# 🔁 Variables in JMeter

Variables are **key to creating flexible, reusable, and dynamic test plans** in JMeter. Instead of hardcoding values, you define variables and reference them throughout the test plan.

**📌 1. What are JMeter Variables?**

🧩 **Definition**:  
Variables in JMeter are **key-value pairs** that get resolved during test execution.

🛠️ Example:

Bash

BASE\_URL = http://example.com

USERNAME = testuser

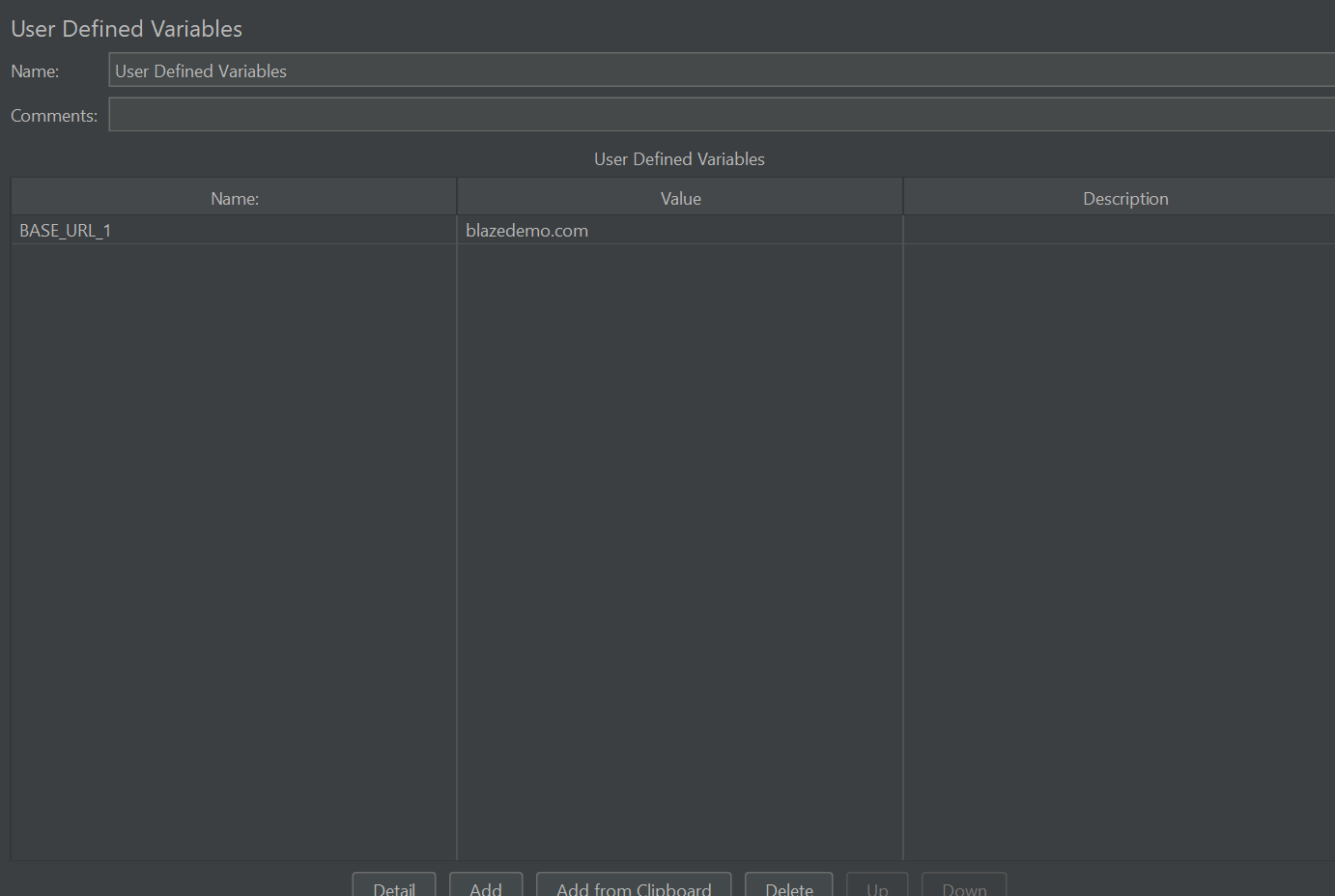
Used as ${BASE\_URL} in HTTP Sampler → JMeter will replace it with http://example.com.

