

# Logging the right way

And why it's important.  
... with ...



# Niranjana Singh

Backend Developer.

*Wanderer in Day,  
Coder in Night*

- Using Django for last 4 years.
  - Engaged with python.
  - Working with Aasaanjobs.com  
(OLX Group)
-

# Contents

- Why bother with logging, and what's it for ?
- Implementation in Django.
- Making logging meaningful and useful.
- Leveraging python default logging.
- Using django\_easy\_logger and using its powers.
- Q & A

# Why Logging ?

*You are gonna realize it when your code fails.*

1. Helps in debugging.
2. Tracking code and understanding what your code does.
3. Helping others to understand what your code does.
4. Optimize your existing code.

# Implementation in Django

- Django uses python built-in module to perform system logging.
- Different logging levels – Info, warning, exception, error, critical.
- Configuring Logging.
  - Handlers
  - Logging destination.
  - Defining logging format

# Meaningful logging

- Write them for others.
- Logging should be story for any user session/ request-response cycle.
- More than words data points matter.
- Use common semantics in logging across microservices.
- Don't over do it.

# Leveraging python logging

- Using multiple handlers.
- Custom Logging formatters.
- Problems you might face while writing logs.
  - What to do to log a particular value every time throughout your project for ex. User\_id ?
  - Logging all helper functions.
  - Hiding sensitive data while logging.

# Django\_easy\_logger

1. Defining global format in settings.
2. Defining global log variables.
3. Hiding sensitive data to get logged.
4. Function Logger.



**Thank you**

---