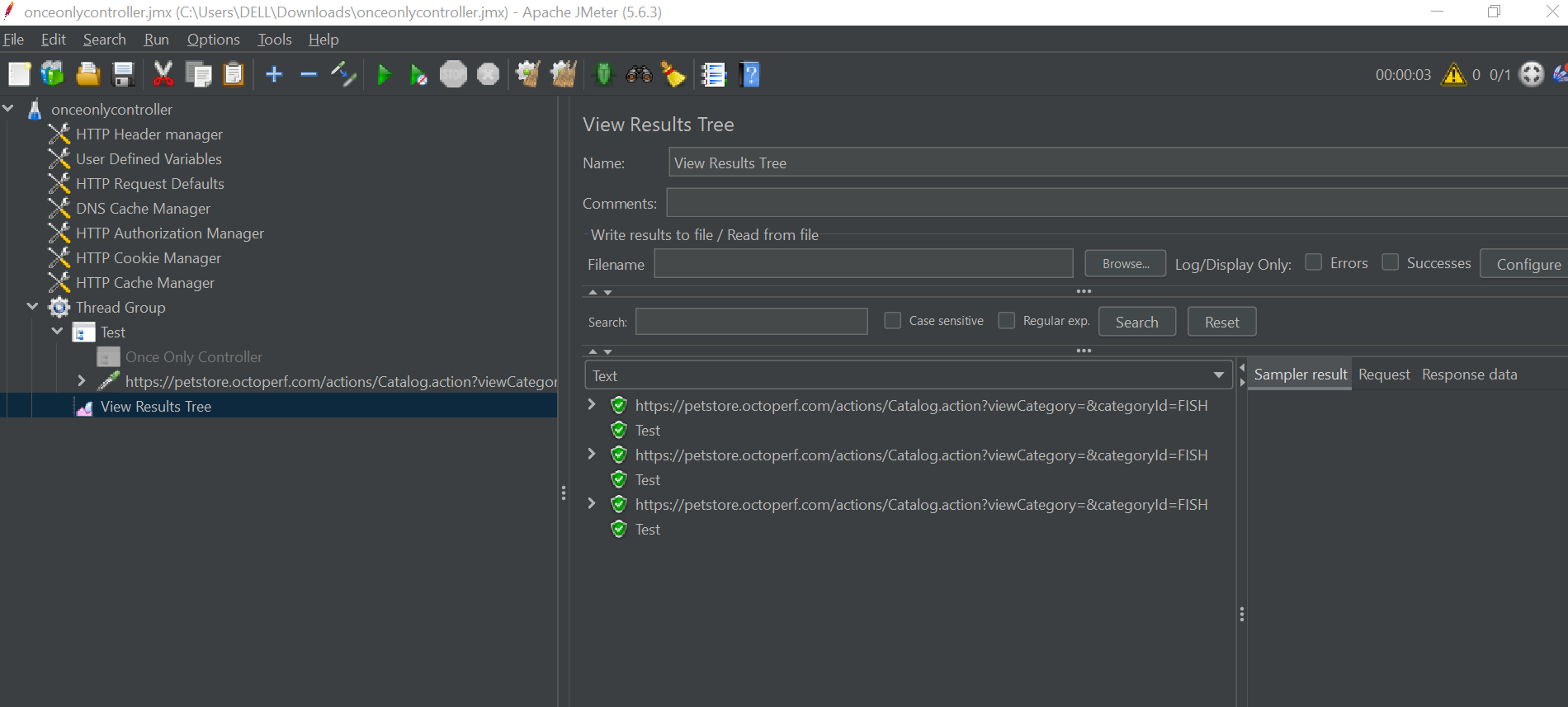
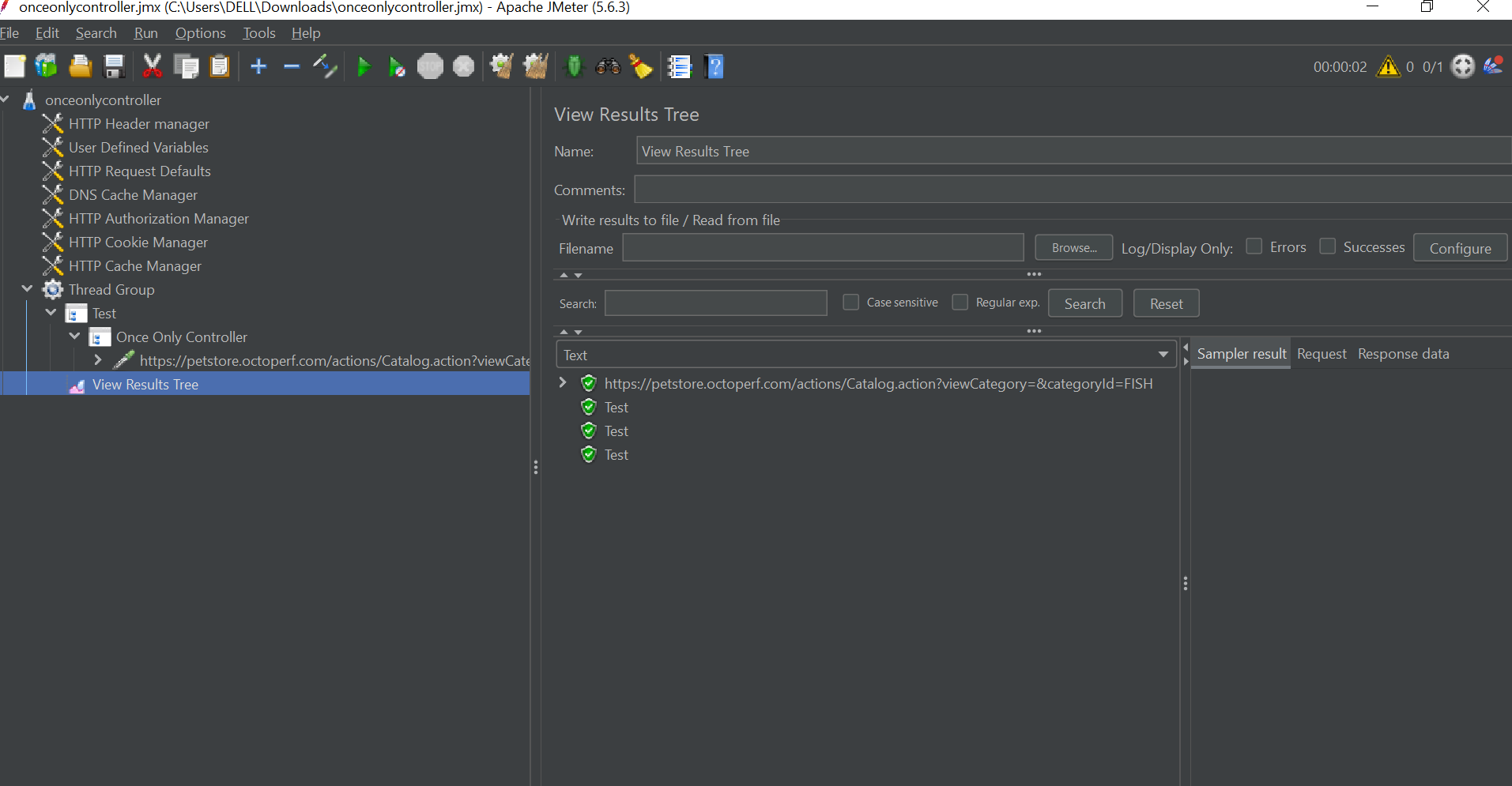
🧩 Once Only Controller







