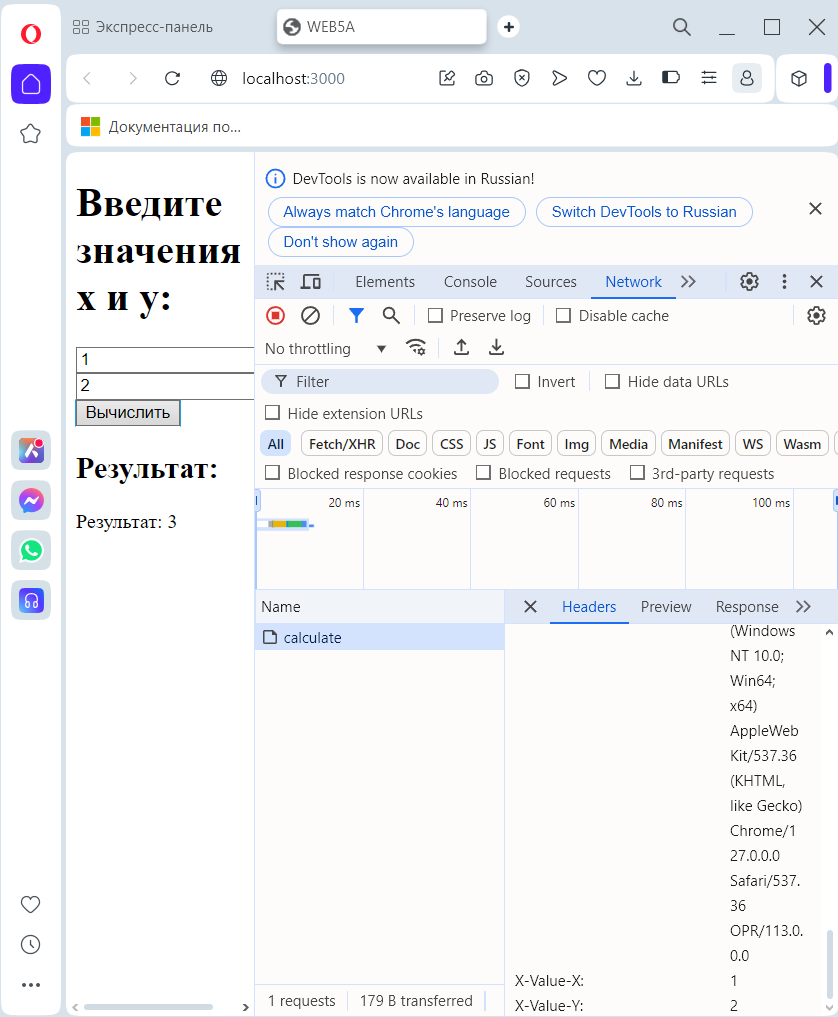
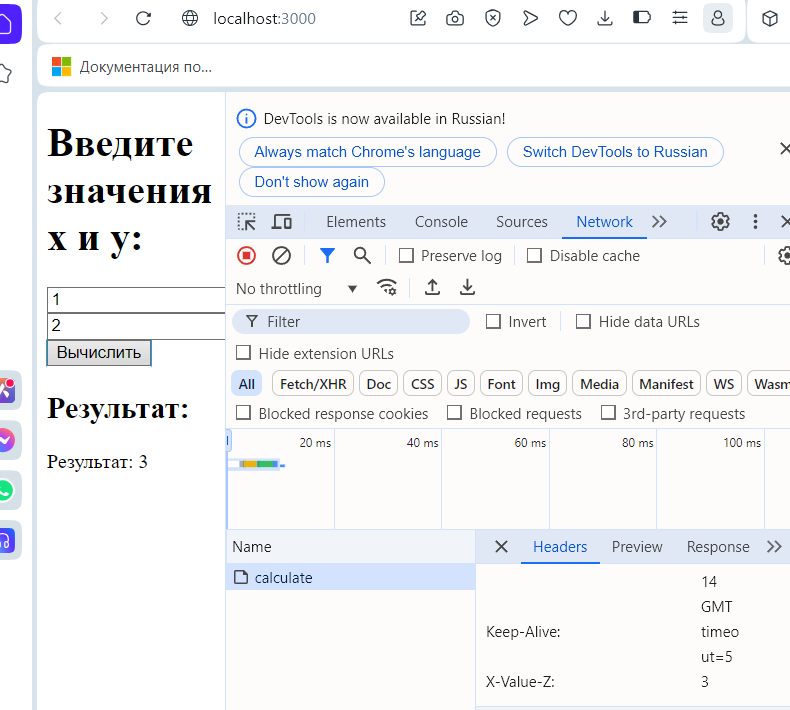
**TРWП**

