Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование протокола HTTP

Студент: Сосновец М. И.

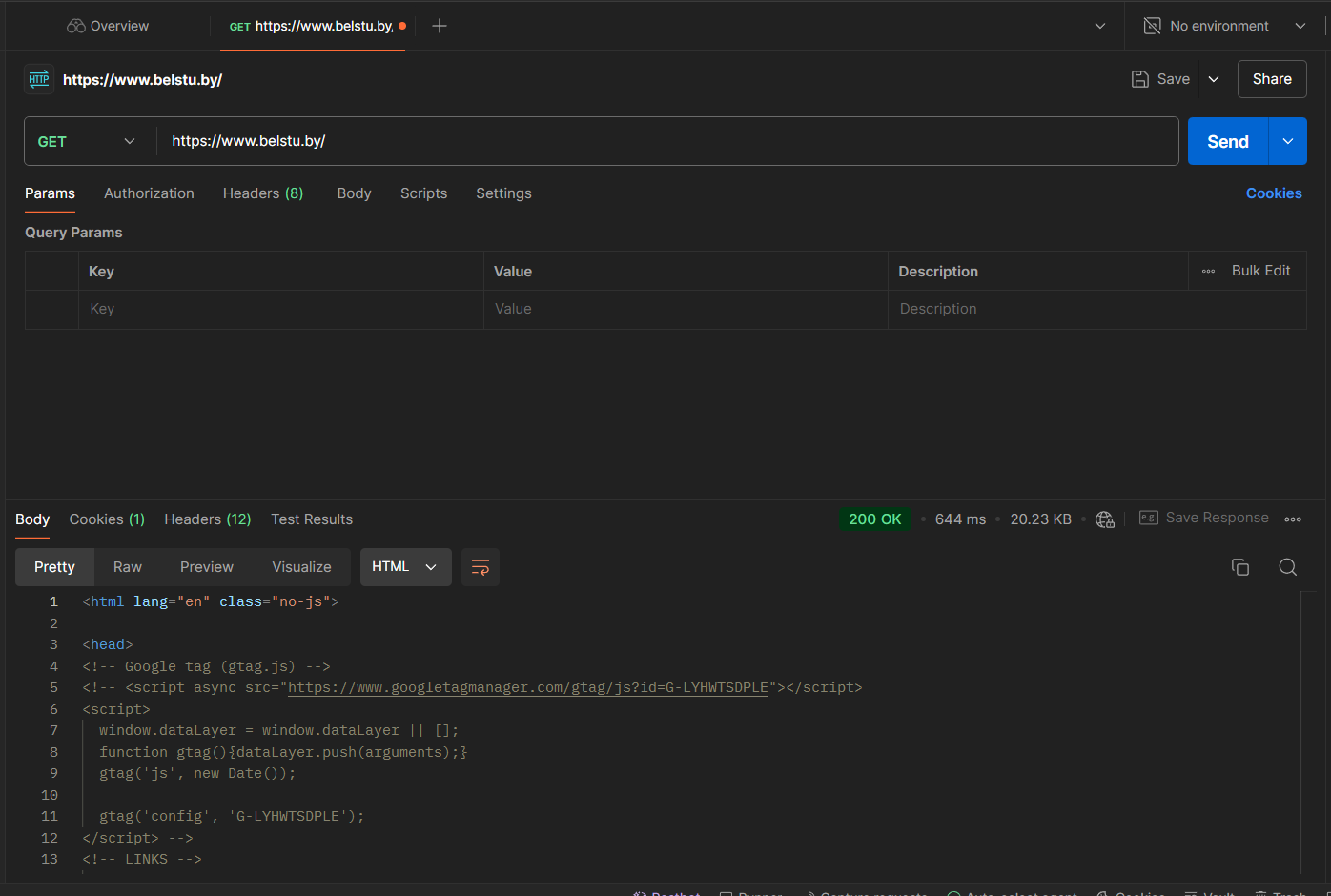
ФИТ 3 курс 4 группа

Минск 2024

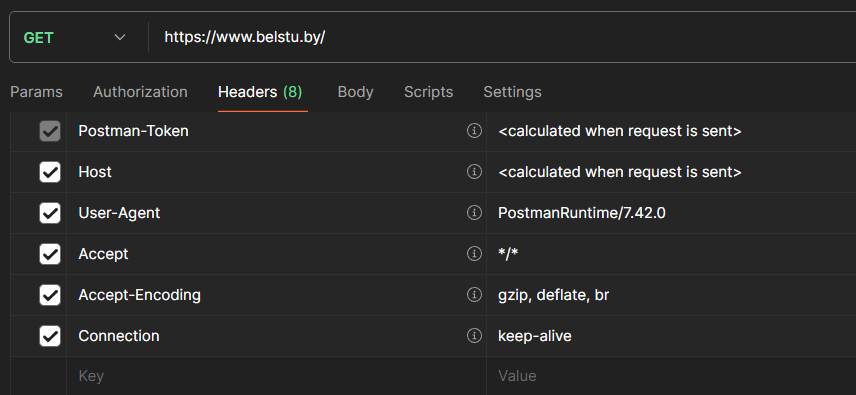
1. Ознакомьтесь с описанием протокола HTTP на сайте IETF.
2. Установите приложение POSTMAN для анализа сообщений протокола HTTP.

**Задание1**

1. **POSTMAN:** выполните GET-запрос к сайту БГТУ: <https://www.belstu.by/>



1. Определите заголовки запроса. Поясните их назначение.



Postman-Token(Request заголовки) – пользовательский заголовок (custom), содержит случайный UUID для каждого запроса (помогает в отладке разработчикам). Значение этого заголовка - это уникальный идентификатор, который генерируется для каждого запроса.

Host(Request заголовки) – указывает хост и номер порта.

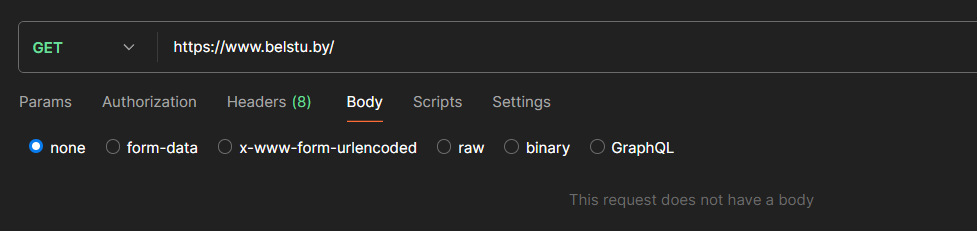
User-Agent(Request заголовки) – содержит информацию о клиенте, который отправляет запрос. Это может включать в себя детали о браузере, операционной системе и другую информацию, которая может быть полезна для сервера при формировании ответа.

Accept(Request заголовки) – список допустимых форматов ресурсов, которые клиент может принять (Если \*/\* - то клиент, у нас это браузер, может принять любой тип контента (html, json, png …).

Accept-Encoding(Request заголовки) – список допустимых кодировок понятные клиенту (Если gzip, deflate, br – то это просто алгоритмы сжатия, инфа по каждому думаю излишне).

Connection(general заголовки) – запрашивает открытый или закрытый тип соединения (Кроме keep-alive еще может быть close (закрыть соединение), update (переключение на другой протокол)).

1. Определите содержимое тела запроса. Поясните.



Тело пустое. Основная идея GET-запросов заключается в том, что они передают информацию через URL, а не через тело запроса.

