

परिचय: JMeter का उपयोग करना

JMeter एक शक्तिशाली ओपन-सोर्स टूल है जो मुख्य रूप से प्रदर्शन परीक्षण के लिए डिज़ाइन किया गया है, हालांकि इसे कार्यात्मक परीक्षण के लिए भी उपयोग किया जा सकता है। यह वेब एप्लिकेशन, एपीआई, डेटाबेस और अन्य सेवाओं के प्रदर्शन को परीक्षण करने के लिए व्यापक रूप से उपयोग किया जाता है, कई उपयोगकर्ताओं को सिमुलेट करने और सिस्टम के लोड के तहत कैसे प्रतिक्रिया करता है, मापने के लिए। नीचे एक चरण-दर-चरण गाइड है कि कैसे JMeter को प्रभावी रूप से उपयोग किया जा सकता है, इंस्टॉलेशन से शुरू करके एक बुनियादी परीक्षण चलाने और परिणामों का विश्लेषण करने तक।

चरण 1: JMeter इंस्टॉल करें

JMeter का उपयोग करने से पहले, आपको इसे अपने मशीन पर इंस्टॉल करना होगा। यहाँ कैसे:

- प्रारंभिक आवश्यकताएँ:** JMeter एक जावा आधारित एप्लिकेशन है, इसलिए आपको जावा डेवलपमेंट किट (JDK) या जावा रनटाइम एनवायरनमेंट (JRE) इंस्टॉल किया जाना चाहिए। संस्करण 8 या उससे ऊपर सिफारिश किया जाता है। आप इसे अपने टर्मिनल या कमांड प्रॉम्प्ट में `java -version` चलाकर सत्यापित कर सकते हैं। अगर जावा इंस्टॉल नहीं है, तो इसे ऑफिसियल जावा वेबसाइट से डाउनलोड और इंस्टॉल करें।
- JMeter डाउनलोड करें:** JMeter वेबसाइट पर जाएं और नवीनतम बाइनरी रिलीज़ (आम तौर पर एक .zip या .tgz फ़ाइल) डाउनलोड करें।
- फ़ाइल को एक्सट्रैक्ट करें:** डाउनलोड की गई फ़ाइल को अपने पसंदीदा डायरेक्टरी में एक्सट्रैक्ट करें (उदाहरण के लिए, `C:\JMeter` या `~/JMeter` आधारित सिस्टम पर)। एक्सट्रैक्ट किया गया फ़ोल्डर को चलाने के लिए सभी फ़ाइलें शामिल करता है।
- JMeter लॉन्च करें:**
 - एक्सट्रैक्ट किया गया डायरेक्टरी के अंदर `bin` फ़ोल्डर में जाएं (उदाहरण के लिए, `C:\JMeter\apache-jmeter-5.x\bin`).
 - उपयुक्त एक्सिक्यूटेबल चलाएं:
 - Windows पर:** `jmeter.bat` पर दबाएं या कमांड प्रॉम्प्ट से चलाएं।
 - Linux/macOS पर:** टर्मिनल में `./jmeter.sh` चलाएं।
 - यह JMeter ग्राफिकल यूजर इंटरफ़ेस (GUI) खोलता है, जहाँ आप अपने टेस्ट प्लान बनाएँगे और प्रबंधित करेंगे।

चरण 2: एक टेस्ट प्लान बनाएं

एक **टेस्ट प्लान** JMeter में आप क्या परीक्षण करना चाहते हैं और कैसे, को परिभाषित करता है। यह आपके प्रदर्शन परीक्षण का नींव है। यहाँ एक बुनियादी टेस्ट प्लान सेट अप करने के लिए:

एक थ्रेड ग्रुप जोड़ें

1. `TestPlan` में, बाएं पैन में **टेस्ट प्लान** नोड पर दाएं क्लिक करें और `TestPlan > Add Thread Group (JMeter)` > `TestPlan` चुनें।
2. थ्रेड ग्रुप को कॉन्फ़िगर करें:
 - **थ्रेड्स (उपयोगकर्ता) की संख्या**: सिमुलेट करने वाले वर्चुअल उपयोगकर्ताओं की संख्या (उदाहरण के लिए, 10).
 - **रैम्प-अप अवधि (सेकंड)**: `TestPlan` सभी थ्रेड्स को शुरू करने में लेता है (उदाहरण के लिए, 10 सेकंड का मतलब है 10 सेकंड में 1 थ्रेड प्रति सेकंड शुरू होता है).
 - **लूप काउंट**: प्रत्येक थ्रेड परीक्षण को कितनी बार दोहराता है (उदाहरण के लिए, 1 एकल रन के लिए, या "TestPlan" चेक करें निरंतर लूपिंग के लिए).

