

WHICH SANDWICH?

PEANUT BUTTER JELLY!

A peanut butter and jelly sandwich is a small miracle. Simple, timeless, and universally recognizable, it somehow achieves harmony through contrast: salty and sweet, sticky and smooth, structured yet soft. Oddly enough, this balance is also what software engineers strive for when building clean codebases. The sandwich is more than food—it's a metaphor.

The bread is the foundation. In code, this means the architecture, directory structure, and naming conventions that support everything else. Good bread doesn't call attention to itself, just as a well-designed project layout doesn't confuse contributors. When the foundation is right, the rest of the experience becomes intuitive.

Peanut butter is the logic layer. It's the sticky part, the substance. This is where your domain knowledge lives—core functions, data transformations, decision-making rules. When applied poorly, peanut butter clumps up, hard to swallow, hard to digest. Likewise, unrefactored code becomes tangled, unreadable, and difficult to test. The solution isn't to remove the complexity but to spread it more evenly. Break things into smaller functions. Use meaningful names. Avoid nesting madness. Good peanut butter spreads thin and even.

Jelly is the interface—the output. It's what people see and taste first. Whether that's a UI, an API response, or logs, your jelly has to complement the logic without overpowering it. Too much jelly and the sandwich slips apart. Similarly, a beautiful front-end tied to messy back-end code is fragile. The key is containment: don't let jelly leak. Don't let interface concerns bleed into core logic.

Dependencies are your ingredients. Just as not all jelly is created equal, not all packages are worth importing. Choose libraries that are well-maintained and well-documented. Trust is earned, even in a sandwich. And just as allergies exist, you must be aware of edge cases and security vulnerabilities in your stack. Don't blindly spread whatever's in the fridge.

Testing is the bite test. You pick up the sandwich and see if it falls apart. A good codebase survives a grip, a chew, a bit of shaking. It doesn't break under pressure. Write tests, not for 100% coverage, but for confidence. Test that the core works—that the peanut butter sticks and the jelly stays where it should.

Finally, documentation is the label on the jar. When someone new opens the fridge (or your repo), they should know what they're reaching for. Clear READMEs, thoughtful comments, simple guides—they all ensure someone else can step in and build their own sandwich without confusion.

Building clean code isn't about elegance for its own sake. It's about making something that works, is pleasurable to use, and that others can pick up and extend. Like a peanut butter and jelly sandwich, good code is timeless, satisfying, and best when shared. The goal isn't perfection—it's nourishment. Code like you sandwich: with care, balance, and a touch of joy.

Bodega Test Page 1

Instructions on Preparation of Sandwich

this is for B.L.T. (Bacon Lettuce Tomato)

(Oz)	B	L	T
Small	0.5	1	1
Medium	1	2	2
Large	2.5	3	2

NOTE: Mayo Optional



Bodega Test Page 2

Instructions on Preparation of Sandwich (P2)

this is for Sandwich Topping Compatibility

	BLT	Club	Ruben
Mayo	✓	✓ -	✓
Mustard	✓		
Salt & Pepper	✓	✓	
[PLACEHOLDER]	✓ +	✓	✓ +

NOTE: NA

Bodega Test Page 3

Instructions on Preparation of Chips (P3)

this is for Sandwich Topping Compatibility

Brand	Sandwich	Coffee	Other
Frito	+\$2.00	+\$3.00	+\$2.00*
Lays	+\$1.50	+\$3.00	+\$2.09
Juantonios	+\$1.00	+\$2.75*	+\$2.50
[OTHER]	+\$0.00	+\$2.50	NA

NOTE: *Free Drink

Bodega Test Page 4

Instructions on Preparation of Chips (P3)

this is for Sandwich Topping Compatibility

[SMALL DRINKS] Drink Name	ID
Coke	04182003
Pepsi	10252005
Dr. Pepper	11221969

NOTE: *Spite not available for Sm.

LEFT TABLE FOR SMALL DRINKS
RIGHT TABLE FOR MED+



[MD-LG+ DRINKS] Drink Name	ID
Coke	04182004
Pepsi	10252006
Dr. Pepper	11221970
Spite	04071963