**🎯 2. Why Use Variables?**

| **Purpose** | **Description** |
| --- | --- |
| 🧪 Parameterization | Replace hardcoded values like URL, username, etc. |
| 🔧 Maintainability | Change one value → reflects everywhere it’s used |
| 🌀 Dynamic Data | Generate random data using functions or scripts (like \_\_Random()) |
| 🔄 Correlation | Capture and reuse values from previous responses (like tokens/session IDs) |

**🧱 3. How to Define Variables in JMeter**

**🧮 A. User Defined Variables (UDV) – *Config Element***



📍 **Scope**: Static variables available throughout the test.

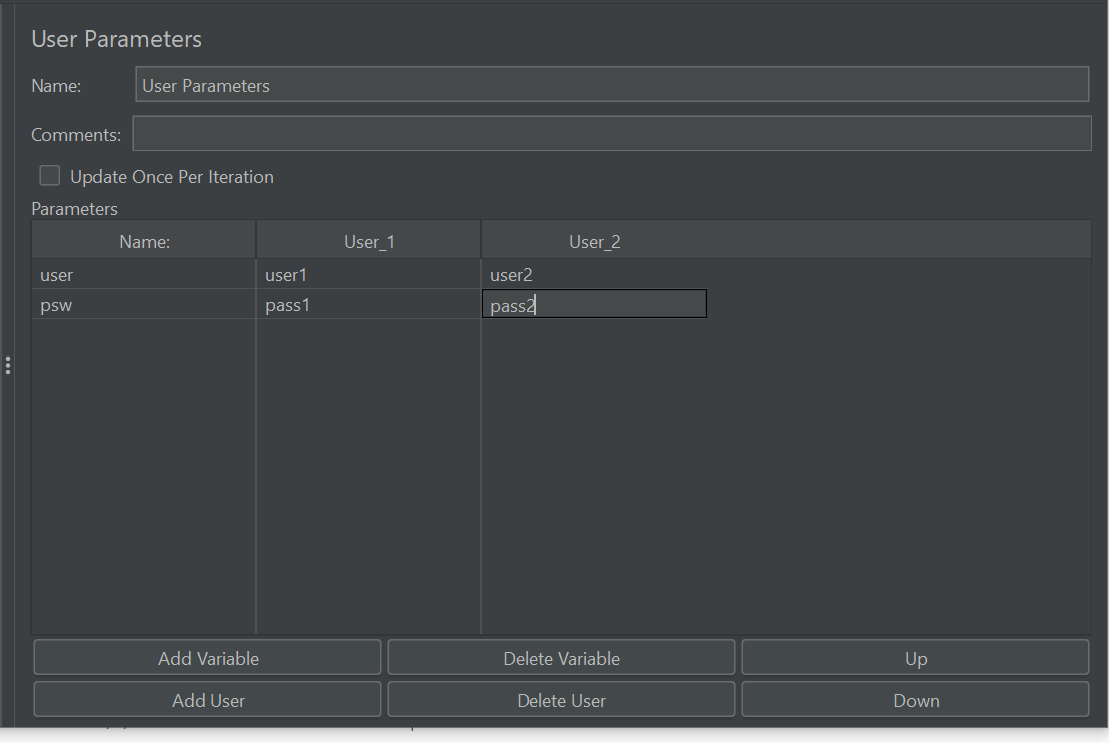
📌 **Add via**:  
Test Plan / Thread Group → Add → Config Element → User Defined Variables

📋 **Example**:

| **Name** | **Value** |
| --- | --- |
| BASE\_URL | <http://example.com> |
| USER\_PATH | /users |

📨 **Reference**: ${BASE\_URL} or ${USER\_PATH}  
💡 **Limitation**: If you use a function like \_\_Random(), it’s evaluated only once at the beginning.

