Got it ✅ You want precise **WSTG notes for “Test Ability to Forge Requests” (WSTG-BUSL-03)**.  
Here’s a clean breakdown 👇

**📝 Notes: Test Ability to Forge Requests**

**🎯 Purpose**

To check whether an attacker can **manipulate or craft their own HTTP requests** to bypass normal workflows, escalate privileges, or abuse business rules.

* Many apps assume requests come only from the legitimate UI → attackers bypass that with forged requests.

**⚡ Common Attack Scenarios**

1. **Hidden Parameters**
   * Modify price field: price=100 → price=1
2. **Unintended Actions**
   * Crafting a request to approve an order without admin role.
3. **CSRF/Session Abuse**
   * Forged requests leveraging victim’s session.
4. **Workflow Skipping**
   * Direct POST to /checkout/confirm without adding items.
5. **ID Manipulation**
   * Accessing /user?id=1234 instead of your own /user?id=1001.

**🔍 How to Test**

1. **Intercept Requests**
   * Use **Burp Suite, OWASP ZAP, Postman** to capture HTTP requests.
2. **Replay & Modify**
   * Change parameters (IDs, amounts, roles).
   * Example: Add unauthorized fields like isAdmin=true.
3. **Bypass Client Controls**
   * Disable JS validations, remove hidden input restrictions.
   * Submit directly via curl or intercepted POST.
4. curl -X POST https://target.com/transfer \
5. -d "amount=10000&to=attacker"
6. **Test Unlinked/Undocumented Endpoints**
   * Try calling endpoints not available in the UI (predictable paths).
7. **Check for Anti-CSRF Tokens**
   * If missing or predictable, forged requests become trivial.

**🛡️ Mitigation**

* Enforce **server-side authorization & validation** (never trust client).
* Use **CSRF tokens** bound to session + action.
* Validate **workflow sequence** (cannot skip required steps).
* Check **referer/origin headers** for sensitive actions.
* Apply **principle of least privilege** in endpoints.

✅ **Key Takeaway:**  
If an app does not validate requests properly, an attacker can **forge HTTP requests** outside the intended UI to bypass logic, escalate privileges, or steal data.

👉 Do you want me to also make a **step-by-step tester’s playbook** (like: capture → replay → modify → bypass) so you can apply it directly in practice?