

Задание 01

1. Разработайте серверное приложение **04-01**, которое принимает HTTP-запросы **GET**, **POST**, **PUT**, **DELETE** на <http://localhost:5000/api/db>.

2. Назначение запросов.

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

3. Для работы с БД в приложении должен быть разработан специальный объект **DB**, который должен располагаться в отдельном модуле.

4. Объект **DB** должен уметь генерировать и **асинхронно** обрабатывать следующие события

GET	Получить все строки таблицы базы данных (БД).
POST	Добавить новую строку в таблицу БД.
PUT	Изменить существующую строку таблицы БД.
DELETE	Удалить существующую строку в таблице БД.

5. Для выполнения операций с БД объект **DB** должен обеспечивать следующие функции

select	Получить все строки таблицы базы данных (БД).
insert	Добавить новую строку в таблицу БД.
update	Изменить существующую строку таблицы БД.
delete	Удалить существующую строку в таблице БД.

6. Строка таблицы содержит:
 - идентификатор (**id**);
 - имя (**name**);
 - дата рождения (**bday**).
7. Для имитации таблицы БД используйте JS-массив, а для имитации операций с БД соответствующие операции с массивом.
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/> приложение (сервер) пересылает клиенту HTML-страницу, которая должна располагаться в том 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 пояснению понятию «событие программного объекта».
15. Объясните механизм генерации и обработки событий в C#.
16. Поясните как самостоятельно реализовать механизм генерации и обработки событий на JS или C++.
17. Какой встроенный механизм используется в **Node.js** для генерации и обработки событий. Поясните принцип его работы.