

## Практическая работа №5

**После установки JMeter, создаем тесты опираясь на методичку:**

Теперь создадим тесты для этого бэкенда в JMeter.

### 1. Создание тестового плана

- Откройте JMeter.
- Создайте новый тестовый план (Test Plan).

### 2. Добавление Thread Group

- Правый клик на Test Plan → Add → Threads (Users) → Thread Group.
- Установите "Number of Threads (users)" на 5, "Ramp-Up Period (seconds)" на 10 и "Loop Count" на 1.

### 3. Добавление HTTP Request Sampler для POST запроса

- Правый клик на Thread Group → Add → Sampler → HTTP Request.
- В настройках HTTP Request:
  - Name: Create Item
  - Server Name or IP: 127.0.0.1
  - Port Number: 8000
  - Method: POST
  - Path: /items/
  - Body Data: Вставьте пример данных:

json

Копировать код

```
{  
  "name": "Item 1",  
  "description": "A description of Item 1",  
  "price": 100.0  
}
```

### 4. Добавление HTTP Request Sampler для GET запроса

- Правый клик на Thread Group → Add → Sampler → HTTP Request.
  - В настройках HTTP Request:
    - Name: Get Items

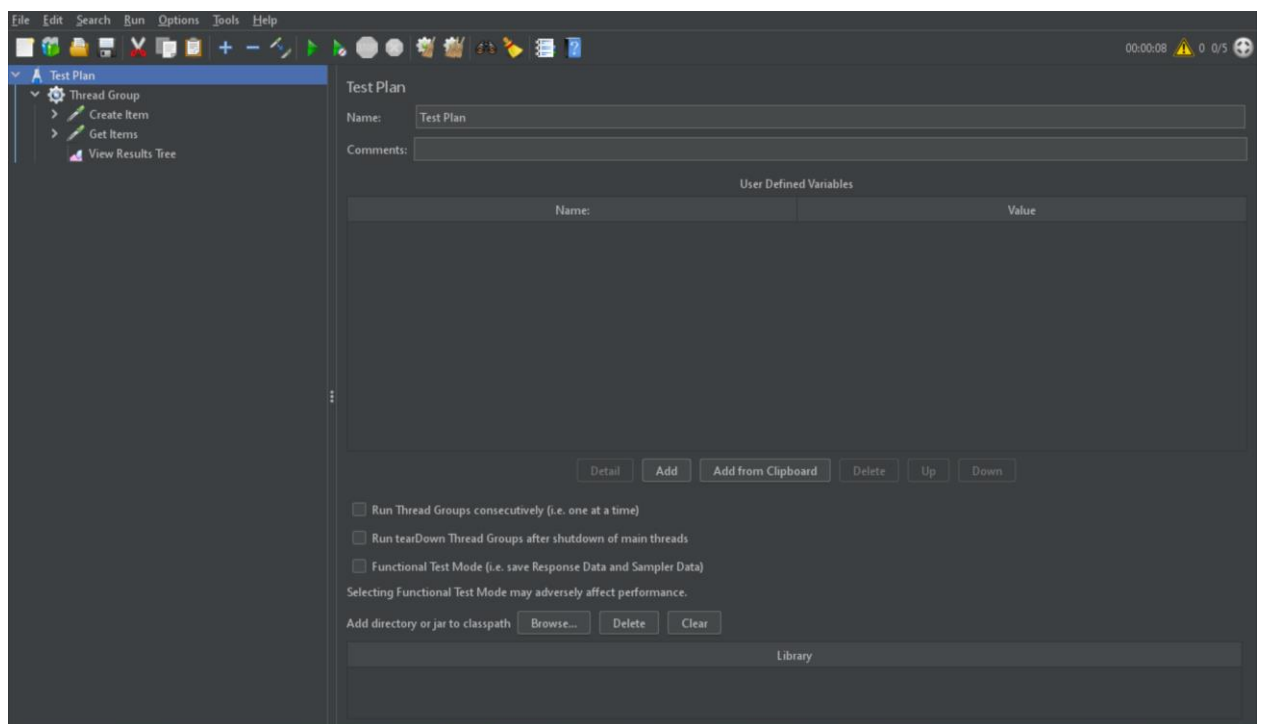
- Server Name or IP: 127.0.0.1
- Port Number: 8000
- Method: GET
- Path: /items/

