## **🔄 Pre-Processors in JMeter**

**📌 What are they?** Pre-Processors execute **before** the sampler they're associated with. They prepare or modify data/settings for the upcoming request.

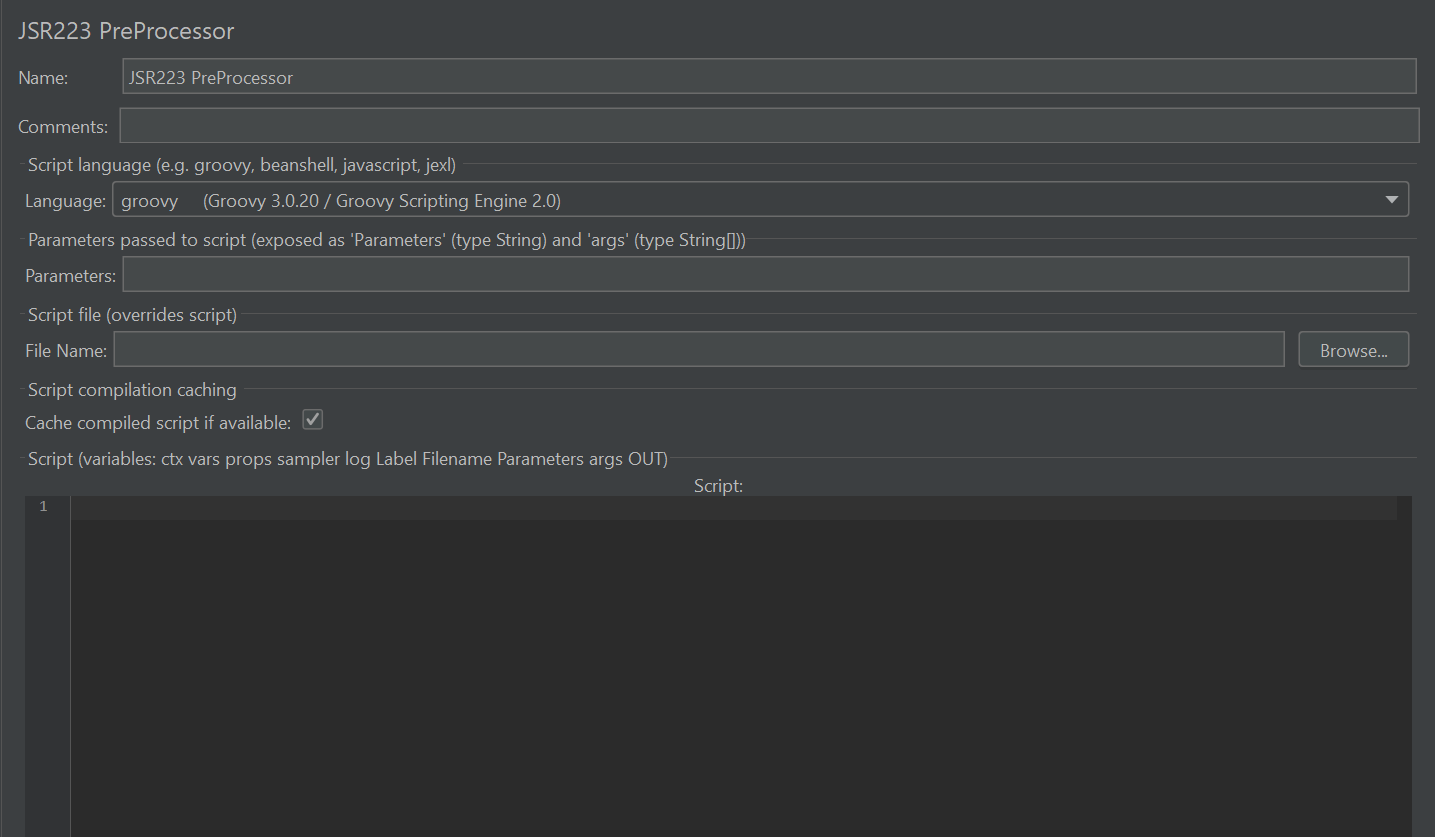
