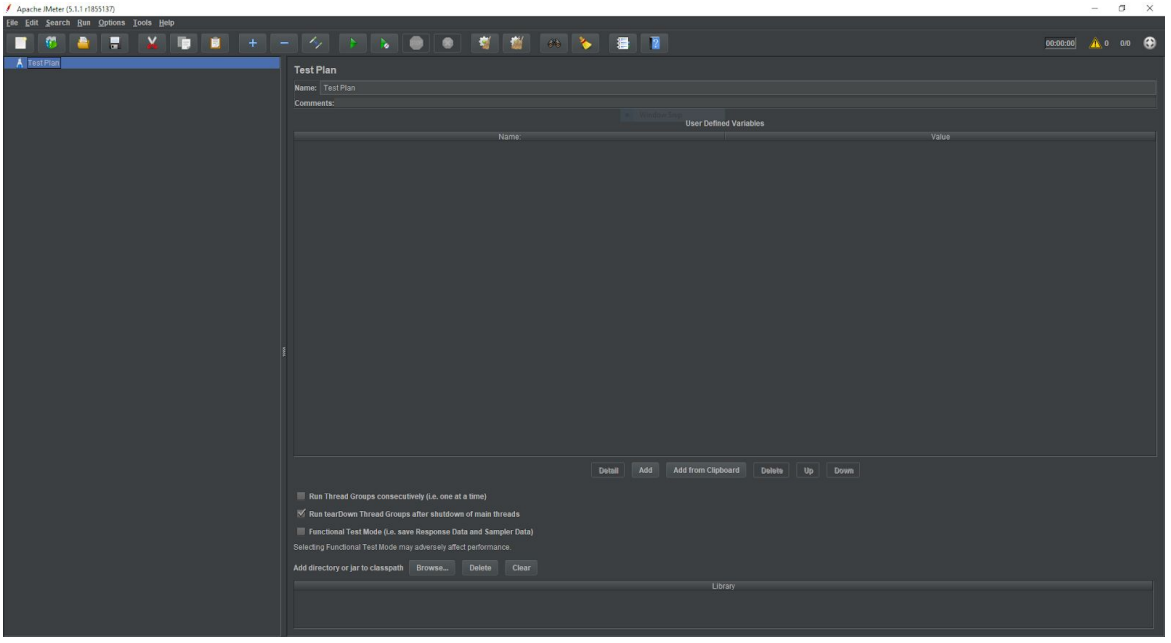
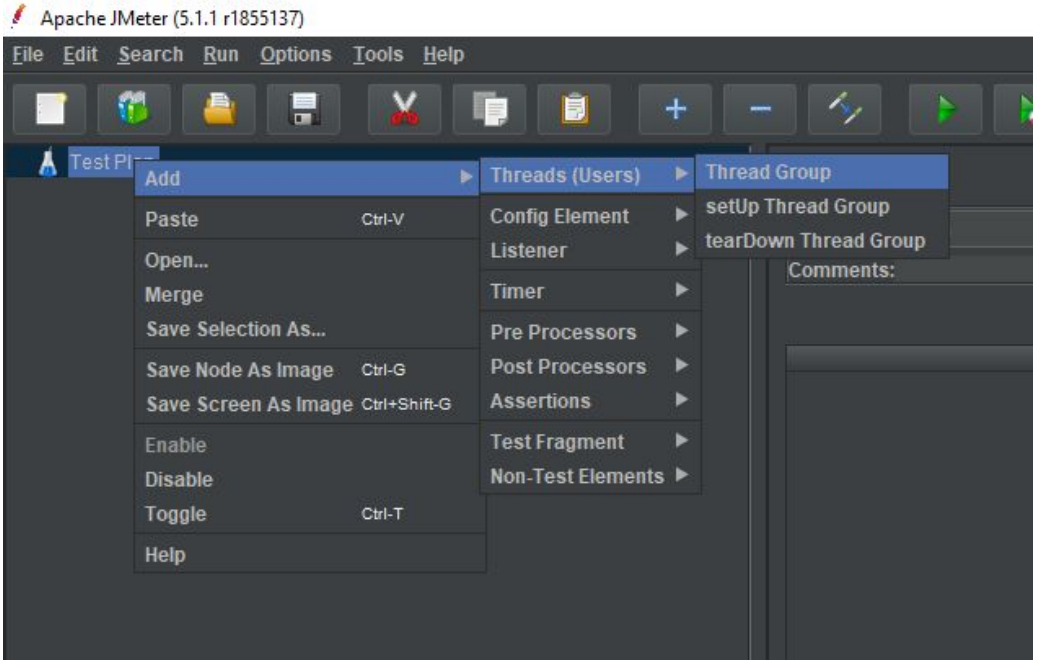
# Exercises

**1st** To start off we will create a simple test that will simulate a single user accessing the frontpage of our website.

