**✅ Ako testovať e-mailovú šablónu s dynamickými údajmi z databázy**

**🧾 Príklad šablóny:**

**Predmet:** Objednávka č. {{order\_id}} bola úspešne prijatá

Dobrý deň, {{customer\_name}},  
ďakujeme za Vašu objednávku. Tu sú jej podrobnosti:

• Produkt: {{product\_name}}  
• Množstvo: {{quantity}}  
• Cena spolu: {{total\_price}} €

Vaša objednávka bude odoslaná na adresu:  
{{delivery\_address}}

S pozdravom,  
Tím {{company\_name}}

**🧪 Čo všetko otestovať:**

**🔹 1. Nahradenie placeholderov**

* Správne načítanie všetkých premenných (napr. {{order\_id}}, {{customer\_name}})
* Ošetrenie prázdnych hodnôt (napr. fallback ako "Zákazník" pri chýbajúcom mene)

**🔹 2. Lokalizácia a jazyková verzia**

* Správny jazyk podľa preferencie používateľa  
  • Over, či systém používa nastavenie jazyka zo správneho zdroja (napr. jazyk v profile používateľa, cookies, nastavenie účtu)  
  • Porovnaj, či jazyk e-mailu zodpovedá tomuto nastaveniu (napr. používateľ má zvolenú nemčinu → predmet aj obsah e-mailu sú v nemčine)
* Preklady statických aj dynamických textov

**🔹 3. Štruktúra a formátovanie e-mailu**

* Korektné HTML zobrazenie v rôznych klientoch (Gmail, Outlook, mobil)

prípadne prehliadačoch (Firefox, Safari, Edge, Chrome)

* Odstupňovanie, odrážky, zarovnanie
* Korektné zobrazenie aj v plain-text verzii (čisto textová verzia bez formátovania, ktorú niektoré e-mailové klienty zobrazujú namiesto HTML)

**🔹 4. Obsahová kontrola dát**

* Zhoduje sa objednávka v e-maile s údajmi v databáze?
* Cena je správne vypočítaná a zaokrúhlená?
* Adresa nie je useknutá?

**🔹 5. Bezpečnosť**

* Žiadne HTML injekcie cez vstupy. HTML injection (HTML injekcia) je situácia, keď útočník alebo chyba v systéme umožní vložiť nebezpečný HTML kód do obsahu e-mailu. Napríklad ak používateľ zadá do mena reťazec ako <script>alert('hack');</script>, a tento sa bez úpravy dostane do e-mailu, môže spôsobiť problémy v prehliadači príjemcu. Takéto vstupy treba filtrovať alebo escapovať, aby sa zobrazovali len ako text.
* E-mail neobsahuje citlivé dáta (napr. číslo karty)

**🔹 6. Spúšťanie a logika odosielania**

* Kedy sa e-mail spustí (po objednávke, po úhrade...)?
* Neodosiela sa viackrát?  
  • Skontroluj logy systému alebo e-mailovej služby (napr. Sendgrid, Mailgun) – koľkokrát bola odoslaná notifikácia pre tú istú objednávku  
  • Pozri záznamy v databáze (napr. tabuľka email\_logs, notifications) – je tam len 1 záznam?  
  • Otestuj v QA prostredí: vytvor objednávku a sleduj, či sa e-mail neodošle opakovane, ak klikneš znova  
  • Over, či systém používa príznak ako email\_sent = true, a či sa tento stav správne aktualizuje
* Čo sa stane pri chybe v šablóne (log, fallback, chyba v systéme)?

**🔹 7. Testy okrajových prípadov**

* Chýbajúci produkt (napr. vymazaný z databázy)  
  Over, čo sa stane, keď sa po vytvorení objednávky produkt z databázy odstráni. E-mailová šablóna by mala buď:  
  • čerpať dáta z momentu objednania (napr. snapshot v inej tabuľke),  
  • alebo bezpečne ošetriť chýbajúce hodnoty (napr. „Produkt bol medzičasom odstránený“).

🔸 Test: vytvor testovaciu objednávku → následne produkt z databázy zmaž → spusti generovanie e-mailu a sleduj, čo sa zobrazí v šablóne.

* Príliš dlhé meno alebo adresa
* Zahraničné znaky a diakritika

**🔹 8. Odhlásenie / footer**

* Obsahuje e-mail možnosť odhlásiť sa z notifikácií (ak má)?
* Správne odkazy, GDPR texty

**🧠 Vysvetlenie pojmu: Fallback**

Záložná hodnota alebo správanie, ktoré systém použije, ak niečo chýba. Napr. ak nie je dostupný preklad, zobrazí sa text v angličtine ako fallback. Pri e-mailových šablónach sa ako fallback často používa napr. "Zákazník" namiesto {{customer\_name}}, ak meno chýba.

**#tester** **#akotestovat** **#manualnytester** **#automatizovanytester** **#ITtester**

**#QA** **#ITkariera**