

РЕАЛИЗАЦИЯ СЕРВЕРНОГО И КЛИЕНТСКОГО РЕНДЕРИНГА В WEB ПРИЛОЖЕНИЯХ

Кораблев Денис Павлович

магистрант, Сибирский государственный университет телекоммуникаций и информатики
СибГУТИ, РФ, г. Новосибирск

Кокорева Елена Викторовна

научный руководитель, канд. техн. наук, доцент, Сибирский государственный университет
телекоммуникаций и информатики СибГУТИ, РФ, г. Новосибирск

IMPLEMENTING SERVER AND CLIENT RENDERING IN WEB APPLICATIONS

Denis Korablev

*Magister, Siberian State University of Telecommunications and Information Sciences, Russia,
Novosibirsk*

Elena Kokoreva

*Candidate of Engineering Sciences, Associate Professor, Siberian State University of
Telecommunications and Information Sciences, Russia, Novosibirsk*

Аннотация. В настоящей статье рассматривается вопрос реализации серверного и клиентского рендеринга для Web приложений с клиент-серверной архитектурой.

Abstract. This article discusses the implementation of server and client rendering for Web applications with a client-server architecture.

Ключевые слова: распределенные приложения; клиент-серверная архитектура; тарификация данных АТС; облачные технологии.

Keywords: distributed applications; client-server architecture; PBX data billing; cloud technologies.

Для оптимальной разработки Web приложения важно выбрать правильный подход и место для реализации бизнес-логики и рендеринга клиентского интерфейса. При определении подхода для рендеринга необходимо тщательно изучить возможные варианты, чтобы при удобстве использования интерфейса для пользователя не прогадать с производительностью клиент-серверной системы.

Рассмотрим процесс создание пользовательского интерфейса для следующих ситуаций:

- Традиционный подход с использованием протокола HTTP (серверный рендеринг)
- Двухнаправленный обмен информацией с помощью технологии Web Socket (клиентский рендеринг).

При серверном рендеринге, представленном на рисунке 1, в ответ на HTTP запрос сервер генерирует весь код HTML страницы и берет на себя полную реализацию всей бизнес-логики приложения. Это исключает необходимость дополнительных запросов, данных со стороны клиента, так как сервер берёт всю работу на себя, прежде чем отправить ответ. Такой подход позволяет добиться малого показателя FCP (First Contentful Paint - первая содержательная отрисовка), а выполнение бизнес-логики на сервере позволяют избежать отправки клиенту большого количества данных, что приведет к меньшему времени показателя TTI (Time To Interactive - время до интерактивности).

При данном подходе отсылаются только текст и ссылки, что хорошо работает на широком диапазоне устройств и сетевых условий, а также откроет возможности различных оптимизаций и возможности кеширования информации в браузере пользователя.



Рисунок 1. Серверный рендеринг HTML страницы

При клиентском рендеринге, представленном на рисунке 2, формирование страниц идет прямо в браузере с помощью технологии WebSocket. Вся логика формирования HTML страниц, бизнес-логика, и т.п. обрабатываются на клиенте, а сервер отправляет только сами необходимые данные для Web приложения. При таком рендеринге сложно поддерживать высокую скорость на слабых клиентских устройствах, но появляется возможность организовать двухнаправленный канал связи с использованием собственного двухнаправленного API, что позволяет минимизировать эту проблему.

