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** информацию о 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.

2. Сервер должен генерировать уведомление, подписавшемуся клиенту, если любая копия файла **StudentList.json** будет изменена.
3. Для проверки корректности обработки http-запросов используйте POSTMAN.

4. Для получения уведомлений от сервера, разработайте соответствующий клиент и продемонстрируйте корректность генерации сервером уведомлений.
5. Примеры структур данных:

StudentList.json – список студентов

```
[

    {"id":1, "name":"Иванов И.И.",   "bday":"2000-12-02", "specility":"ПОИТ"},
    {"id":2, "name":"Петров П.П.",   "bday":"2001-11-01", "specility":"ИСиТ"},
    {"id":2, "name":"Сидорова С.С.", "bday":"2001-11-01", "specility":"ДЭВИ"}

]
```

Информация о студенте

```
{"id":25, "name":"Алексей А.А.", "bday":"1999-09-01", "specility":"ПОИБМС"}
```

Информация об ошибках

```
{
    "error": 1,
    "message":"ошибка чтения файла D:/Files/StudentList.json"
}
```

```
{
    "error": 2,
    "message":"студент с id равным 22 не найден"
}
```

```
{
    "error": 3,
    "message":"студент с id равным 22 уже есть"
}
```

Задание 02. Ответьте на следующие вопросы

6. Поясните понятие «файл».
7. Поясните понятие «файловая система».
8. Перечислите типы файловых систем.
9. Поясните понятие «поток данных».
10. Поясните понятие «системные потоки данных».
11. Перечислите типы потоков данных, поддерживаемых Node.js.