

**Задание 01**

1. Разработайте серверное приложение **04-01**, которое принимает HTTP-запросы **GET, POST, PUT, DELETE** на <http://localhost:5000/api/db>.
2. Для имитации таблицы базы данных (БД) используйте JS-массив, а для имитации операций с «БД» соответствующие операции с массивом.
3. Назначение запросов.

<b>GET</b>	Получить все строки таблицы «БД». В теле ответа возвращается массив строк в json-формате.
<b>POST</b>	Добавить новую строку в таблицу «БД»; строка передается серверу в json-формате. В теле ответа возвращается измененная строка в json-формате.
<b>PUT</b>	Изменить существующую строку таблицы «БД»; строка передается серверу в json-формате.
<b>DELETE</b>	Удалить существующую строку (по id) в таблице «БД»; серверу передается id (в query-строке). В теле ответа возвращается удаленная строка в json-формате. Пример: <a href="http://localhost:5000/api/db?id=25">http://localhost:5000/api/db?id=25</a>

4. Для работы с «БД» в приложении должен быть разработан специальный объект **DB**, который должен располагаться в отдельном модуле.
5. Объект **DB** должен уметь генерировать и **асинхронно** обрабатывать следующие события:

<b>GET</b>	Получить все строки таблицы «БД».
<b>POST</b>	Добавить новую строку в таблицу «БД».
<b>PUT</b>	Изменить существующую строку таблицы «БД».
<b>DELETE</b>	Удалить существующую строку в таблице «БД».

6. Для выполнения операций с «БД» объект **DB** должен обеспечивать следующие функции:

<b>select</b>	Получить все строки таблицы «БД».
<b>insert</b>	Добавить новую строку в таблицу «БД».
<b>update</b>	Изменить существующую строку таблицы «БД».
<b>delete</b>	Удалить существующую строку в таблице «БД».

7. Строка таблицы содержит:
- идентификатор (**id**);
  - имя (**name**);
  - дата рождения (**bday**).
8. Проверьте работоспособность приложения с помощью **POSTMAN**.

### **Задание 02**

9. Разработайте серверное приложение **04-02**, используя код приложения **04-01**.
10. Дополнительно к функциональности, унаследованной от **04-01**, приложение **04-02** должно отвечать на GET-запросы к <http://localhost:5000/>.
11. В ответ на GET-запрос к <http://localhost:5000/> сервер пересылает клиенту HTML-страницу, которая должна располагаться в той же директории, что и js-файл приложения.
12. HTML-страница должна обеспечивать ввод необходимых данных, отправку всех типов запросов к серверу и обработку полученных результатов с применением функции **fetch**.
13. Проверьте работоспособность приложения с помощью браузера.

### **Задание 03.** Ответьте на следующие вопросы.

14. Дайте пояснение понятию «событие программного объекта».
15. Какой встроенный механизм используется в **Node.js** для генерации и обработки событий. Поясните принцип его работы.