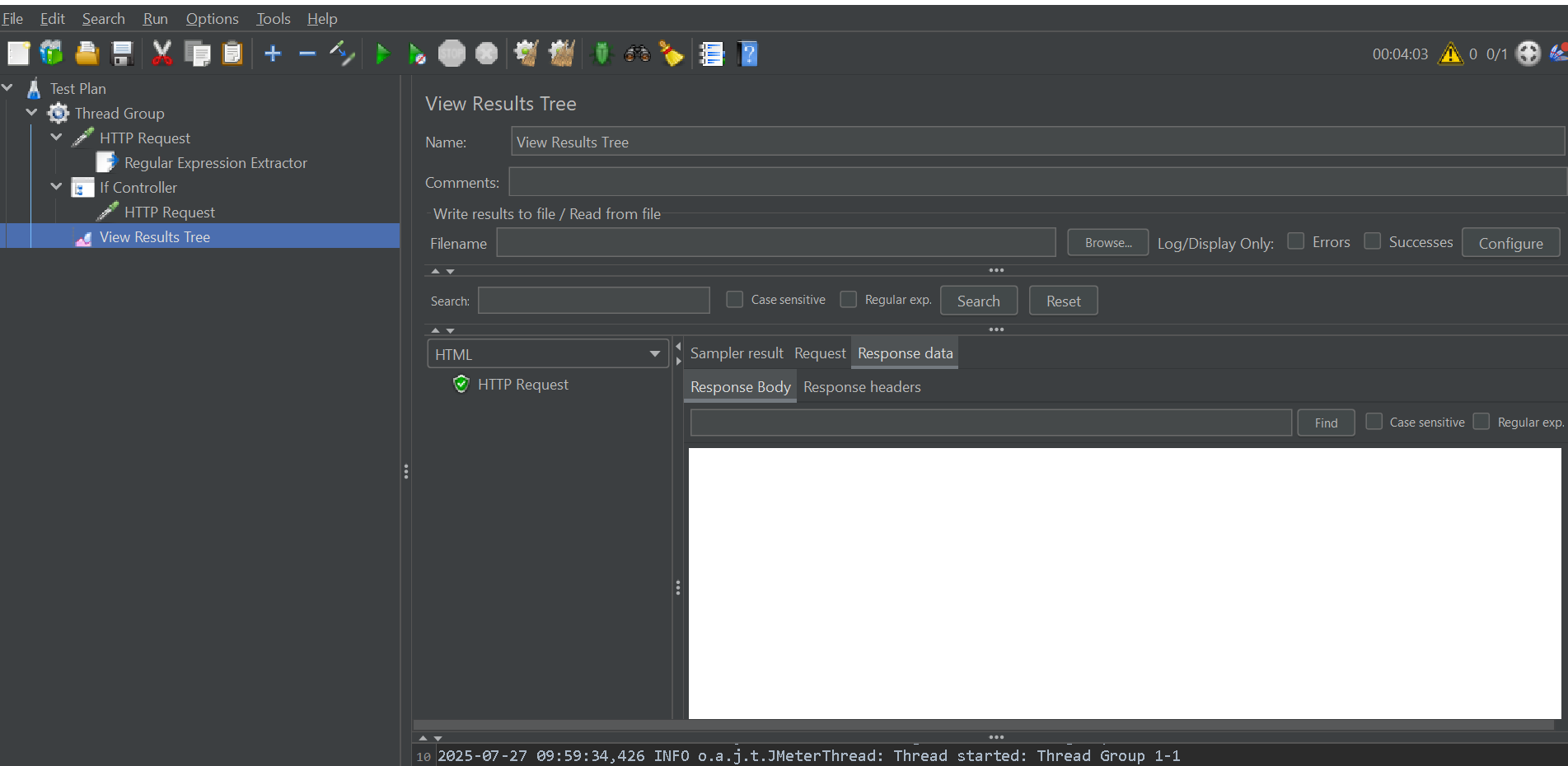
🔧 Controllers in JMeter





Controllers are essential components in Apache JMeter that determine **how and when** requests (Samplers) are executed. They bring **logic and structure** to your test plan, enabling you to simulate **real-world user behaviors** and **complex interaction flows**.