**ПОИТ-3**

**Лабораторная 3.Asynchronous JavaScript and XML (AJAX) (4 часа)**

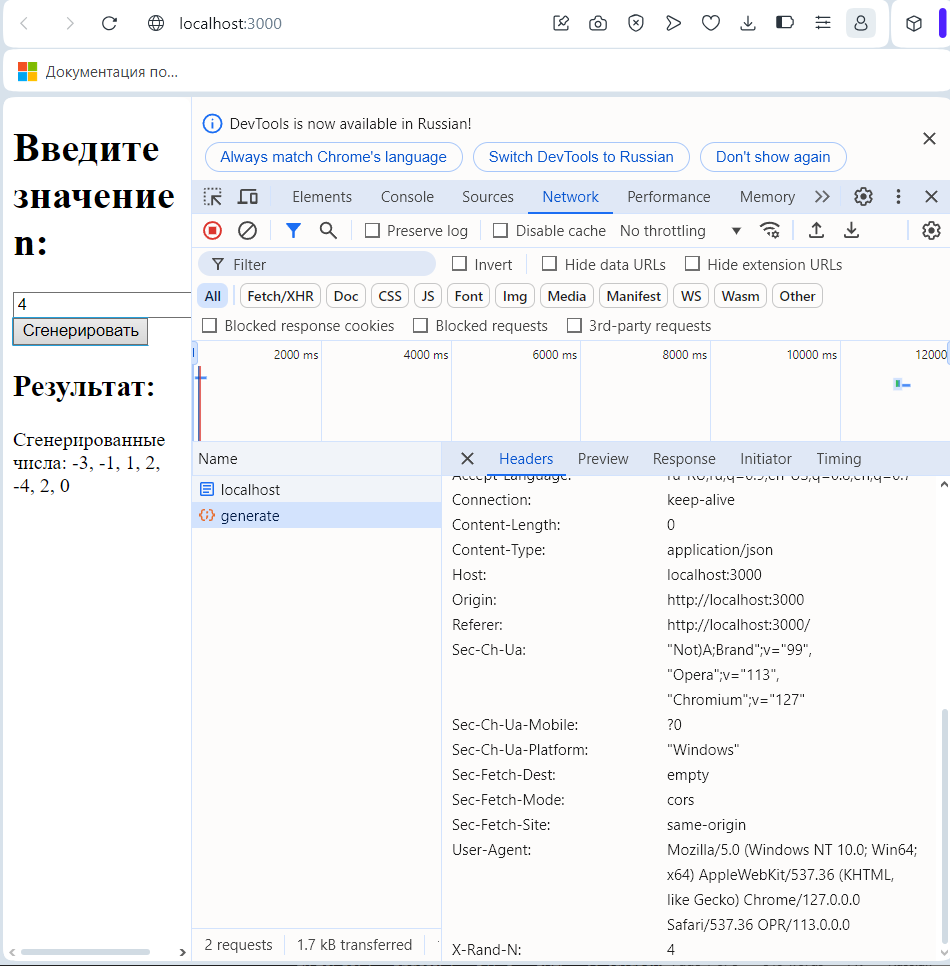
**Задание A. Применение заголовков для обмена между сценарием JavaScript и серверным приложением.**

1. Разработайте web-приложение **WEB5A**, содержащая HTTP-обработчик post-запроса.
2. HTTP-обработчик должен принимать от клиента запросы содержащие заголовки **X-Value-x** и **X-Value-y**; заголовки содержат целочисленные значения; HTTP-обработчик должен сформировать и отправить ответ с заголовком **X-Value-z,** содержащим сумму целочисленных значений полученных в запросе.
3. Разработайте форму для ввода значений **x**, **y** и отображения **z**. Разработайте сценарий JavaScript, который упаковывает **x** и **y**  в заголовки с именем **X-Value-x** и **X-Value-y** в http-запрос и оправляет его HTTP-обработчику c помощью post-запроса**,** а после этого принимает ответ, распаковывает заголовок ответа **X-Value-z** и отображает его с помощью тега, соответствующего **z**.

****

**Задание B. Применение JSON для обмена между сценарием JavaScript серверным приложением.**

1. Разработайте web-приложение WEB5B, содержащая HTTP-обработчик post-запроса.
2. HTTP-обработчик должен принимает заголовок **X-Rand-N** целочисленным значением **n** исформировать ряд целых случайных чисел из интервала **(-n, n)**. Количество сформированных чисел в ряду должно быть случайным из интервала **[5,10]**. Полученный ряд необходимо отправить клиенту в формате JSON.
3. Разработайте форму для ввода значений **n** и вывода ответа. Разработайте сценарий JavaScript, который упаковывает значение **n**  в заголовок с именем **X-Rand-N** http-запроса и оправляет его HTTP-обработчику**,** а после этого принимает ответ, разбирает полученный json-текст в тело тега, предназначенного для отображения результата.



**Задание\_C. Применение асинхронной обработки.**

1. Сведите 2 формы, сделанные в заданиях A,B в одну.
2. Разработайте web-приложение WEBD, содержащее три HTTP-обработчика post-запросов, аналогичных разработанным в заданиях A, B.
3. Установите в вызове метода **open** объекта **XMLHTTPRequest** значение третьего параметра (асинхронность) в **false**.
4. Внесите изменения в разработанные HTTP-обработчики таким образом, чтобы они выполняли http-ответ с задержкой: A(10 сек), B (1 сек).
5. Выполните приложение и объясните эффект.
6. Верните значение параметра в методе **open** в состояние **true**.
7. Проверьте возможность асинхронной работы вашего приложения (не дожидаясь результата в первой форме, сразу выполнять вторую и третью). Объясните полученный результат.

npm init -y

npm install koa koa-router

npm install koa-bodyparser