### **✅ When to Use:**

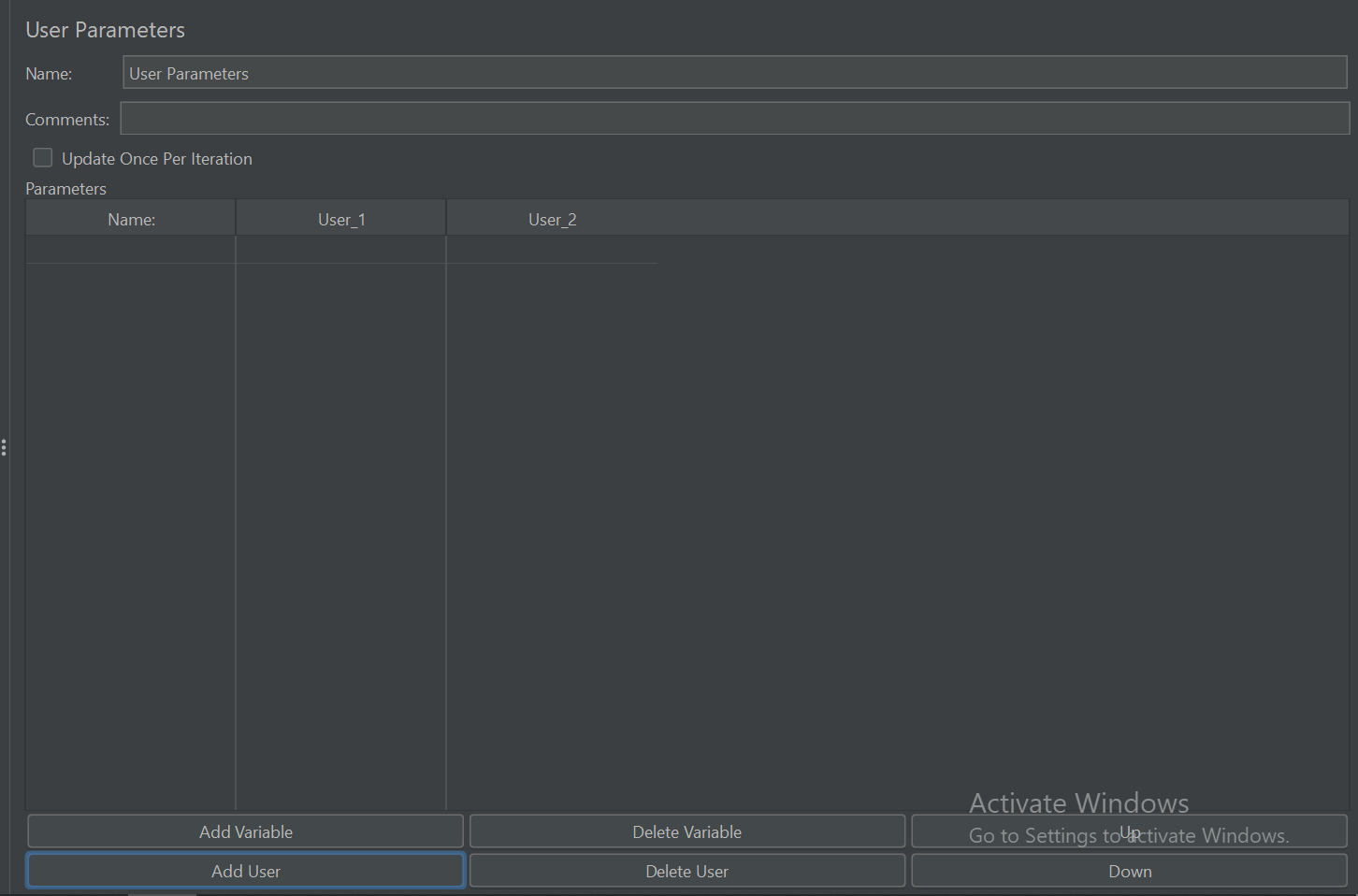
* 🔧 Data preparation
* ⚙️ Modify sampler inputs
* 📋 Pre-conditions or validations
* 🧪 Generate dynamic data