1. Start JMeter and this should be what you see:




2. Our first step will be creating a Thread Group, which is responsible with creating multiple threads to execute our scripts. To create one, right-click your “Test Plan”, *Add -> Threads(User) -> Thread Group*



3. Now you can begin to customize your threads! There are 3 main properties that we will be using for these exercises:

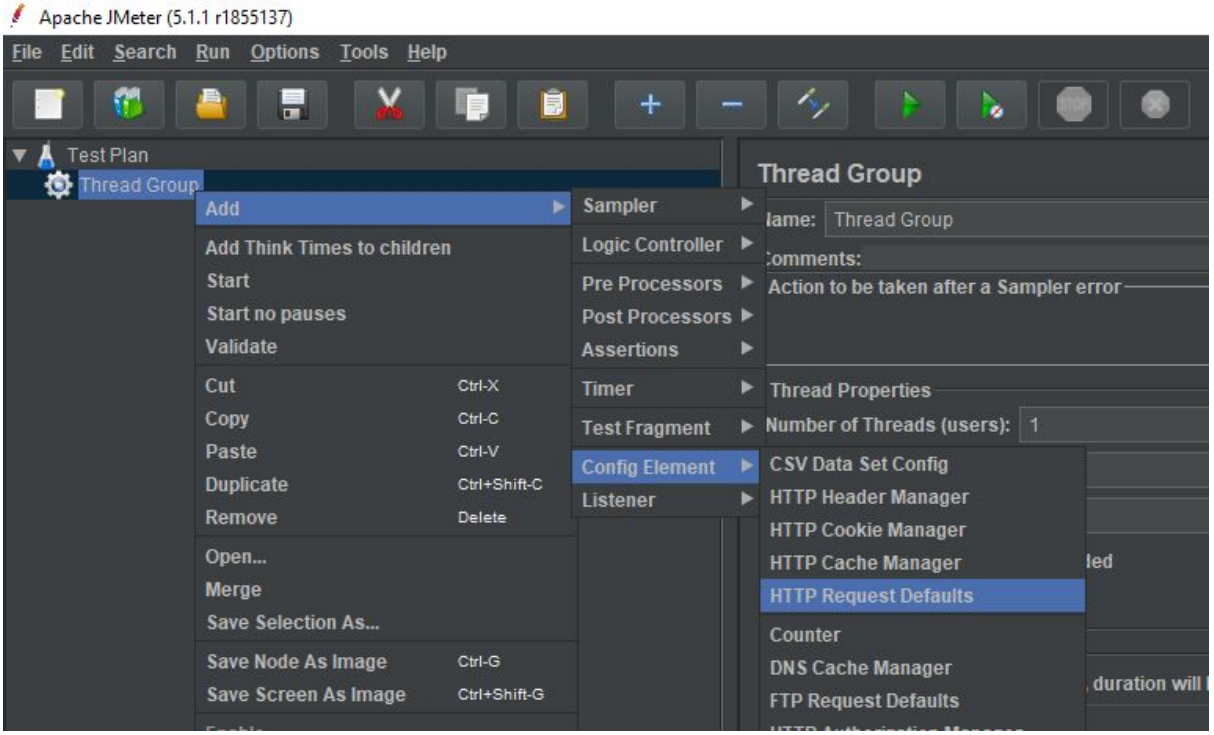
**Number of Threads (users)** - the nº of active threads that will be executing our script.

**Ramp-Up Period (in seconds)** - The amount of time it will take for all the threads to be started. Higher values allow you to check the performance while the throughput increases slowly. Lower values allow you to check the performance through quick spikes.

