**Глава 2. ПРОГРАММНЫЙ ИНТЕРФЕЙС GEOLOCATION**

**2.1. Назначение Geolocation API и принципы его работы**

Geolocation API – программный интерфейс, входящий в состав стандарта HTML5 и позволяющий определить местоположение пользователя. Так как эта информация может нарушить конфиденциальность, прежде чем сервер сможет получить данную информацию, пользователь должен явным образом подтвердить свое разрешение. Местоположение задается географическими координатами (широтой и долготой), заданными в градусах.

Процедура определения местоположения встроена в браузер, поэтому Geolocation API представляет собой набор функций, позволяющий обратиться к этому механизму. Каждый тип браузера имеет свою процедуру определения координат, поэтому в общем случае разработчик не может заранее знать каким способом они были получены. Однако, можно перечислить способы, с помощью которых браузер может вычислить координаты пользователя.

**IP-адрес**. Для получения информации о местоположении пользователя может быть использована внешняя база данных IP-адресов, связывающая IP-адрес с его географическим расположением. Например, браузеры Opera, Chrome и FireFox используют для доступа к такой базе данных сервис Google Location Service. Преимуществом этого метода является то, что он работает на всех устройствах, недостатком – местоположение в большинстве случаев может быть определено с точностью до местоположения офиса Интернет-провайдера.

**Global Positioning System (GPS)**. Большинство современных мобильных устройств оснащены оборудованием, позволяющим с помощью спутников GPS достаточно точно определить его местоположение. Главными недостатками этого способа является необходимость устройству «видеть небо», а также низкая скорость его работы – процесс определения координат может занять много времени.

**WiFi**. При WiFi-позиционировании задействуется одна или несколько точек доступа WiFi. Данный способ является быстрым и работает в помещениях, однако обладает тем же недостатком, что метод IP-позиционирования.

**Мобильный телефон**. Если браузер работает в мобильном телефоне, то он может использовать информацию о расстоянии (оценка по уровню сигнала) до нескольких телефонных вышек, координаты которых известны. Недостатком этого метода является то, что он работает только в зоне покрытия сотовой связи.

Если браузер достаточно «умен», то он может использовать несколько способов определения местоположения.

**2.2. Применение Geolocation API**

Рассмотрим пример простого одностраничного web-приложения, демонстрирующего применение Geolocation API. На рис. 2.1 представлен внешний вид страницы этого приложения.

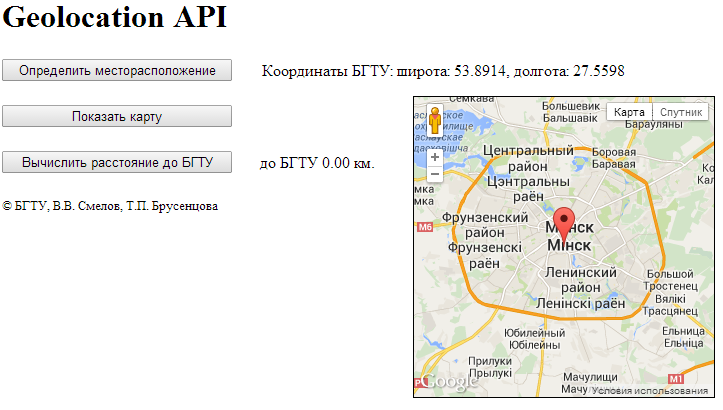
****

Рис. 2.1. Внешний вид страницы web-приложения, применяющего Geolocation API

В начальном состоянии страница отображает географические координаты Белорусского государственного технологического университета (БГТУ); карту с маркером текущего местоположения, сформированных с помощью картографического сервиса Google Maps [7], а также расстояние в километрах до БГТУ.

Приложение управляется с помощью трех клавиш.

При нажатии клавиши «Определить месторасположение» (рис. 2.1.) с помощью Geolocation API осуществляется запрос к браузеру для получения текущих координат. Браузер запрашивает разрешение у пользователя на выполнение этой операции (рис. 2.2) и если оно получено, определяет текущие широту и долготу, а также точность вычисления координат. Соответствующая информация выводится в окно браузера (рис. 2.3).

****

Рис. 2.2. Запрос браузера на разрешение получить

текущие географические координаты

****

Рис. 2.3. При нажатии клавиши «Определить месторасположение» выводится информация о текущем местоположении

При нажатии клавиши «Показать карту» (рис. 2.1.) с помощью Google Maps API в окне браузера отображается карта с маркером, отмечающим место соответствующее текущим координатам (рис. 2.4).