**👥 B. User Parameters – *PreProcessor***



📍 **Scope**: Thread-specific, supports **multiple values** per thread (virtual user).

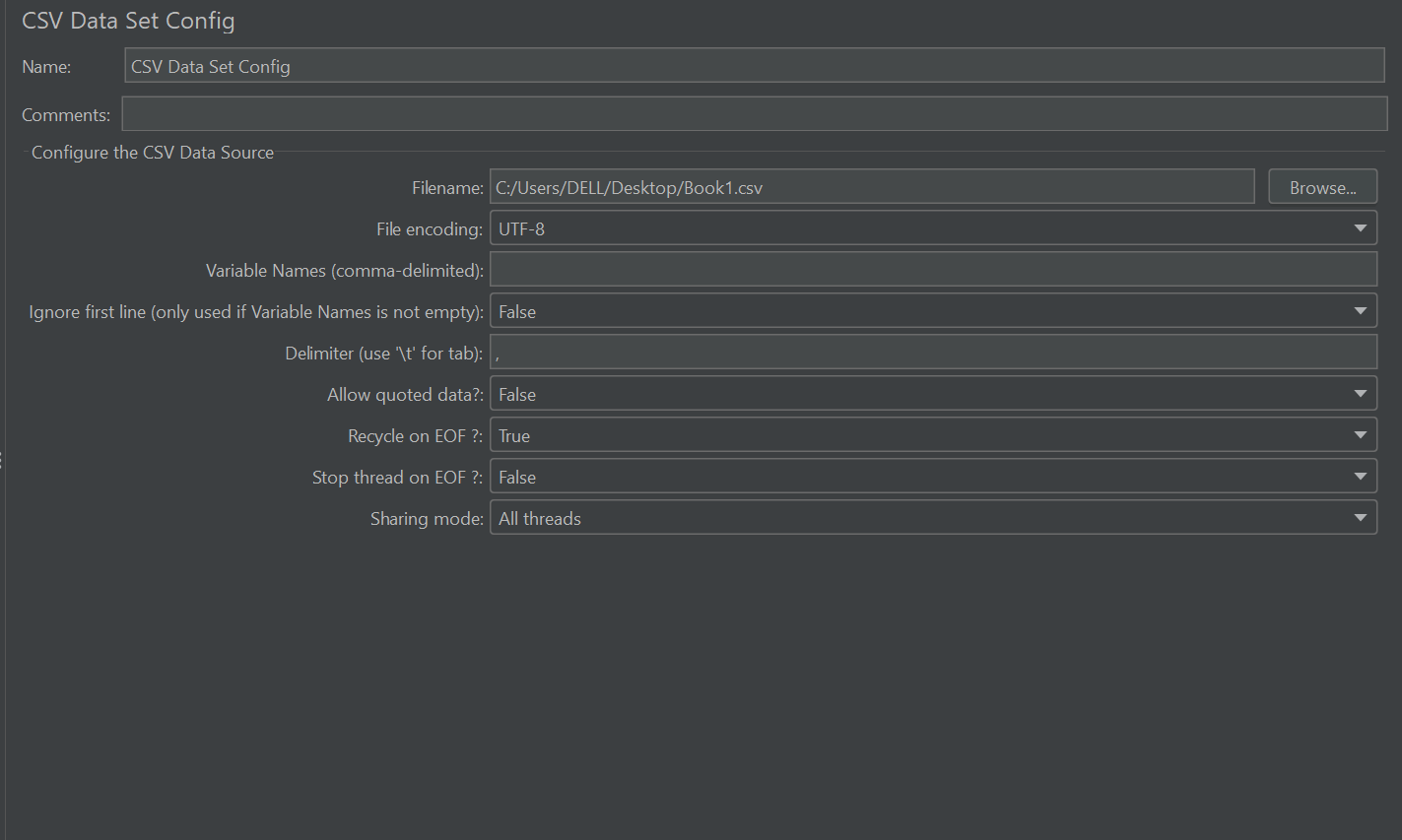
📌 **Add via**:  
Sampler/Controller → Add → PreProcessor → User Parameters

