

УДК 004.421

**ОПРЕДЕЛЕНИЕ ТОЧНОГО МЕСТОПОЛОЖЕНИЯ ПОЛЬЗОВАТЕЛЯ САЙТА  
С ИСПОЛЬЗОВАНИЕМ JAVASCRIPT****Н.И. ЛАПКОВСКИЙ**  
(Представлено: Д.В. ПЯТКИН)

Большинство современных устройств способны определять свое местоположение с помощью GPS, WiFi или IP-геолокации. Разработчики могут использовать эту информацию, чтобы предоставить лучшие варианты поиска, расположенные поблизости магазины и реализовать все виды полезных взаимодействий с картами в своих приложениях и на веб-сайтах.

В статье представлен простой способ, основанный на чистом JavaScript, для доступа к местонахождению устройства, не полагаясь на какие-либо внешние зависимости или сторонние сервисы.

**Источники информации о местоположении.** JavaScript предлагает простой, но мощный инструмент для определения местоположения устройств в виде API геолокации. Он состоит из небольшого набора простых в использовании методов, которые могут получить положение устройства с помощью всех трех из упомянутых выше служб:

- GPS – в первую очередь на мобильных устройствах, очень точный (до 10 метров);
- WiFi – доступен на большинстве подключенных к интернету устройств, также очень точный;
- IP-геолокация – ограничена регионом и часто ненадежна, используется как сценарий наихудшего случая, когда два других способа терпят неудачу.

Когда запрашиваются геоданные, браузер пытается использовать все три вышеупомянутых варианта в зависимости от того, что доступно. Результаты из источника WiFi используются чаще всего, поскольку он быстрее, чем GPS, и намного точнее, чем IP-геолокация [1].

**Использование API геолокации.** Geolocation API имеет почти полную межбраузерную поддержку, но чтобы убедиться, что он доступен для наших пользователей, лучше перед тем, как что-либо делать, проверить, существует ли объект геолокации в интерфейсе Window.navigator. Пример представлен в листинге 1.

```
if (navigator.geolocation) {  
    // геолокация возможна  
}  
else {  
    //геолокация не поддерживается  
}
```

Внутри объекта navigator.geolocation находятся все методы для API:

- Geolocation.getCurrentPosition () – определяет текущее местоположение устройства;
- Geolocation.watchPosition () – прослушивает изменения в местоположении и запускает обратный вызов при каждом движении;
- Geolocation.clearWatch () – удаляет обработчик события watchPosition.

**Листинг 1 – Проверка существования объекта геолокации в интерфейсе Window.navigator**

Методы getCurrentPosition () и watchPosition () используются практически одинаково. Они оба работают асинхронно, пытаясь получить положение устройства и в зависимости от результат попытки запустить обратный вызов успешного завершения или обратный вызов ошибки, если она имеется [2]. Реализация метода getCurrentPosition () представлена в листинге 2.

```
navigator.geolocation.getCurrentPosition(  
  
    // Success callback  
    function(position) {  
  
        /* Объект position содержит различную информацию о местоположении устройства:  
        position = {  
            coords: {  
                latitude – Географическая широта в десятичных градусах.  
                longitude – Географическая долгота в десятичных градусах.            }  
        }  
    }  
);
```

```
altitude - Высота в метрах относительно уровня моря.  
accuracy - Возможная погрешность для координат в метрах.  
altitudeAccuracy - Возможная погрешность для высоты в  
метрах.  
heading - Направление устройства в градусах относи-  
тельно севера.  
speed - Скорость устройства в метрах в секунду.  
}  
timestamp - Время, когда местоположение было получено.  
}  
*/  
  
},  
  
// Optional error callback  
function(error) {  
  
    /*  
    В объекте error хранится причина неудачной попытки:  
  
    error = {  
        code - Код ошибки, представляющий её тип  
            1 - PERMISSION_DENIED  
            2 - POSITION_UNAVAILABLE  
            3 - TIMEOUT  
  
        message - Детали об ошибке в читаемом для человека формате.  
    }  
    */  
  
}  
  
);
```

Листинг 2. – Реализация метода `getCurrentPosition()`

Использовать Geolocation API довольно просто — нам просто нужно вызвать правильный метод, подождать, пока он вернет координаты, и затем сделать с ними все, что мы захотим.

**Разрешение пользователя.** Поскольку Geolocation API предоставляет глубоко личную информацию, когда приложение пытается получить к ней доступ в первый раз, появляется диалоговое окно с запросом разрешения. Это гарантирует, что пользователи не будут раскрывать свои личные данные, если они явно не разрешат это (рис. 1).

