**JMeter Configuration for n11.com Search Module Load Testing**

**Test Plan Configuration:**

In the JMeter Test Plan for testing the search functionality on [n11](https://www.n11.com) the following configurations were applied:

**1. User Defined Variables:**

- Define any variables such as URLs, product names, or file paths for reusability across the test plan.

**2. HTTP Request Defaults\*\*:**

- Set the base URL to avoid repeating the host for each request (`https://www.n11.com`).

**3. DNS Cache Manager:**

- Used to cache DNS lookups and reduce overhead during multiple requests.

**4. HTTP Authorization Manager:**

- Manages credentials if the site requires login or any authentication during testing.

**5. HTTP Cache Manager:**

- Configures the behavior of client-side caching to simulate real browser interactions more effectively.

**6. HTTP Header Manager:**

- Adds necessary headers such as `User-Agent`, `Accept`, and other HTTP headers.

**7. HTTP Cookie Manager:**

- Manages cookies like a real browser to ensure session persistence during tests.

**Reporting Tools:**

**Three types of listeners were added for comprehensive reporting:**

* Aggregate Report: Provides a summary of key metrics such as response times and throughput. <https://prnt.sc/aQhxkSGBKi_F>
* View Results in Table: Displays detailed results for each individual request in a tabular format. <https://prnt.sc/aQhxkSGBKi_F>
* View Results Tree: Offers a comprehensive view, showing request and response data for each test step. <https://prnt.sc/kH0uGf2nZ7VR>

**JMeter Test Scenarios:**

**Preparation:**

Home page is created to check effectiveness of [n11](https://www.n11.com) to verify that the system was running and responding with status `200`.

**Thread Group Configuration:**

* Number of Threads (Users): 1
* Ramp-up period: 1 second
* Loop Count: 1

Adjustments can be made to the number of users, ramp-up time, and loop count to simulate the system's response to multi-user or repetitive operations. For example:

* Thread Group: 100 Threads (users)
* Ramp-up Time: 10 seconds

- This configuration will simulate 100 users starting within 10 seconds, at a rate of 10 users per second.

**Scenarios:**

**Scenario 1:** Basic Search (Typical User Behavior)

**Action:** Perform a `GET` request on [n11](https://www.n11.com).

**Search Term:** Use a keyword such as "dress" for the search.

**Expected Response:**

* HTTP status code should be `200`.
* Successful loading of search results.

**Scenario 2:** Search Module with Different Variables

**Action:** Perform a search using various keywords from a `.csv` file (e.g., "hat", "dress", "glasses").

**Expected Response:**

* HTTP status code should be `200`.
* All keywords should return results within acceptable response times.

**Scenario 3:** Rapid Search Test

**Action:** Perform multiple rapid searches for "dress" in quick succession.

**Expected Response:**

* HTTP status code should be `200`.
* Duration Assertion: Expect response time to be less than 2000 milliseconds.

**Scenario 4:** Empty Search Test

**Action:** Perform a search with an empty `q` parameter (`q=`).

**Expected Response**:

* HTTP status code should be `200`.
* The server should handle the empty search correctly.
* Duration Assertion: Expect response time to be less than 2500 milliseconds.

**Scenario 5:** Searching with Unnecessary Spaces

Action: Search for a term with leading or trailing spaces (e.g., " laptop ").

**Expected Response**:

* HTTP status code should be `200`.
* The system should trim unnecessary spaces and return accurate results.
* Duration Assertion: Response time should be less than 2500 milliseconds.

**Scenario 6:** Multiple Search Terms Test

**Action:** Search using multiple keywords (e.g., "laptop mouse cat food").

**Expected Response:**

* HTTP status code should be `200`.
* The system should return results for all terms.
* Duration Assertion: Response time should be less than 2500 milliseconds.

**Scenario 7:** Searching in Different Languages

**Action:** Search using product names in various languages (e.g., Turkish, English).

**Expected Response:**

* HTTP status code should be `200`.
* The system should handle searches in different languages efficiently.
* Duration Assertion: Response time should be less than 2500 milliseconds.

**Scenario 8:** Numeric Search Terms

**Action:** Perform a search using numbers or decimals (e.g., "12345", "199.99").

**Expected Response:**

* HTTP status code should be `200`.
* The system should return relevant results based on numeric terms.

**Scenario 9:** Special Characters in Search

**Action:** Perform a search using special characters (e.g., "@", "#", "&", "\*").

**Expected Response:**

* HTTP status code should be `200`.
* The system should handle special characters gracefully without errors.

**Scenario 10:** Rapid Search with Different Variables\*\*

Action: Perform rapid searches using various keywords from a `.csv` file (e.g., "hat", "dress", "glasses").

**Expected Response:**

* HTTP status code should be `200`.
* Duration Assertion: Responses should be returned within 2000 milliseconds.

**Additional Enhancements:**

For each scenario, \*\*Response Assertions\*\* were added to ensure the system returned the correct status code (200 OK). Additionally, \*\*Duration Assertions\*\* were used to ensure responses were within acceptable performance limits (e.g., less than 2000 ms for rapid searches).

With this setup, the search module can be thoroughly tested for performance, responsiveness, and correct behavior across various input scenarios. Each test simulates real-world conditions, ensuring the system can handle load effectively while delivering accurate results.