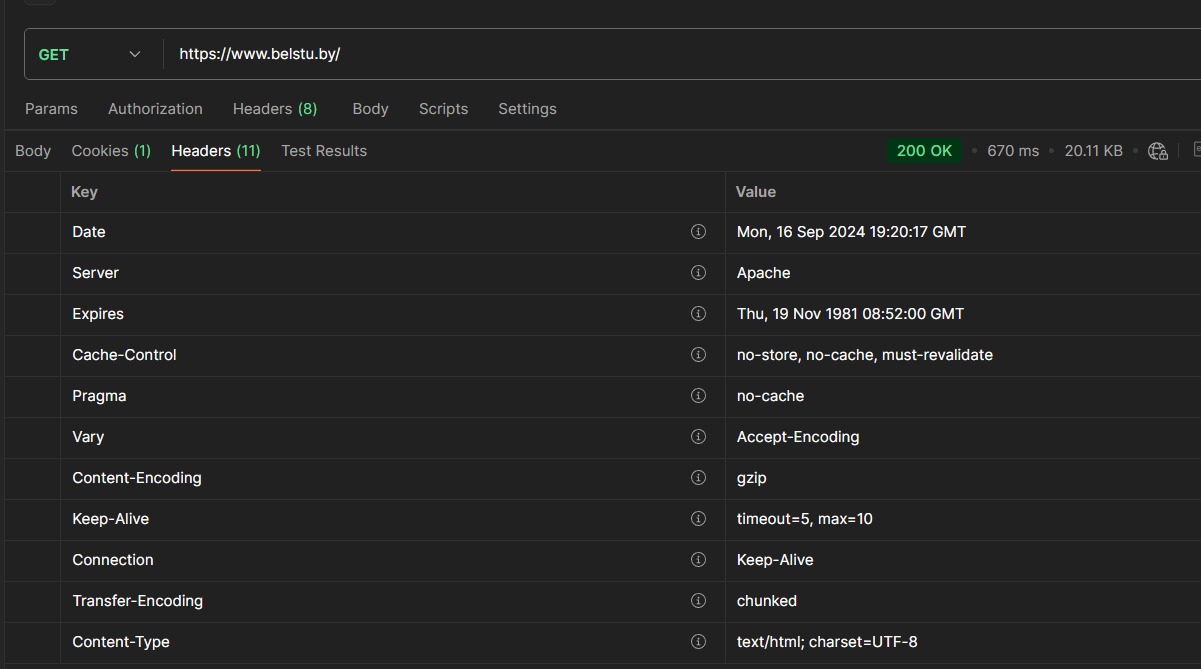
1. Определите статус ответа. Поясните значение.



200 OK – запрос успешно выполнен

Статус ответа дает информацию о том, как был обработан запрос

1. Определите заголовки ответа. Поясните их назначение.



**General заголовки:**

Date – дата и время генерации отклика.

Cache-Control – используется для задания инструкций кеширования как для запросов, так и для ответов (Может содержать: no-store (запрещает кеширование), no-cache (необходимость отправить запрос на сервер для валидации ресурса перед использованием закешированных данных), must-revalidate (кешированный ответ должен быть проверен на актуальность перед использованием)).

Pragma – устаревший заголовок, который используется для обратной совместимости с HTTP/1.0, означает то же, что и Cache-Control: no-cache (указывает на необходимость отправить запрос на сервер для валидации ресурса перед использованием закешированных данных).

Keep-Alive – пользовательский заголовок (custom), работающий вместе с Connection. Указывает сколько по времени или количество запросов проживет соединение с сервером. (Не поддерживается в HTTP/2 и HTTP/3)

Connection – определяет останется ли соединение открытым (keep-alive) или закрытым (close) после завершения текущей транзакции. (Не поддерживается в HTTP/2 и HTTP/3)

Transfer-Encoding – определяет форму кодирования, используемую для безопасной передачи payload body пользователю.

**Response заголовки:**

Server – содержит информацию как (с помощью чего) сервер обрабатывает запросы (В данном случае, у меня на сервере работает веб-сервер Apache, еще может быть nginx, GWS, Microsoft IIS).

Set-Cookie – используется для отправки куки от сервера к пользователю (при повторном запросе заголовок отсутствует).

Expires – определяет дату и время, когда клиенту следует считать копию ресурса устаревшей. Прошедшая или невалидная дата, например 0, обозначает, что ресурс уже устарел.

Vary – заголовок используется для отображения запрашиваемого содержимого страницы при ее кэшировании. Наиболее актуально использование данного заголовка при наличии нескольких версий динамических или статических веб-страниц (мобильных, десктопных, мультиязычных версий) под одним URL.

