Лабораторная работа 06

ПСКП

**Задание 00**

1. Разработайте приложение (сервер) **06-00** предназначенное для обработки следующих запросов.

|  |  |  |
| --- | --- | --- |
| HTTP-метод | URI | Задание |
| GET | /connection?set=***set*** | 01 |
| GET | /headers | 02 |
| GET | /parameter?x=***x***&y=***y*** | 03 |
| GET | /parameter/***x***/***y*** | 04 |
| GET | /socket | 05 |
| GET | /resp-status?code=***c***&mess=***m*** | 06 |
| POST | /formparameter | 07 |
| POST | /json | 08 |
| POST | /xml | 09 |
| GET | /files | 10 |
| GET | /files/***filename*** | 11 |
| GET/POST | /upload | 12 |

**Задание 01** /connection?set=***set***

1. При GET-запросе **/connection** в окно браузера вывести текущее значение параметра **KeepAliveTimeout.**
2. При GET-запросе **/connection?set=*set*** установить новое значение системного параметра **KeepAliveTimeout = *set*** и вывести в окно браузера сообщение, что установлено новое значение параметра **KepAliveTimeout=*set***.
3. Продемонстрируйте влияние системного параметра **KeepAliveTimeout** на работу приложения**.**

**Задание 02** /headers

1. Отобразите в окне браузера все заголовки запроса и ответа.
2. Объясните назначение каждого заголовка.

**Задание 03** /parameter?x=***x***&y=***y***

1. Проанализируйте значения параметров **x** и **y**.
2. Если **x** и **y** имеют числовые значения, то выведите в окно браузера сумму, разность, произведение и частное этих чисел
3. Иначе выведите сообщение об ошибке.

**Задание 04** /parameter/***x***/***y***

1. Проанализируйте значения параметров **x** и **y**.
2. Если **x** и **y** имеют числовые значения, то выведите в окно браузера сумму, разность, произведение и частное этих чисел.
3. Иначе выведите **URI**.

**Задание 05** /socket

1. При получении этого запроса, в окно браузера выведите ip- адрес, порт клиента и ip-адрес и порт сервера.

**Задание 06** resp-status?code=***c***&mess=***m***

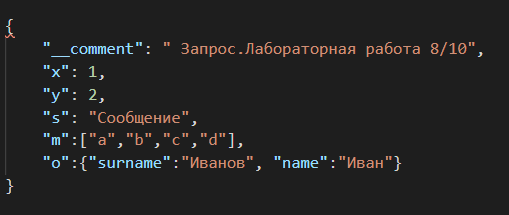
1. При получении этого запроса, сформируйте ответ, имеющий статус, заданный значением ***с***, и пояснение к статусу, заданное значением ***m***.

**Задание 07** /formparameter

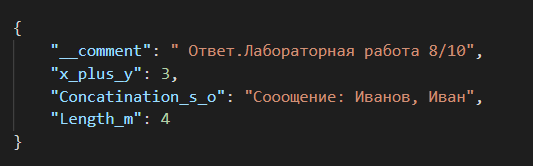
1. Используйте HTML-форму, включающую теги **input** с **type**: **text**, **number, date, checkbox, radiobutton,** тег **textarea**, а также два тега **input type=submit**, имеющих одно и тоже имя, но разные значения.
2. В окно браузера выведите значения параметров, полученных в запросе.

**Задание 08** /json

1. Принимайте POST-запросы, содержащие данные в json-формате и отправляйте ответы в json-формате.
2. Сообщение в запросе имеет следующую структуру:



1. Сообщение в ответе имеет следующую структуру:



Поле **x+y** ответасодержитсумму полей **x** и **y** запроса.

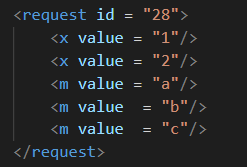
Поле **Concatination\_s\_o** ответа-конкатенациюполей **s** и свойств объекта **о** запроса.

Поле **Length\_m** ответа-количество элементов в массиве **m** запроса.

