Реализовать сервер, который:

хранит массив объектов типа { time: number, value: number }, который инициализируется из файла data.csv;

максимальное число элементов массива 1000;

каждую секунду сервер генерирует новое случайное значение в диапазоне 0 - 100;

новое значение и время добавляется в массив объектов, при этом массив сохраняется в файл data.csv;

Сервер имеет следующий роутинг:

GET api/current возвращает текущее значени (первый элемент массива) { time: number, value: number }

GET api/history возвращает массив всех значений [{ time: number, value: number }]

Сервер имеет возможность подключения клиентов по WebSocket

при каждом подключении клиента сервер отсылает подключившемуся клиенту массив всех значений [{ time: number, value: number }]

при каждом создании нового значения сервер рассылает всем клиентам новое значение { time: number, value: number }

Сервер хранит в директории public две страницы html (RestExample.html, WsExample.html)

Страница RestExample.html работает с сервером по REST

при запуске инициируется запрос на сервер GET api/history для получения всех значений [{ time: number, value: number }]

при запуске создается интервал в котором каждую секунду осуществляется запрос GET api/current

на странице присутствуют два графика (canvas и svg)

графики в реальном времени отображают значения полученные с сервера (обновление происходит при получении текущего значения)

Страница WsExample.html работает с сервером по WebSocket

при запуске инициируется connect к серверу по ws

в ответ на connect сервер присылает массив всех значений [{ time: number, value: number }]

на каждый onmessage с сервера приходит новое значение

на странице присутствуют два графика (canvas и svg)

графики в реальном времени отображают значения полученные с сервера (обновление происходит при получении текущего значения)