

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

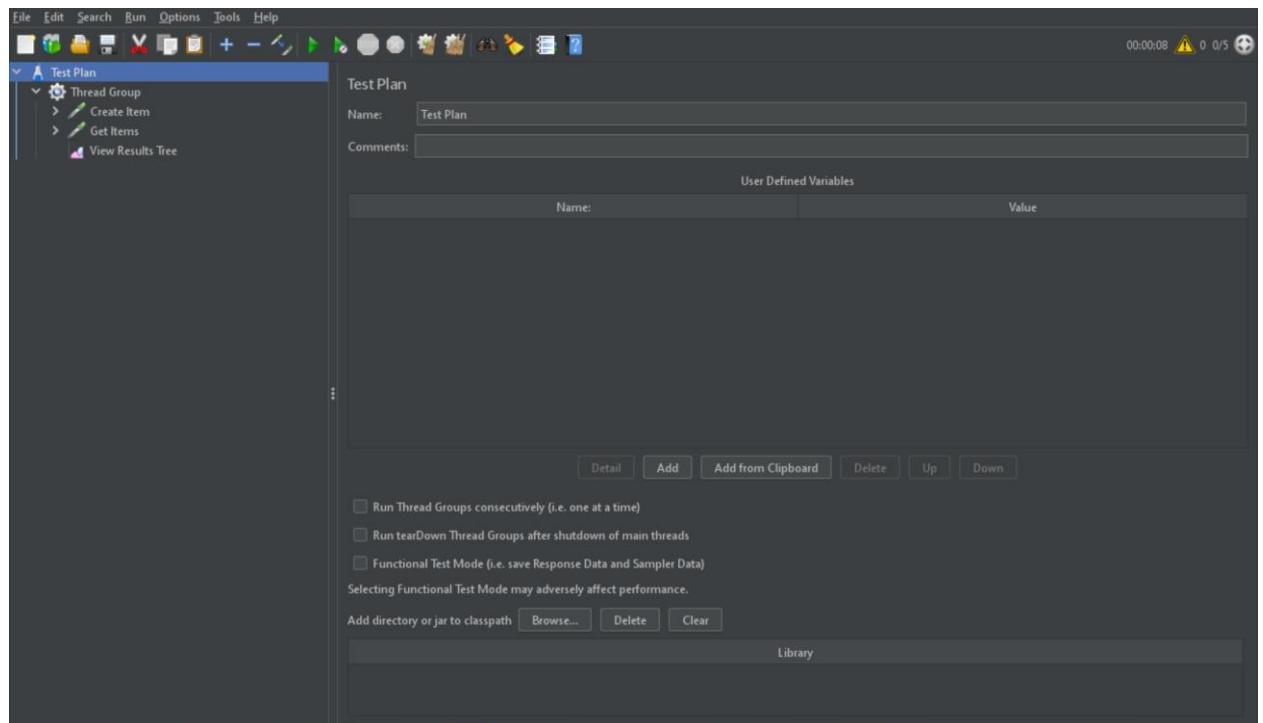
- Server Name or IP: 127.0.0.1
- Port Number: 8000
- Method: GET
- Path: /items/