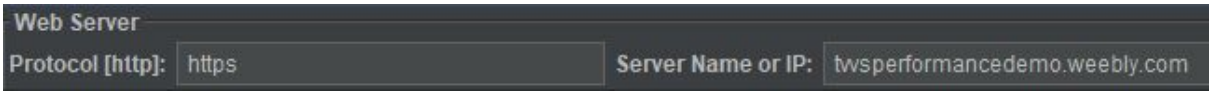
**Loop count** - the nº of times the threads will execute the scripts. Use 'Forever' to check the performance while under constant pressure.

4. Since we are simulating a single user accessing the homepage once, let's set the properties like this:

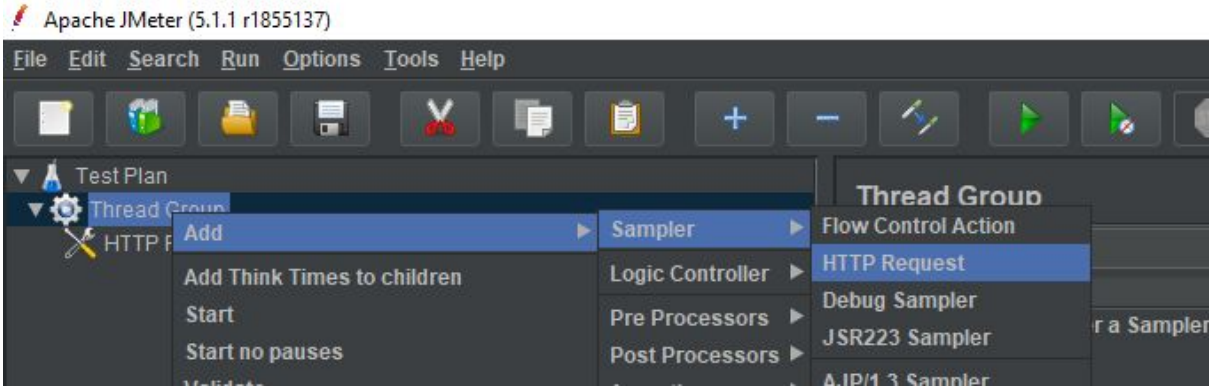

5. Now we will start defining how we will access our website and homepage. For this we need to create an **HTTP Request Defaults**. Just right-click your **Thread Group** and *Add -> Config Element -> HTTP Request Defaults*.



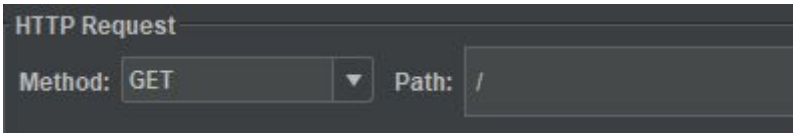
6. We will use this **HTTP Request Defaults** to set our website's endpoint as the default. Our website is [https://tvvsperformancedemo.weebly.com/](https://tvvsperformancedemo.weebly.com/), so set your Protocol to **https** and Server Name to **tvvsperformancedemo.weebly.com**:



