

Практическая работа «WebSocket API» по курсу JavaEE.

В рамках данного задания необходимо разработать механизм обновления/актуализации наиболее востребованных данных, отображаемых в пользовательском интерфейсе. При этом выбранный архитектурный подход приложения не имеет никакого значения (важно, чтобы он был одним из разновидностей MV* шаблонов).

Работа будет включать следующие этапы:

1. Создание серверной части WebSocket API
2. Создание клиентской части WebSocket API (JavaScript/Java)

Выполнение данной работы предполагает расширение возможностей существующего web-приложения из предыдущих домашних работ.

Итак, в рамках задания предполагается:

- Для существующей функциональности получения информации о курсе валют и новостях разработать Server Endpoint, который с периодичностью в 5 минут запрашивает информацию у «конечных» сервисов и в случае найденных расхождений от закешированных на сервере производить уведомление клиентской стороны. Периодичность следует вынести как параметризируемое значение, поддающееся изменению. При передаче данных рекомендуется использовать формат JSON.
- Внести изменения клиентского кода страниц/модулей новостей и курса валют для возможности интеграции с серверной частью с целью получения обновленных данных и их последующего отображения для всех подключенных пользователей, без необходимости обновления страниц, используя возможности JS.
- Аналогичный механизм обновления предусмотреть для dashboard статистической информации (адаптировать клиент/сервер).
- Разработать аналогичные веб-сокеты клиенты, используя возможности платформы Java (опционально).

В качестве альтернативного задания предлагается разработать многопользовательскую чат-комнату средствами платформы JavaEE, используя возможности технологии веб-сокетов.