A **Logic Controller** that ensures its **child elements run only once per thread**, regardless of how many iterations the thread executes.

**🎯 Core Purpose**

To execute certain actions **once** per **virtual user** (thread) — typically at the **start of a user session**, even if the test loops multiple times.

**⚙️ How It Works**

| **Behavior** | **Description** |
| --- | --- |
| **Per-thread execution** | Each thread executes the controller's contents **only once** during its life |
| **Loop-independent** | Executes once regardless of parent **loop count** |
| **Triggered on first visit** | Activated when the thread **first encounters** it |
| **Subsequent encounters** | Skipped silently in later loop iterations |

**📌 Common Use Cases**

| **Scenario** | **Why Use Once Only Controller** |
| --- | --- |
| ✅ **User Login** | Users log in once, then perform actions while authenticated |
| ✅ **Session Initialization** | Set up session tokens, headers, or cookies once |
| ✅ **Home Page Visit** | Navigate to a landing page only at session start |
| ✅ **Per-user test data** | Generate unique user data only once |
| ✅ **Database session setup** | Avoid repeated connection or session establishment |

**🧱 Example Test Plan Structure**

text

CopyEdit

Test Plan

└─ Thread Group (Users) [Threads: 5, Loop Count: 10]

├─ Once Only Controller (User Login)

│ ├─ HTTP Request - GET /login\_page

│ ├─ RegEx Extractor (CSRF token)

│ ├─ HTTP Request - POST /login

│ └─ Assertion (Verify login success)

└─ Simple Controller (Shopping Session)

├─ HTTP Request - GET /products

└─ HTTP Request - POST /cart/add\_item

**🧾 Execution Timeline (for one thread):**

| **Iteration** | **What Happens** |
| --- | --- |
| **1** | Executes Login (Once Only Controller), then Shopping Session |
| **2–10** | Skips Login (Once Only Controller), only executes Shopping Session actions |

**🧭 Steps to Add and Configure**

**➕ Add the Controller**

1. Right-click on your **Thread Group**.
2. Navigate to Add > Logic Controller > Once Only Controller.

**➕ Add Child Elements**

* Insert your login/setup requests or data generators inside the Once Only Controller.

⚠️ **Important Placement Rule**:  
To execute only **once per thread**, the Once Only Controller should be **directly under the Thread Group** (not inside another Loop Controller).

**🧠 Best Practices**

| **Tip** | **Why It Matters** |
| --- | --- |
| ✅ **Keep it at top-level** | Ensures true “once per user” behavior |
| ❌ **Avoid nesting in other loops** | Otherwise, it may execute more than once |
| 🔄 **Use with Sessions/Tokens** | Works great for establishing authentication/session state |
| 🧪 **Combine with Debug Sampler** | Easily verify its one-time execution |
| 💾 **Add Cookie Manager (if needed)** | To preserve session across multiple loops |

**🔍 Quick Comparison: Once Only vs. Loop Controller**

| **Feature** | **Once Only Controller** | **Loop Controller** |
| --- | --- | --- |
| Executes only once per... | **Thread** | **Each iteration** (custom count) |
| Repeats on loop? | ❌ Never | ✅ Yes (controlled by loop count) |
| Common Use Case | Login, Setup | Repeating actions like browsing, data entry |

**✅ Summary**

The **Once Only Controller** is ideal for modeling **realistic user sessions** by executing critical setup logic **only once per thread**. It’s essential for actions like:

* Logging in
* Session initialization
* Setting cookies or tokens
* Unique data setup