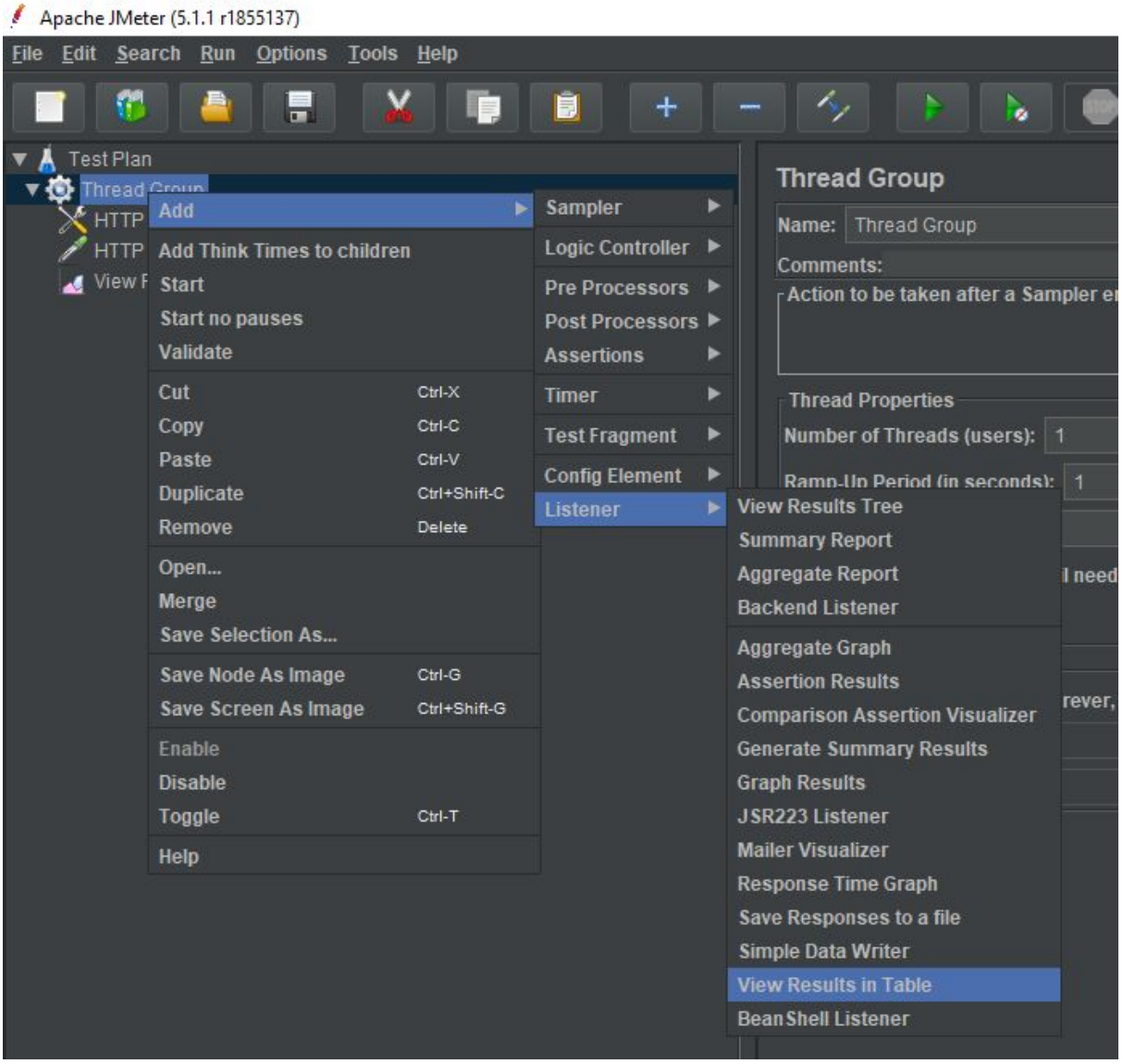

7. Now that we know where our website, we need to say where our homepage is. For this we will create an **HTTP Request**, just right-click *Thread Group -> Add -> Sampler -> HTTP Request*:




8. And now we can define our HTTP Request to our liking. Let's start with a simple GET request to our homepage at path **/** .

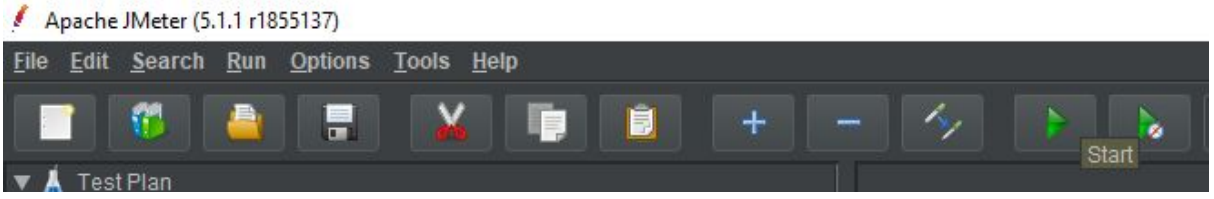



9. We are almost ready now! All we need to do is listen to our requests to check their status. We do this with a Listener, so let's create one like this: *right-click Thread Group -> Add -> Listener -> View Results in Table*

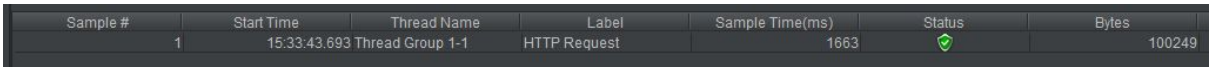



JMeter has many types of listeners for your objectives, we are using the simple **View Results in Table** since it is a simple and straightforward way of seeing our requests and their status, even in large numbers.

