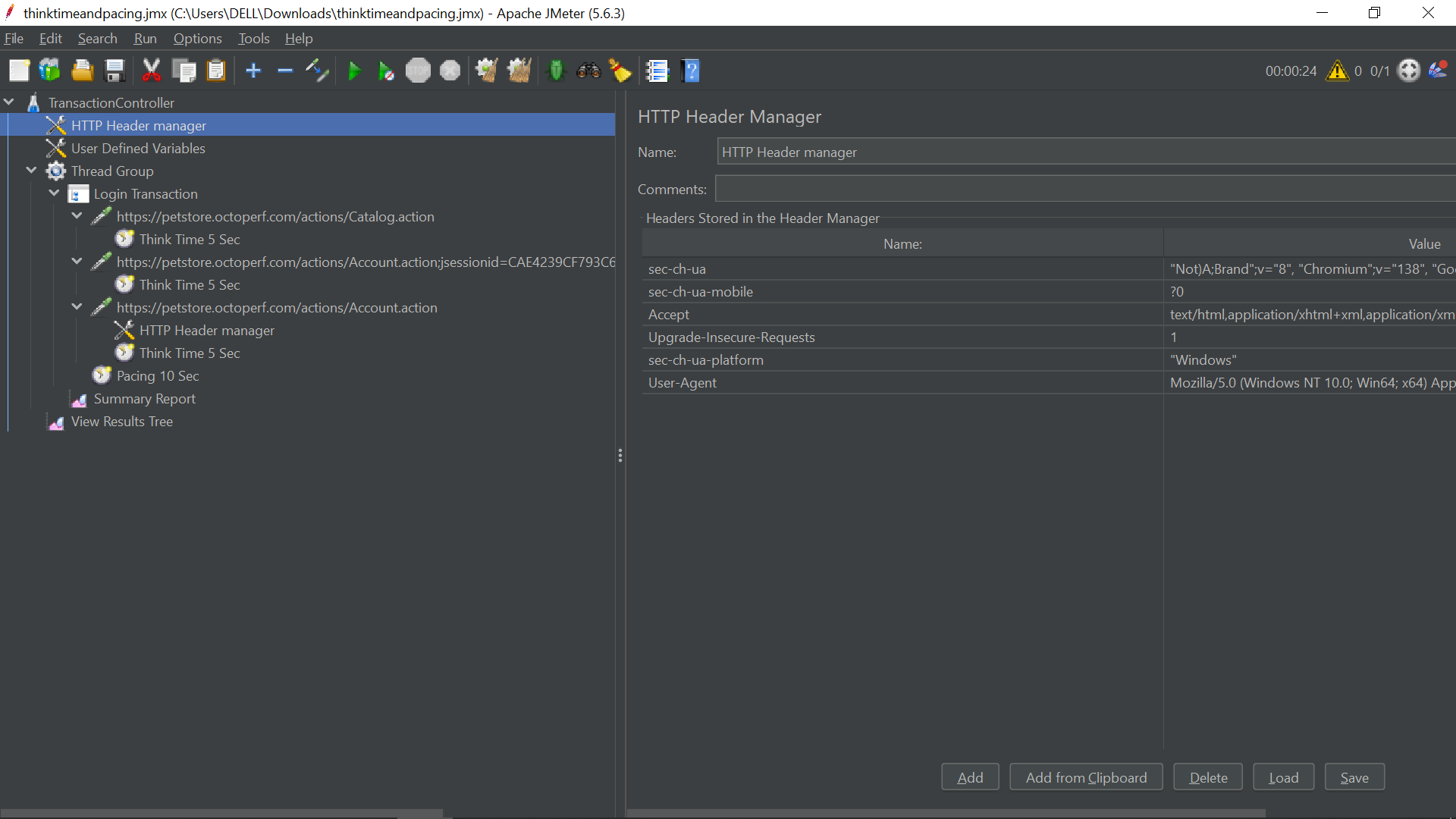
🧠 Think Time vs ⏱️ Pacing





**Simulate Real User Behavior with Precision**

**1️⃣ What Are Think Time and Pacing?**

**🔹 Think Time**

The natural delay a user introduces between **steps** in a workflow.

