# **📘 Parameterization in JMeter**

Creating Realistic, Scalable & Dynamic Test Plans

## **🎯 What Is Parameterization?**

Parameterization in JMeter means **replacing hardcoded values** with **dynamic or variable inputs** to:

* Simulate real-world user behavior
* Support data-driven testing
* Avoid caching effects
* Scale your load test across multiple virtual users (VUs)

## **🤔 Why Is Parameterization Essential?**

| **🔍 Purpose** | **✅ Benefit** |
| --- | --- |
| **Realism** | Simulates how real users enter varied data |
| **Prevent Caching** | Reduces repeated identical requests that can skew load |
| **Data-Driven Testing** | Tests app behavior across multiple input sets |
| **Reusability** | One test plan, many data sets or environments |
| **Scalability** | Each VU can use unique data to avoid collisions or redundancy |

## **📌 Common Use Cases**

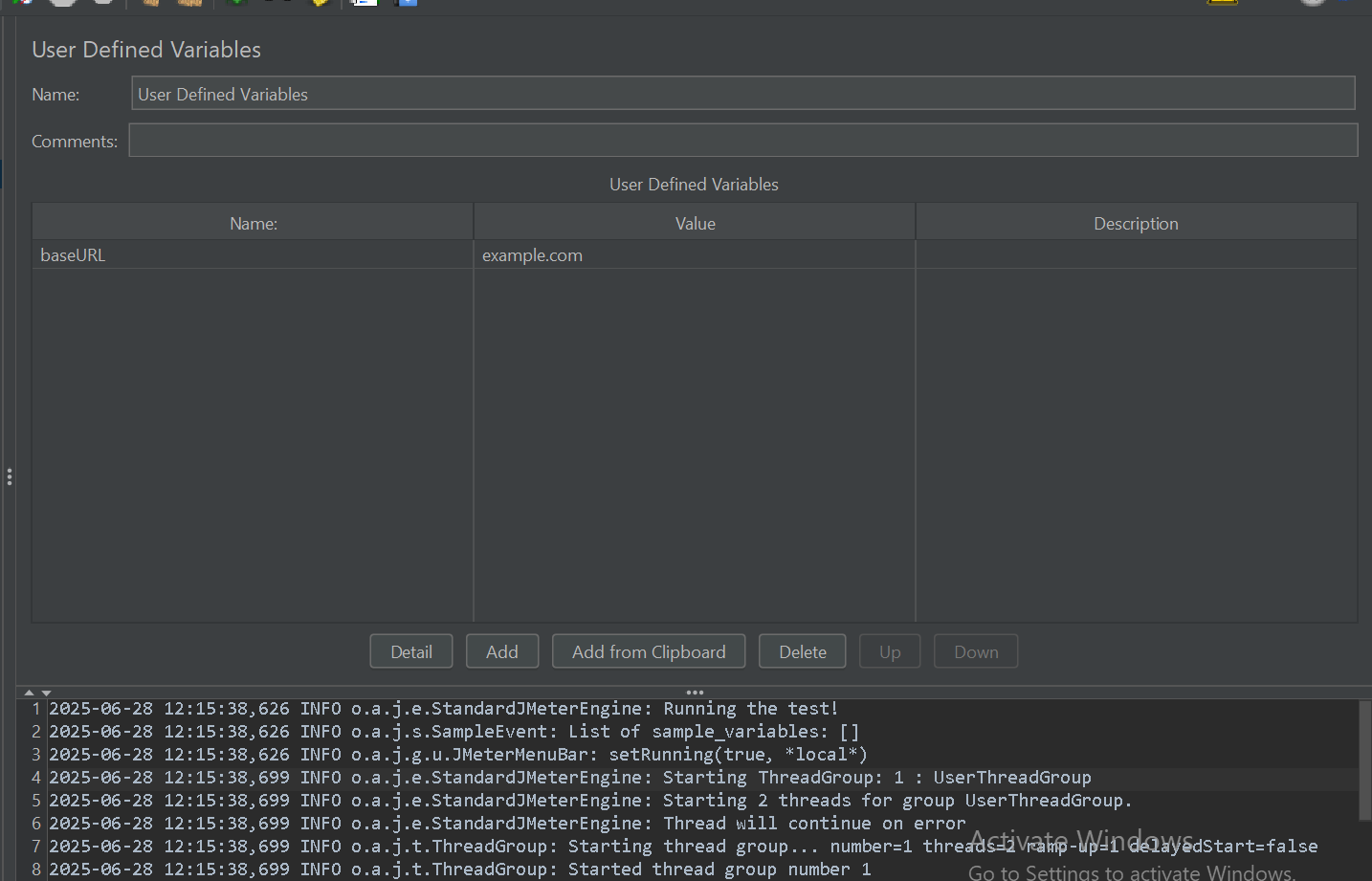
* 🔐 **Login Credentials**: Different user IDs and passwords
* 🔎 **Search Queries**: Varying keywords
* 🛒 **Product IDs**: Simulate browsing/purchasing different items
* 🧾 **Form Inputs**: Different names, emails, addresses
* 🌐 **URLs/Endpoints**: API parameterization
* 🕒 **Timestamps/IDs**: Dynamic or unique data values