Процесс кэширования призван сократить время и ресурсы при передаче данных от браузера к первоначальному серверу.

Vary позволяет отдавать на запрос одного и того же URL разные ответы в зависимости от различных факторов и условий:

* Версии сайта (для десктопа или мобильных платформ);
* Географии или языка пользователя;
* Статуса пользователя (регистрация на сайте, оплата подписки, наличие товаров в корзине) и др.

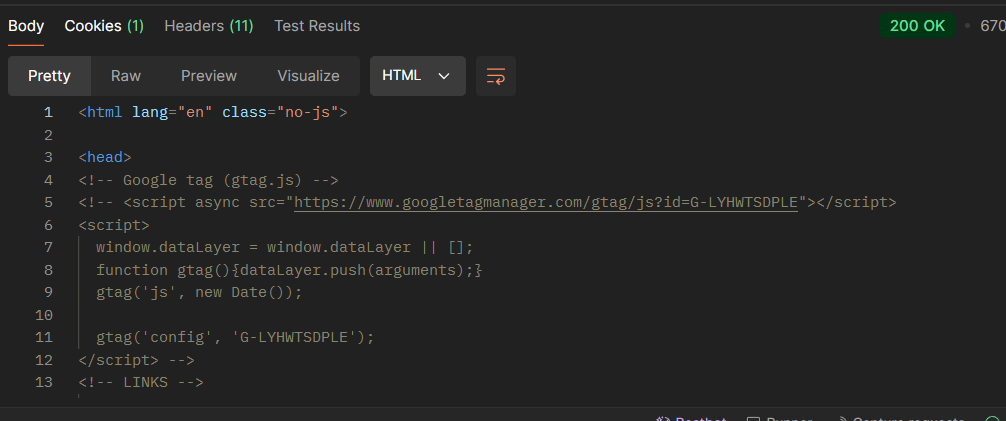
(Если установлен Accept-Encoding, то сервер может отправлять разные версии ресурса в зависимости от того, какие кодировки сжатия поддерживаются клиентом, может еще быть User-Agent (разные версии в зависимости от, к примеру, браузера), Accept-Language (разные версии в зависимости от языка пользователя), Origin (в зависимости от источника запроса).

**Entity заголовки:**

Content-Encoding – определяет способ кодирования содержимого сущности при передачи.

Content-Type – указывает на тип мультимедийного ресурса, передаваемого в теле сообщения