थ्रेड ग्रुप उपयोगकर्ता ट्रैफिक को सिमुलेट करता है। उदाहरण के लिए, 10 थ्रेड्स के साथ 10 सेकंड रैम्प-अप और 1 लूप का मतलब है 10 उपयोगकर्ता 10 सेकंड में एप्लिकेशन पर हिट करेंगे, प्रत्येक परीक्षण एक बार करेंगे।

एक सैंपलर जोड़ें

सैंपलर्स `TestPlan` द्वारा टारगेट सिस्टम को भेजे गए अनुरोधों को परिभाषित करते हैं। वेब परीक्षण के लिए सबसे आम है `TestPlan` अनुरोध सैंपलर: 1. थ्रेड ग्रुप पर दाएं क्लिक करें और `TestPlan > Add Sampler (JMeter)` > `TestPlan` चुनें। 2. `TestPlan` अनुरोध को कॉन्फ़िगर करें: - **प्रोटोकॉल**: `http` या `https`. - **सर्वर नाम या आईपी**: टारगेट सिस्टम का डोमेन या आईपी (उदाहरण के लिए, `example.com`). - **पोर्ट नंबर**: आम तौर पर `TestPlan` के लिए 80 या `TestPlan` के लिए 443 (अगर मानक तो खाली छोड़ें). - **मैथड**: अनुरोध प्रकार के अनुसार GET, POST आदि. - **पथ**: विशिष्ट एंडपॉइंट या पेज (उदाहरण के लिए, / होमपेज के लिए). - आवश्यकता के अनुसार पैरामीटर या अनुरोध बॉडी जोड़ें (उदाहरण के लिए, `TestPlan` अनुरोधों के लिए).

इस सैंपलर `TestPlan` को बताता है कि प्रत्येक वर्चुअल उपयोगकर्ता क्या कार्य करनी चाहिए।

एक लिसनर जोड़ें

लिसनर्स परीक्षण परिणामों को इकट्ठा और प्रदर्शित करते हैं: 1. थ्रेड ग्रुप पर दाएं क्लिक करें और `TestPlan > Add Listener (JMeter)` > `TestPlan` `TestPlan` (या किसी अन्य लिसनर जैसे `TestPlan` `TestPlan`) चुनें। 2. `TestPlan` `TestPlan` प्रत्येक अनुरोध के लिए विस्तृत परिणाम दिखाता है, जिसमें प्रतिक्रिया समय, स्थिति कोड और प्रतिक्रिया डेटा शामिल हैं।

लिसनर्स आपके एप्लिकेशन के प्रदर्शन को परीक्षण के दौरान विश्लेषण करने के लिए आवश्यक हैं।

टेस्ट प्लान को सेइव करें

`TestPlan` > `TestPlan` `TestPlan` पर क्लिक करें और अपने `.jmx` फ़ाइल को सेइव करें (उदाहरण के लिए, `mytest.jmx`). इससे आप इसे बाद में पुनः उपयोग या संशोधित कर सकते हैं।

चरण 3: टेस्ट चलाएं

अपने टेस्ट को चलाने के लिए: 1. `TestPlan` में, टूलबार में हरी `TestPlan` बटन (□) पर क्लिक करें, या `TestPlan` > `TestPlan` पर जाएं। 2. `TestPlan` थ्रेड ग्रुप में परिभाषित उपयोगकर्ताओं को सिमुलेट करेगा, परिभाषित `TestPlan` अनुरोध भेजेगा। 3. लिसनर (उदाहरण के लिए, `TestPlan` `TestPlan`) में परिणामों को परीक्षण चलने के साथ भरते देखें।

छोटी परीक्षाओं के लिए, लोड के माध्यम से चलाना ठीक है। बड़े परीक्षाओं के लिए, नीचे “उन्नत उपयोग” खंड में नॉन-संक्रमण मोड के बारे में देखें।

चरण 4: परिणामों का विश्लेषण करें

परीक्षण पूरा होने के बाद, लिसनर का उपयोग परिणामों को समीक्षित करने के लिए करें: - **संक्षिप्त परिणाम**: प्रत्येक अनुरोध के सफल/असफल, प्रतिक्रिया समय और प्रतिक्रिया डेटा दिखाता है। - **विस्तृत परिणाम**: औसत प्रतिक्रिया समय, थ्रूपुट (सेकंड प्रति अनुरोध) और त्रुटि दर जैसे संग्रहित मेट्रिक्स प्रदान करता है।

ये मेट्रिक्स आपको एप्लिकेशन के प्रदर्शन का मूल्यांकन करने में मदद करते हैं (उदाहरण के लिए, लोड के तहत कितनी तेजी से प्रतिक्रिया करता है या कितने उपयोगकर्ताओं पर विफल हो जाता है)।

उदाहरण: एक सरल वेब पेज का परीक्षण

example.com को 10 उपयोगकर्ताओं के साथ परीक्षण करें: 1. **ब्रॉउज़र** लॉन्च करें। 2. एक थ्रेड ग्रुप जोड़ें: - थ्रेड्स: 10 - रैम्प-अप: 10 सेकंड - लूप काउंट: 1 3. एक **सर्वर अनुरोध** सैंपलर जोड़ें: - प्रोटोकॉल: http - सर्वर नाम: example.com - मेथड: GET - पथ: / 4. एक **लिसनर** जोड़ें। 5. सेटव और परीक्षण चलाएं। 6. **परिणाम** में परिणामों को देखें प्रतिक्रिया समय और स्थिति कोड (उदाहरण के लिए, 200 OK) देखें।