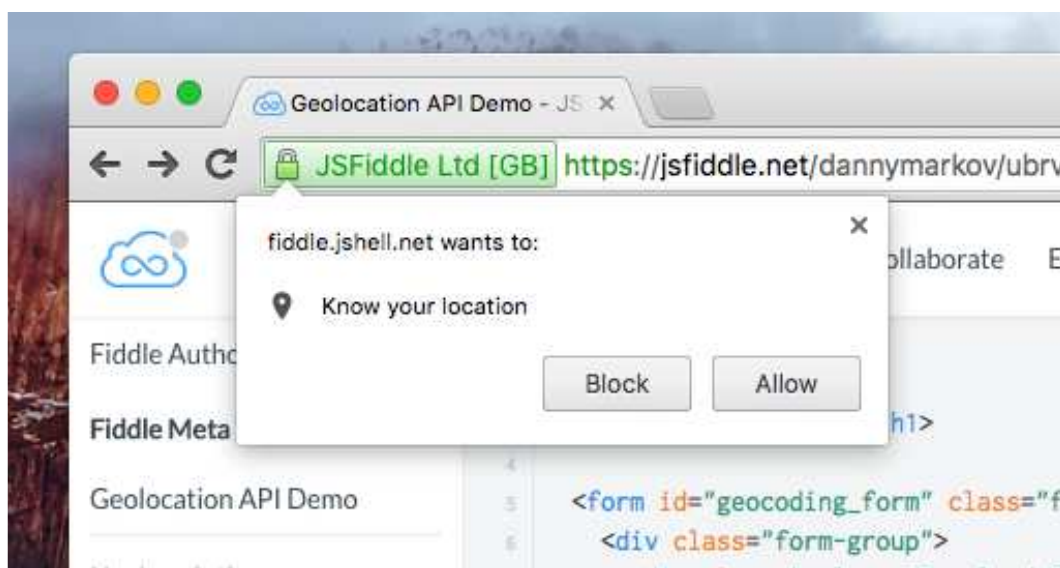


Рисунок 1. – Диалоговое окно с запросом разрешения использования данных геолокации

Обычно браузер сам заботится об отображении диалога, но разрешение также может быть запрошено разработчиком программно. Это иногда необходимо, так как однажды запрещенный исходный диалог, сгенерированный браузером, не появляется во второй раз.

Другая защитная мера - использование HTTPS-соединения. Из-за новой политики веб-безопасности Google Chrome (как для настольных, так и для мобильных версий) больше не позволяет незащищенным хостам запускать API геолокации. Вместо этого разработчики, которые хотят использовать эту функцию, должны обслуживать свои приложения по протоколу HTTPS, таким образом сводя к минимуму риски кражи или злоупотребления данными людей [3].

**Демо приложение.** В листинге 3 показан исходный код простого приложения, демонстрирующего некоторые возможности API. Оно состоит из кнопки, которая при нажатии захватывает координаты устройства и передает их в плагин GMaps, указывая точное местоположение на карте.

```
findMeButton.on('click', function(){

    navigator.geolocation.getCurrentPosition(function(position) {

        // Получаем координаты текущей позиции.
        var lat = position.coords.latitude;
        var lng = position.coords.longitude;

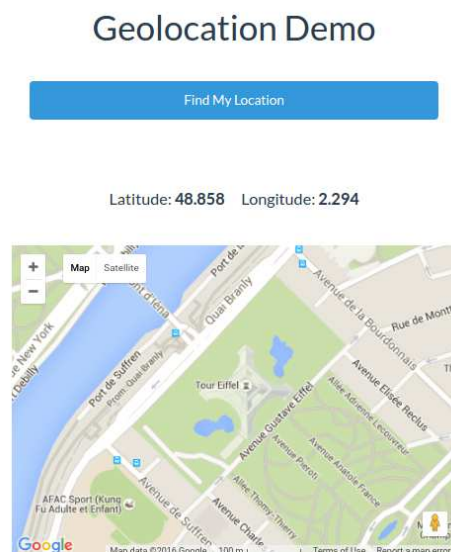
        // Создаём новую карту и устанавливаем маркер на местоположе-
ние устройства.
        var map = new GMaps({
            el: '#map',
            lat: lat,
            lng: lng
        });

        map.addMarker({
            lat: lat,
            lng: lng
        });

    });

});
```

**Листинг 3. – Исходный код простого приложения,  
демонстрирующего некоторые возможности Geolocation API**



**Рисунок 2. – Приложение, демонстрирующее возможности Geolocation API**

**Заключение.** В ходе данного исследования был рассмотрен способ, основанный на чистом JavaScript, для доступа к местонахождению устройства, не полагаясь на какие-либо внешние зависимости или сторонние сервисы. Так же было написано простое веб-приложение, демонстрирующее возможности Geolocation API.

#### ЛИТЕРАТУРА

1. HTML5 Geolocation [Электронный ресурс] – Режим доступа: [https://www.w3schools.com/html/html5\\_geolocation.asp](https://www.w3schools.com/html/html5_geolocation.asp) - Дата доступа: 22.09.2019.
2. Geolocation: Displaying User or Device Position on Maps [Электронный ресурс] – Режим доступа: <https://developers.google.com/maps/documentation/javascript/geolocation> - Дата доступа: 23.09.2019.
3. Geolocation API Removed from Unsecured Origins in Chrome 50 [Электронный ресурс] – Режим доступа: <https://developers.google.com/web/updates/2016/04/geolocation-on-secure-contexts-only> – Дата доступа: 23.09.2019.
4. Introduction: callbacks [Электронный ресурс] – Режим доступа: <https://javascript.info/callbacks> – Дата доступа: 23.09.2019.
5. MDN web docks [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/> - Дата доступа: 23.09.2019.