



Best Practices For Writing Automation Test Code

- Follow **Programming Language Guidelines**(method name-getPayload(), class name- APIEndpoints, package name-com.testingsumo.backend.tests, variable name-AuthToken)
- Follow OOps Concepts Wherever Possible- **Abstraction(base classes), Inheritance(multiple implementation of same things/multiple inheritance), Polymorphism(many forms with something different), Data Hiding(hide unnecessary/sensitive info), Encapsulation(Bind small entities into a single large entity)**
- Reduce code duplicacy (think before writing new code, can i use/make change in existing code?)
- Increase code reusability
- Make your code generic wherever possible
- Leave **no hardcoded data** in source code
- Keep your static data outside the source code
- Keep your dynamic data dynamic in testcode (fetch it from db queries or scripts)
- Test your code properly, use IDE options such as call heirarcy or show usage to test your changes end 2 end
- Use Extensive logging- everything which is part of source code should be analyzed from logs without looking at the source code
- Generate and save failure proofs outside the src code- videos/data/screenshots/logs

- Focus on making your code scalable and faster without compromising the code quality
- Your code should be platform and system independent
- Use as many assertions as possible focus on automated testing rather than automation
- Leave no hardcoded data in source code
- Always think for the future, separate out tech dependencies so that migration to new tech is easy in case it is needed
- Keep your tests independent for better results in multithreading unless they are related (example publisher subscriber related tests)
- Use Proper Documentation
- Create code which is can be easily read and modified by others