📋 **Example**:

| **Variable Name** | **User 1** | **User 2** |
| --- | --- | --- |
| username | user1 | user2 |
| password | pass1 | pass2 |

📨 **Reference**: ${username}, ${password}  
🔄 **Dynamic**: Values change per iteration if enabled.

**📂 C. CSV Data Set Config – *Config Element***



📍 **Scope**: Read data from external .csv file.

📌 **Add via**:  
Test Plan / Thread Group → Add → Config Element → CSV Data Set Config

📋 **CSV File Sample**:

pgsql

username,password

user1,pass1

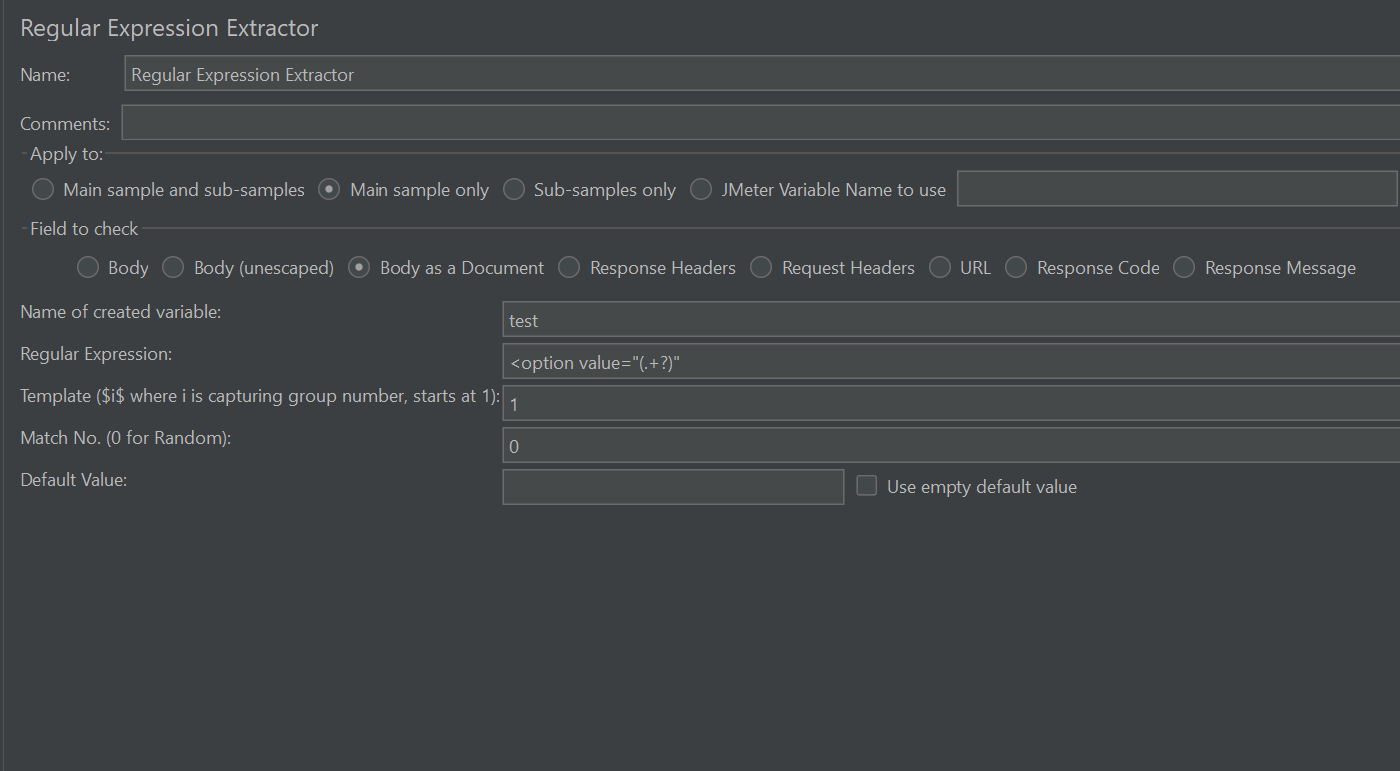
user2,pass2

🛠️ **Settings**:

* File path
* Variable names
* Delimiter (default ,)
* Recycle on EOF
* Sharing Mode

📨 **Reference**: ${username}, ${password}

**🧪 D. Regular Expression Extractor – *PostProcessor***



📍 **Scope**: Extracts value from **response** using regex.

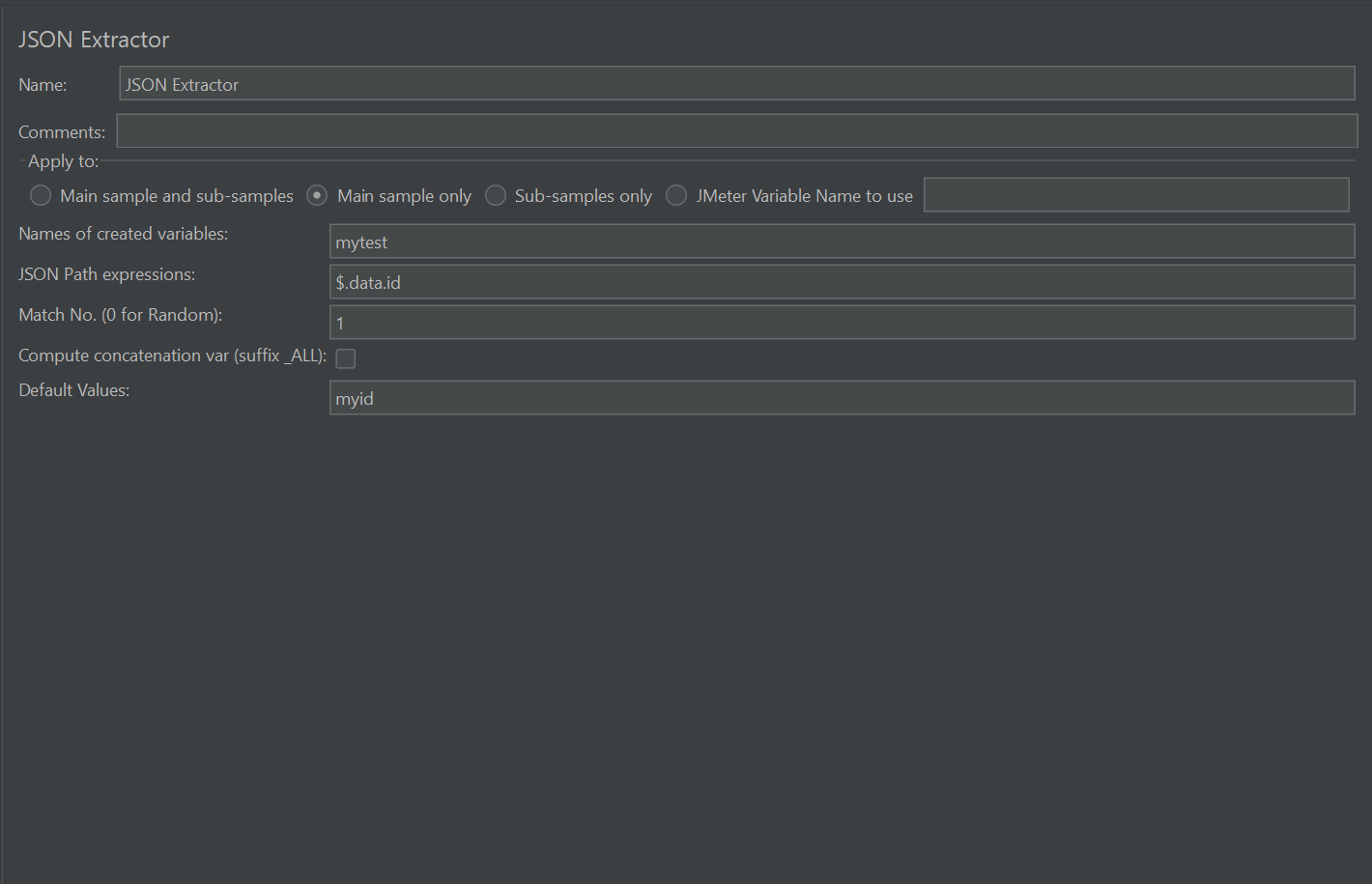
📌 **Add via**:  
Sampler → Add → PostProcessor → Regular Expression Extractor

