**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе № 7**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | **Нагрузочное тестирование** | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Тестирование информационных систем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Филатов Егор Александрович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-311 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г**

**Лабораторная работа №7.** **«Нагрузочное тестирование»**

Пишем модель данных для тестового веб-приложения

dataclass позволяет записать класс в упрощенном виде.



Реализовываем паттерн репозиторий для управления моделью

Пишем уже сами эндпоинты нашего приложения



Запускаем



Для начала надо написать конфигурацию для Jmeter в файле users\_api\_test.jmx

<?xml version="1.0" encoding="UTF-8"?>

<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.5">

<hashTree>

<TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="User API Test Plan" enabled="true">

<stringProp name="TestPlan.comments"></stringProp>

<boolProp name="TestPlan.functional\_mode">false</boolProp>

<boolProp name="TestPlan.tearDown\_on\_shutdown">true</boolProp>

<boolProp name="TestPlan.serialize\_threadgroups">false</boolProp>

<elementProp name="TestPlan.user\_defined\_variables" elementType="Arguments" guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">

<collectionProp name="Arguments.arguments"/>

</elementProp>

<stringProp name="TestPlan.user\_define\_classpath"></stringProp>

</TestPlan>

<hashTree>

<ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Users API Thread Group" enabled="true">

<stringProp name="ThreadGroup.on\_sample\_error">continue</stringProp>

<elementProp name="ThreadGroup.main\_controller" elementType="LoopController" guiclass="LoopControlPanel" testclass="LoopController" testname="Loop Controller" enabled="true">

<boolProp name="LoopController.continue\_forever">false</boolProp>

<stringProp name="LoopController.loops">10</stringProp>

</elementProp>

<stringProp name="ThreadGroup.num\_threads">100</stringProp>

<stringProp name="ThreadGroup.ramp\_time">10</stringProp>

<boolProp name="ThreadGroup.scheduler">false</boolProp>

<stringProp name="ThreadGroup.duration"></stringProp>

<stringProp name="ThreadGroup.delay"></stringProp>

<boolProp name="ThreadGroup.same\_user\_on\_next\_iteration">true</boolProp>

</ThreadGroup>

<hashTree>

<ConfigTestElement guiclass="HttpDefaultsGui" testclass="ConfigTestElement" testname="HTTP Request Defaults" enabled="true">

<elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">

<collectionProp name="Arguments.arguments"/>

</elementProp>

<stringProp name="HTTPSampler.domain">127.0.0.1</stringProp>

<stringProp name="HTTPSampler.port">8000</stringProp>

<stringProp name="HTTPSampler.protocol">http</stringProp>

<stringProp name="HTTPSampler.contentEncoding"></stringProp>

<stringProp name="HTTPSampler.path"></stringProp>

<stringProp name="HTTPSampler.concurrentPool">6</stringProp>

<stringProp name="HTTPSampler.connect\_timeout"></stringProp>

<stringProp name="HTTPSampler.response\_timeout"></stringProp>

</ConfigTestElement>

<hashTree/>

<HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="GET All Users" enabled="true">

<elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">

<collectionProp name="Arguments.arguments"/>

</elementProp>

<stringProp name="HTTPSampler.domain"></stringProp>

<stringProp name="HTTPSampler.port"></stringProp>

<stringProp name="HTTPSampler.protocol"></stringProp>

<stringProp name="HTTPSampler.contentEncoding"></stringProp>

<stringProp name="HTTPSampler.path">/users</stringProp>

<stringProp name="HTTPSampler.method">GET</stringProp>

<boolProp name="HTTPSampler.follow\_redirects">true</boolProp>

<boolProp name="HTTPSampler.auto\_redirects">false</boolProp>

<boolProp name="HTTPSampler.use\_keepalive">true</boolProp>

<boolProp name="HTTPSampler.DO\_MULTIPART\_POST">false</boolProp>

<stringProp name="HTTPSampler.embedded\_url\_re"></stringProp>

<stringProp name="HTTPSampler.connect\_timeout"></stringProp>

<stringProp name="HTTPSampler.response\_timeout"></stringProp>

</HTTPSamplerProxy>

<hashTree/>

<HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="GET Single User" enabled="true">

<elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">

<collectionProp name="Arguments.arguments"/>

</elementProp>

<stringProp name="HTTPSampler.domain"></stringProp>

<stringProp name="HTTPSampler.port"></stringProp>

<stringProp name="HTTPSampler.protocol"></stringProp>

<stringProp name="HTTPSampler.contentEncoding"></stringProp>

<stringProp name="HTTPSampler.path">/users/1</stringProp>

<stringProp name="HTTPSampler.method">GET</stringProp>

<boolProp name="HTTPSampler.follow\_redirects">true</boolProp>

<boolProp name="HTTPSampler.auto\_redirects">false</boolProp>

<boolProp name="HTTPSampler.use\_keepalive">true</boolProp>

<boolProp name="HTTPSampler.DO\_MULTIPART\_POST">false</boolProp>

<stringProp name="HTTPSampler.embedded\_url\_re"></stringProp>

<stringProp name="HTTPSampler.connect\_timeout"></stringProp>

<stringProp name="HTTPSampler.response\_timeout"></stringProp>

</HTTPSamplerProxy>

<hashTree/>

<HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="POST New User" enabled="true">

<elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">

<collectionProp name="Arguments.arguments">

<elementProp name="id" elementType="HTTPArgument">

<boolProp name="HTTPArgument.always\_encode">false</boolProp>

<stringProp name="Argument.value">${\_\_Random(100,999)}</stringProp>

<stringProp name="Argument.metadata">=</stringProp>

<boolProp name="HTTPArgument.use\_equals">true</boolProp>

<stringProp name="Argument.name">id</stringProp>

</elementProp>

<elementProp name="name" elementType="HTTPArgument">

<boolProp name="HTTPArgument.always\_encode">false</boolProp>

<stringProp name="Argument.value">User\_${\_\_Random(1,1000)}</stringProp>

<stringProp name="Argument.metadata">=</stringProp>

<boolProp name="HTTPArgument.use\_equals">true</boolProp>

<stringProp name="Argument.name">name</stringProp>

</elementProp>

<elementProp name="age" elementType="HTTPArgument">

<boolProp name="HTTPArgument.always\_encode">false</boolProp>

<stringProp name="Argument.value">${\_\_Random(18,60)}</stringProp>

<stringProp name="Argument.metadata">=</stringProp>

<boolProp name="HTTPArgument.use\_equals">true</boolProp>

<stringProp name="Argument.name">age</stringProp>

</elementProp>

</collectionProp>

</elementProp>

<stringProp name="HTTPSampler.domain"></stringProp>

<stringProp name="HTTPSampler.port"></stringProp>

<stringProp name="HTTPSampler.protocol"></stringProp>

<stringProp name="HTTPSampler.contentEncoding"></stringProp>

<stringProp name="HTTPSampler.path">/users</stringProp>

<stringProp name="HTTPSampler.method">POST</stringProp>

<boolProp name="HTTPSampler.follow\_redirects">true</boolProp>

<boolProp name="HTTPSampler.auto\_redirects">false</boolProp>

<boolProp name="HTTPSampler.use\_keepalive">true</boolProp>

<boolProp name="HTTPSampler.DO\_MULTIPART\_POST">false</boolProp>

<stringProp name="HTTPSampler.embedded\_url\_re"></stringProp>

<stringProp name="HTTPSampler.connect\_timeout"></stringProp>

<stringProp name="HTTPSampler.response\_timeout"></stringProp>

</HTTPSamplerProxy>

<hashTree/>

<ResultCollector guiclass="ViewResultsFullVisualizer" testclass="ResultCollector" testname="View Results Tree" enabled="true">

<boolProp name="ResultCollector.error\_logging">false</boolProp>

<objProp>

<name>saveConfig</name>

<value class="SampleSaveConfiguration">

<time>true</time>

<latency>true</latency>

<timestamp>true</timestamp>

<success>true</success>

<label>true</label>

<code>true</code>

<message>true</message>

<threadName>true</threadName>

<dataType>true</dataType>

<encoding>false</encoding>

<assertions>true</assertions>

<subresults>true</subresults>

<responseData>false</responseData>

<samplerData>false</samplerData>

<xml>false</xml>

<fieldNames>true</fieldNames>

<responseHeaders>false</responseHeaders>

<requestHeaders>false</requestHeaders>

<responseDataOnError>false</responseDataOnError>

<saveAssertionResultsFailureMessage>true</saveAssertionResultsFailureMessage>

<assertionsResultsToSave>0</assertionsResultsToSave>

<bytes>true</bytes>

<sentBytes>true</sentBytes>

<url>true</url>

<threadCounts>true</threadCounts>

<idleTime>true</idleTime>

<connectTime>true</connectTime>

</value>

</objProp>

<stringProp name="filename"></stringProp>

</ResultCollector>

<hashTree/>

<ResultCollector guiclass="SummaryReport" testclass="ResultCollector" testname="Summary Report" enabled="true">

