

# Logging with Django

Fun with Examples

---

```
import logging
logger = logging.getLogger('springboard')
logger.info('Name: {}'.format('Indrajit'))
```

# Logging components

- **Loggers**
  - Interface for logging from code
- **Handlers**
  - Route the logs to the right destination
- **Filters**
  - Fine grained controls. Allows skipping/modifying of message before logging
- **Formatters**
  - Specify formatting and metadata on each log record

# Logging Levels

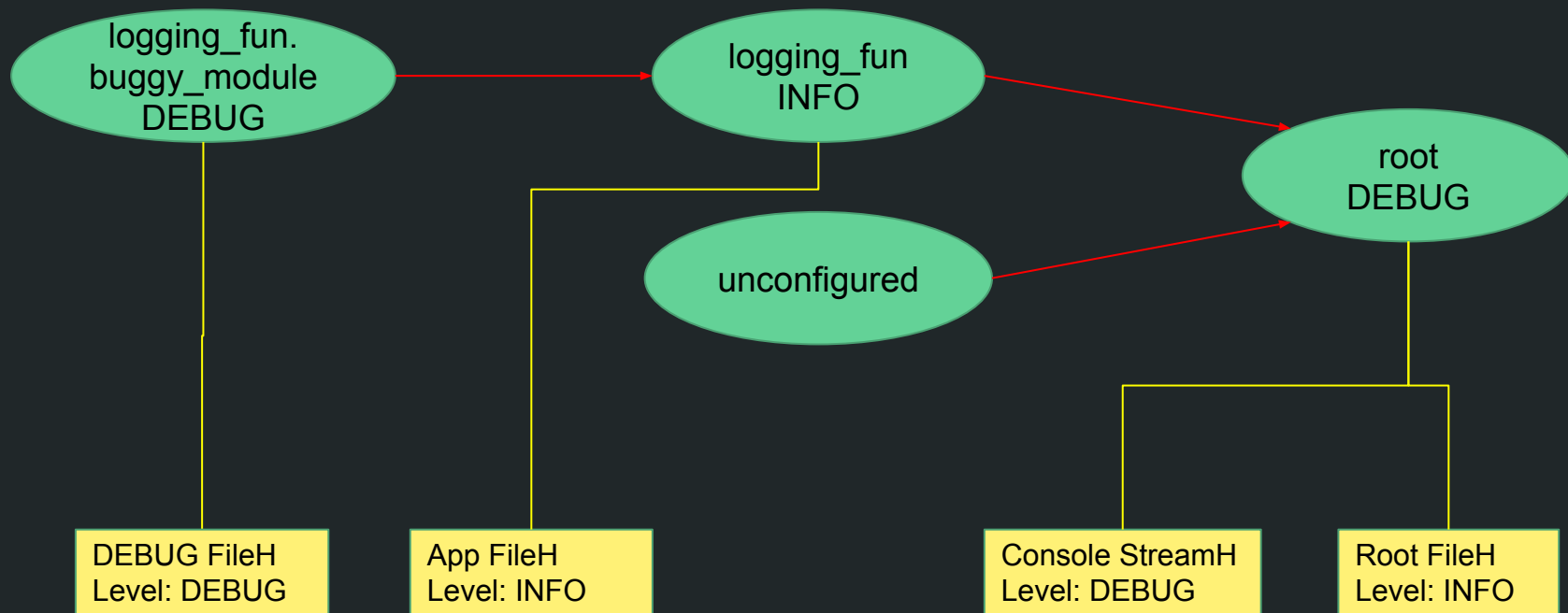
- **DEBUG:** Low level system information for debugging purposes
  - `logger.debug()`
- **INFO:** General system information
  - `logger.info()`
- **WARNING:** Information describing a minor problem that has occurred.
  - `logger.warning()`
- **ERROR:** Information describing a major problem that has occurred.
  - `logger.error()`
  - `logger.exception()`: Creates an ERROR level logging message wrapping the current exception stack frame.
- **CRITICAL:** Information describing a critical problem that has occurred.
  - `logger.critical()`