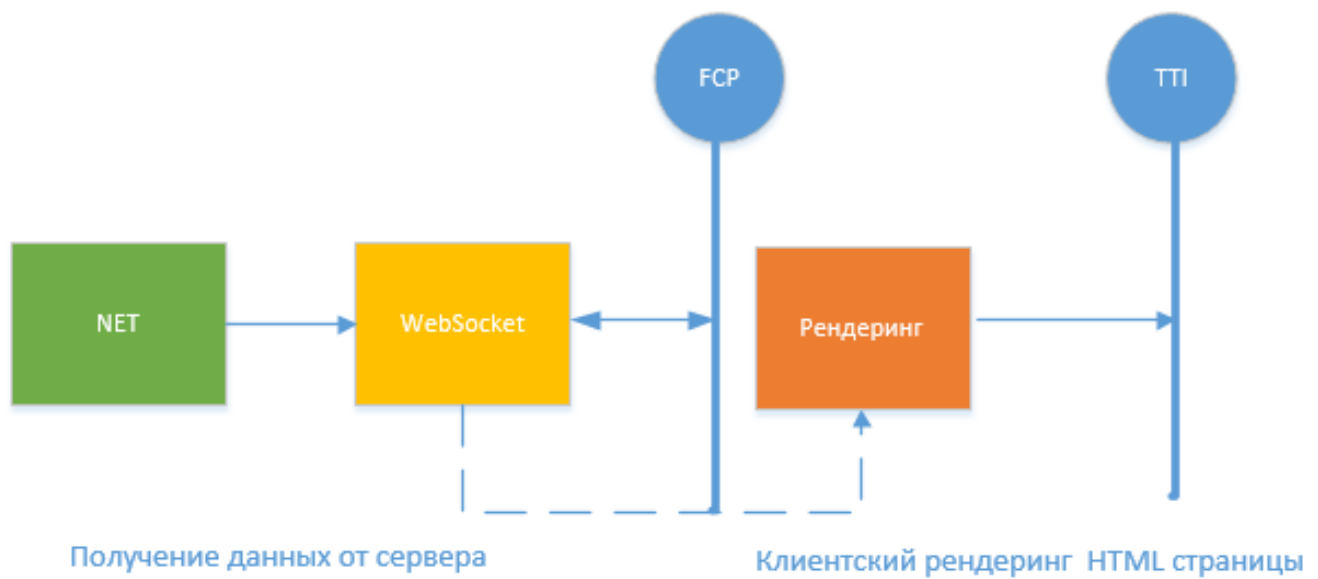


Рисунок 2. Клиентский рендеринг HTML-страницы

Итоговая таблица №1 сравнения вариантов формирования пользовательского интерфейса, представленного на Рисунке 3 и Рисунке 4, представлена ниже:

Таблица 1.

Сравнение преимуществ клиентского и серверного рендеринга

	Показатель FCP	Показатель TTI	Интерактивность участия пользователя
Серверный рендеринг	Зависит от скорости сети	Высокая скорость	Низкая
Клиентский рендеринг	Доступен при загрузке клиента	Более низкая скорость	Высокая

Avaya Monitoring 58500 x +

127.0.0.1:8000/finddate.html?day=13&month=5&year=2019&key=CalledNumber&opkey=0&value=#



Оглавление.