<boolProp name="ResultCollector.error\_logging">false</boolProp>

<objProp>

<name>saveConfig</name>

<value class="SampleSaveConfiguration">

<time>true</time>

<latency>true</latency>

<timestamp>true</timestamp>

<success>true</success>

<label>true</label>

<code>true</code>

<message>true</message>

<threadName>true</threadName>

<dataType>true</dataType>

<encoding>false</encoding>

<assertions>true</assertions>

<subresults>true</subresults>

<responseData>false</responseData>

<samplerData>false</samplerData>

<xml>false</xml>

<fieldNames>true</fieldNames>

<responseHeaders>false</responseHeaders>

<requestHeaders>false</requestHeaders>

<responseDataOnError>false</responseDataOnError>

<saveAssertionResultsFailureMessage>true</saveAssertionResultsFailureMessage>

<assertionsResultsToSave>0</assertionsResultsToSave>

<bytes>true</bytes>

<sentBytes>true</sentBytes>

<url>true</url>

<threadCounts>true</threadCounts>

<idleTime>true</idleTime>

<connectTime>true</connectTime>

</value>

</objProp>

<stringProp name="filename"></stringProp>

</ResultCollector>

<hashTree/>

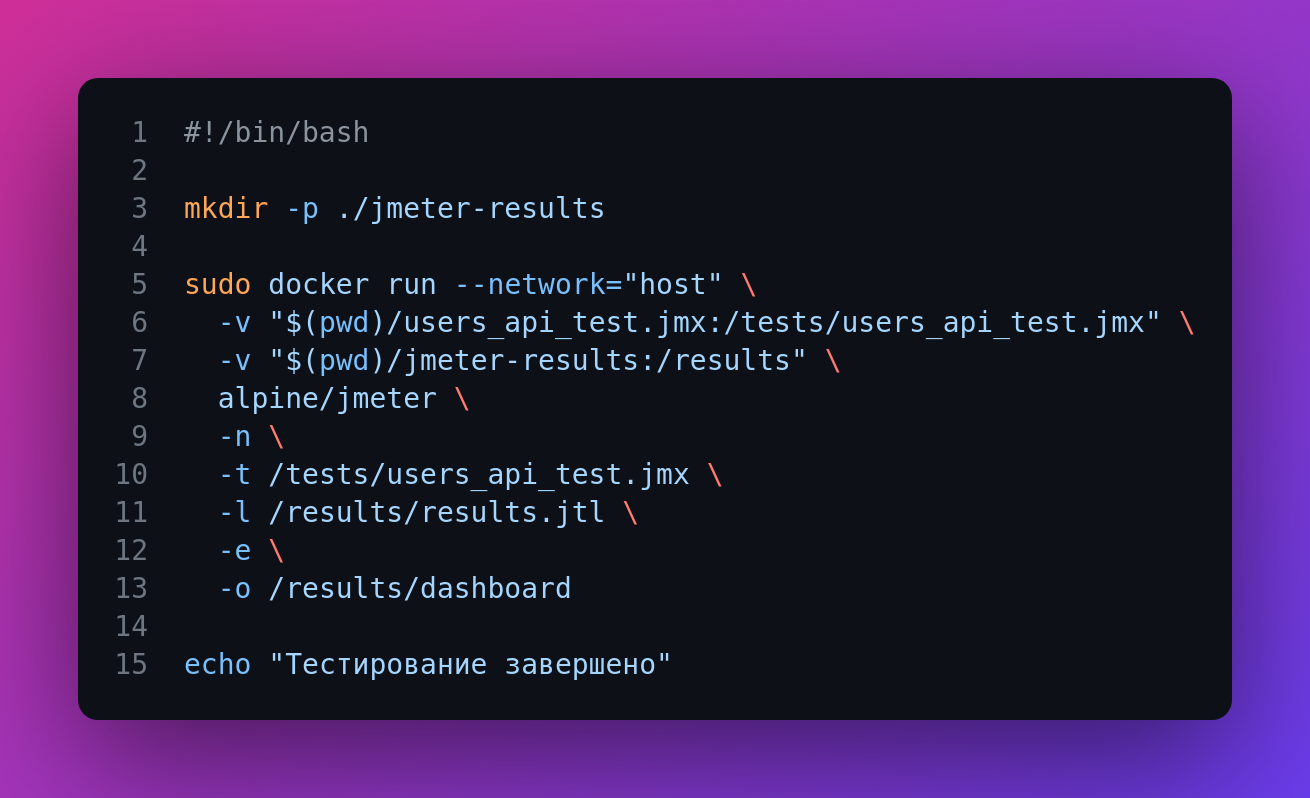
</hashTree>

</hashTree>

</hashTree>

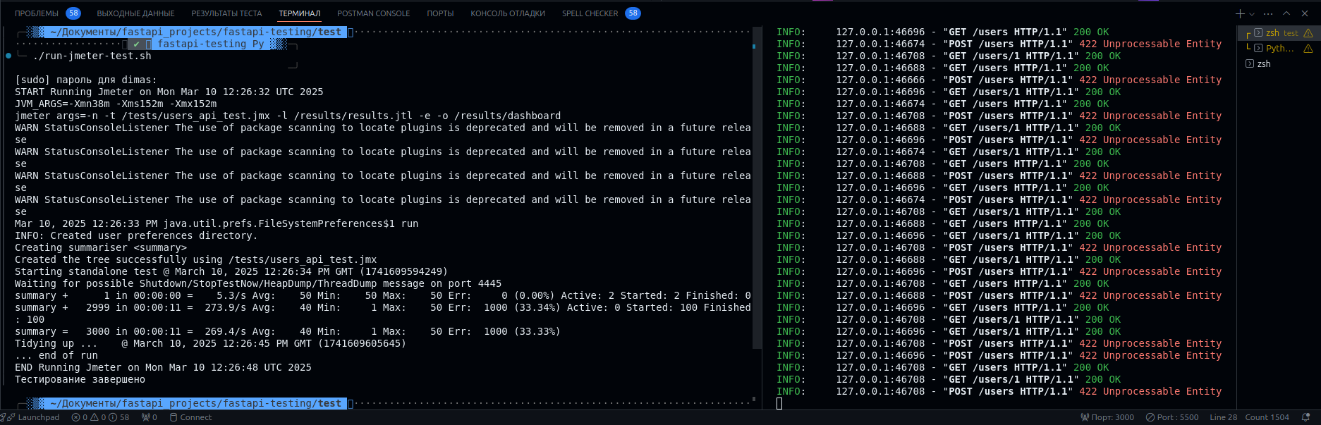
</jmeterTestPlan>

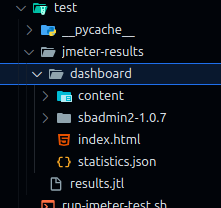
Далее требуется написать bash-скрипт, который запустит нам все это дело