**🧩 Two Main Types of Controllers in JMeter**

1. **Samplers**  
   These represent the actual requests JMeter sends to a server (e.g., HTTP, FTP, JDBC). They define the type of request and the associated data—making them the *fundamental units of work* in any test plan.
2. **Logic Controllers**  
   These control **how and when** Samplers (and other Logic Controllers) are executed. They help you simulate decisions, repetitions, and alternate flows, much like users interacting unpredictably with your application.

**🎯 Why Are Controllers Important?**

* **Simulate Real-World Behavior**  
  Real users don’t just follow a linear flow. They repeat actions, take conditional paths, or act randomly. Logic Controllers help mimic this behavior.
* **Organize Test Plans**  
  Group and structure requests logically, making complex test plans easier to maintain and understand.
* **Control Execution Flow**  
  Define loops, conditions, and execution sequences to replicate precise user journeys.
* **Generate Business-Level Reports**  
  Some controllers (e.g., Transaction Controller) give a higher-level performance view that business stakeholders care about.

**🔁 Common Logic Controllers**

**1. Simple Controller**

* Groups elements without affecting execution.
* Great for organizing test plan sections.

**2. Loop Controller**

* Repeats its child elements a specified number of times.
* Ideal for repeated user actions like browsing or refreshing.

**3. Once Only Controller**

* Runs its child elements **once per thread**, even if loops are present.
* Perfect for login actions that happen only once per user session.

**4. If Controller**

* Executes child elements **only if** a condition evaluates to true.
* Enables conditional logic (e.g., check if login was successful).

**5. While Controller**

* Executes child elements repeatedly **while** a condition holds true.
* Stops once the condition becomes false or upon sampler failure.

**6. Switch Controller**

* Similar to a *switch-case* in programming.
* Executes only the child element matching a switch value (index or variable).

**7. Random Controller**

* Executes **one random** child per iteration.
* Simulates users making random selections (e.g., different products or paths).

**8. Random Order Controller**

* Executes **all** child elements in a **random order**.
* Useful for non-sequential access patterns.

**9. Interleave Controller**

* Alternates between child elements **one at a time** across iterations.
* Useful for testing round-robin or staggered access.

**10. Transaction Controller**

* Groups multiple samplers into a single transaction.
* Measures total time for a business action (e.g., Login + Dashboard Load).
* Can produce a parent sample for consolidated reporting.

**11. Throughput Controller**

* Controls **how often** its children execute based on a percentage or number of iterations.
* Helps simulate real-world traffic proportions.

**12. Runtime Controller**

* Runs child elements for a **specified duration** (in seconds), regardless of loop settings.

**13. Recording Controller**

* Captures and organizes recorded requests during test recording.
* Typically used with the HTTP(S) Test Script Recorder.

**14. Module Controller**

* Enables modular test plans.
* Refers to another controller (like a Test Fragment or Simple Controller) elsewhere in the test plan.

**15. Include Controller**

* Imports and executes an **external .jmx file** into the current test.
* Useful for reusing common components across projects.

**16. Critical Section Controller**

* Ensures **only one thread** executes its children at a time.
* Prevents race conditions or shared resource conflicts.

**🧠 How Controllers Influence Test Execution**

| **Aspect** | **How Controllers Help** |
| --- | --- |
| **Flow Control** | Dictate the execution order of requests |
| **Iteration** | Loop, While, and Runtime controllers handle repetitions |
| **Conditional Logic** | If and Switch allow selective execution |
| **Randomization** | Random/Interleave/Random Order add variability |
| **Synchronization** | Critical Section manages concurrency |
| **Business Metrics** | Transaction Controller aggregates timings for business actions |