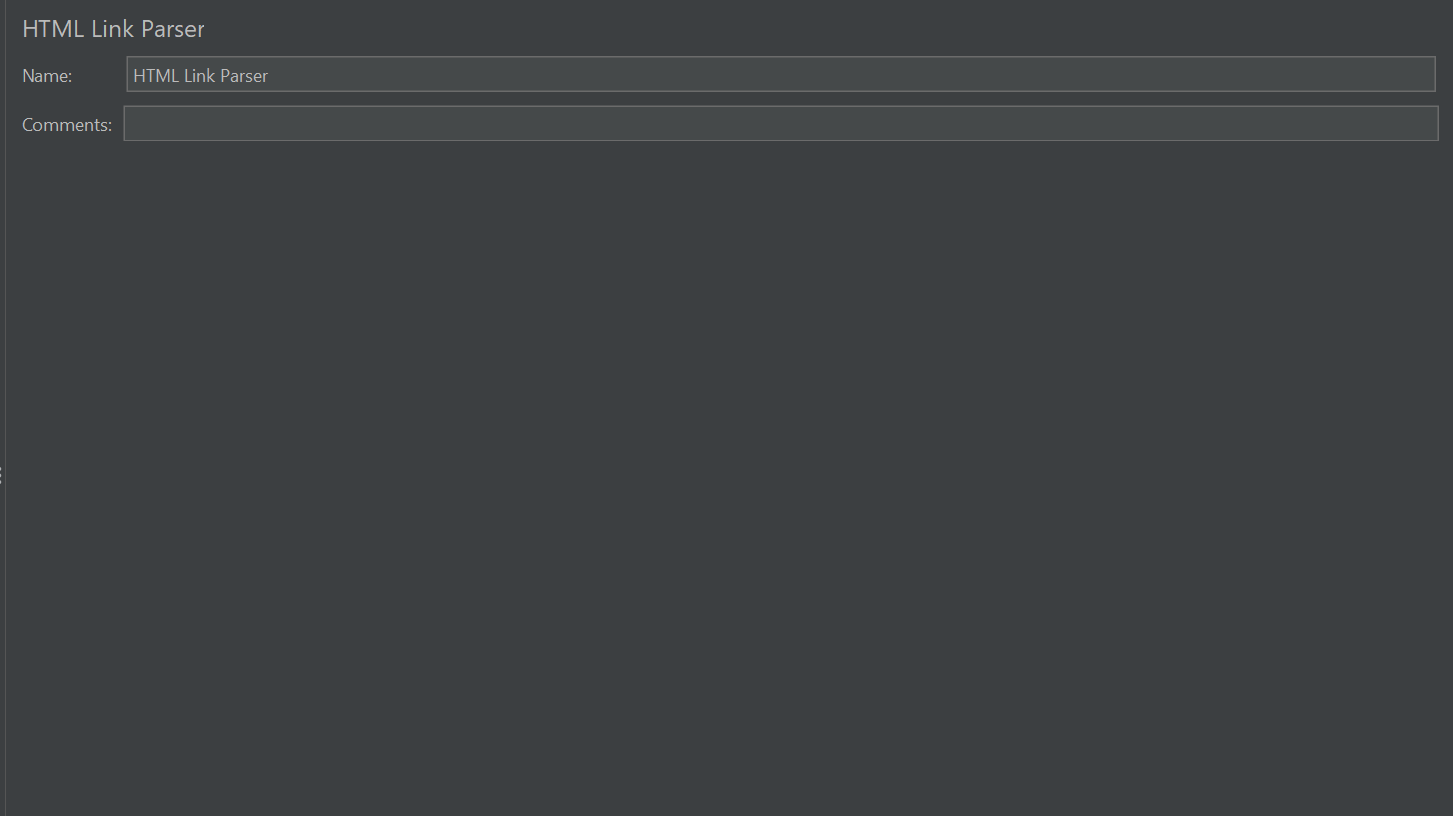
### **🛠️ Common Pre-Processor Types**

| **🧩 Pre-Processor** | **💡 Purpose** | **🧪 Example** |
| --- | --- | --- |
| **User Parameters** | Define user-specific variables | Set different username/password per virtual user |
| **JDBC PreProcessor** | Run SQL to fetch data | Get user\_id from DB for next request |
| **JSR223 PreProcessor** | Execute custom logic (Groovy preferred) | Encrypt password or generate session ID |
| **Sample Timeout** | Enforce execution time cap | Timeout API after 5 seconds |
| **HTML Link Parser** | Extract links from HTML | Click random product link after homepage load |
| **HTTP URL Rewriting Modifier** | Handle session IDs in URL | Append session ID in all next URLs |