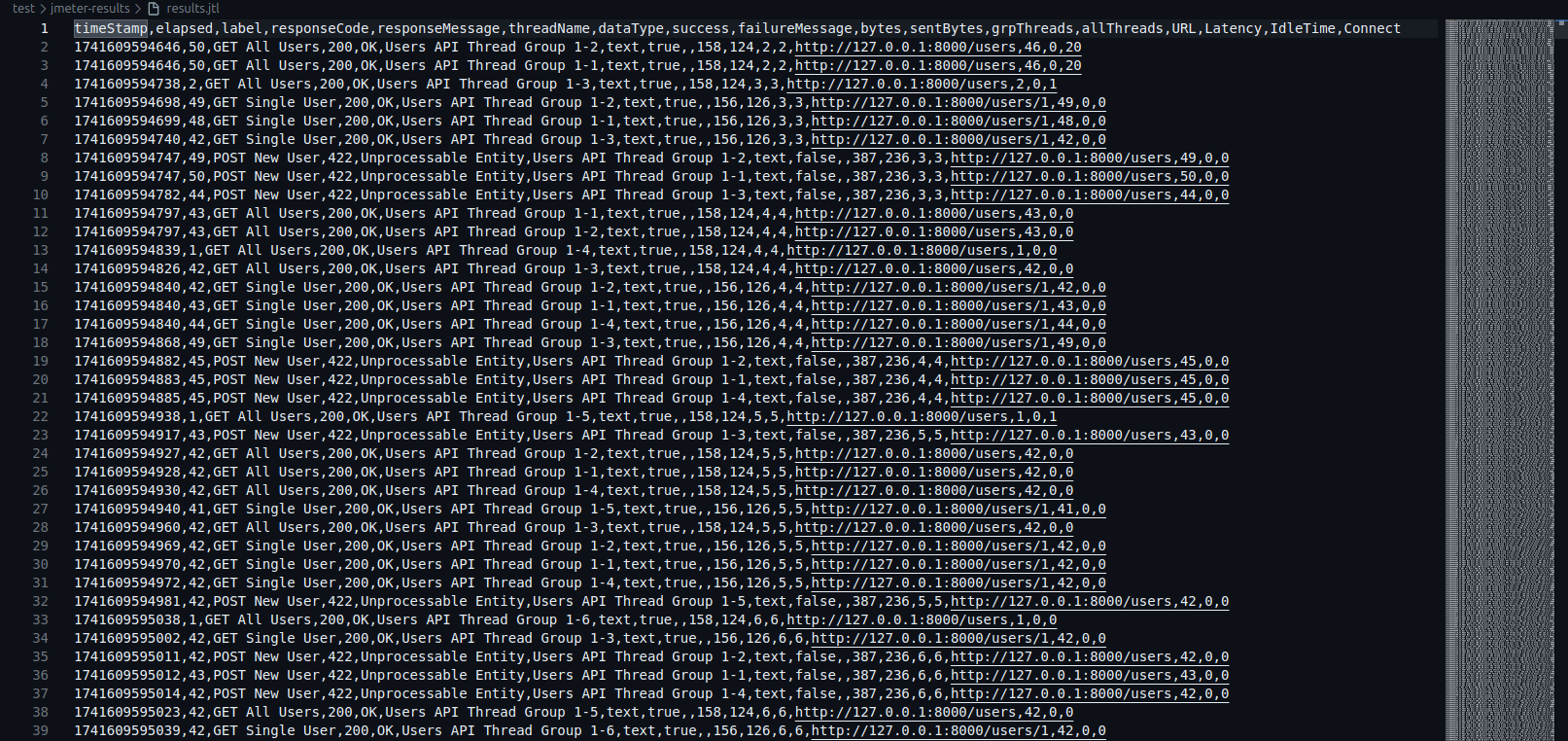
Для работы требуется наличие докера в системе

