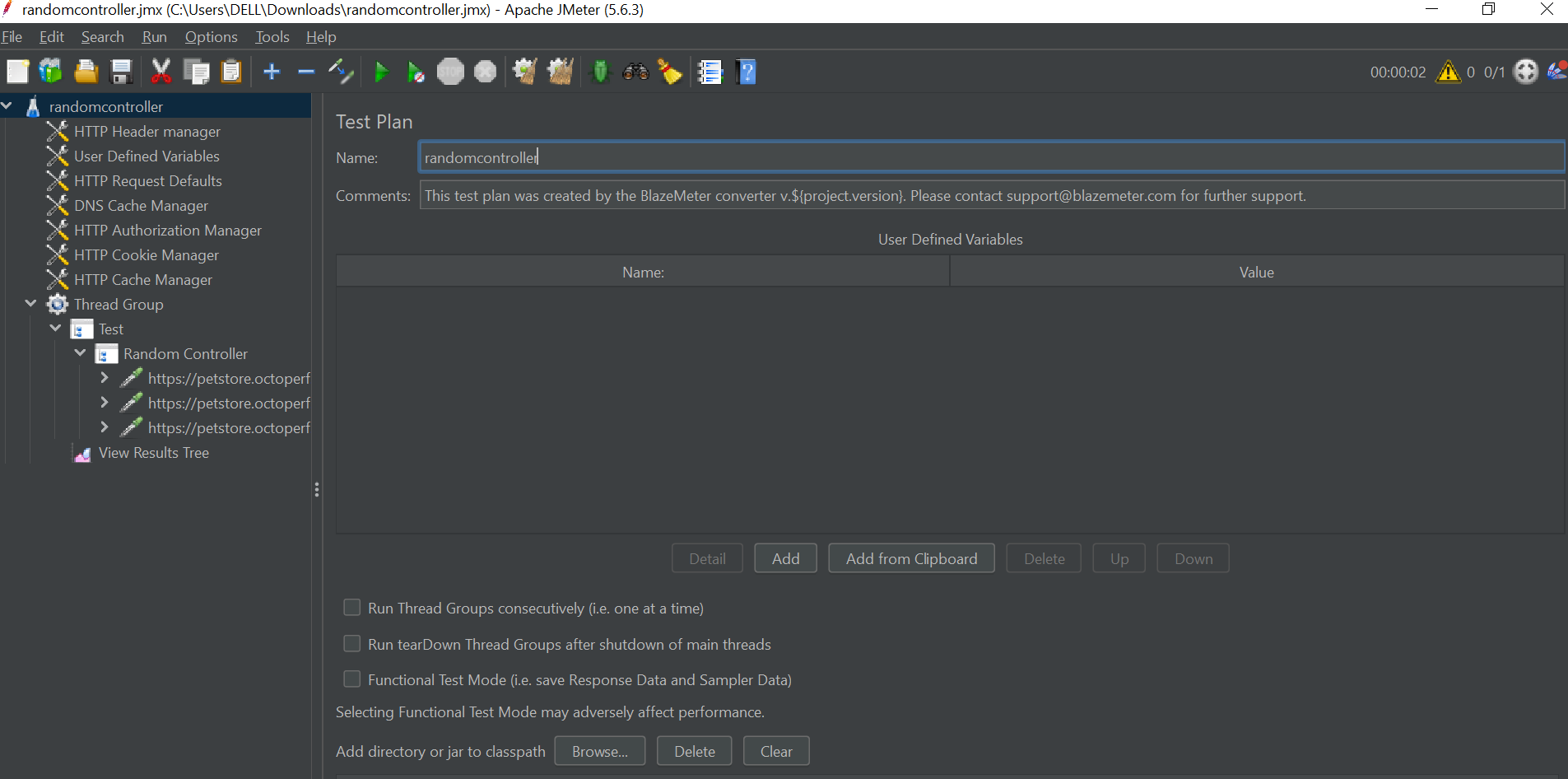
🎲 Random Controller





The **Random Controller** is a **Logic Controller** that **executes only one** of its direct children **randomly per iteration** of its parent (e.g., Thread Group or Loop Controller). It is designed to simulate **unpredictable, real-world user behavior**.

**🔧 How It Works**

| **Feature** | **Behavior** |
| --- | --- |
| Executes Per Iteration | On each parent loop, picks and runs **one** child element randomly |
| Selection Mechanism | Random — **equal chance** for each child |
| Remaining Children | All others are skipped in that iteration |
| Reset Behavior | On next loop, the process repeats, independent of previous picks |

🔄 It behaves like a digital dice roll—each iteration is independent of the previous one.

**📌 Common Use Cases**

| **Scenario** | **Why Random Controller Works Well** |
| --- | --- |
| 🔍 **Product Browsing** | Randomly simulate visits to /laptop, /tablet, /smartphone |
| 🧭 **Non-linear Navigation** | Users don’t follow fixed paths—simulate erratic user journeys |
| 🧪 **Testing Search Variability** | Rotate between different search terms randomly |
| 🕳 **Dead End Simulation** | Include an empty or dummy sampler to simulate users abandoning a page |
| 🧩 **Multiple UI Options** | Test alternate UI paths, e.g., click "Add to Cart", "Wishlist", or "Share" |

**🏗️ Example Test Plan Structure**

text

CopyEdit

Test Plan

└─ Thread Group (Threads: 1, Loop Count: 5)

├─ Once Only Controller (Login Flow)

│ ├─ HTTP Request - GET /login

│ └─ HTTP Request - POST /login

└─ Random Controller (Browse Random Products)

├─ HTTP Request - GET /products/laptop

├─ HTTP Request - GET /products/tablet

└─ HTTP Request - GET /products/smartphone

**🧾 Execution Flow (Sample):**

| **Iteration** | **Request Executed** |
| --- | --- |
| 1 | /products/tablet |
| 2 | /products/smartphone |
| 3 | /products/laptop |
| 4 | /products/laptop |
| 5 | /products/tablet |

🔁 You never know which request will be picked. Each is selected with **equal probability**, per iteration.

**⚙️ How to Add It**

1. **Right-click** on your **Thread Group** (or any Logic Controller).
2. Navigate to: Add > Logic Controller > Random Controller.
3. **Add Samplers or Controllers** beneath it (order does *not* matter).

**⚠️ No Configuration Needed**

The Random Controller has no configurable properties (aside from its name). Its randomness is built-in.

**🧠 Important Considerations**

| **Tip** | **Why It Matters** |
| --- | --- |
| ❌ Do Not Expect Round-Robin | Use **Interleave Controller** if you want ordered cycling |
| ✅ Can Include Other Controllers | Randomly trigger **Simple Controllers**, even with multiple steps |
| ⚖️ Uniform Distribution | Each child has an **equal chance** of selection |
| ✅ Works with CSV Data | Each child can use different CSV values if data sharing is managed well |

**🔁 Comparison Table: Random vs. Interleave vs. Random Order**

| **Controller** | **Children Executed per Iteration** | **Selection Type** | **Execution Pattern** |
| --- | --- | --- | --- |
| **Random Controller** | 1 | Random | Varies every iteration |
| **Interleave Controller** | 1 | Sequential | Round-robin (e.g., A → B → C) |
| **Random Order Controller** | All | Random Order | All children, shuffled each time |

**✅ Summary**

Use the **Random Controller** to:

* Simulate users choosing one out of several options.
* Add realism to your test plans by **randomizing user flows**.
* **Avoid repetition** and **predictability** in load tests.

It’s a lightweight and powerful tool for building dynamic user behavior patterns in your JMeter test scenarios.