📋 **Example**:

* Response: {"token":"abc123"}
* Regex: "token":"(.+?)"
* Reference Name: auth\_token

📨 **Reference**: ${auth\_token}

**📊 E. JSON Extractor – *PostProcessor***



📍 **Scope**: Extract values from JSON using **JSONPath**.

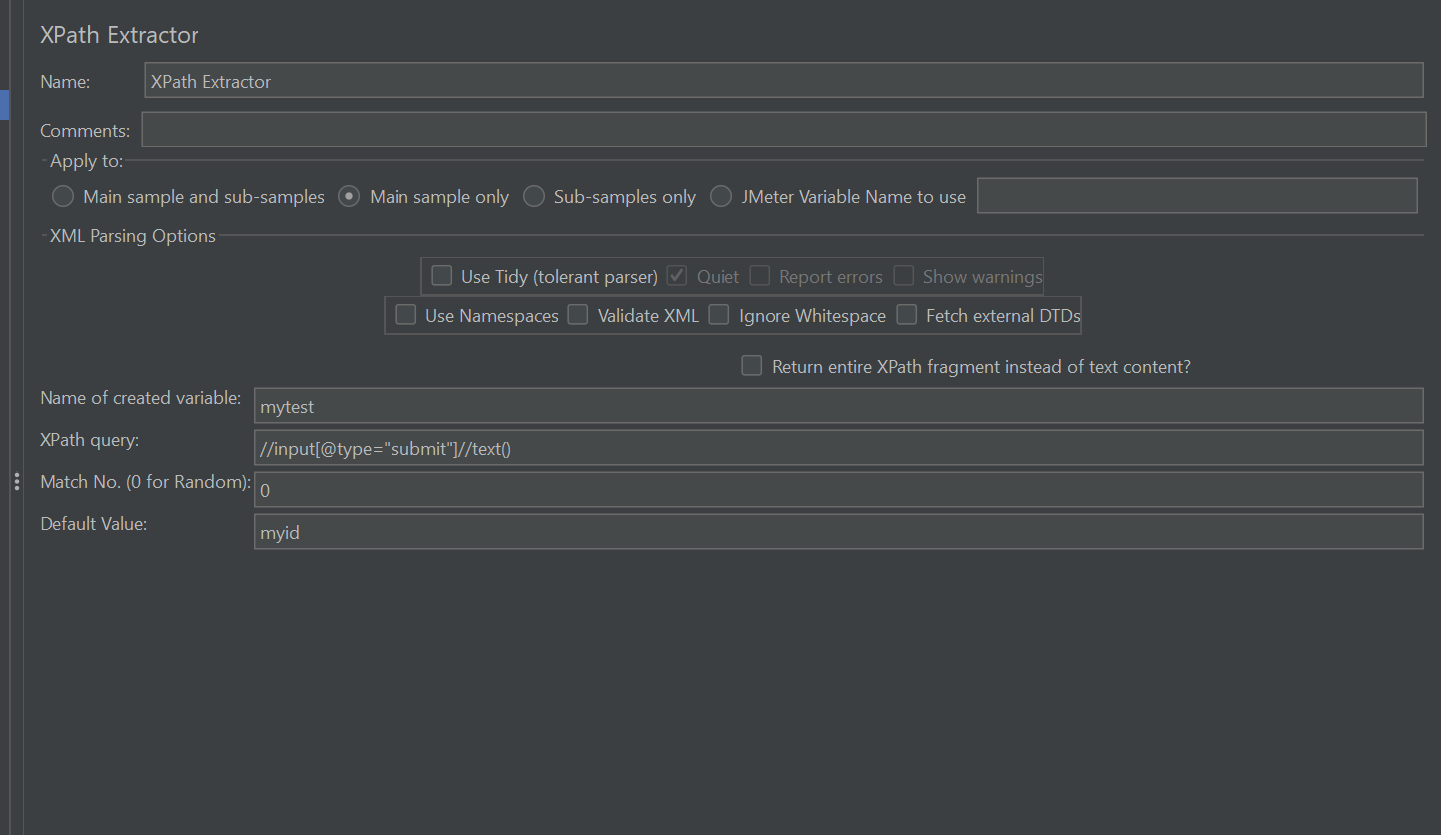
📌 **Add via**:  
Sampler → Add → PostProcessor → JSON Extractor