## 5. Добавление Listener

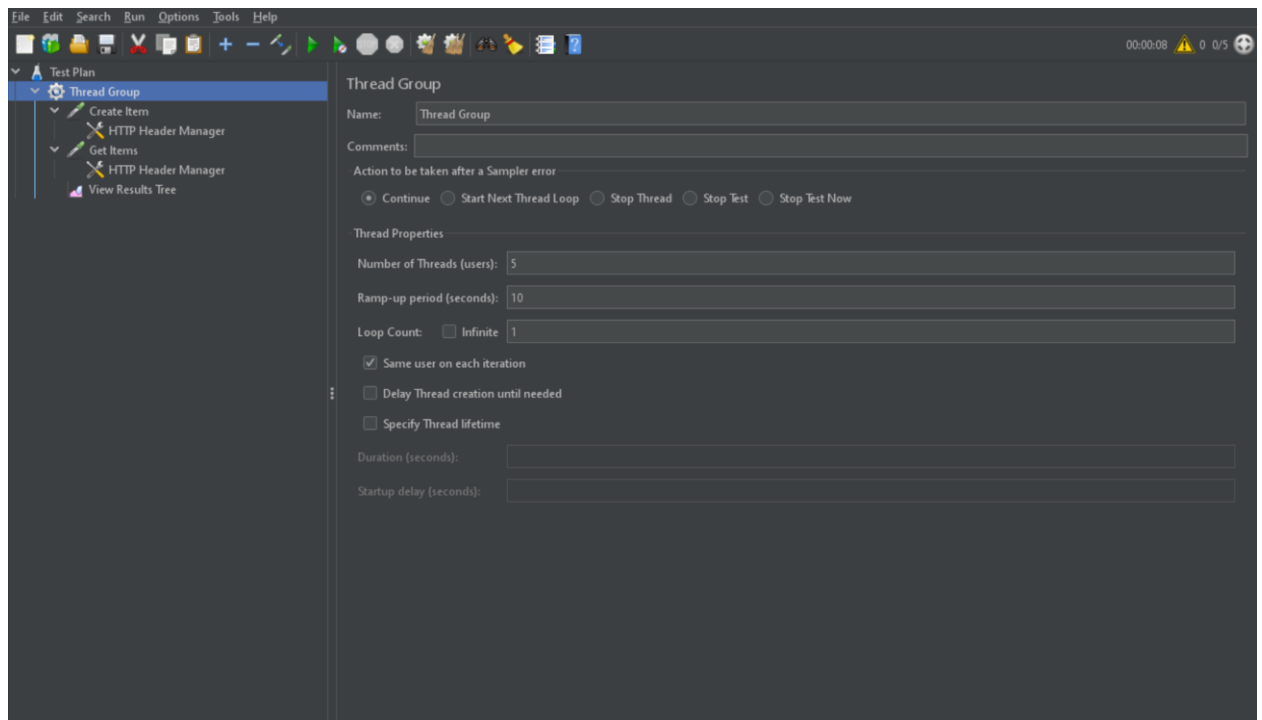
- Правый клик на Thread Group → Add → Listener → View Results Tree.