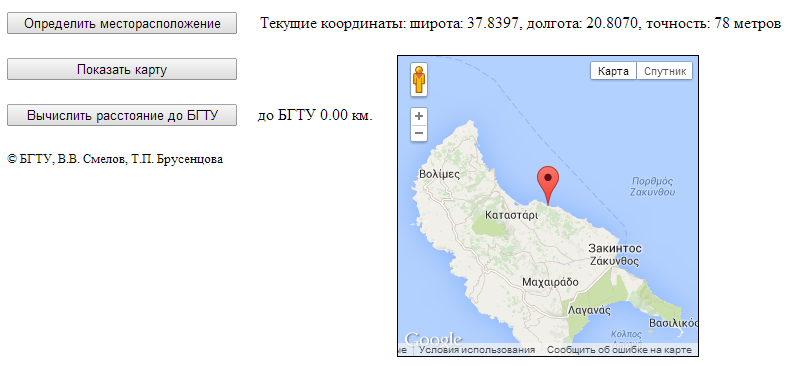
****

Рис. 2.4. При нажатии клавиши «Показать карту» выводится карта с маркером, указывающим место, соответствующее текущим координатам

С помощью клавиши «Вычислить расстояние до БГТУ» осуществляется вычисление расстояния между текущим местоположением и БГТУ и выводится соответствующая информация (рис. 2.5).

****

Рис. 2.5. При нажатии клавиши «Вычислить расстояние до БГТУ» вычисляется и выводится расстояние от текущего местоположения до БГТУ

На рис. 2.6 представлен исходный код html-страницы, внешний вид которой отражен на рис. 2.1. Элементы <section> свернуты. Их содержимое будет рассматриваться ниже.

.

Рис. 2.6. Исходный код html-страницы, представленной на рис. 2.1

В исходном коде на рис. 2.6 применяется три элемента <script>. Первый элемент предназначен для загрузки JavaScript-кода Google Map API. Второй загружает JavaScript-код, используемый на этой странице.

Содержимое последнего, третьего, элемента <script> представлено на рис. 2.7.

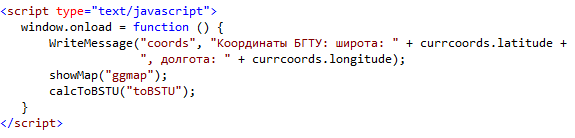


Рис. 2.7. Содержимое третьего элемента <script> (рис. 2.6)

В сценарии на рис. 2.7 задается функция, выполняющаяся при каждой загрузке html-страницы (событие onload объекта window). Функция предназначена для начальной инициализации информации на странице: выводит координаты БГТУ (функция WriteMessage), карту (showMap) и расстояние до БГТУ (calcBSTU).

На рис. 2.8 представлена фрагмент файла Geolocation.js, загружаемый с вторым элементом <script> (рис.2.6). Фрагмент содержит объявление глобальных переменных и функцию WriteMessage, предназначенную для вывода строки в элемент, идентификатор которого задан первым параметром.

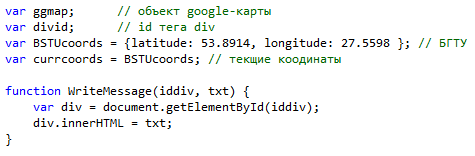
****

Рис. 2.8. Фрагмент файла Geolocation.js: объявление глобальных

переменных и функции WriteMessage

Обратите внимание, как заданы координаты БГТУ – в виде JavaScript-объекта со свойствами latitude (широта в градусах) и longitude (долгота в градусах). При необходимости , координаты объектов на поверхности Земли могут быть получены с помощью сервиса Google Maps (рис. 2.9).

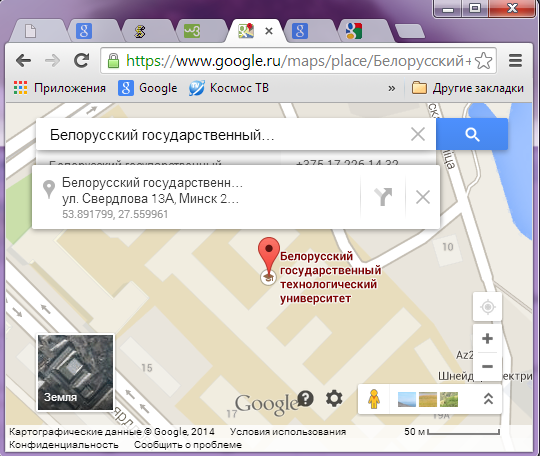
****

Рис. 2.9. Координаты объектов на Земле, можно получить с помощью сервиса Google Maps

**2.2.1. Определение текущих координат**

Вычисление текущих координат осуществляется при нажатии клавиши «Определить месторасположение» (рис. 2.1). Элемент <input type=”button”> отображающий эту клавишу содержится внутри первого элемента <section> (рис. 2.6). Содержимое этой секции приведено на рис.2.10.