## **🛠️ Ways to Achieve Parameterization in JMeter**

### **1️⃣ User Defined Variables (UDVs)**

📌 *Static configuration values shared globally*

* **Location**:  
   Test Plan > Add > Config Element > User Defined Variables
* **Syntax**:  
   Define baseURL = http://example.com → use ${baseURL}
* **Best For**:  
   Hostnames, ports, tokens, global configs



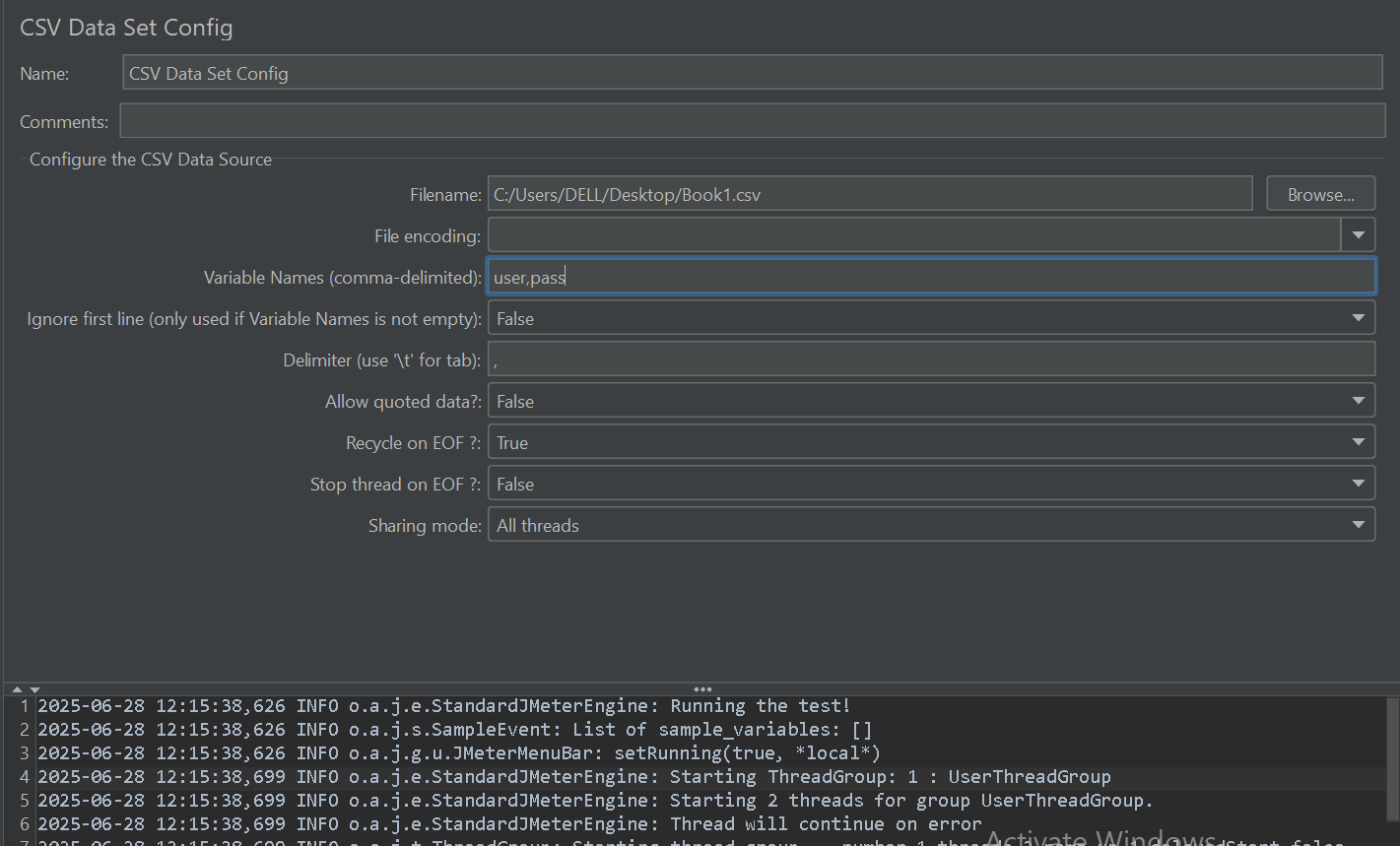
### **2️⃣ CSV Data Set Config (Most Common for Data-Driven Testing)**