# Handlers

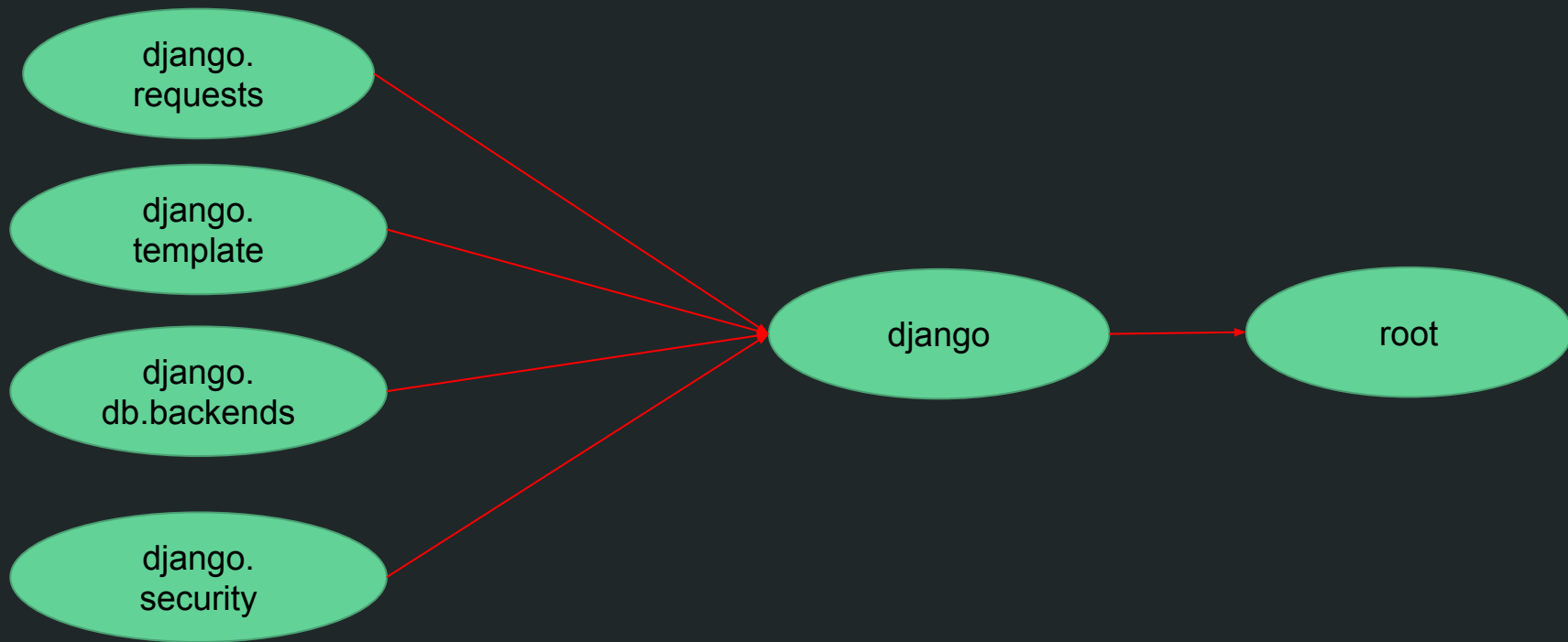
- NullHandler - absolutely nothing
- StreamHandler - sys.stdout, sys.stderr
- FileHandler - file
- RotatingFileHandler - file (rollover as file.1, file.2)
- TimedRotatingFileHandler - file (rollover after a specified time-period)
- AdminEmailHandler - sends an email to the site admins for each log
- .
- .
- .
- .
- .
- .

# Logging Architecture

1. Find the right Logger
2. Check log level and skip/continue
3. Look for handlers
4. For each handler, check level and log/skip
5. Propagate?



# Django Logging



# Formatters

- %(asctime)s
- %(levelname)s
- %(lineno)d
- %(module)s
- %(process)d
- %(thread)d
- django-request-id

<https://docs.python.org/3/library/logging.html#logrecord-attributes>

# Filters

- %(asctime)s
- %(levelname)s
- %(lineno)d
- %(module)s
- %(process)d
- %(thread)d
- django-request-id

<https://docs.python.org/3/library/logging.html#logrecord-attributes>



# Thank you!

---

Q&A