## 







## **🔁 Post-Processors in JMeter**

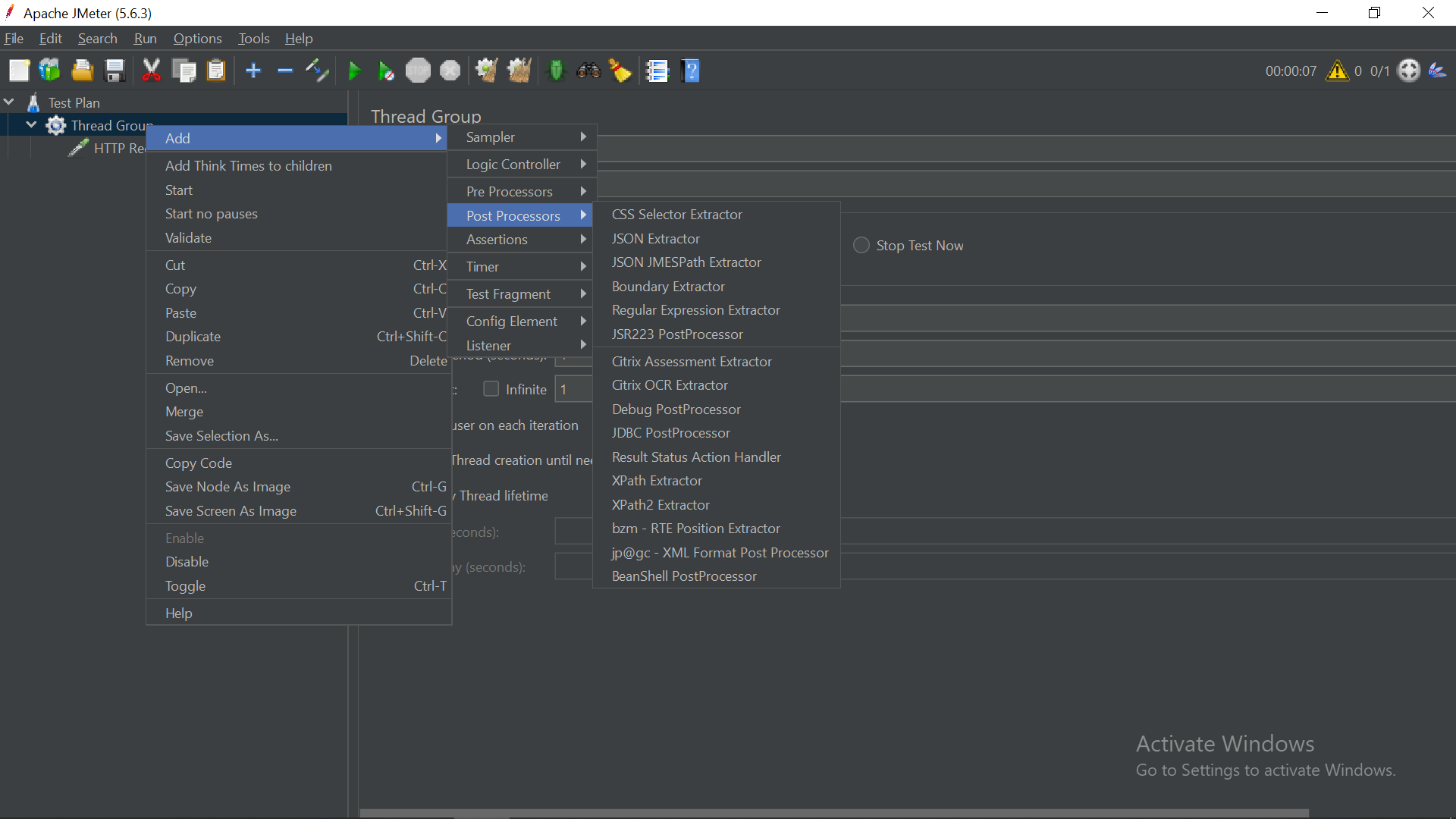
**📌 What are they?** Post-Processors execute **after** the sampler and are primarily used for extracting or validating response data.

