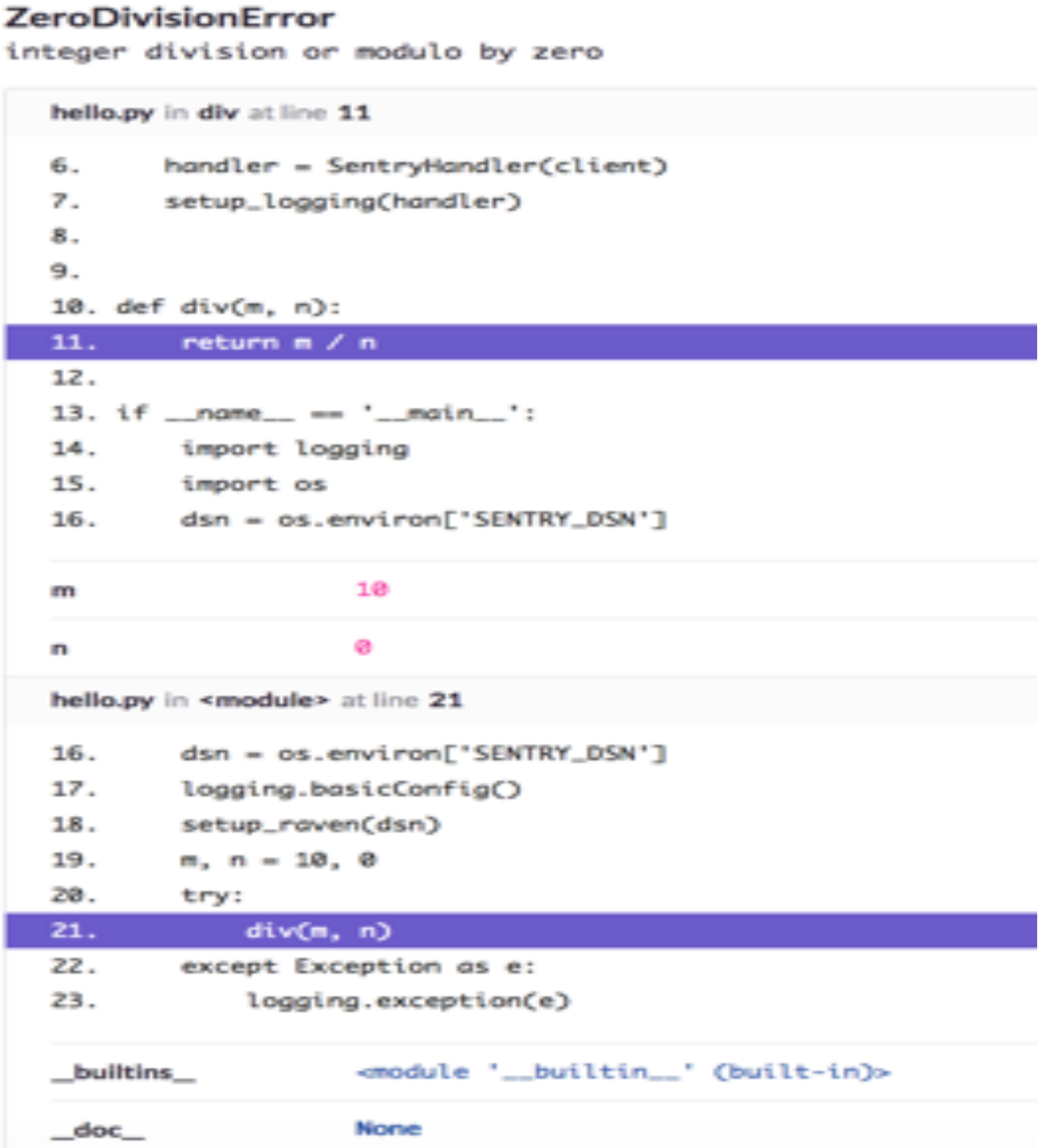
  * 尤其是对 Python 支持好

Without Sentry



```
Traceback (most recent call last):
  File "hello.py", line 21, in <module>
    div(m, n)
  File "hello.py", line 11, in div
    return m / n
ZeroDivisionError: integer division or modulo by zero
```

With Sentry



ZeroDivisionError

integer division or modulo by zero

hello.py in **div** at line **11**

```
6.      handler = SentryHandler(client)
7.      setup_logging(handler)
8.
9.
10. def div(m, n):
11.     return m / n
12.
13. if __name__ == '__main__':
14.     import logging
15.     import os
16.     dsn = os.environ['SENTRY_DSN']
```

```
m                   10

n                   0
```

hello.py in **<module>** at line **21**

```
16.     dsn = os.environ['SENTRY_DSN']
17.     logging.basicConfig()
18.     setup_raven(dsn)
19.     m, n = 10, 0
20.     try:
21.         div(m, n)
22.     except Exception as e:
23.         logging.exception(e)
```

```
__builtins__        <module '__builtin__' (built-in)>

__doc__             None
```

Sentry 简介

Sentry 简介

- 一个错误记录和汇聚平台
- 始于 2010 由 Disqus 开源
- 源码地址：https://github.com/getsentry
- 服务端使用 Python 开发（https://github.com/getsentry/sentry watch: 458, star: 10940, fork: 1302 commits: 17140, contributors: 276, master update：2016-11-16, 截止到 2016-11-16）
- 官方支持多种编程语言的客户端: Python/JavaScript/Node.js/Java/Go/PHP/Ruby/…

Hello World

Hello World(Python)

```
def setup_raven(dsn):
    from raven import Client
    from raven.conf import setup_logging
    from raven.handlers.logging import SentryHandler
    client = Client(dsn)
    handler = SentryHandler(client)
    setup_logging(handler)


def div(m, n):
    return m / n

if __name__ == '__main__':
    import logging
    import os
    dsn = os.environ['SENTRY_DSN']
    logging.basicConfig()
    setup_raven(dsn)
    m, n = 10, 0
    try:
        div(m, n)
    except Exception as e:
        logging.exception(e)
```

Hello World(Flask)

```
from flask import Flask

app = Flask(__name__)


def setup_raven(dsn, app):
    from raven.contrib.flask import Sentry
    sentry = Sentry(dsn=dsn)
    sentry.init_app(app)


@app.route('/div/<int:m>/<int:n>')
def div(m, n):
    return m / n

if __name__ == '__main__':
    import os
    dsn = os.environ['SENTRY_DSN']
    setup_raven(dsn, app)
    app.run()
```

Hello World(Node.js)

```
•   var app = require('express')();
•   var raven = require('raven');

•   var raven_dsn = process.env.SENTRY_DSN;

•   // The request handler must be the first item
•   app.use(raven.middleware.express.requestHandler(raven_dsn));

•   app.get('/div/:m/:n', function mainHandler(req, res) {
•       var m = parseInt(req.params.m);
•       var n = parseInt(req.params.n);
•       var result = m / n;
•       throw new Error('Broke!');
•   });

•   // The error handler must be before any other error middleware
•   app.use(raven.middleware.express.errorHandler(raven_dsn));

•   app.listen(3000);
```

Thank You!