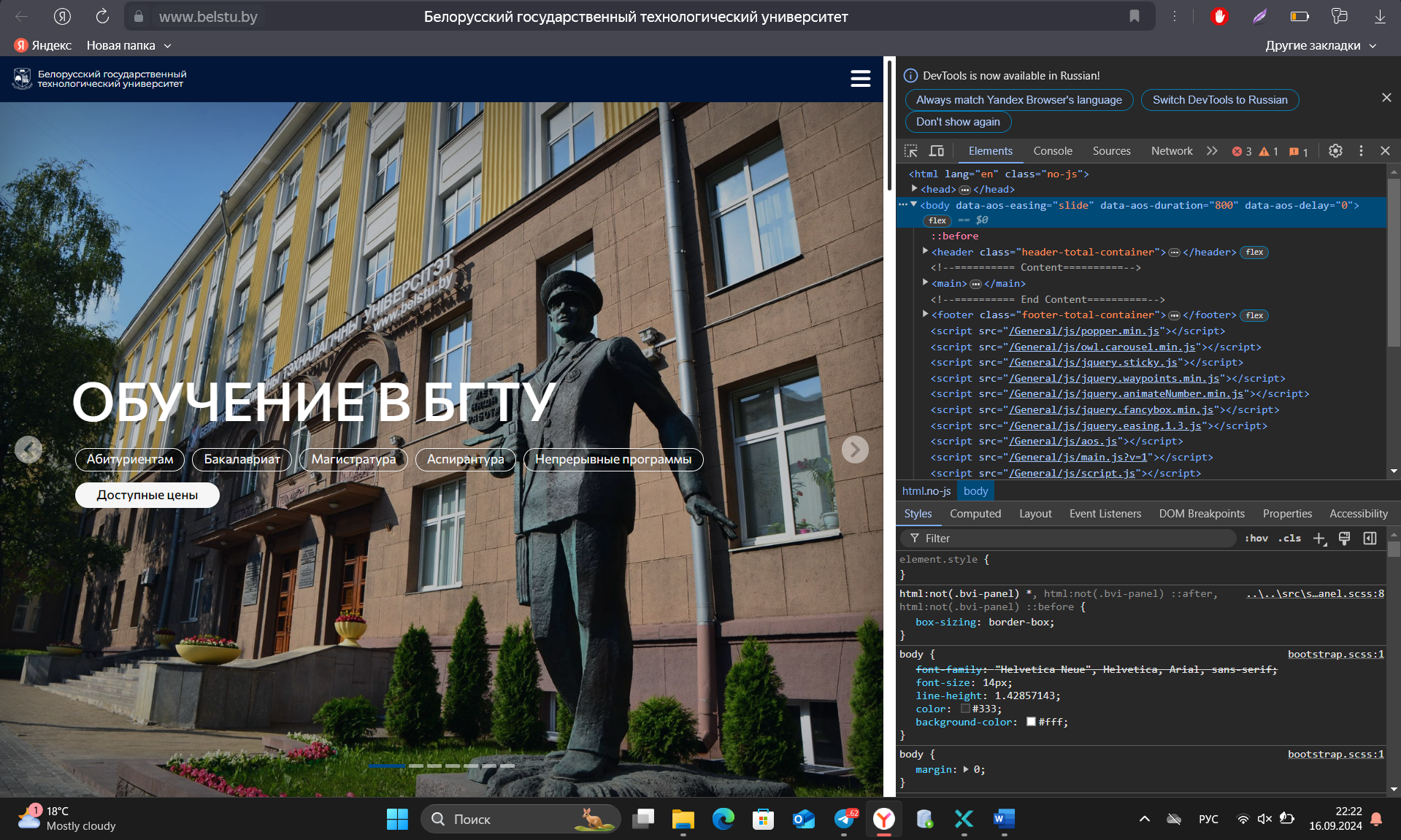
1. Определите содержимое тела ответа. Поясните.



В теле ответа будет HTML-код сайта belstu.by

**Задание2**

1. **WEB BROWSER**: выполните GET-запрос к сайту БГТУ: <https://www.belstu.by/>
2. Используйте «**Инструмент разработчи**ка» браузера



1. Определите заголовки запроса. Определите содержимое тела запроса.

**Request заголовки:**

Accept-Language – список языков поддерживаемых клиентом.

Sec-Ch-Ua – отправляет информацию о браузере.

Sec-Ch-Ua-Mobile – указывает браузер мобильной версии или нет.

Sec-Ch-Ua-Platform – указывает ОС клиента.

Sec-Fetch-Dest – указывает, как и где будут использоваться полученные данные (тип).

Sec-Fetch-Mode – указывает режим запроса (в общих чертах это позволяет серверу различать запросы, исходящие от пользователя, перемещающегося между HTML-страницами, и запросы на загрузку изображений и других ресурсов).

Sec-Fetch-Site – указывает на связь между источником инициатора запроса и источником запрошенного ресурса.

Sec-Fetch-User – указывает, что запрос инициирован пользователем.

Upgrade-Insecure-Requests – указывает, что браузер желает перенаправить запрос с HTTP на HTTPS.

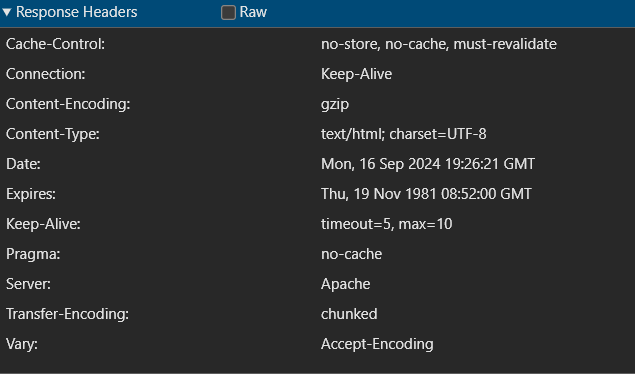
1. Определите содержимое тела запроса.

