LOGGERS

→ Loggers / Logging is used for debugging purposes.

By using loggers we can see –

- which API users hit
- which data was processed
- where failures occured.
- who deleted/updated student.
- what went wrong.

→ There are **5 log levels**

LEVEL	USED for	Example
1 Trace	Very detailed internal flow	Variables changing inside loops.
2 Debug	Developer debugging info.	Entering Exiting Methods
3 Info	Important Applicat^n Events	Student created successfully.
4 Warn	Potential Issues	"Email format is Strange but allowed.
5 Error	Actual failure	Exceptions, DB failure.

90% of time we use Info, warn, Error only.

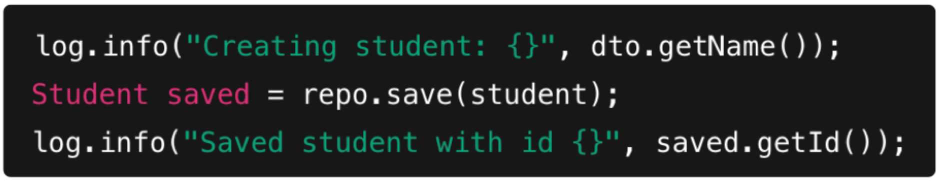
Where do we add logs?

(A) Service layer → Mandatory.

- Info logs → for something imp

- Info logs → for something imp.
- Warn logs → for suspicious activities
- Error logs → when something fails in service

Example →

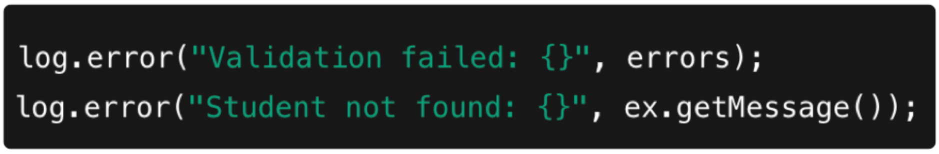


```
log.info("Creating student: {}", dto.getName());
Student saved = repo.save(student);
log.info("Saved student with id {}", saved.getId());
```

(B) Exception Handeling → Mandatory.

- This is where all errors land
- Product? debugging gets easy

Example →

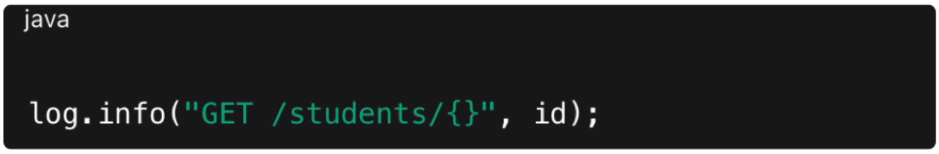


```
log.error("Validation failed: {}", errors);
log.error("Student not found: {}", ex.getMessage());
```

(C) In controller → Recommended.

- only used to request entry.

Example →



java

```
log.info("GET /students/{}", id);
```