10. We are now ready to see our test in action! Just select your **View Results in Table** so you can see the results coming in and click the **Run** icon above:



11. Now you should see your first result! Hopefully everything went well and you can see the green status icon, if not please ask for help to check what went wrong.



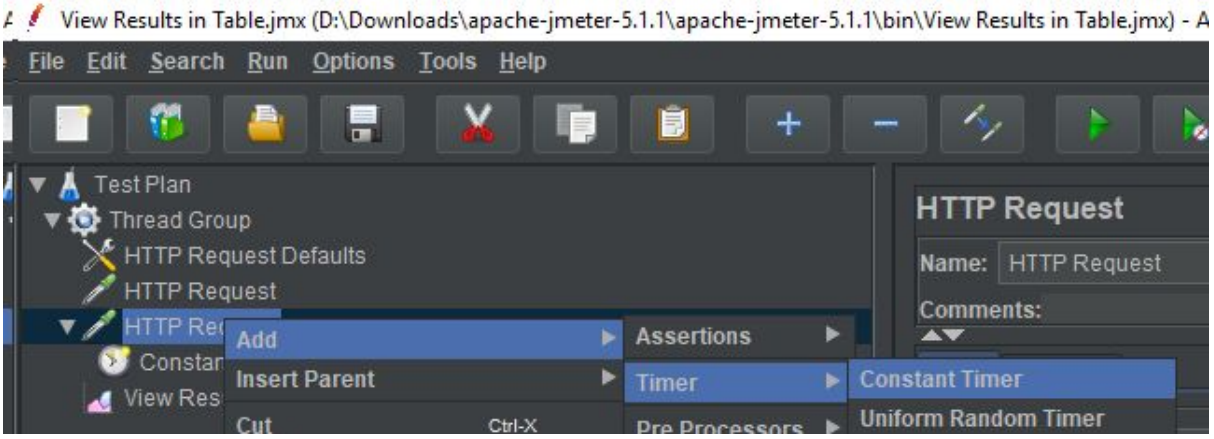

**2nd** Now that we have the basics in check, we can start truly testing our website's limits and performance. For our second exercise we will test our website under a constant use by about 10 users. We will also simulate a user changing to another page after a certain delay.

1. Let's begin by changing our **Thread Pool**'s properties to simulate 10 users, with a Ramp-Up of 10 seconds and loop **Forever**.

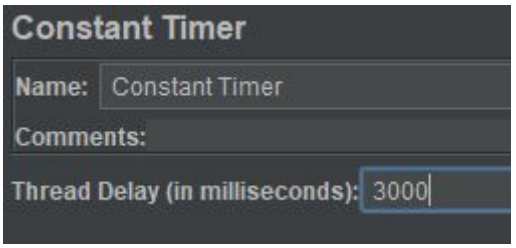
The Ramp-Up of 10 seconds gives us some time to see the requests start stacking up slowly.

2. Next let's create a new **HTTP Request** to access another page. Add another one to your Thread Pool and use the path: **/services**

3. Now we can use another of JMeter's great tools to simulate user input, which are Timers. We will use a **Constant Timer** to insert a delay between accessing the 2 pages. Add a Constant Timer by: right-clicking your new *HTTP Request-> Add -> Timer -> Constant Timer*




Set a Thread Delay of 3000ms to delay that request for 3 seconds.



4. Now we are ready to see our test in action! Go back to the results view and click the run button again to watch as the threads start and the results come in. Take a close look at the nº of threads starting up due to our Ramp-Up property as well:


*Top right corner, 4 out of 10 threads running*

5. If everything goes well we should see plenty of results coming in and no issues whatsoever. This is as expected, since having 10 users frequently changing between pages of your website is very little traffic, even for a simple free website like ours.

**3rd** Now that we showed you some basic features from JMeter, we want you to try testing the limits of the website. Can you find a way to either crash or be locked out from the website? Give it a try and ask for help if you aren't managing it!

**Good job!** You are now free to explore the other great functionalities that JMeter has to offer! Here are a few examples for you to play around with:

- HTTP Cookie Manager
- HTTP Cache Manager
- Uniform Random Timer
- Summary Report
- Aggregate Graph
- Graph Results

Just search the web for the documentation on these features or try to figure it out yourself while testing with them.