Тело пустое

1. Определите статус ответа.



1. Определите заголовки.

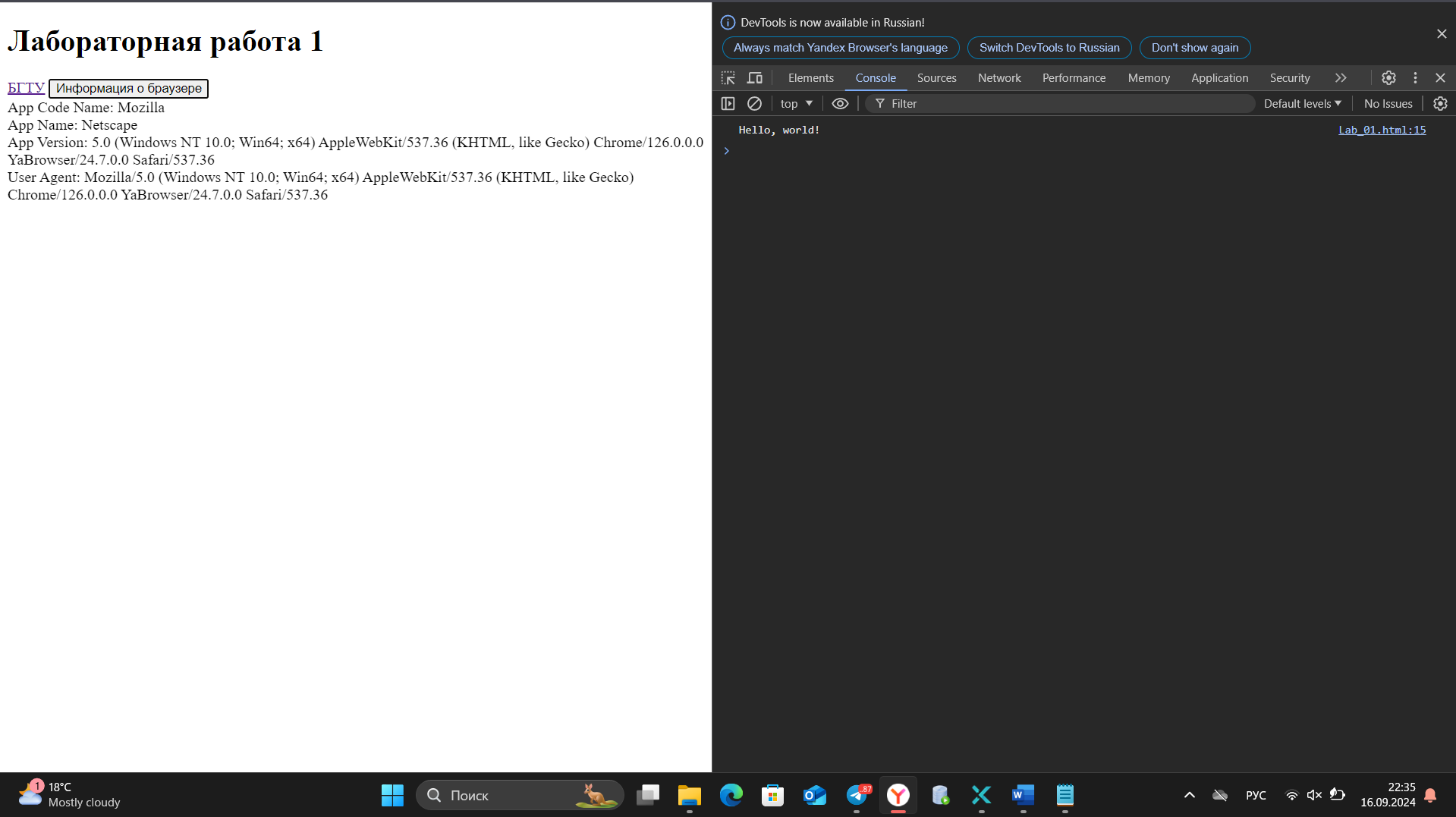


1. Определите содержимое тела ответа.