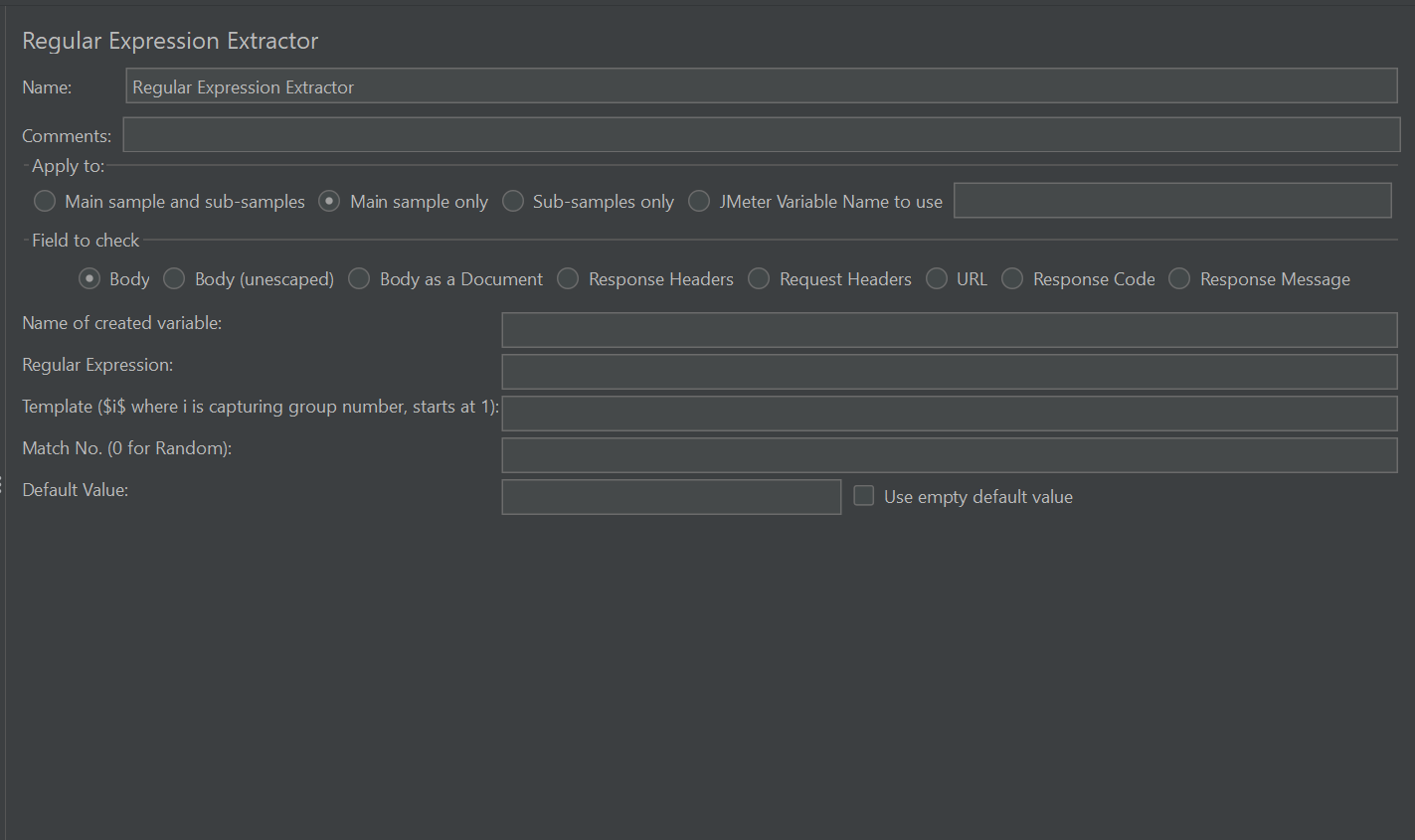
### **✅ When to Use:**

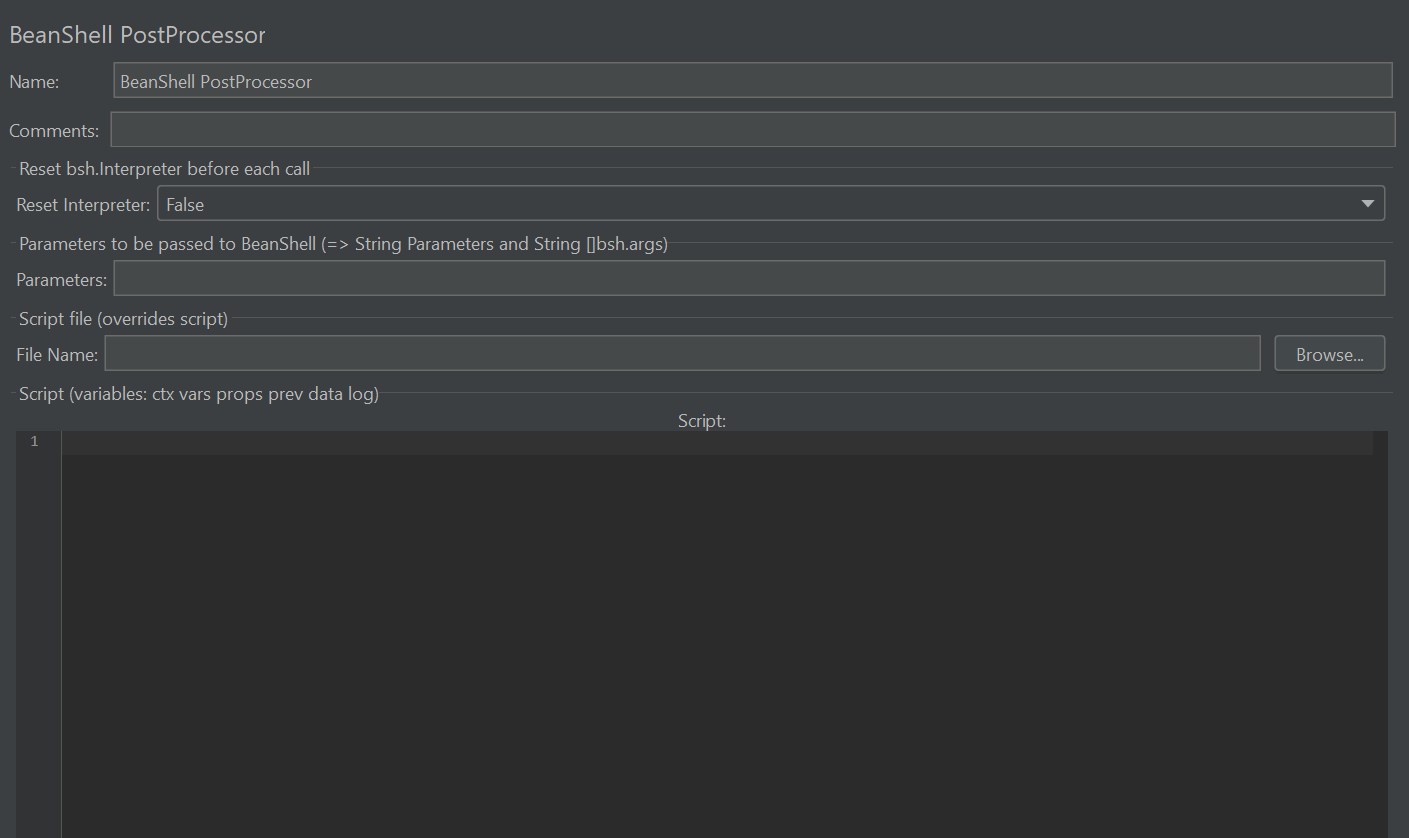
* 🧬 Extract values (for correlation)
* 📊 Validate or log response data
* 🔄 Modify variables dynamically
* ❌ Handle errors

### **🛠️ Common Post-Processor Types**

| **🧩 Post-Processor** | **💡 Purpose** | **🧪 Example** |
| --- | --- | --- |
| **Regular Expression Extractor** | Extract with regex | Capture session token from login response |
| **JSON Extractor** | Use JSONPath for parsing JSON | Get product\_id from API JSON response |
| **CSS/jQuery Extractor** | Extract from HTML via selectors | Extract price from a <span class='price'> |
| **XPath Extractor** | Parse XML/HTML with XPath | Get value of XML node <user\_id> |
| **JSR223 PostProcessor** | Custom response logic | Combine multiple IDs or parse JSON |
| **Boundary Extractor** | Text between markers | Get value between START\_ID= and &END\_ID= |
| **Result Status Action Handler** | Define actions based on success/failure | Stop thread on failed login |







## **🧭 Key Differences and Execution Flow**

| **🔍 Criteria** | **🔄 Pre-Processor** | **🔁 Post-Processor** |
| --- | --- | --- |
| **Execution Timing** | Before sampler | After sampler |
| **Purpose** | Prepare/modify request | Handle/parse response |
| **Scope** | Sampler/Controller/Thread Group | Same |
| **Order of Execution** | Top → bottom in tree | Top → bottom in tree |

## **📘 Tip: How to Use Pre/Post-Processors Effectively**

* 🎯 Always scope them carefully — placing under a **Sampler** ensures they only apply where needed.
* 🧩 Use **JSR223** with Groovy when performance and flexibility matter.
* 🧪 For dynamic websites or APIs, combine **Pre-processors** (for token generation) with **Post-processors** (for token extraction).