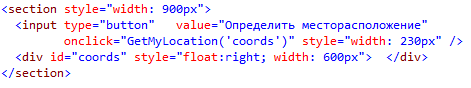


Рис. 2.10. Фрагмент html-страницы web-приложения: cодержимое первого элемента <section> (рис. 2.6)

При нажатии клавиши (рис. 2.10) вызывается функция GetMyLocation, расположенная в файле Geolocation.js и представленная на рис. 2.11.

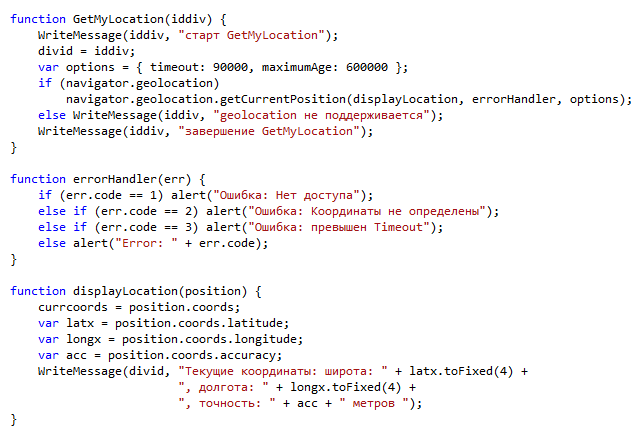


Рис. 2.11. Фрагмент файла Geolocation.js: объявление функции GetMyLocation

Функция GetMyLocation принимает один параметр: идентификатор элемента, который будет содержать результат выполнения функции. Обратите внимание: при вызове функции, ей передается идентификатор элемента <div>, расположенного ниже <input type=”button”> (рис. 2.10).

Собственно получение координат осуществляется с помощью функции navigator.geolocation.GetCurrentPosition (далее для краткости просто GetCurrentPosition ). Функция принимает три параметра: функция, которая будет вызвана при успешном получении координат; функция, которая будет вызвана в случае возникновении ошибки; дополнительные параметры (опции) функции.

При успешном получении координат GetCurrentPosition вызывает функцию displayLocation (заданную первым параметром), принимающую единственный параметр – JavaScript-объект в свойстве coords, которого содержаться координаты: latitude (широта), longitude (долгота), а также accuracy – величину погрешности (рис. 2.11).

Если в процессе выполнении функции GetCurrentPosition возникла ошибка, то вызывается функция ErrorHandler (заданную вторым параметром). Функция ErrorHandler принимает один параметр – JavaScript-объект, в свойстве code которого содержится код ошибки (рис. 2.11).

Третий параметр функции GetCurrentPosition – JavaScript-объект, свойства, которого задают дополнительные параметры функции: timeout – максимальное количество миллисекунд, которые отводятся браузеру на определение координат; maximumAge – максимальный «возраст» координат в миллисекундах (интервал кэширования текущих координат браузером).

Обратите внимание, каким образом в функции GetMyLocation осуществляется проверка работоспособности Geolocation API без применения библиотеки Modernizr (см. 1.8).

**2.2.2. Отображение карты с помощью Google Maps API**

Для отображения карты применяется клавиша «Показать карту» (рис. 2.1). Элемент <input type=”button”> отображающий эту клавишу содержится внутри второго элемента <section> (рис. 2.6). Содержимое этой секции приведено на рис.2.12.

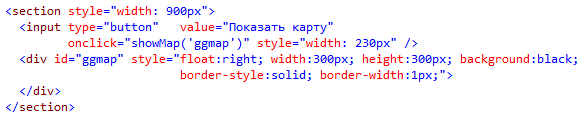
****

Рис. 2.12. Фрагмент htl-страницы web-приложения: cодержимое второго элемента <section> (рис. 2.6)

При нажатии клавиши (рис. 2.12) вызывается функция showMap, расположенная в файле Geolocation.js и представленная на рис. 2.13. Функция showMap принимает два параметра: идентификатор элемента, который будет содержать результат выполнения функции (карту); JavaScript-объект, содержащий координаты точки Земли, отображаемой на карте. Второй параметр не является обязательным, если от отсутствует, то будут использоваться текущие координаты, хранящиеся в глобальной переменной currcoords (рис. 2.8). Обратите внимание: при вызове функции, ей передается один параметр – идентификатор элемента <div>, расположенный ниже <input type=”button”> (рис. 2.12).