Веб-страница

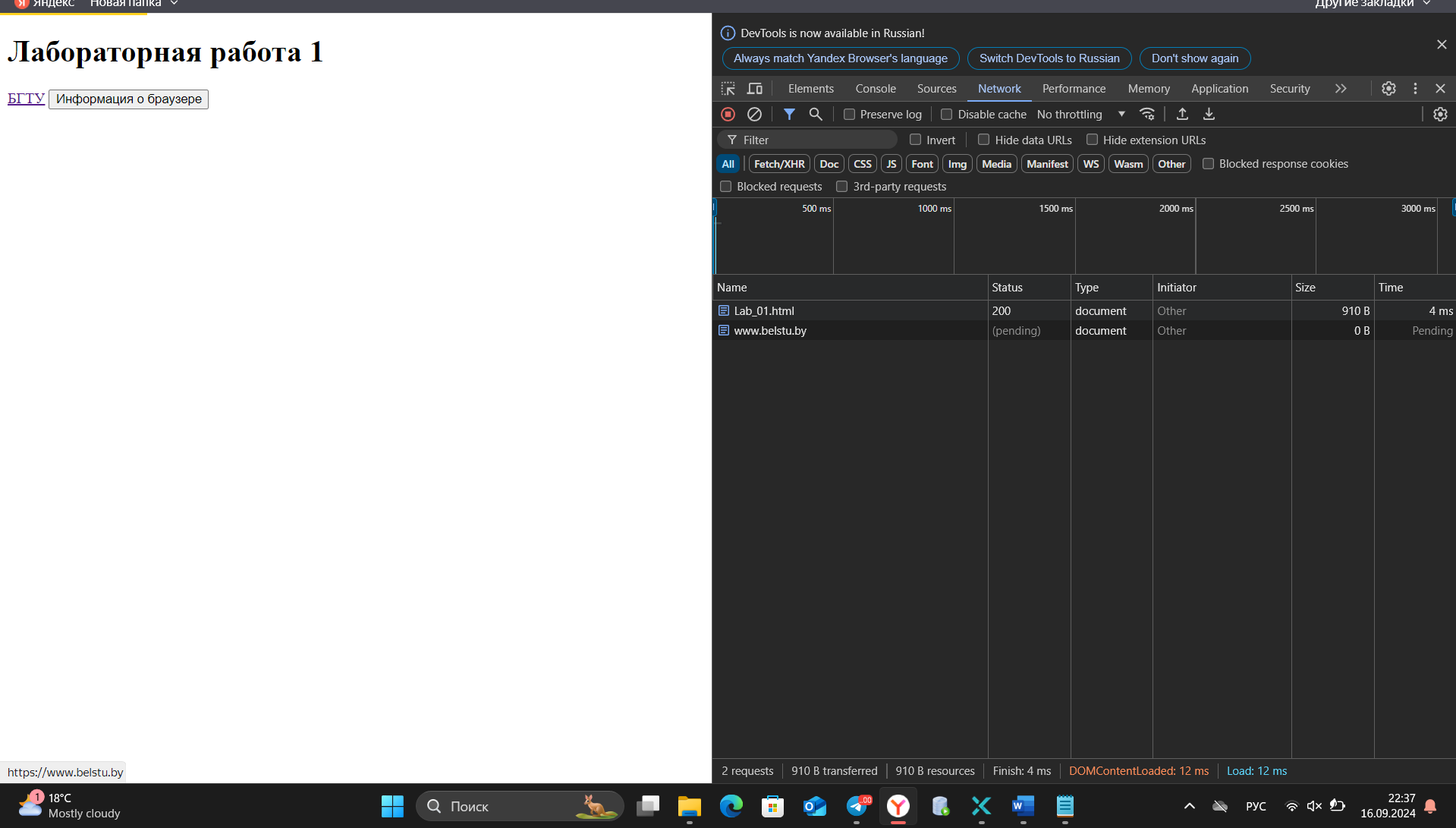
**Задание3**

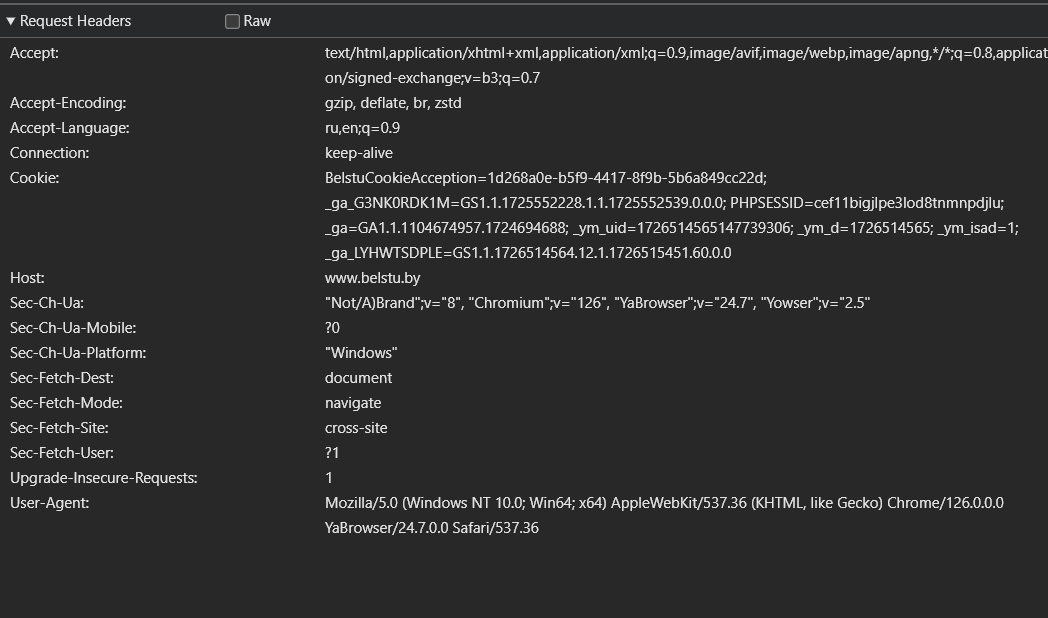
1. Разработайте html-страницу с помощью Блокнота, которая в браузере отображается следующим образом. Страница по ссылке БГТУ должна переходить на сайт БГТУ, при нажатии кнопки высвечивается информация о браузере, а в консоль разработчика должно выводиться Hello World.



1. Загрузите разработанную страницу в браузер и выполните запрос к сайту БГТУ с помощью ссылки.

Выполните анализа запроса и ответа с помощью «**Инструмент разработчи**ка» браузера





Изменилось только значение заголовка Sec-Fetch-Site. Было none (означает, что операция была инициирована пользователем, например: ввод URL-адреса в адресную строку, открытие закладки или перетаскивание файла в окно браузера), стало cross-site (инициатор запроса и сервер, на котором размещен ресурс, имеют разные сайты).

**Задание4. Ответьте на вопросы**

1. В каком документе описывает протокол HTTP?

RFC (Request for Comments) 2616

1. Какого уровня HTTP-протокол в моделях OSI/ISO, TCP/IP?

На прикладном уровне модели OSI/ISO и на уровне приложений в модели TCP/IP.

1. Что такое web-приложение?

**Веб-приложение** – клиент-серверное приложение, в котором клиент и сервер взаимодействуют с помощью протокола HTTP.

1. Какие порты использует HTTP-протокол?

80 – HTTP

443 – HTTPS

1. Что такое RFC?

RFC (Request for Comments) – это серия документов, содержащих технические спецификации и стандарты, широко применяемые в Интернете.

1. Почему HTTP-протокол называют ассиметричным?

Потому что сообщения Request и Response отличаются (т.к. роли сервера и клиента разные при обмене сообщениями)

1. Что такое MIME?

MIME: Multipurpose Internet Mail Extensions - многоцелевые расширения Internet-почты. Используется и как стандарт кодирования Internet-сообщений.