Запрос

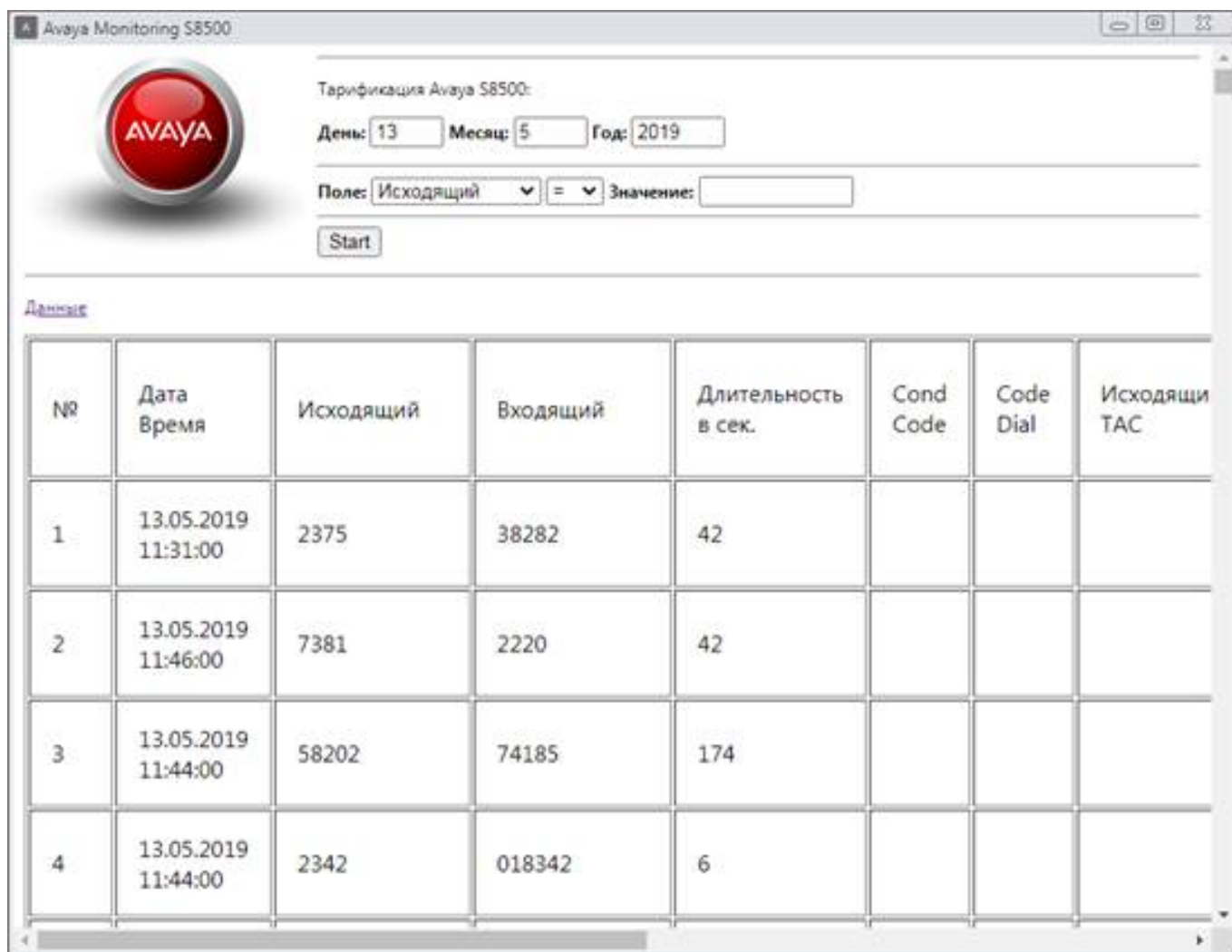
День: Месяц: Год: Поле: = Значение:

Мониторинг. Количество записей: 261.

[таблица](#)

№	Дата Время	Исходящий	Входящий	Длительность в сек.	Cond Code	Code Dial	Исходящий TAC
1	13.05.2019 11:31:00	2375	38282	42			
2	13.05.2019 11:46:00	7381	2220	42			
3	13.05.2019 11:44:00	58202	74185	174			
4	13.05.2019 11:44:00	2342	018342	6			

Рисунок 3 Загрузка пользовательского интерфейса Серверным рендерингом



Avaya Monitoring S8500

Тарификация Avaya S8500:

День: 13 Месяц: 5 Год: 2019

Поле: Исходящий = Значение:

Start

Данные

№	Дата Время	Исходящий	Входящий	Длительность в сек.	Cond Code	Code Dial	Исходящий TAC
1	13.05.2019 11:31:00	2375	38282	42			
2	13.05.2019 11:46:00	7381	2220	42			
3	13.05.2019 11:44:00	58202	74185	174			
4	13.05.2019 11:44:00	2342	018342	6			

Рисунок 4. Загрузка пользовательского интерфейса Клиентским рендерингом с помощью расширения Google Chrome

Список литературы:

1. Разработка интернет-приложений: учеб. пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под науч. ред. Л. Г. Доросинского. — М. : Издательство Юрайт, 2017 ; Екатеринбург : Изд-во Урал. ун-та. — 90 с. — Серия : Университеты России.
2. First Contentful Paint (FCP) [Электронный ресурс]. URL: <https://web.dev/fcp/> (дата обращения: 11.01.2021).
3. Time to Interactive (TTI) [Электронный ресурс]. URL: <https://web.dev/tti/> (дата обращения: 11.01.2021)