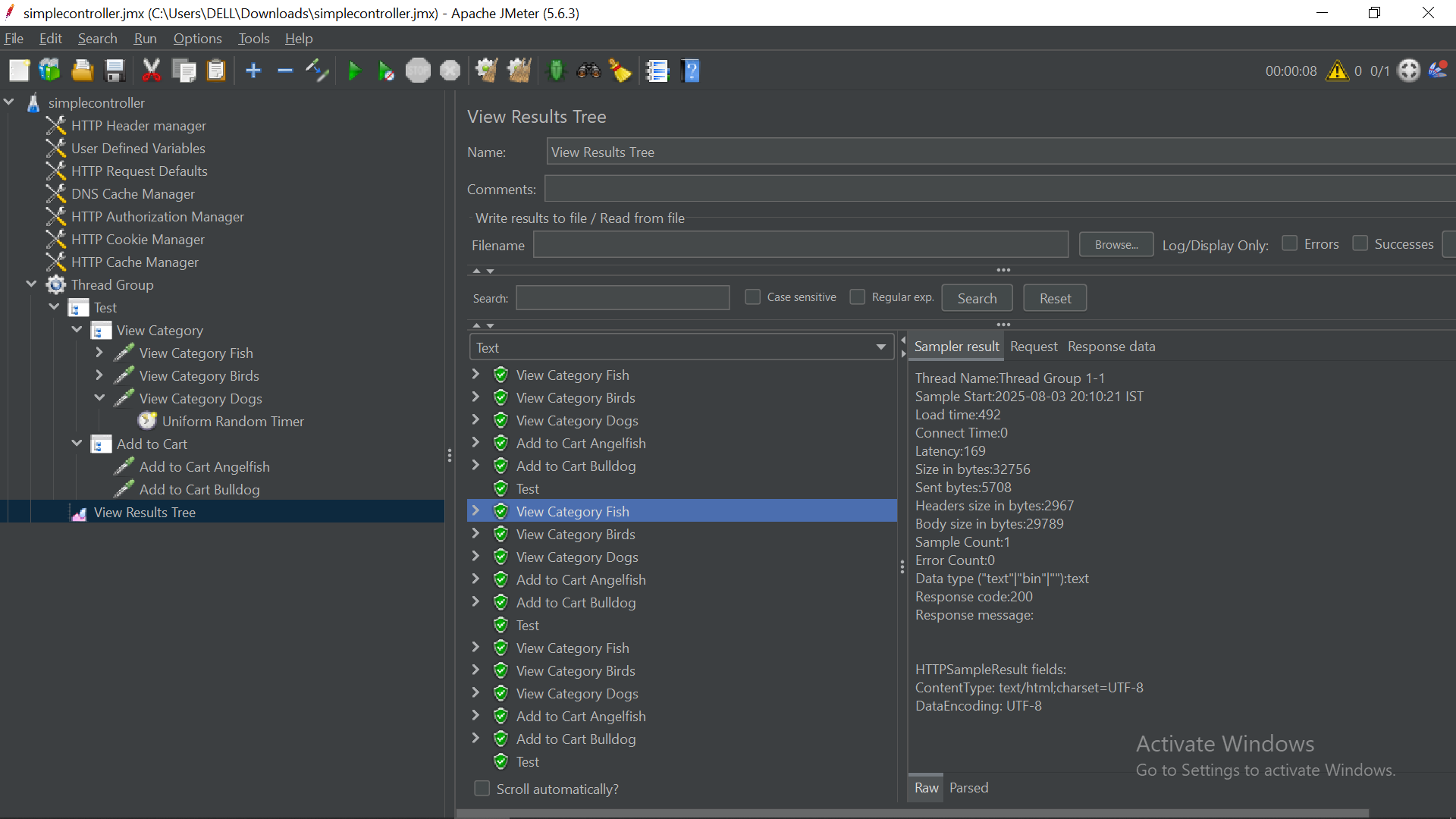
🧩 Simple Controller





The **Simple Controller** is one of the most fundamental and straightforward logic controllers in JMeter. It serves primarily as an **organizational tool** rather than adding any logic to test execution.

**🔧 What It Does**

* **Acts as a container**: The Simple Controller groups samplers (e.g., HTTP requests) and other test elements like preprocessors or assertions.
* **No additional logic**: It does **not** introduce looping, branching, or any conditional execution.
* **Sequential execution**: All elements inside are executed **in order**, just as they appear.

Think of it as a **folder** to logically separate or bundle related steps within a test plan.

**🧠 Why Use It?**

* **Improved readability**: Helps break down large test plans into manageable parts.
* **Logical grouping**: You can represent real-world user flows clearly (e.g., Login → Search → Checkout).
* **Scoped configuration**: Configuration elements like **HTTP Header Manager** or **User Defined Variables** placed alongside a Simple Controller only affect the elements under that controller.

**🪜 How to Add**

1. Right-click on a **Thread Group** or existing controller.
2. Navigate to **Add > Logic Controller > Simple Controller**.

**📦 Example: E-commerce Test Plan Structure**

java

CopyEdit

Test Plan

└── Thread Group

├── Simple Controller (User Login)

│ ├── HTTP Request (Home Page)

│ └── HTTP Request (Login POST)

│

├── Simple Controller (Product Search and View)

│ ├── HTTP Request (Search Page)

│ └── HTTP Request (Product Details Page)

│

└── Simple Controller (Checkout Process)

├── HTTP Request (Add to Cart)

└── HTTP Request (Checkout)

In this example, each Simple Controller encapsulates a specific **user journey**, making the test easier to follow and manage.

**✅ Key Points to Remember**

* Executes children **sequentially**.
* **Does not add logic** (no looping, conditions, etc.).
* Great for **modularizing** and **scoping** tests.
* Ideal for use with tools like **Test Fragment**, **Include Controller**, or when combining with **Enable/Disable** features.