5. Добавление Listener

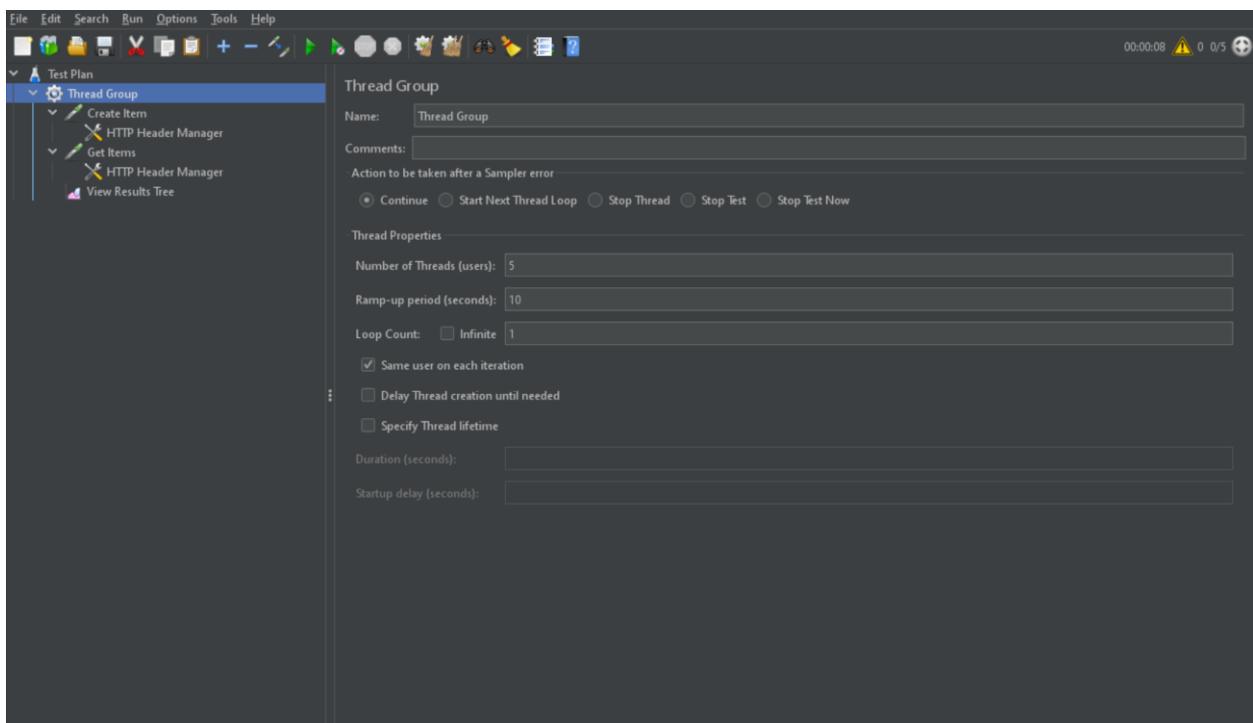
- Правый клик на Thread Group → Add → Listener → View Results Tree.

6. Запуск тестов

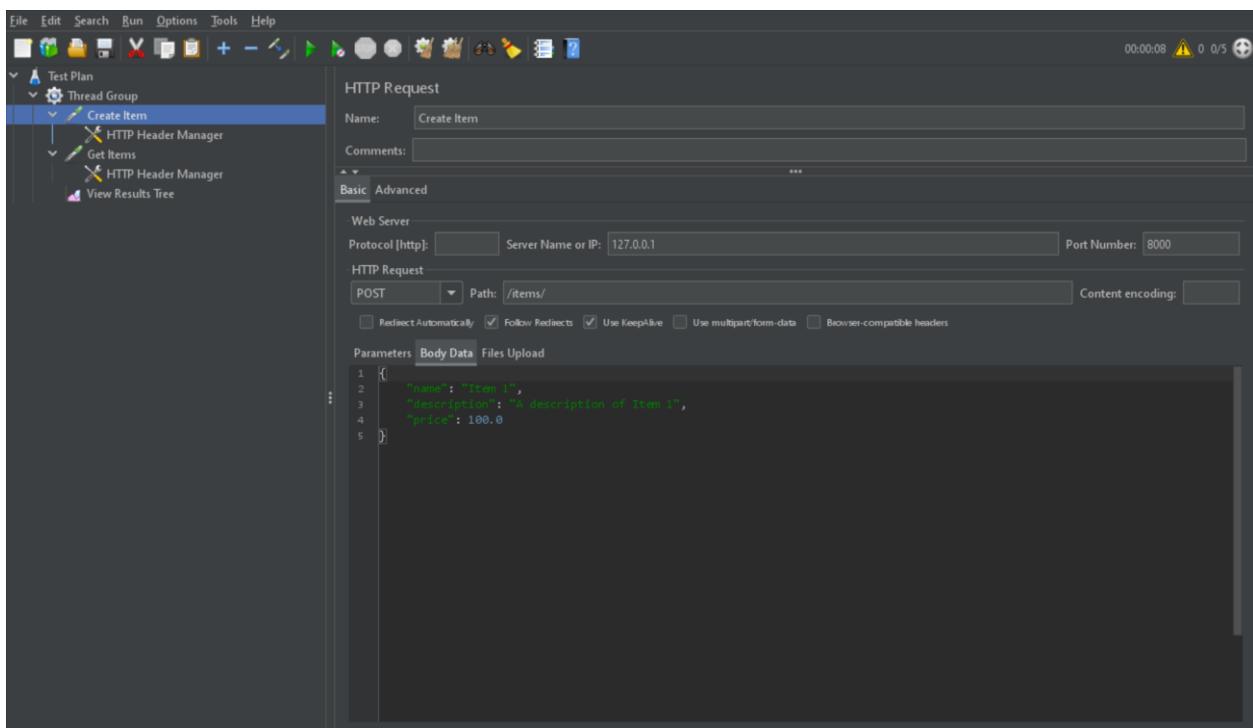
- Нажмите кнопку "Start" на панели инструментов JMeter.
- После завершения теста, просмотрите результаты в "View Results Tree".