🧍 Example: A user logs in, spends a few seconds reviewing their dashboard, and then clicks on “Search.”  
This **pause between actions** is the think time.

**🔹 Pacing**

The delay between **iterations** of a full user journey.

🕐 Example: After completing a search and logging out, the user waits 10 seconds before repeating the workflow.  
This **pause between flows** is the pacing.

**✅ Why They Matter**

| **Purpose** | **Description** |
| --- | --- |
| 🎯 Realism | Mimics real user behavior (humans pause and think!) |
| 📊 Load Modeling | Controls Transactions Per Second (TPS) more accurately |
| 🔒 Session Testing | Helps surface session timeout and token expiration issues |
| 📉 Avoid Overload | Prevents unrealistic hammering of the server (which JMeter users often do!) |

**2️⃣ Practical Demo in JMeter**

**🔁 Scenario: Simulate the following user journey:**

1. Navigate to Homepage
2. Login
3. Search for a product
4. Logout

We’ll implement **Think Time** between steps, and **Pacing** between iterations.

**📐 Step-by-Step Setup**

**Step 1: Create the Test Plan**

* **Add Thread Group**  
  Test Plan → Add → Threads (Users) → Thread Group
  + Threads (users): 1
  + Ramp-Up: 1
  + Loop Count: 2
* **Add HTTP Request Samplers**  
  For each step (Homepage, Login, Search, Logout):  
  Thread Group → Add → Sampler → HTTP Request

**Step 2: Add Think Time (Between Actions)**

Add **Uniform Random Timer** before Login, Search, and Logout:

📍 **Where?**  
Direct child of Thread Group, placed **before** each request.

🛠️ **How?**  
Add → Timer → Uniform Random Timer

* Constant Delay Offset (ms): 1000 (1 sec)
* Random Delay Max (ms): 5000 (up to 5 sec)

💡 This results in a **1–6 second** random delay before each action — simulating natural user pause.

**Step 3: Add Pacing (Between Iterations)**

Add a **Constant Timer** **after** the last HTTP request:

📍 **Where?**  
Direct child of Thread Group, **after Logout request**.

🛠️ **How?**  
Add → Timer → Constant Timer

* Delay (ms): 10000 (10 seconds)

This enforces a **10-second pause** between each full business flow.

**✅ Final Test Plan Hierarchy**

pgsql

CopyEdit

Test Plan

└── Thread Group (Loop Count: 2)

├── Homepage HTTP Request

├── Uniform Random Timer (Think Time 1–6s)

├── Login HTTP Request

├── Uniform Random Timer (Think Time 1–6s)

├── Search HTTP Request

├── Uniform Random Timer (Think Time 1–6s)

├── Logout HTTP Request

└── Constant Timer (Pacing 10s)

**Step 4: Run and Monitor**

* **Add View Results Tree**: Good for development/debugging  
  Add → Listener → View Results Tree
* **Add Aggregate Report**: For performance stats  
  Add → Listener → Aggregate Report

🚀 **Run the test**:  
You will see:

* Random delays between individual requests (Think Time)
* Fixed delay before loop restarts (Pacing)

**💡 Pro Tips**

| **Tool** | **Use Case** |
| --- | --- |
| ⏳ **Constant Throughput Timer** | Maintain TPS/TPM target instead of fixed pacing |
| ⚙️ **JSR223 Timer** | Use Groovy for highly custom delays |
| 📦 **Throughput Controller** | Control request flow by probability or percentages |
| 🧪 **Timer Scoping** | Timers affect **siblings and children** — watch placement! |

**🧠 Summary: Think Time vs Pacing**

| **Feature** | **Think Time** | **Pacing** |
| --- | --- | --- |
| Scope | Between steps within 1 iteration | Between iterations |
| Goal | Simulate user reading/thinking | Control test intensity / loop spacing |
| Element | Uniform Random Timer (or others) | Constant Timer (after full flow) |