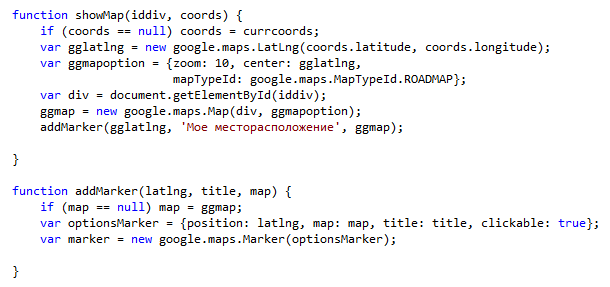
****

Рис. 2.13. Фрагмент файла Geolocation.js: объявление функции showMap

Получение и запись карты в заданный элемент осуществляется с помощью функции google.maps.Map (далее для краткости Map), расположенной в js-файле, загруженным первым элементом <script> на html-странице (рис. 2.6). Функция принимает два параметра: идентификатор элемента (в примере <div>), в теле которого в итоге будет размещена карта; JavaScript-объект, свойства которого содержат дополнительные параметры функции. Объект второго параметра формируется до вызова функции Map и включает следующие свойства: zoom – масштаб, изменяется от 1 до 21, более высокие значения соответствуют большему увеличению; center – координаты центра карты; mapTypeId – тип карты (в примере выбрана карта дорог). Для приведения координат в формат используемый Google Maps API, применяется функция google.maps.LatLng, которая вызывается для формирования значения свойства center.

Отображение на карте маркера выполняется с помощью функции addMarker, которая может быть вызвана, сразу после выполнения функции Map (рис. 2.13).

Функция addMarker принимает три параметра: JavaScript-объект, содержащий координаты маркера; текст, который будет отображаться при наведении курсора на маркер; JavaScript-объект, возвращенный функцией Map.

Непосредственно формирование и отображение маркера осуществляется функцией google.maps.Marker, принимающей один параметр, представляющий собой JavaScript-объект. Объект включает следующие свойства: position – координаты курсора в формате Google Maps API; map – JavaScript-объект, возвращенный функцией Map; title – текст-подсказка; clickable – логическое значение (true или false), разрешающее или запрещающее выполнять клик мышью.

**2.2.3. Вычисление расстояния между двумя точками на Земле**

Для вычисление и отображения расстояния между точкой на Земле, соответствующей текущим координатам и БГТУ применяется клавиша «Вычислить расстояние до БГТУ» (рис. 2.1). Элемент <input type=”button”> отображающий эту клавишу содержится внутри третьего элемента <section> (рис. 2.6). Содержимое этой секции приведено на рис. 2.14.

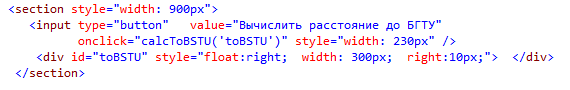


Рис. 2.14. Фрагмент htl-страницы web-приложения: cодержимое третьего элемента <section> (рис. 2.6)

При нажатии клавиши (рис. 2.14) вызывается функция calcToBSTU, расположенная в файле Geolocation.js и представленная на рис. 2.15. Функция calcToBSTU принимает два параметра: идентификатор элемента, который будет содержать результат выполнения функции; координаты удаленной от БГТУ точки. Второй параметр не является обязательным, если он не задан, берутся текущие координаты.

Для вычисления расстояния между двумя заданными точками, функция calcToBSTU вызывает функцию calcDistance, код которой позаимствован авторами в [8].

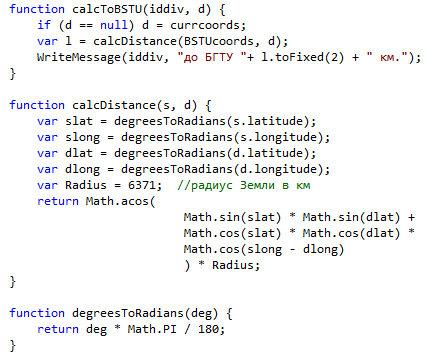


Рис. 2.14. Фрагмент файла Geolocation.js: объявление

функции calcToBSTU

Функция calcDistance принимает два параметра, содержащие координаты двух точек. В результате работы функция к точке вызова возвращает расстояние между двумя заданными точками на Земле, выраженное в километрах. Для перевода градусной меры в радианы функцией calcDistance вызывается вспомогательная функция degressToDadians.