📋 **Example**:

* Response: {"user":{"id":"101"}}
* JSON Path: $.user.id
* Variable Name: userId

📨 **Reference**: ${userId}

**🗂️ F. XPath Extractor – *PostProcessor***



📍 **Scope**: Extracts value from **XML/HTML** using XPath.

📌 **Add via**:  
Sampler → Add → PostProcessor → XPath Extractor

📋 **Example**:

* XPath: //user/id/text()
* Reference Name: userId

📨 **Reference**: ${userId}

**🧬 G. JSR223 Elements – *Pre/Post Processor, Sampler, etc.***

📍 **Scope**: Dynamic scripting with **Groovy** (recommended), BeanShell, JavaScript.

📌 **Add via**:  
Add → PreProcessor / PostProcessor / Sampler → JSR223 (Groovy)

🛠️ **Set Variable**:

groovy

CopyEdit

vars.put("city", "Beijing")

🔍 **Get Variable**:

groovy

CopyEdit

def city = vars.get("city")

⚠️ Use vars for thread-local access, props for global properties.

**🧭 4. Variable Scope & Lifetime**

| **Variable Type** | **Scope** | **Access Syntax** | **Description** |
| --- | --- | --- | --- |
| vars | Thread-local | vars.get("key") | Each thread gets its own copy of the variable |
| props (Property) | Global across all threads | props.get("key") or ${\_\_P(key)} | Shared among all thread groups (use with caution) |

**📥 5. Referencing Variables**

🎯 **Syntax**: ${VariableName}

📋 **Example Use in HTTP Sampler**:

* **BASE\_URL**: http://mywebapp.com
* **USER\_PATH**: /users

📨 Resulting URL: ${BASE\_URL}${USER\_PATH}/login  
➡️ Becomes: http://mywebapp.com/users/login

**🧠 Final Tips**

✅ Use **User Defined Variables** for static data  
✅ Use **CSV Data Set Config** for test data  
✅ Use **Extractors** (Regex/JSON/XPath) for **correlation**  
✅ Use **JSR223** for full **dynamic control**  
✅ Always use ${varName} for referencing