Далее запускаем скрипт

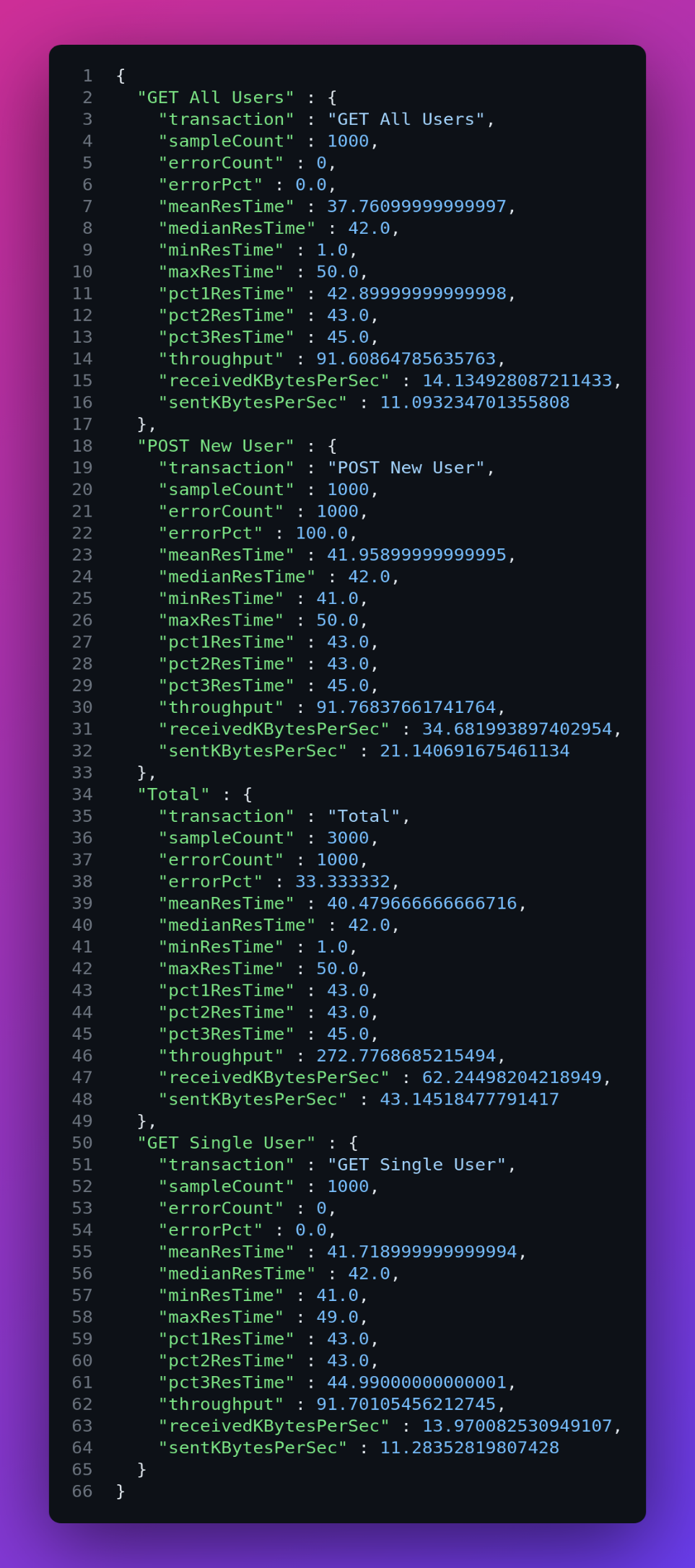
Также видим отчет в папке jmeter-results