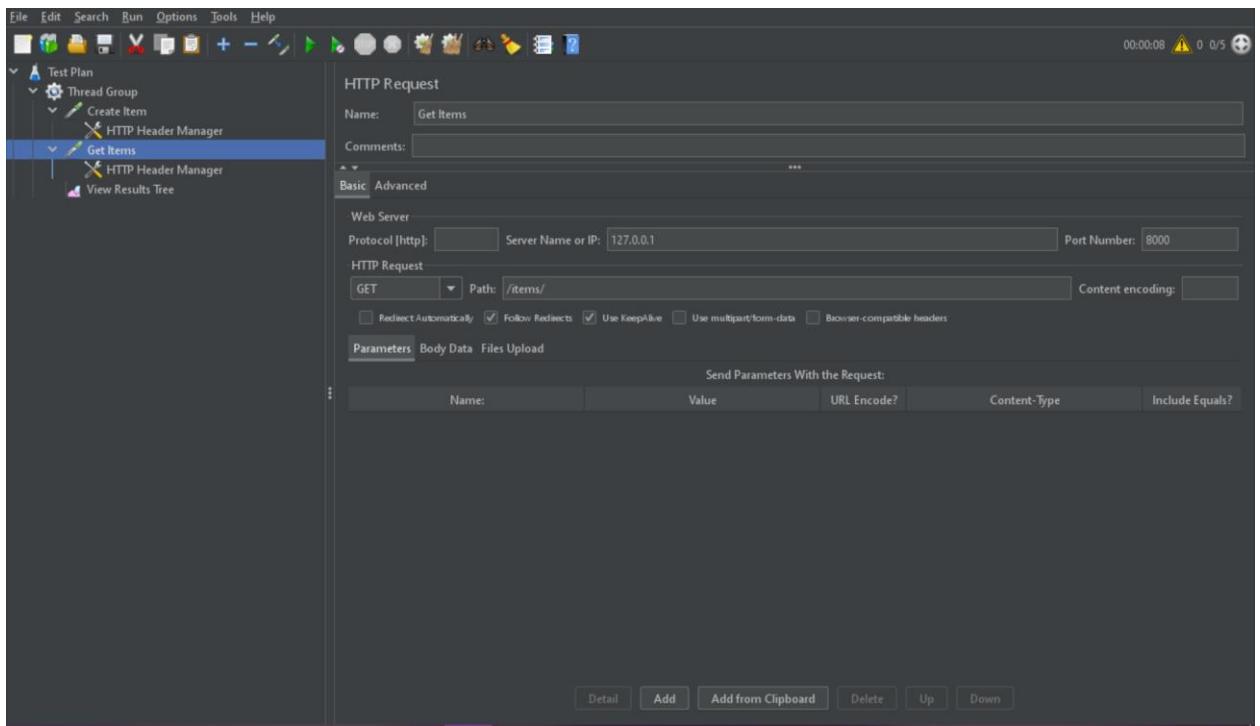
1. Создаем тестовый план



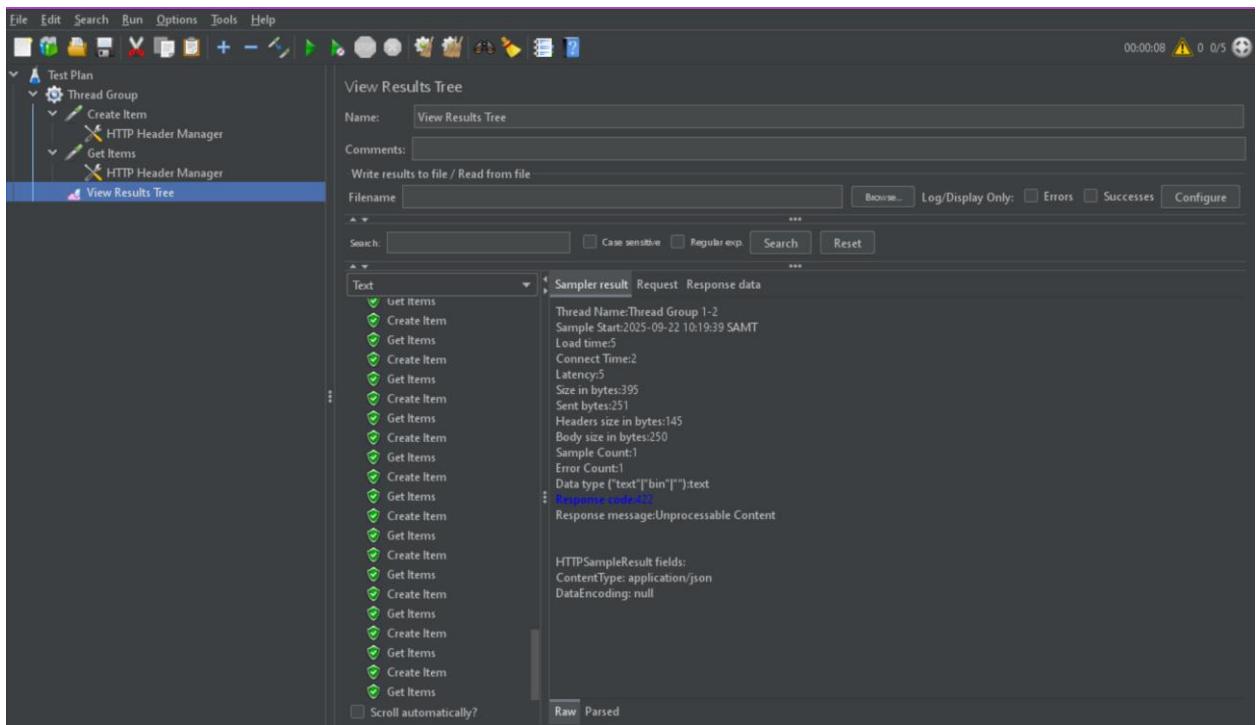
2.Добавляем Thread Group



3.Добавляем HTTP Request Sampler для POST запроса



4.Добавляем HTTP Request Sampler для GET запроса



5.Добавляем Listener

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from typing import List, Optional
import sqlite3

app = FastAPI()

# Database setup
def init_db():
    conn = sqlite3.connect('auction.db')
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS items (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT NOT NULL,
            description TEXT,
    """)
    conn.commit()

    return conn

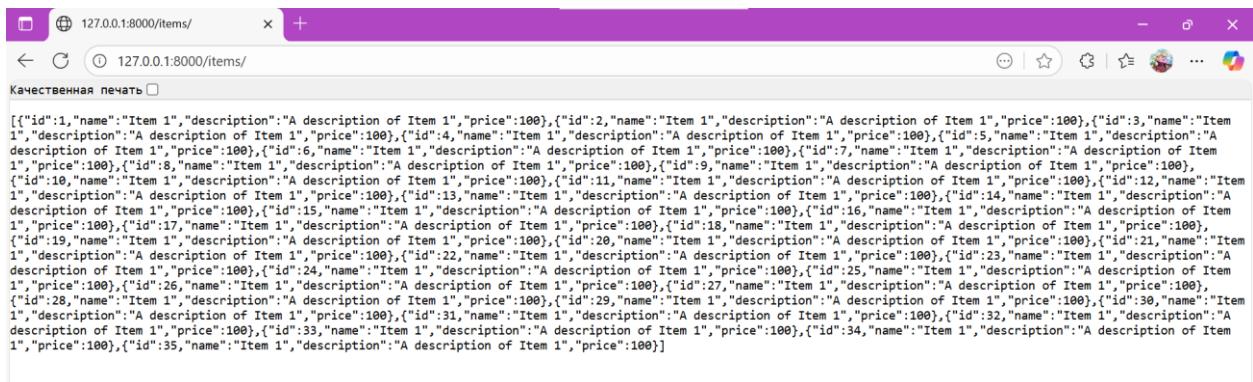
app.add_event_handler("startup", init_db)

@app.get("/items/")
async def read_items():
    conn = init_db()
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM items")
    items = cursor.fetchall()
    cursor.close()
    conn.close()
    return items
```

Run Main

```
INFO:     Started server process [11256]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     127.0.0.1:51873 - "POST /items/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:51873 - "GET /items/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:51878 - "POST /items/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:51878 - "GET /items/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:51880 - "POST /items/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:51880 - "GET /items/ HTTP/1.1" 200 OK
```

6. Запускаем тест



7. Получаем данные