**🧪 Testes no Postman – Validação de Requisições**

🔸 1. **Status code: Code is 200**  
**O que faz: Verifica se o resultado http é 200.**  
**Exemplo:**

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});  
**Print:**





🔸 2. **Response body: Contains string**  
**O que faz:** Verifica se há a String especifica na verificação  
**Exemplo:**

pm.test("Body matches string", function () {

pm.expect(pm.response.text()).to.include("available");

});  
**Print:**





🔸 3. **Response body: JSON value check**  
**O que faz:** Valida um valor específico no JSON da resposta.  
**Exemplo:**

pm.test("Your test name", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.name).to.eql("Firulais");

});  
**Print:**





🔸 4. **Response body: Is equal to a string**  
**O que faz:** Compara o corpo da resposta com uma string exata.  
**Exemplo:**

pm.test("Body is correct (string comparison)", function () {

pm.response.to.have.body('{"id":1,"category":{"id":1,"name":"cat"},"name":"dog","photoUrls":[],"tags":[],"status":"sold"}');

});  
**Print:**





🔸 5. **Response headers: Content-Type header check**  
**O que faz:** Verifica se o header Content-Type é application/json  
**Exemplo:**

//Response headers: Content-Type header check

pm.test("Content-Type is present", function () {

pm.response.to.have.header("Content-Type");

});

**Print:**





🔸 6. **Response time is less than 200ms**  
**O que faz:** Valida se o tempo de resposta foi menor que 200ms.  
**Exemplo:**

//Response time is less than 200ms

pm.test("Response time is less than 200ms", function () {

pm.expect(pm.response.responseTime).to.be.below(100);

});  
**Print:**





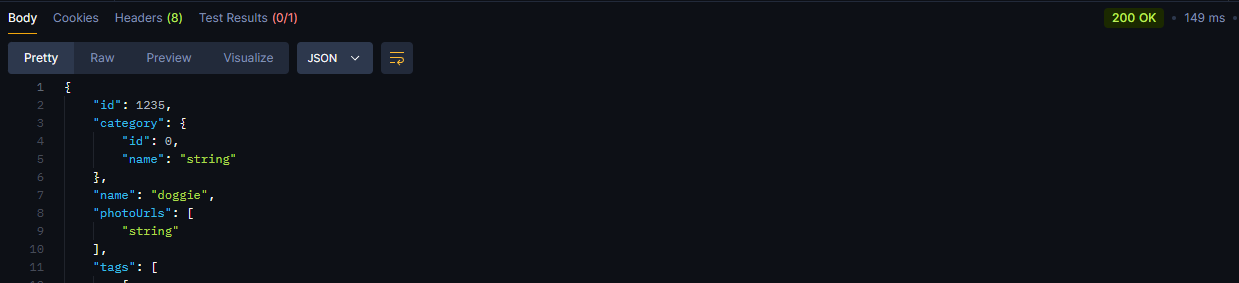
🔸 7. **Status code: Successful POST request**  
**O que faz:** Verifica se o status code é 201  
**Exemplo:**

pm.test("Successful POST request", function () {

pm.expect(pm.response.code).to.be.oneOf([201, 202]);

});  
**Print:**





🔸 8. **Status code: Code name has string**  
**O que faz:** Verifica se o nome do status code contém "OK".  
**Exemplo:**

pm.test("Status code name has string", function () {

pm.response.to.have.status("OK");

});  
**Print:**





🔸 9. **Response body: Convert XML body to a JSON Object**  
**O que faz:** Converte um corpo XML em JSON (não aplicável à Petstore, pois retorna JSON).  
**Exemplo:**

var jsonObject = xml2Json(responseBody);  
**Print:**



🔸 10. **Use Tiny Validator for JSON data**  
**O que faz:** Valida json contra um schema.  
**Exemplo:**

var schema = {

"type": "object",

"properties": {

"id": { "type": "integer" },

"name": { "type": "string" },

"status": { "type": "string" }

},

"required": ["id", "name", "status"]

};

var jsonData = pm.response.json();

pm.test("Validação do schema com tv4", function () {

pm.expect(tv4.validate(jsonData, schema)).to.be.true;

});  
**Print:**





**🔧 Variáveis no Postman**

🔸 11. **Get a global variable**  
**O que faz:**   
**Exemplo:**

const globalVar = pm.globals.get("var\_name");  
**Print:**

🔸 12. **Get a collection variable**  
**O que faz:**   
**Exemplo:**

const collectionVar = pm.collectionVariables.get("var\_name");  
**Print:**

🔸 13. **Get an environment variable**  
**O que faz:**  
**Exemplo:**

const collectionVar = pm.collectionVariables.get("var\_name");  
**Print:**

🔸 14. **Get a variable**  
**O que faz:**  
**Exemplo:**

const dynamicVar = pm.variables.get("var\_name");  
**Print:**

🔸 15. **Set a global variable**  
**O que faz:**  
**Exemplo:**

pm.globals.set("global\_var", "value");  
**Print:**

**🧼 Limpar Variáveis no Postman**

🔸 16. **Clear a collection variable**  
**O que faz:**  
**Exemplo:**

pm.collectionVariables.unset("var\_name");  
**Print:**

🔸 17. **Clear an environment variable**  
**O que faz:**  
**Exemplo:**

pm.environment.unset("var\_name");  
**Print:**

🔸 18. **Clear a local variable**  
**O que faz:**  
**Exemplo:**

pm.variables.unset("var\_name");  
**Print:**

**🔁 Fluxo de Requisição**

🔸 19. **Send an HTTP request**  
**O que faz:**  
**Exemplo:**

pm.sendRequest("https://petstore.swagger.io/v2/pet/1", function (err, res) {

console.log(res.json());

});  
**Print:**