Лабораторная работа 12

ПСКП

ПОИТ-3

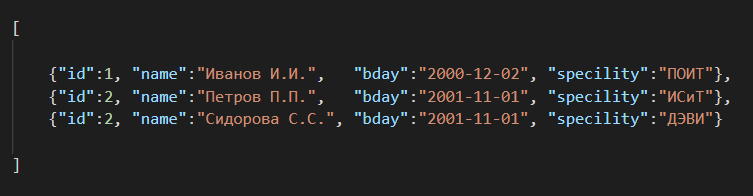
**Задание 01**

1. Разработайте серверное приложение 12-01, обрабатывающий запросы представленные в следующей таблице

|  |  |  |
| --- | --- | --- |
| Метод | URI | Назначение |
| GET | / | отправка клиенту полного списка студентов в формате json из файла **StudentList.json** |
| GET | /***n*** | отправка клиенту информацию о студенте из с идентификатором **id** равным **n** в формате json; если в файле **StudentList.json** такого студента нет, то клиенту сообщение об ошибке. |
| POST | / | клиент отправляет серверу информацию о новом студенте в формате json; сервер дополняет список студентов в файле **StudentList.json**;если в файле **StudentList.json** уже есть информация о студенте с таким же **id**, то клиенту отправляется сообщение об ошибке; если операция выполнена успешно, то клиенту возвращается информацию о добавленном студенте (в том же виде) в json-формате |
| PUT | / | клиент отправляет серверу информацию о студенте в формате json; сервер находит в **StudentList.json** информацию о студенте с таким же **id** и заменяет ее; если в **StudentList.json** не найдена информация с заданным **id,** то клиенту отправляется соответствующее сообщение об ошибке; если операция выполнена успешно, то клиенту возвращается информацию об измененном студенте (новые данные) в json-формате |
| DELETE | /n | сервер удаляет из **StudentList.json** информациюо студенте с **id** равным **n**; если в **StudentList.json** не найдена информация с заданным **id,** то клиенту отправляется соответствующее сообщение об ошибке; если операция выполнена успешно, то клиенту возвращается информацию об удаленном студенте (удаленные данные) в json-формате |
| POST | /backup | при получении запроса, сервер копирует файл **StudentList.json;** имя копии ***YYYYMMDDHHSS***\_**StudentList.json**, где ***YYYYMMDDHHSS*** и текущая и время; копирование осуществляется с задержкой в 2 сек. |
| DELETE | /backup/  ***yyyyddmm*** | сервер удаляет все копии файла **StudentList.json** дата создания которых старше заданной ***yyyyddmm*** |
| GET | /backup | сервер оправляет клиенту список копий файла **StudentList.json** в json-формате. |

1. Сервер должен генерировать уведомление, подписавшемуся клиенту, если любая копия файла **StudentList.json** будет изменена.
2. Для проверки корректности обработки http-запросов используйте POSTMAN.
3. Для получения уведомлений от сервера, разработайте соответствующий клиент и продемонстрируйте корректность генерации сервером уведомлений.
4. Примеры структур данных:

**StudentList.json –** список студентов



**Задание 02.**Ответьте на следующие вопросы

1. Поясните понятие «файл».

Именованная часть диска

1. Поясните понятие «файловая система».

Набор спецификаций, кот.отвеч.за создание/удаление/ чтение/запись и др.дейсвтия над файловой информацией, а также за управление доступом к файлу и упр-ние рес.кот.исп.файлами

1. Перечислите типы файловых систем.

FAT, HTFS, ext2/3 и др.

Для носителей с произвольным доступом (FAT32)

Для носителей с последовательным доступом (QIC)

Для оптических носителей (UDF, HFS)

Виртуальные файловые системы (AEFS)

Сетевые файловые системы (NFS, CIFS)

Специализированные файловые системы (ZFS)

1. Поясните понятие «поток данных».

Абстракция, используемая для чтения или записи файлов, сокетов и др.в единой манере.

1. Поясните понятие «системные потоки данных».

(м.использовать без подключения доп.библиотек)

1. Перечислите типы потоков данных, поддерживаемых Node.js.

Readable – для чтения

Writable – способ записи

Dublex – поток из кот.м.и читать, и писать в него

Transform – м.изменять д-е при чтении/запис