



IBTISSAM OULAHCHOUM

WRITING CLEAN, SECURE

NODE.JS

APIS — A CHECKLIST
YOU'LL ACTUALLY USE!





IBTISSAM OULAHCHOUM

1. STRUCTURE YOUR PROJECT LIKE A PRO

CLEAN FOLDERS = CLEAN THINKING

- controllers/ — business logic
- routes/ — API endpoints
- services/ — data and external API handling
- middlewares/ — logic for validation, authentication, etc.
- models/ — database schemas
- utils/ — helper functions



Pro Tip:

Boring folder structures are the best ones!





IBTISSAM OULAHCHOUM

2. VALIDATE ALL INCOMING DATA

- Never trust user input (not even your own)
- Validate headers, queries, and bodies
- Use Joi, Zod, or express-validator

```
const Joi = require('joi');

const userSchema = Joi.object({
  email: Joi.string().email().required(),
  password: Joi.string().min(8).required(),
});
```





IBTISSAM OULAHCHOUM

3. HANDLE ERRORS LIKE A PRO

- Centralize error handling
- Never leak stack traces or sensitive info to users.
- Separate client (4xx) and server (5xx) errors



```
app.use((err, req, res, next) => {  
  console.error(err); // Log it  
  res.status(err.status || 500).json({  
    message: err.isOperational ? err.message : 'Something went wrong.',  
  });  
});
```





IBTISSAM OULAHCHOUM

4.SECURE YOUR API LIKE A BANK VAULT

- Helmet.js: Secure HTTP headers.
- Rate limiting: Prevent abuse (e.g., express-rate-limit).
- CORS: Strictly configure allowed origins.
- Authentication: Use JWT or OAuth2, not your own DIY tokens.
- Input Sanitization: Avoid SQL Injection / XSS (e.g., xss-clean)



```
const helmet = require('helmet');  
app.use(helmet());
```





IBTISSAM OULAHCHOUM

5. USE ENVIRONMENT VARIABLES

- Store secrets outside your code
- Use dotenv (or Convict)
- Never push .env files to GitHub

Tip:

- Your .env = your toothbrush 🪥 → Don't share it!





IBTISSAM OULAHCHOUM

6. VERSION YOUR API

- Prefix routes: /api/v1/users
- Plan backward compatibility
- Deprecate old versions gracefully — with warning headers or sunset notices.

7. WRITE TESTS (YES, YOU SHOULD)

- Unit tests for logic
- Integration tests for endpoints
- Tests = living documentation!





IBTISSAM OULAHCHOUM

8. LOG LIKE A DETECTIVE

- Use structured logging (e.g., Winston, Pino).
- Log important events: logins, failed requests, database errors.
- Avoid logging sensitive user data!

```
const winston = require('winston');

const logger = winston.createLogger({
  transports: [new winston.transports.Console()],
});
```





IBTISSAM OULAHCHOUM

9.KEEP DEPENDENCIES UP-TO-DATE

- Run npm audit regularly
- Use Dependabot / Snyk
- Stay on Node.js LTS versions

10.DOCUMENT YOUR API

- Use Swagger or Postman Collections
- Document authentication, errors, examples
- Update docs when you update your code



SECURE APIS AREN'T A TREND ,THEY'RE ESSENTIAL !!

👉 Share this checklist with your team!

👉 Got more tips? Drop them in the comments!



IBTISSAM
OULAHCHOUM