



# SENTRY

as a way not to be afraid

Do you know about Sentry?

Do you know you can use it for free?

Do you know you can self-host it?

Do you use Sentry?

My toy project problems:

- unexpected SQLite behaviour

(we always don't know something about our tools)

- unexpected user behaviour

(there will always be a way to break our application)

## Disadvantages of logs:

- several machines + different environments
- noisy (“expectable errors” along with important ones)

## In the end:

- no one looks through logs for a long time
- rare mistakes are lost

and not everything  
settles in the logs:  
query string VS body

Notifications?

No, Sentry won't bombard you with notifications:

- only first time and resolved errors will push notifications
- notification frequency is configurable

and you can integrate  
with Slack, Telegram etc

To avoid another “dead logs” instance:

- from time to time fix even non-critical bugs
- set "resolved" status automatically in (two) days after the last error occurrence

Sentry allows:

- to be notified about errors instantly as soon as they occur
- not to crawl through the logs with `tail+grep`
- not to search necessary logs on different servers
- not to miss an important new error or regression

# Sentry could be used for free

(you could use the cloud or self-host it)



## Some Sentry use cases:

- monitor problems in third-party services
- monitor known errors
- don't be afraid to do refactoring
- track obsolete clients
- some kind of monitoring and statistics gathering

Divide errors into groups:

```
def before_send(event, hint):  
    if hint and hasattr(hint['log_record'], 'fingerprint'):  
        event['fingerprint'] = hint['log_record'].fingerprint  
    return event
```

```
sentry_sdk.init(  
    dsn='https://<key>@sentry.io/<project>',  
    integrations=[FlaskIntegration()],  
    before_send=before_send  
)
```

```
logger.error('Special users! uid=%s', uid,  
            extra={'fingerprint': ['oh-users', uid]})
```

## Some Sentry use cases:

- monitor problems in third-party services
- monitor known errors
- don't be afraid to do refactoring
- track obsolete clients
- some kind of monitoring and statistics gathering

Front-end too

use Sentry

=

know what's going on with your project

+

be sure it works

+

don't be afraid to make changes

integrate Sentry in the  
development phase

Used:

- <https://sentry.io/>
- <https://develop.sentry.dev/self-hosted/>
- <https://github.com/baverman/flask-app-template>
- <http://flask.palletsprojects.com/en/1.1.x/>
- <https://uwsgi-docs.readthedocs.io/en/latest/>

@lauvenhelz

@lautips