Лабораторная работа 02

ПвИ

**Задание 01**

1. Разработайте серверное приложение **02-01**, которое на запрос [**http://localhost:5000**](http://localhost:5000)**/html** возвращает ответ, в теле которого содержится содержимое файла **index.html** (расположен в той же папке, что и скрипт **02-01.js**).
2. Файл **index.html** содержит статическую HTML-разметку с данными студента (факультет, курс, группа, фамилия, имя и отчество).
3. С помощью браузера проверьте работоспособность приложения.

**Задание 02**

1. Разработайте серверное приложение **02-02**, которое на GET-запрос [**http://localhost:5000**](http://localhost:5000)**/png** возвращает ответ, в теле которого содержится содержимое файла **pic.png** (расположен в той же папке, что и скрипт **02-02.js**).
2. Файл **pic.png** содержит любое изображение в формате png.
3. С помощью браузера проверьте работоспособность приложения.

**Задание 03**

1. Разработайте серверное приложение **02-03**, которое на GET-запрос [**http://localhost:5000/api/name**](http://localhost:5000/api/name)возвращает ответ, в теле которого содержится ваша фамилия имя и отчество в текстовом виде (text/plain).
2. C помощью браузера проверьте работоспособность приложения.
3. Проверьте работоспособность приложения с помощью **POSTMAN**.

**Задание 04**

1. Разработайте серверное приложение **02-04**, которое на запрос к [**http://localhost:5000/xmlhttprequest**](http://localhost:5000/xmlhttprequest)возвращает ответ, в теле которого содержится содержимое файла **xmlhttprequest.html** (расположен в той же папке, что и скрипт **02-04.js**).
2. Файл **xmlhttprequest.html** содержит HTML-разметку и JS.
3. С помощью объекта **XMLHTTPRequest** выполняется асинхронный GET-запрос к [**http://localhost:5000/api/name**](http://localhost:5000/api/name) иполученный ответ (фамилия, имя и отчество) динамически встраивается в HTML-разметку страницы.

**Задание 05**

1. Разработайте серверное приложение **02-05**, которое на запрос к [**http://localhost:5000/fetch**](http://localhost:5000/fetch)возвращает ответ, в теле которого содержится содержимое файла **fetch.html** (расположен в той же папке, что и скрипт **02-05.js**).
2. Файл **fetch.html** содержит HTML-разметку и JS.
3. С помощью объекта **fetch** выполняется асинхронный GET-запрос к [**http://localhost:5000/api/name**](http://localhost:5000/api/name)иполученный ответ (фамилия, имя и отчество) динамически встраивается в HTML-разметку страницы.

**Задание 06**

1. Разработайте серверное приложение **02-06**, которое на запрос к [**http://localhost:5000/jquery**](http://localhost:5000/jquery)возвращает ответ, в теле которого содержится содержимое файла **jquery.html** (расположен в той же папке, что и скрипт **02-06.js**).
2. Файл **jquery.html** содержит HTML-разметку и JS.
3. С помощью **jQuery.Ajax** (функция библиотеки jQuery, обеспечьте в **jquery.html** соответствующий <script> c jQuery CDN: [https://code.jquery.com](https://code.jquery.com/)) выполняется асинхронный GET-запрос к [**http://localhost:5000/api/name**](http://localhost:5000/api/name)иполученный ответ (фамилия, имя и отчество) динамически встраивается в HTML-разметку страницы.

**Задание 07.Ответьте на следующие вопросы**

1. Перечислите основные свойства **NODE.JS**
2. Что такое **npm**?
3. Поясните назначение HTTP-заголовка **Content-Type**.
4. Поясните назначение функции **require**.
5. Поясните понятие «**Модуль Node.js**».
6. Поясните понятие «**Node.js built-in modules**» («**Node.js Core modules**»).
7. Какой модуль **NODE.JS** обеспечивает работу с протоколом HTTP?
8. Какой модуль **NODE.JS** обеспечивает работу с файловой системой?
9. Основные свойства Node.js:

* Однопоточность и событийный цикл: Node.js работает в одном потоке и использует событийно-ориентированную модель, что позволяет эффективно обрабатывать множество одновременных запросов без блокировки основного потока.
* Неблокирующий ввод-вывод (Non-blocking I/O): Node.js использует асинхронные операции ввода-вывода, что позволяет обрабатывать запросы эффективно без блокировки.
* Модульная система: Node.js имеет мощную модульную систему, которая позволяет разбивать приложения на модули для повторного использования кода и упрощения разработки.
* Высокая производительность: Благодаря использованию неблокирующего ввода-вывода и эффективной обработке одновременных запросов, Node.js обеспечивает высокую производительность и масштабируемость.
* Кросс-платформенность: Node.js поддерживает различные операционные системы, такие как Windows, macOS и Linux, что позволяет разработчикам создавать приложения, которые могут работать на разных платформах.

1. npm (Node Package Manager) - это менеджер пакетов для Node.js. Он предоставляет среду для установки, управления и распространения пакетов JavaScript, которые могут быть использованы в проектах на Node.js. npm позволяет легко добавлять зависимости в проекты, управлять версиями пакетов и обновлять их.
2. HTTP-заголовок Content-Type определяет тип данных, которые отправляются или ожидаются в теле HTTP-запроса или ответа. Content-Type указывает серверу или клиенту, как правильно интерпретировать данные. Например, Content-Type может указывать, что данные являются HTML, JSON, XML или изображением. Это помогает обеспечить правильную обработку и интерпретацию содержимого.
3. Функция require - это встроенная функция в Node.js, которая используется для подключения модулей в других модулях. Она позволяет импортировать функциональность, определенную в других модулях, чтобы использовать ее в текущем модуле. Функция require принимает путь к модулю и возвращает экспортированные объекты или функции из этого модуля.
4. Модуль Node.js - это файл, содержащий код на JavaScript, который экспортирует определенные объекты, функции или переменные для использования в других модулях. Модули помогают разбивать код на отдельные компоненты с четкими границами и повторно использовать код между различными частями приложения. Каждый модуль имеет свою область видимости, что позволяет изолировать код и предотвращает конфликты имен.
5. Встроенные модули Node.js (Node.js Core modules) - это набор модулей, предоставляемых Node.js "из коробки" без необходимости установки дополнительных пакетов. Эти модули включаются в распространение Node.js и предоставляют функциональность для работы с различными аспектами, такими как файловая система, сеть, потоки, криптография и другие. Некоторые из встроенных модулей включают fs
6. . Модуль Node.js, который обеспечивает работу с протоколом HTTP, называется http. Он предоставляет классы и методы для создания HTTP-серверов и клиентов в Node.js. Модуль http позволяет создавать и обрабатывать HTTP-запросы и ответы, устанавливать заголовки, работать с параметрами запросов и многое другое.
7. Модуль Node.js, который обеспечивает работу с файловой системой, называется fs. Он предоставляет функции для выполнения операций с файлами и директориями, такие как чтение, запись, удаление, переименование и многое другое. Модуль fs позволяет взаимодействовать с файловой системой на компьютере, на котором выполняется Node.js, и осуществлять операции ввода-вывода с файлами.