इस सरल परीक्षण example.com को 10 एक साथ उपयोगकर्ताओं के साथ कैसे प्रदर्शन करता है, मापता है।

उन्नत उपयोग

जटिल सीनारियो के लिए, **कॉन्फिगर** अतिरिक्त विशेषताएं प्रदान करता है:

पैरामीटराइजेशन

कॉन्फिगर का उपयोग करके, अपने परीक्षण में अलग डेटा (उदाहरण के लिए, उपयोगकर्ता नाम, पासवर्ड) को एक **फ़ाइल** से फीड करें: 1. थ्रेड ग्रुप में **कॉन्फिगर** > **कॉन्फिगर** जोड़ें। 2. फ़ाइल पथ और वेरिएबल नामों (उदाहरण के लिए, \${username}) को स्पेसिफ़ाई करें, फिर इन वेरिएबल्स का उपयोग अपने **सर्वर अनुरोध** में करें।

कॉरिलेशन

दिनामी मानों (उदाहरण के लिए, सेशन आईडी, टोकन) को संभालें: 1. एक अनुरोध के बाद **कॉन्फिगर** > **कॉन्फिगर** या **कॉन्फिगर** जोड़ें। 2. एक मान (उदाहरण के लिए, एक टोकन) को एक वेरिएबल (उदाहरण के लिए, \${token}) में निकालें। 3. \${token} को अगले अनुरोधों में उपयोग करें (उदाहरण के लिए, हेडर या पैरामीटर में)।

असर्टशन्स

प्रतिक्रियाओं को सत्यापित करें: 1. सैंपलर में `assert` > `assertEquals` जोड़ें। 2. विशेष टेक्स्ट, प्रतिक्रिया कोड (उदाहरण के लिए, 200) या `assert` परीक्षण के लिए `assertEquals` का उपयोग करें।

नॉन-गैर मोड

बड़े परीक्षणों या ऑटोमेशन के लिए: - कमांड लाइन से `java` चलाएं: `jmeter -n -t mytest.jmx -l results.jtl -n: नॉन-गैर मोड. -t: टेस्ट प्लान फ़ाइल. -l: परिणाम लॉग फ़ाइल (उदाहरण के लिए, .jtl). - एक report रिपोर्ट जनरेट करें: jmeter -n -t mytest.jmx -l results.jtl -e -o report_folder -e: रिपोर्ट जनरेट करें. -o: report रिपोर्ट के लिए आउटपुट फ़ोल्डर.`

वितरित परीक्षण

उच्च लोड के लिए: 1. अलग मशीनों पर कई `slave` इंस्टेंस सेट अप करें। 2. मास्टर मशीन की `jmeter.properties` फ़ाइल को स्लेव आईपी शामिल करने के लिए कॉन्फ़िगर करें। 3. मास्टर से परीक्षण चलाएं लोड जनरेशन को को-ऑर्डिनेट करने के लिए।

अतिरिक्त टिप्स

- **सहयोगी प्रोटोकॉल:** `Http` के अलावा, `FTP`, `SMTP`, `POP` और अधिक को समर्थन करता है, जिससे इसे विभिन्न सिस्टम के लिए बहुमुखी बनाता है।
- **सर्वर मॉनिटरिंग:** `Http` को सर्वर-साइड टूल्स (उदाहरण के लिए, `WebLogic`) के साथ जोड़ें `Http`, मेमोरी और डेटाबेस प्रदर्शन को परीक्षण के दौरान मॉनिटर करने के लिए।
- **वास्तविक सिमुलेशन:** `Http` `Http` `Http` या `Http` जोड़ें ब्राउज़र व्यवहार या उपयोगकर्ता सोचने का समय को नकल करने के लिए।
- **कार्यात्मक परीक्षण:** असर्टशन्स का उपयोग करें प्रतिक्रिया की सटीकता को सत्यापित करने के लिए, केवल प्रदर्शन नहीं।

निष्कर्ष

`Http` का उपयोग करने के लिए: 1. इसे जावा के साथ इंस्टॉल करें और `Http` लॉन्च करें। 2. एक थ्रेड ग्रुप, सैंपलर्स और लिसनर्स के साथ एक टेस्ट प्लान बनाएं। 3. टेस्ट तत्वों (उपयोगकर्ता, अनुरोध आदि) को कॉन्फ़िगर करें। 4. परीक्षण चलाएं और परिणामों का विश्लेषण करें।

एक सरल शुरुआत के लिए, ऊपर दिखाए गए तरीके से एक वेबपेज जैसे `example.com` का परीक्षण करें। जैसे-जैसे आप अनुभव प्राप्त करते हैं, उन्नत विशेषताओं जैसे पैरामीटराइजेशन, कॉरिलेशन और वितरित परीक्षण का पता लगाएं जटिल, वास्तविक दुनिया के सीनारियो को संभालने के लिए। `Http` की लचीलापन और विस्तृत दस्तावेज़ीकरण इसे नए और उन्नत परीक्षकों के लिए एक मूल्यवान टूल बनाते हैं।