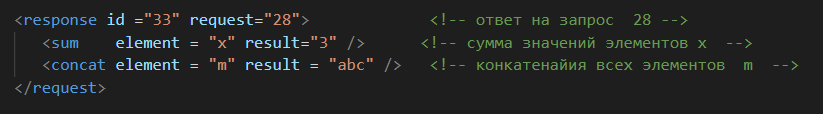
1. Проверьте работоспособность приложения с помощью POSTMAN.

**Задание 09 /**xml

1. Принимайте POST-запросы, содержащие данные в xml-формате и отправляйте ответы в xml-формате.
2. Сообщение в запросе имеет следующую структуру.



1. Количество элементов **x** и **m** в запросе может быть произвольным.
2. Сообщение в ответе имеет следующую структуру:



1. Элемент **sum** в ответе один и содержит сумму всех значений элементов **x** (в атрибуте **result**).
2. Элемент **concat** в ответе один и содержит сумму всех значений элементов **m** (в атрибуте **result**).
3. Проверьте работоспособность приложения с помощью POSTMAN.

**Задание 10** /files

1. В ответ на запрос высылается ответ с заголовком **X-static-files-count:** ***n***, где ***n*** -количество файлов в директории **static*.*** Используйте функции модуля **fs**.
2. Проверьте работоспособность приложения с помощью POSTMAN.

**Задание 11** /files/***filename***

1. В ответ на запрос высылается ответ, пересылающий файл с именем ***filename*** из директории **static.**
2. Если файл ***filename*** ненайден возвращается ответ со статусом 404.
3. Проверьте работоспособность приложения с помощью браузера.

**Задание 12** /upload

1. В ответ на GET-запрос к **/upload** высылается web-форма, позволяющая отправить POST-запрос к **/upload,** присылающийсерверу файл.
2. Сервер сохраняет файл в директории **static.**
3. Проверьте работоспособность приложения с помощью браузера.

**Задание 13** Ответьте на следующие вопросы

1. Поясните назначение заголовка **Content-Type.**

используется для указания типа содержимого, передаваемого в теле сообщения. Он состоит из типа и подтипа, разделенных косой чертой

1. Поясните назначение заголовка **Accept.**

используется для указания типов содержимого, которые клиент может понимать. Он состоит из одного или нескольких MIME-типов, разделенных запятыми. Например, Accept: text/html, application/xhtml+xml, application/xml;q=0.9, \*/\*;q=0.8 указывает, что клиент может понимать HTML-документы, XHTML-документы и XML-документы, но предпочитает HTML-документы

1. Для чего используется значение **multipart/form-data** заголовка **Content-Type.**

используется для передачи файлов и других двоичных данных вместе с формой HTML. Оно указывает, что тело сообщения содержит несколько частей, каждая из которых представляет отдельное поле формы. Каждая часть содержит заголовок Content-Disposition, который указывает имя поля формы, и заголовок Content-Type, который указывает тип содержимого части

1. Как с помощью тега **form**, обеспечить значение **multipart/form-data** заголовка **Content-Type.**

нужно добавить атрибут enctype="multipart/form-data" в тег form

1. Какое значение заголовка **Content-Type** отправляется тегом **form** в запросе по умолчанию.

application/x-www-form-urlencoded. Это означает, что данные формы будут закодированы в URL-кодированном виде и отправлены в теле запроса

1. Где и в каком формате передаются параметры в GET-запросе?

параметры передаются в строке запроса после символа вопроса ?

1. Где и в каком формате передаются параметры в POST-запросе?

передаются в теле сообщения. Они могут быть закодированы в различных форматах, таких как URL-кодирование, JSON или XML

1. Поясните понятие **JSON**?

формат обмена данными, основанный на синтаксисе объектов JavaScript

1. Поясните понятие **XML**?

Extensible Markup Language) - это язык разметки, используемый для хранения и передачи данных. Он представляет собой текст в формате ключ-значение, где ключи представлены тегами, а значения - содержимым тегов