Задание 1

**Заголовки запроса**

**Cookie**: содержит сохраненные [файлы cookie HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies) , связанные с сервером

**Postman-Token**:  это заголовок HTTP, который используется для обеспечения идемпотентности запросов в клиенте Postman или других инструментах для тестирования API.

**Host**: содержит имя домена, для которого предназначен запрос и, опционально, номер порта.

**User-Agent**: строка с характеристиками, по которым сервера и сетевые узлы могут определить тип приложения, операционную систему, производителя и/или версию [пользовательского агента](https://developer.mozilla.org/ru/docs/Glossary/User_agent).

**Accept**: указывает, какие типы контента, выраженные как [MIME типы](https://developer.mozilla.org/ru/docs/Web/HTTP/Basics_of_HTTP/MIME_types), клиент может понять

**Accept-Encoding**: Список форматов сжатия данных, которые поддерживает клиент.

**Connection**: определяет, остаётся ли сетевое соединение активным после завершения текущей транзакции (запроса). Если в запросе отправлено значение keep-alive, то соединение остаётся и не завершается, позволяя выполнять последующие запросы на тот же сервер

**Тело запроса**

Отсутствует. У GET-запросов его не бывает.

**Заголовки ответа**

**Date**: Дата и время отправки запроса

**Server**: Информация о том как сервер принимает запросы (тип сервера)

**Expires**: Cодержит дату/время, по истечении которой ответ сервера считается устаревшим

**Cache-Control**: Определяет правила кэширования

**Pragma**: это заголовок, зависящий от реализации, который может иметь различные эффекты в цепочке запрос-ответ. Он используется для обратной совместимости с кешами HTTP / 1.0, где заголовок Cache-Control HTTP / 1.1 ещё не присутствует

**Vary**: это механизм используемый для отображения различных представлений ресурса по тому же URI, так чтобы клиент мог указать, что лучше подходит для пользователя (например, желаемый язык документа, формат изображения, или кодировку текста).

**Content-Encoding**: это сущность заголовка, используемая для сжатия медиа-типа. При наличии её значение определяет кодировку, применённую к сущности body

**Keep-Alive**: позволяет отправителю подсказать, как можно использовать соединение для установки тайм-аута и максимального количества запросов

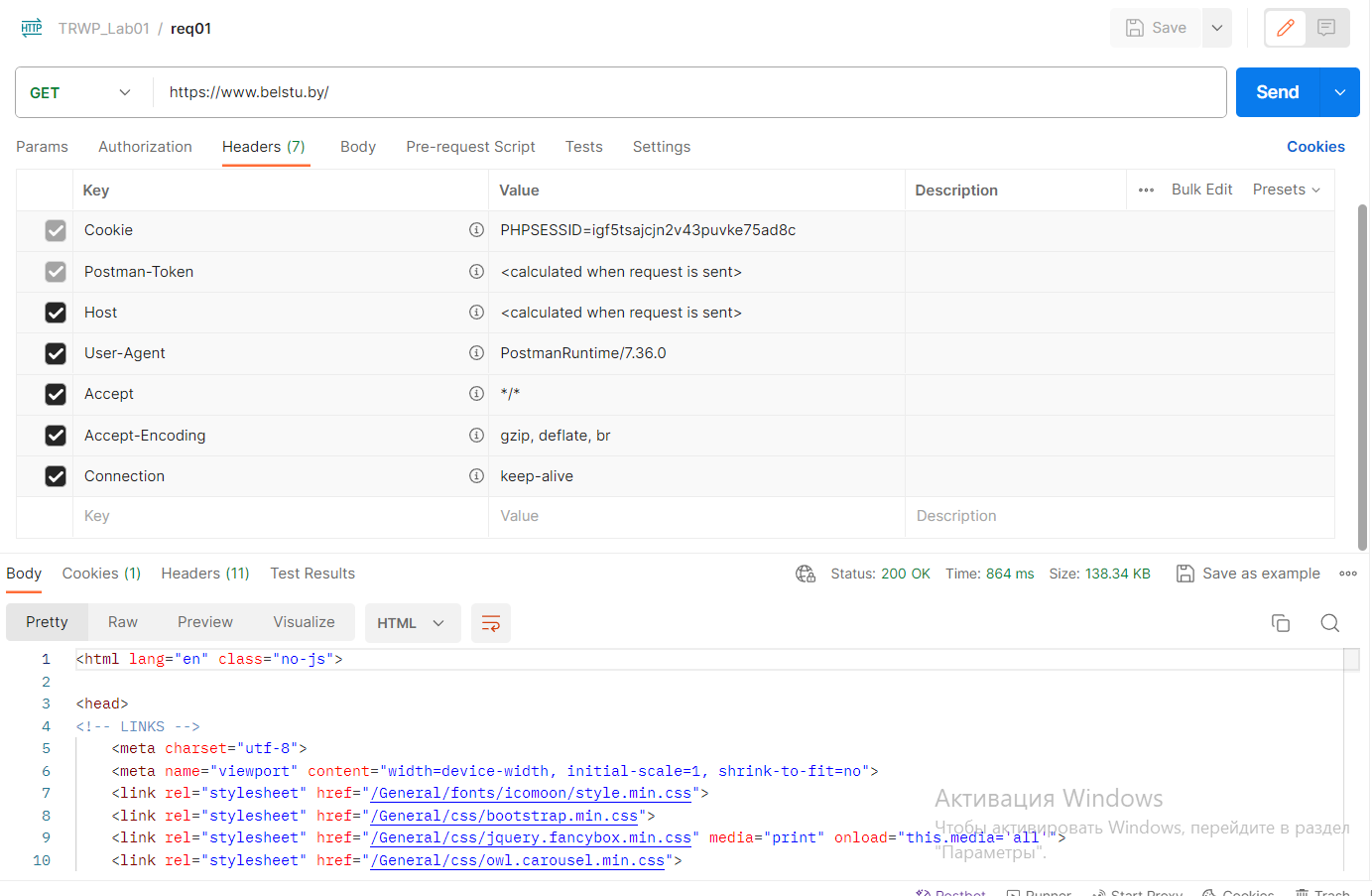
**Connection**: определяет, остаётся ли сетевое соединение активным после завершения текущей транзакции (запроса). Если в запросе отправлено значение keep-alive, то соединение остаётся и не завершается, позволяя выполнять последующие запросы на тот же сервер.

**Transfer-Encoding**: определяет форму кодирования, используемую для безоп асной передачи [тела полезной нагрузки](https://developer.mozilla.org/en-US/docs/Glossary/Payload_body) пользователю.

**Content-Type**: Позволяет клиенту определить MIME тип документа.

**Тело ответа**

HTML код страницы.



Вопросы

1. RFC 2616 – HyperText Transfer Protocol – HTTP/1.1
2. Прикладной
3. Клиент-серверная архитектура, протокол HTTP
4. 80, 443
5. Документы со спецификацией протокола. (Request for Comments)
6. Сервер может отправлять только ответы, а клиент запросы.
7. Стандарт для идентификации типов данных. (Multipurpose internet mail extensios)