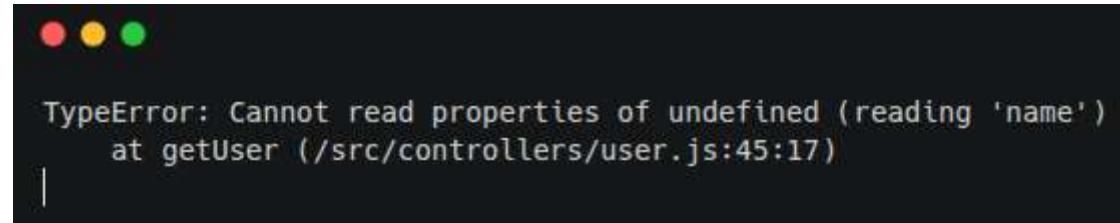


Difference Between Development, Test, Staging & Production Environments

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



A screenshot of a terminal window on a dark background. In the top-left corner, there are three small colored circles: red, yellow, and green. The main area of the terminal displays the following error message:

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

Errors in Development environment

```
{  
  "status": 500,  
  "message": "Database connection failed"  
}
```

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

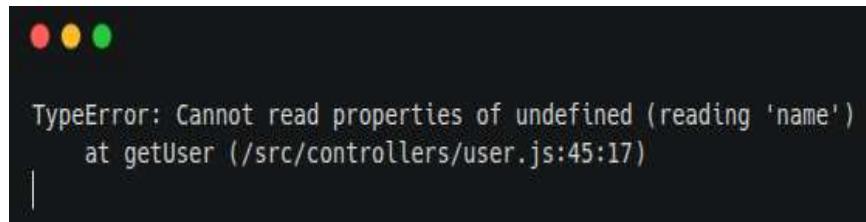
- Replica of production environment
- Used for final verification before deployment
- Realistic data & configuration settings
- Used for UAT, performance tests, release testing

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)
```



```
{
  "statusCode": 500,
  "message": "Internal server error",
  "error": "Server Error"
}
```

Staging Environment Errors

- Log errors in the server.



A screenshot of a terminal window with a dark background. In the top left corner, there are three colored dots: red, yellow, and green. Below them, a code snippet is displayed:

```
catch(exception)
{
    console.log("Error:", exception)
}
```

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

Bad gateway Error code 502

Visit [cloudflare.com](#) for more information.
2024-01-21 03:40:09 UTC

The diagram illustrates the flow of data from the user's browser to the host server. It shows three nodes: 'You' (Browser) which is 'Working' (green checkmark), 'Cloudflare' (Working) which is also 'Working' (green checkmark), and 'Host' which has an 'Error' (red X). The 'Host' node is preceded by a black redacted bar.

You
Browser
Working

Amsterdam
Cloudflare
Working

Host
Error

What happened?
The web server reported a bad gateway error.

What can I do?
Please try again in a few minutes.

Production Environment Errors

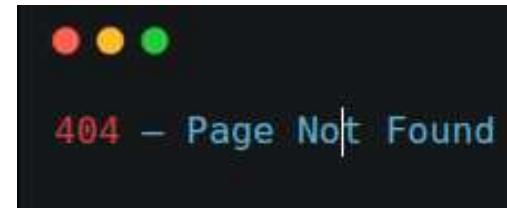
Dev

- 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

The screenshot shows the htop command-line interface running on a terminal window. The title bar indicates the current directory is ~[project]/bin/htop. The top section displays system statistics: CPU usage (1.16% / 7.01%), tasks (55 total, 3 running), load average (0.64, 0.38, 0.29), uptime (05:19:59), and battery status (35.5% running on A/C). Below this is a detailed process list with columns for PID, USER, PR, NI, VIRT, RES, SHR, S, CPU%, %MEM, TIME+, and Command. The processes listed include various system daemons like gmain, abindbus-daemon, gibus, pulseaudio, xfsettingsd, xfce4-power-manager, nm-applet, xfdesktop, and Thunar-daemon, along with user sessions (hishan) and system services (root). The bottom of the screen features a menu bar with options like Help, F2:setup, Search, F4:filter, F5:sorted, F6:collapse, F8:rice, F9:full, and F10:quit.

PID	USER	PR	NI	VIRT	RES	SHR	S	CPU%	%MEM	TIME+	Command
5177	hishan	20	0	35820	5000	4592	S	0.0	0.1	0:00.00	gmain
5176	hishan	20	0	2952	2000	1976	S	0.0	0.0	0:00.05	/bin/dbus-daemon --config-file=/System/Settings/at-spi2/ac
5175	hishan	20	0	35820	5000	4592	S	0.0	0.1	0:00.00	gibus
5168	root	20	0	34956	6224	5236	S	0.0	0.1	0:02.30	/usr/lib/pulseaudio/pulseaudio
5178	root	20	0	34956	6224	5236	S	0.0	0.1	0:00.00	gibus
5169	root	20	0	34956	6224	5236	S	0.0	0.1	0:00.00	gmain
5165	hishan	20	0	1770	12896	6764	S	0.0	0.2	0:47.75	/usr/bin/pulseaudio --start --log-target=syslog
5309	hishan	20	0	1770	12896	6764	S	0.0	0.2	0:00.00	alsa-source-ALC
5308	hishan	20	0	1770	12896	6764	S	0.0	0.2	0:00.00	alsa-sink-ALC36
5100	hishan	20	0	1770	12896	6764	S	0.0	0.2	0:00.01	alsa-source-ALC
5174	hishan	20	0	1770	12896	6764	S	0.0	0.2	0:45.67	alsa-sink-ALC36
5168	hishan	20	0	32288	11616	10624	S	0.7	0.1	0:00.67	xfsettingsd
5167	hishan	20	0	32288	11616	10624	S	0.0	0.1	0:00.53	gnome
5159	hishan	20	0	35826	17196	14320	S	0.0	0.2	0:01.12	xfce4-power-manager
5161	hishan	20	0	35876	17196	14320	S	0.0	0.2	0:00.00	gnome
5150	hishan	20	0	64048	31912	20820	S	0.0	0.4	0:00.68	nm-applet
5207	hishan	20	0	64348	31912	22820	S	0.0	0.4	0:00.00	gnome
5146	hishan	20	0	46952	22548	16712	S	0.0	0.3	0:01.52	xfdesktop
5211	hishan	20	0	46952	22548	16712	S	0.0	0.3	0:00.53	gnome
5144	hishan	20	0	33156	13072	12216	S	0.0	0.2	0:00.02	Thunar --daemon
5153	hishan	20	0	33156	13072	12216	S	0.0	0.2	0:00.00	gnome
5142	hishan	20	0	39972	21724	17800	S	0.0	0.3	0:04.26	xfce4-panel
19805	hishan	20	0	18388	0680	7012	S	0.0	0.1	0:00.14	urxvt -cr green -fn *--lode-* -fb *--lode-* -fi *--lode-* -fb
19807	hishan	20	0	8700	5008	3780	S	0.0	0.1	0:00.09	zsh

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