

## ОСОБЕННОСТИ УЧЕТА ДАННЫХ ГЕОЛОКАЦИИ ПРИ ПРОЕКТИРОВАНИИ ПРОГРАММНЫХ СРЕДСТВ

**Введение.** В настоящее время для того, чтобы приобрести какой-либо полиграфический продукт, можно посетить соответствующий магазин или же купить в интернет-магазине. В этих случаях необходимо знать где находится магазин. Для решения этой проблемы люди используют веб-картографию, с помощью которой можно найти не только географическое положение магазина, но и построить кратчайший маршрут от текущей геолокации пользователя до магазина.

Данная работа предназначена для изучения действия систем геолокации в веб-приложениях. Все приведенные примеры будут с использованием карт *Google Maps*.

**Основная часть.** При разработке веб-версии интернет магазина был использован серверный фреймворк *Django*, который позволил создать *API* (программный интерфейс приложения), в которых описана вся логика приложения. Для создания пользовательского интерфейса была использована *JavaScript*-библиотека *React.js*.

Основной проблемой при проектировании веб-приложений с геолокацией является поддержание постоянного соединения с сервером. Так как серверный фреймворк *Django* использует *HTTP* протокол, который работает по принципу «клиент-сервер». Например, пользователь запросил информацию о странице, отправив серверу *GET* метод *HTTP* протокола, который передается в специальный заголовок. Таким образом, количество запросов было бы очень велико при получении данных о геолокации. Для решения этой проблемы используется *WebSocket* предназначенный для обмена информацией между клиентом и сервером в режиме реального времени. Протокол *WebSocket* работает над *TCP*, таким образом, при соединении браузер отправляет по *HTTP* специальные заголовки, спрашивая поддерживает ли сервер *WebSocket*. Если поддерживает, то дальше *HTTP* прекращается и общение между клиентом и сервером происходит по протоколу *WebSocket*.

У объекта *WebSocket* есть четыре вызова. Один из них вызывается при получении данных и остальные три при изменениях в состо-

янии соединения. Также у этого объекта есть метод который позволя-ет отправлять данные.

Для того, чтобы получить текущую геолокацию пользователя нужно использовать веб-интерфейс *Navigator*. Интерфейс *Navigator* представляет собой состояние и особенности пользовательского аген-та. Этот интерфейс содержит множество методов, один из них *Navigator Geolocation.geolocation*. Возвращает *Geolocation* объект, позволяющий получить доступ к местоположению. У объекта *Geolocation* есть метод *getCurrentPosition*, который включает в себя два вызова и опции:

–*Success callback* (успешный вызов) — вызов функции которая принимает объект *Geolocation Position* в качестве единственного входного параметра. Этот параметр содержит свойство *coords* которое возвращает объект с данными о широте и долготе.

–*Error callback* (вызов ошибки) — вызов функции которая при-нимает объект *GeolocationPositionError* в качестве единственного входного параметра.

–*Options (опции)* — включает в себя набор параметров *maximumAge* (миллисекунды) максимальное время кэширование пози-ции; *timeout* (миллисекунды) время до вызова ошибки; *enableHighAccuracy* (логический оператор) если *true* эквивалент 1, то-гда будет использоваться высокая точность поиска геопозиции.

Таким образом, зная широту и долготу можно получить теку-щую геопозицию (рисунок 1).

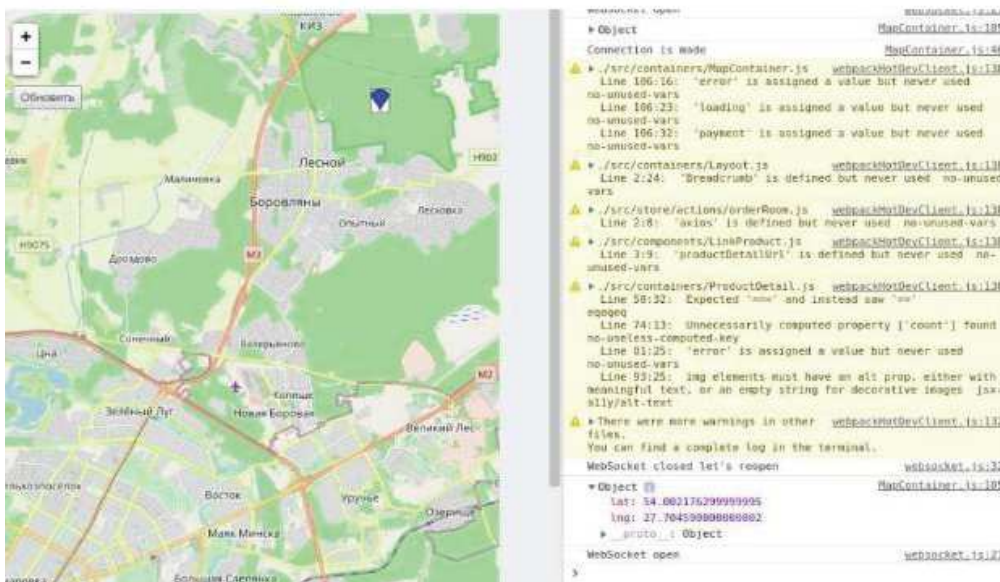


Рисунок 1 – Карта в веб-приложении