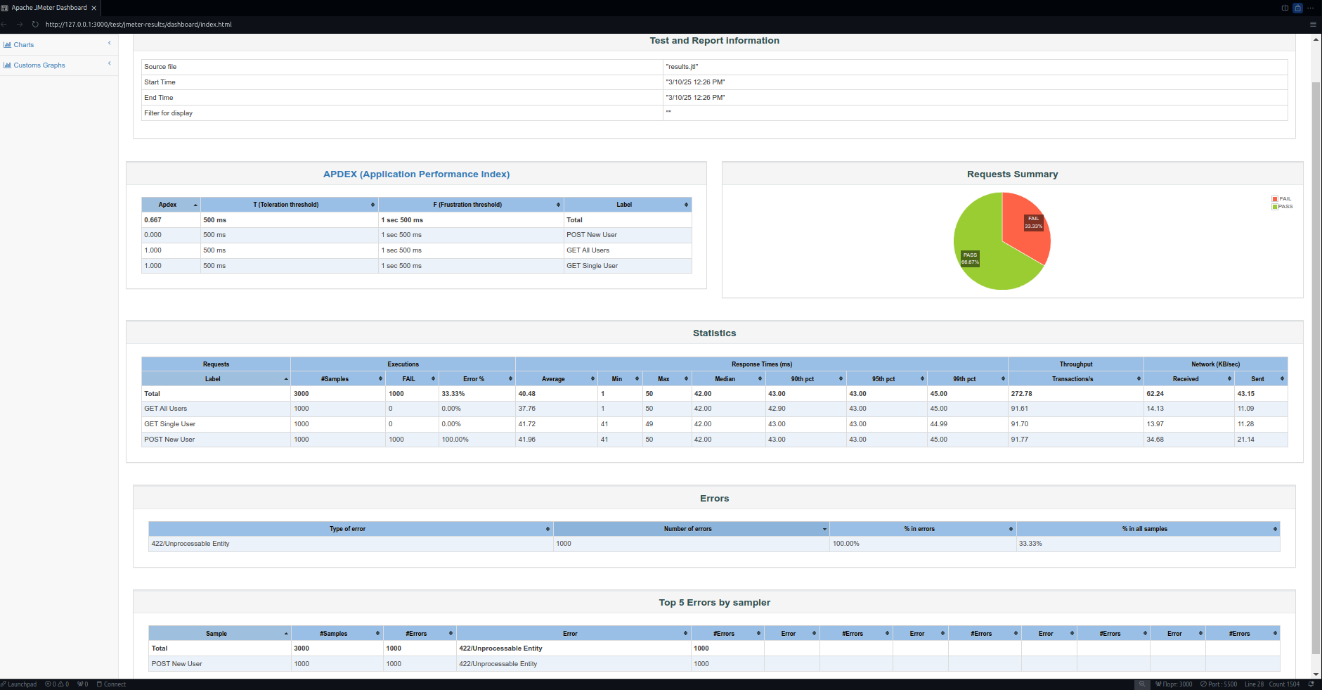
Смотрим датасет результатов



Смотрим среднюю статистику на основе датасета



Заходим на созданную страницу с дашбордом

Смотрим время ответа

