

# JMeter の使い方

JMeter は、ウェブアプリケーションの負荷テストとパフォーマンスの測定のために設計されたオープンソースのソフトウェアツールです。ただし、データベースや FTP サーバーなど他の種類のシステムもテストできます。以下は、ウェブアプリケーションを中心に JMeter を使用してパフォーマンステストを開始するための明確なステップバイステップガイドです。

---

## ステップ 1: JMeter のインストール

- **前提条件:** JMeter は Java ベースのアプリケーションなので、マシンに Java（バージョン 8 以降）がインストールされている必要があります。コマンドラインで `java -version` を実行して確認できます。
  - **ダウンロード:** Apache JMeter ウェブサイトにアクセスし、最新バージョン（.zip または .tgz ファイル）をダウンロードします。
  - **インストール:** ダウンロードしたファイルを好きなディレクトリに展開します（例：Windows の `C:\JMeter` または Linux/Mac の `/opt/jmeter`）。追加のインストール手順はありません。
- 

## ステップ 2: JMeter の起動

- JMeter フォルダ内の `bin` ディレクトリに移動します（例：`C:\JMeter\apache-jmeter-x.x\bin`）。
  - **Windows:** `jmeter.bat` をダブルクリックするか、コマンドラインから実行します。
  - **Linux/Mac:** ターミナルを開き、`bin` ディレクトリに移動し、`./jmeter.sh` を実行します。
  - グラフィカルユーザーインターフェース（GUI）が開き、JMeter のワークベンチが表示されます。
- 

## ステップ 3: テストプランの作成

- **テストプラン** はパフォーマンステストの基盤です。テストの内容と方法を示します。
  - JMeter GUI の左ペインにテストプランが既に表示されています。右クリックして名前を変更するか（例：「Web パフォーマンステスト」）そのままにしておきます。
- 

## ステップ 4: スレッドグループの追加

- **スレッドグループ** は、サーバーにリクエストを送信する仮想ユーザーをシミュレートします。

- テストプラン > **Add > Threads (Users) > Thread Group** の順に右クリックします。
  - 設定:
    - **Number of Threads (users):** 仮想ユーザーの数を設定します (例: 10)。
    - **Ramp-Up Period (seconds):** すべてのスレッドを開始するのにかかる時間 (例: 10 秒は 1 秒ごとに 1 スレッド)。
    - **Loop Count:** テストを繰り返す回数 (例: 1 または「Forever」を選択して連続テスト)。
- 

## ステップ 5: サンプラーの追加

- **サンプラー**はサーバーに送信されるリクエストを定義します。ウェブテストには HTTP リクエストサンプラーを使用します。
  - スレッドグループ > **Add > Sampler > HTTP Request** の順に右クリックします。
  - 設定:
    - **Server Name or IP:** ターゲットウェブサイトをを入力します (例: example.com)。
    - **Path:** エンドポイントを指定します (例: /login)。
    - **Method:** テストシナリオに基づいて GET、POST などを選択します。
- 

## ステップ 6: リスナーの追加

- **リスナー**はテスト結果を表示および分析します。
  - スレッドグループ > **Add > Listener >** (例: **View Results Tree** または **Summary Report**) の順に右クリックします。
  - 人気のあるオプション:
    - **View Results Tree:** リクエスト/レスポンスデータの詳細を表示します。
    - **Summary Report:** 平均応答時間やエラー率などの集計メトリクスを提供します。
- 

## ステップ 7: テストの設定

- テストを追加の要素で強化します (オプションですが有用です):
    - **タイマー:** リクエスト間に遅延を追加します (例: スレッドグループ > **Add > Timer > Constant Timer**)。
    - **アサーション:** サーバーのレスポンスを検証します (例: HTTP リクエスト > **Add > Assertions > Response Assertion**)。
    - **Config Elements:** 変数や HTTP のデフォルトを設定します (例: **HTTP Request Defaults**)。
-

## ステップ 8: テストの実行

- テストプランを保存します (**File > Save**) として .jmx ファイルを再利用します。
  - ツールバーの緑の **Run** ボタン (三角形) をクリックするか、**Run > Start** に移動します。
  - JMeter はスレッドグループの設定に基づいてテストを実行します。
- 

## ステップ 9: 結果の分析

- テストが完了したら、リスナーを確認します：
    - **View Results Tree**: 個々のリクエストの成功/失敗を確認します。
    - **Summary Report**: スループット、応答時間、エラー率などのメトリクスをレビューします。
  - ボトルネック (例: 応答時間が遅い) や調査が必要なエラーを確認します。
- 

## ステップ 10: テストの改善

- 結果に基づいてテストプランを調整します：
    - スレッドの数を増減させます。
    - より現実的なシナリオを追加します (例: 複数の HTTP リクエスト)。
    - 正確性のためにタイマーやアサーションを調整します。
  - 保存してテストを再実行し、改善を検証します。
- 

## 追加のヒント

- **保存と読み込み**: .jmx ファイルを使用してテストプランを保存し、**File > Open** から後で読み込みます。
- **GUI 対非 GUI**: 大規模なテストの場合、システムリソースを節約するために非 GUI モードで JMeter を実行します (例: `jmeter -n -t test.jmx -l results.jtl`)。
- **さらに学ぶ**: 経験を積むにつれて、パラメータ化 (動的データ用) や相関 (動的ウェブアプリ用) などの高度な機能を探索します。

このガイドは、JMeter を効果的に使用するための基本を提供します。練習を重ねることで、特定のパフォーマンステストのニーズに合わせてテストをカスタマイズすることができます！