📁 *Reads data line-by-line from a CSV file*

* **Location**:  
   Thread Group > Add > Config Element > CSV Data Set Config
* **Key Settings**:  
  + Filename: users.csv
  + Variable Names: username,password,productID
  + Recycle on EOF: True/False
  + Sharing Mode: All threads or Current thread

**Example CSV**:  
  
 csv  
CopyEdit  
user1,pass1,P101

user2,pass2,P102

* **Use in HTTP Request**:  
   ${username}, ${password}, ${productID}
* **Best For**:  
   Login sets, product catalog, forms, etc.
* 
* 0

### **3️⃣ JMeter Functions**

⚙️ *Generate dynamic data during runtime*

| **Function** | **Description** |
| --- | --- |
| ${\_\_Random(1,100)} | Random integer |
| ${\_\_RandomString(8,abc123)} | Random 8-char string |
| ${\_\_UUID()} | Universally unique ID |
| ${\_\_counter(TRUE, myCount)} | Global increment counter |
| ${\_\_time(yyyyMMdd\_HHmmss)} | Current timestamp |
| ${\_\_CSVRead(file.csv, 0)} | Read value from CSV |
| ${\_\_P(prop,default)} | Load property from CLI or file |

* **Best For**:  
   Quick test data like OTPs, emails, user IDs, etc.

### **4️⃣ Extractors (For Dynamic Server Data)**

🔄 *Correlation: Capture dynamic values from responses*

**Regular Expression Extractor** html  
CopyEdit  
Response: <input name="csrf\_token" value="abc123"/>

Regex: name="csrf\_token" value="(.+?)"

Variable: csrfToken

* Use as: ${csrfToken}
* **JSON Extractor** Parse tokens or IDs from JSON
* **CSS/JQuery Extractor** Use HTML selectors to extract form fields or hidden inputs
* **Best For**:  
   Session tokens, CSRFs, order IDs, etc.

### **5️⃣ User Parameters**

🎚️ *Set different values per thread or iteration manually*

* **Location**:  
   Sampler > Add > Pre-Processor > User Parameters

Define sets like:  
  
 text  
CopyEdit  
username: user1, user2

password: pass1, pass2

* **Best For**:  
   Small-scale iteration-specific data

### **6️⃣ JSR223 with Groovy (Advanced Logic)**

💻 *Write custom scripts for dynamic or complex values*

* **Location**:  
   Add JSR223 Sampler, PreProcessor, or PostProcessor

**Example**: Generate unique email  
  
 groovy  
CopyEdit  
import java.text.SimpleDateFormat

def sdf = new SimpleDateFormat("yyyyMMddHHmmss")

def timestamp = sdf.format(new Date())

def email = "user\_" + timestamp + "@test.com"

vars.put("uniqueEmail", email)

* Use in request: ${uniqueEmail}
* **Best For**:  
   Complex logic, external data sources, calculated fields

## **🧭 Choosing the Right Parameterization Strategy**

| **Use Case** | **Recommended Technique** |
| --- | --- |
| Static config (base URL, tokens) | **User Defined Variables** |
| Login, search, form data | **CSV Data Set Config** |
| Random OTP, email, counter | **JMeter Functions** |
| Dynamic session/token from response | **Extractors (Regex/JSON/CSS)** |
| Limited fixed variation | **User Parameters** |
| Scripting needs or external APIs | **JSR223 with Groovy** |

## **✅ Example Test Case Flow**

1. 🧾 **CSV Data Set Config** — Load usernames and passwords
2. 🔐 **Login Request** — Uses ${username} & ${password}
3. 🔄 **Regex Extractor** — Extracts dynamic token
4. 🧮 **JSR223 PreProcessor** — Builds custom order ID
5. 🛍️ **Submit Order Request** — Sends all combined data