## 6. Запуск тестов

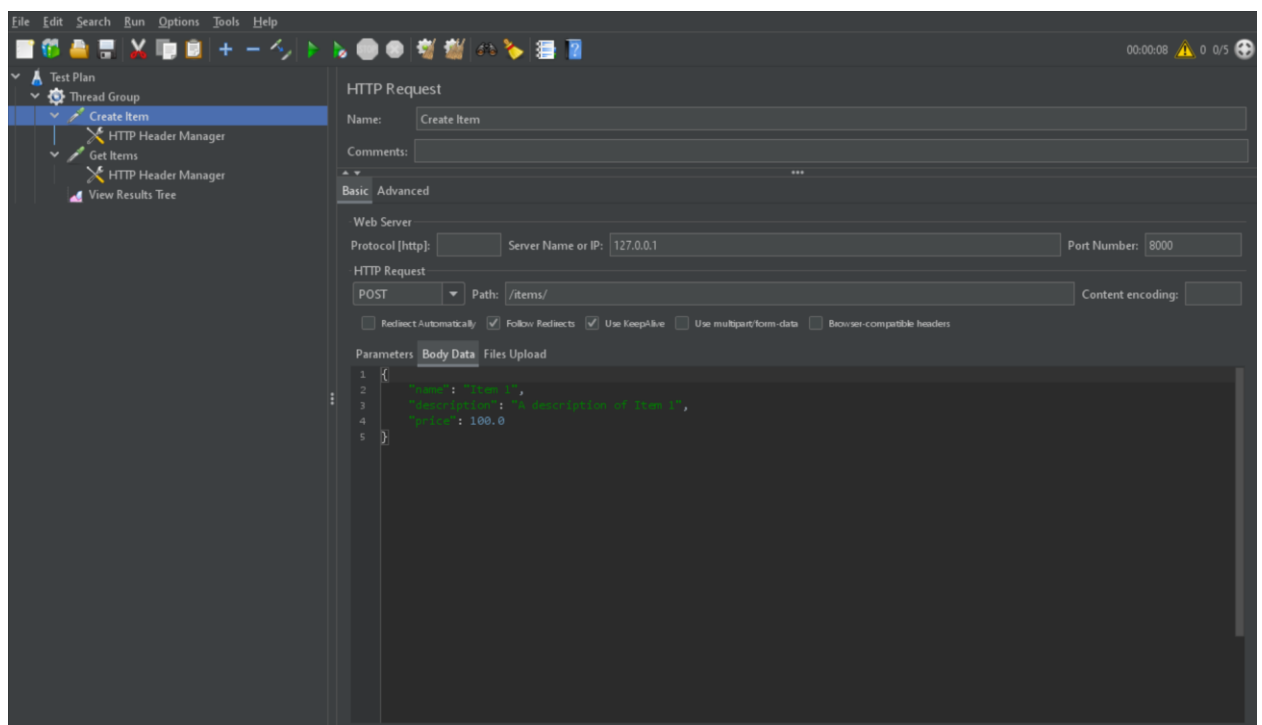
- Нажмите кнопку "Start" на панели инструментов JMeter.
- После завершения теста, просмотрите результаты в "View Results Tree".



## 1. Создаем тестовый план



## 2.Добавляем *Thread Group*



## 3.Добавляем *HTTP Request Sampler* для *POST* запроса



```
5.py x
1 from fastapi import FastAPI, HTTPException
2 from pydantic import BaseModel
3 from typing import List, Optional
4 import sqlite3
5
6 app = FastAPI()
7
8 # Database setup
9 def init_db():
10     conn = sqlite3.connect('auction.db')
11     cursor = conn.cursor()
12     cursor.execute("""
13     CREATE TABLE IF NOT EXISTS items (
14         id INTEGER PRIMARY KEY AUTOINCREMENT,
15         name TEXT NOT NULL,
16         description TEXT,
17     )
18
19 Run Main x
20
21 INFO: Started server process [11256]
22 INFO: Waiting for application startup.
23 INFO: Application startup complete.
24 INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
25 INFO: 127.0.0.1:51873 - "POST /items/ HTTP/1.1" 200 OK
26 INFO: 127.0.0.1:51873 - "GET /items/ HTTP/1.1" 200 OK
27 INFO: 127.0.0.1:51878 - "POST /items/ HTTP/1.1" 200 OK
28 INFO: 127.0.0.1:51878 - "GET /items/ HTTP/1.1" 200 OK
29 INFO: 127.0.0.1:51880 - "POST /items/ HTTP/1.1" 200 OK
30 TNFO: 127.0.0.1:51880 - "GET /items/ HTTP/1.1" 200 OK
```

## 6. Запускаем тест

```
127.0.0.1:8000/items/ x +
127.0.0.1:8000/items/
Качественная печать
[{"id":1,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":2,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":3,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":4,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":5,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":6,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":7,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":8,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":9,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":10,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":11,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":12,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":13,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":14,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":15,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":16,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":17,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":18,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":19,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":20,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":21,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":22,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":23,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":24,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":25,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":26,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":27,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":28,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":29,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":30,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":31,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":32,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":33,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":34,"name":"Item 1","description":"A description of Item 1","price":100}, {"id":35,"name":"Item 1","description":"A description of Item 1","price":100}]
```

## 7. Получаем данные