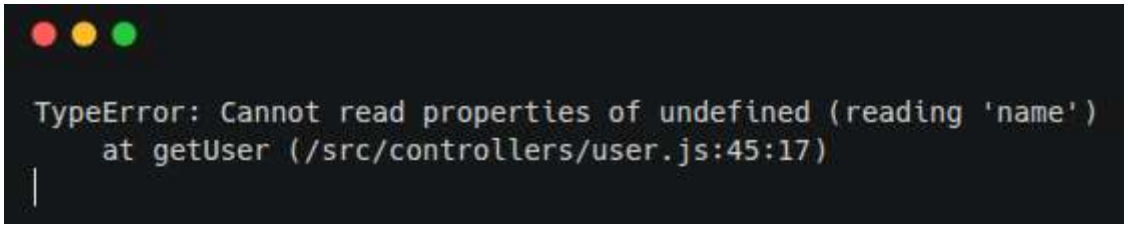
Difference Between Development, Test, Staging & Production Environments

Development Environment

- Used by developers to write, debug, and build code
- Frequent code changes; unstable but flexible
- Contains mock data or local databases
- Tools: IDEs, debuggers, version control branches

Errors in Development environment



Errors in Development environment

Testing Tools Used in Development Environment

- **Unit Testing Frameworks**
  - JavaScript / TypeScript - Jest
  - Python - PyTest
  - Java - JUnit
- **API Testing Tools**
  - Postman
  - Swagger
- **Static Code Analysis Tools**
  - ESLint (JS/TS)
  - TS Error Checker (TypeScript Compiler)
  - Pylint (Python)

Testing Tools Used in Development Environment

- **Build & CI Tools**
  - Jenkins
  - GitHub Actions
- **Debugging**
  - IDE

Test Environment

- Used by QA/Testers to validate functionality
- Contains controlled test data
- Runs automated/manual tests
- More stable than Dev but still isolated
- QA engineers perform functional, integration, performance, and security testing.
- Errors are shown for debugging purpose.

Test Environment tools

- **Functional Testing Tools**
  - **E.g., selenium**
  - **Test end-to-end features.**
- **API Testing Tools**
  - **QA uses these for validating backend services.**
  - **e.g., Postman**
- **Performance Testing Tools**
  - **To measure speed, load, scalability, and stress behavior.**
  - **e.g., JMeter**
- **Bug Tracking Tools**
  - **logging, tracking, assigning, and verifying bugs.**
  - **Jira**

Staging Environment

- Replica of production environment
- Used for final verification before deployment
- Realistic data & configuration settings
- Used for UAT, performance tests, release testing

Staging Environment Errors

- Sensitive internal details are hidden
- Errors are stored in monitoring tools instead of printed on screen

Staging Environment Errors

- **Hide sensitive information (Dev vs Staging)**

```
TypeError: Cannot read properties of undefined (reading 'name')
    at getUser (/src/controllers/user.js:45:17)
```

Staging Environment Errors

- Log errors in the server.

Testing Tools in Staging Env

- Performance & Load Testing Tools
  - Used to validate system behavior under almost real traffic before deployment.
  - Ensure staging can handle production-level load.
  - JMeter
- Integration & End-to-End Testing Tools
  - Selenium
- API Contract Testing Tools
  - Check if the backend and frontend meet agreed API contracts before production.
  - Postman (automated collections)

Production Environment

- Live environment used by real end users
- High stability, uptime, and performance required
- Contains real customer data
- Errors here impact business and users directly
- Tested by end users or selected group of users(before release)

Production Environment Errors

- User-Friendly Error Message (NO technical details)
- This prevents leaking:
  - Stack traces
  - File names
  - Server/internal logic
  - Sensitive data

Production Environment Errors

Production Environment Errors

Dev

- Detailed error message with stack trace
- Contains route name, file, line number

Production

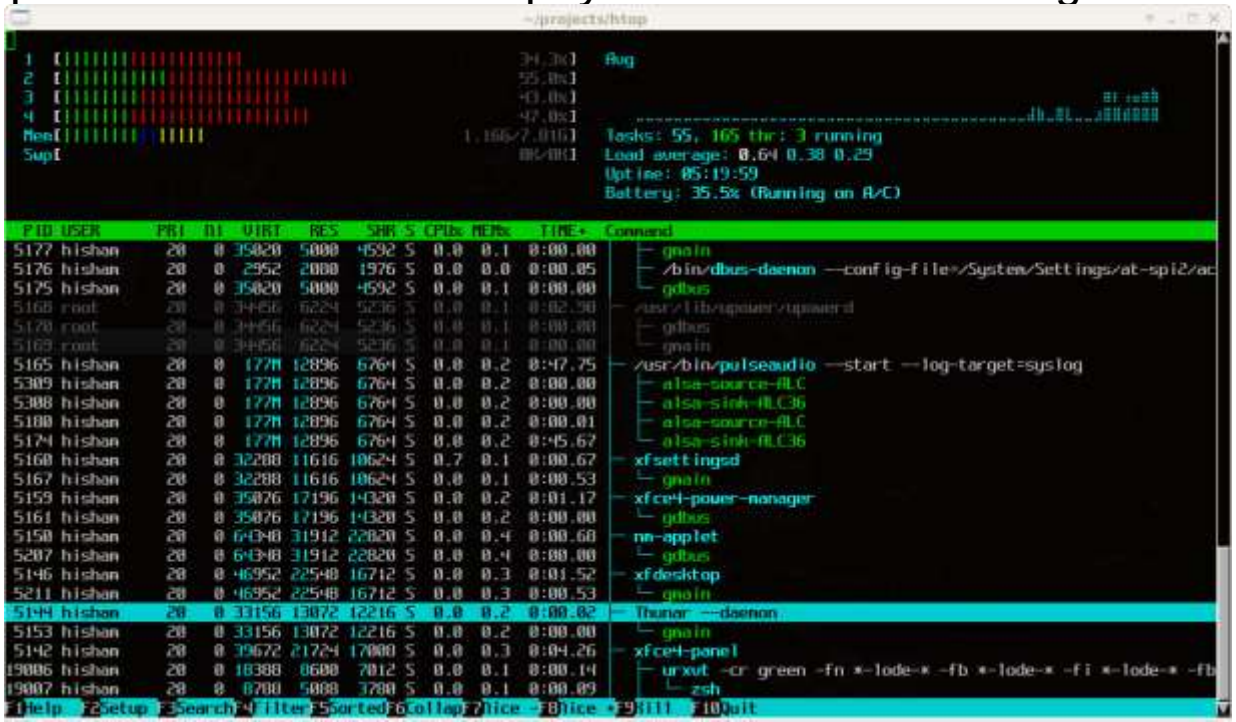
- Clean 404 page, no technical info
- Log errors internally on the server.

Production Environment Testing tools

- Production uses monitoring, observability, on-device testing, synthetic checks, and error tracking tools
- AWS CloudWatch
- Azure Monitor

Production Environment Testing tools

E.g., the htop command-line tool for physical server monitoring



Overview

- Development → Frequently changing, unstable, developer-centric
- Testing → Stable for test cycles, tester-centric
- Staging → Near-production, pre-release validation
- Production → Stable, secure, real users & data