

Web-App, Firefox OS App alles nur ein wenig JavaScript

Google Maps und Tracking

Ein paar wenige Dateien und schon ist die Web-App fertig, oder?

Diese Web-App stellt folgende Funktionen zur Verfügung:

- Google-Maps Kartenansicht mit Umschaltmöglichkeit zur Satellitenbild-Ansicht
- Zentrierung der Karte zum aktuellen Standort
- Aufnehmen der Koordinaten in eine Liste, zur Tracking-Anzeige in der Karte und zum Download als GPS-Datei im GPX Format
- Löschen / Zurücksetzen der aufgenommenen Werte
- Anzeige der aktuellen Werte Longitude, Latitude, Altitude, Speed, Accuracy und Anzahl der eingehenden Werte im Verhältnis zu den aufgenommenen Werten
- ungenaue als auch doppelte Werte werden ignoriert und nicht in die Werteliste aufgenommen



Screenshot der aktiven App

und die erzeugten  [GPX-Daten](#) (473,5 kByte) 30.12.2018 21:29

Es ist noch eine Funktion zur Anzeige aufgenommener Tracks bzw. von anderer Seite zur Verfügung gestellter GPX-Dateien angedacht, des Weiteren einige Eingabefelder zur Beschreibung der GPX-Datei. (ist noch in Arbeit)

Die komplette Definition der App wird in einer manifest Datei abgelegt, zu beachten sind hier besonders die Rechte, je nach Typ stehen auch unter Umständen nicht alle Möglichkeiten zur Verfügung. Unter Firefox OS kann man z.B. nicht beim Typ Web auf die SD-Card zugreifen, also ist später unter Umständen ein anderer Weg einzuplanen, in diesem Fall der Download. Die meisten Punkte erklären sich hoffentlich von alleine.

 [manifest.webapp](#) (1,02 kByte) 22.03.2020 22:35

```
{
  "version": "0.3.0",
  "name": "Gocher Maps Web App",
  "description": "Firefox OS Maps Web App",
  "launch_path": "/index.htm",
  "icons": {
    "16": "/img/icons/icon16x16.png",
    "48": "/img/icons/icon48x48.png",
    "60": "/img/icons/icon60x60.png",
    "128": "/img/icons/icon128x128.png",
    "512": "/img/icons/icon512x512.png"
  }
}
```

```

},
"developer": {
  "name": "Udo Schmal",
  "url": "http://www.gocher.me"
},
"type": "web",
"permissions": {
  "geolocation": {
    "description": "Needed for the app to get positions from the device."
  },
  "device-storage:sdcard": { "access": "readwrite" }
},
"installs_allowed_from": [
  "*"
],
"locales": {
  "en": {
    "description": "Firefox OS Maps Web App",
    "developer": {
      "name": "Udo Schmal",
      "url": "http://www.gocher.me"
    }
  },
  "de": {
    "description": "Firefox OS Maps Web App",
    "developer": {
      "name": "Udo Schmal",
      "url": "http://www.gocher.me"
    }
  }
},
"default_locale": "en"
}

```

In der index.html der eigentlichen Seite werden in dieser App lediglich die benötigten JavaScripts und Stylesheets eingebunden.

 [index.htm](#) HTML (804 Bytes) 06.05.2021 22:54

```

<!DOCTYPE html >
<html>
  <head>
    <meta charset="utf-8" />
    <title>Gocher Maps Web App</title>
    <meta name="description" content="Firefox OS Maps Web App" />
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no,
width=device-width, minimum-scale=1, maximum-scale=1" />
    <link rel="stylesheet" href="app.css" />
    <meta http-equiv="Content-Security-Policy" content="default-src 'self'
https://maps.googleapis.com/; script-src 'self' https://maps.googleapis.com/;
style-src 'self' https://maps.googleapis.com/; img-src 'self' data:
https://maps.googleapis.com/ " />

```

```
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?v=3"></script>
<script type="text/javascript" src="app.js" defer></script>
</head>
<body role="application"></body>
</html>
```

 [app.css](#) StyleSheet (7,94 kByte) 22.03.2020 22:35

```
[role="toolbar"] {height:4rem; width:100%; position:fixed; bottom:0; left:0;
```

```
z-index:100; background:rgba(0,0,0,0.85);}
[role="toolbar"] ul {float:left; list-style:none; padding:0; margin:0;}
[role="toolbar"] ul:last-child {float:right;}
[role="toolbar"] li {float:left;}
[role="toolbar"] button {width:5.5rem; height:4rem; border:none; font-size:0; background:transparent no-repeat 50% 50% / 3rem auto; padding:0; border-radius:0;}
[role="toolbar"] button:active, [role="toolbar"] button.active {background-color:#008aaa;}
[role="toolbar"] .pack-icon-mark {background-image:url();}
[role="toolbar"] .pack-icon-share {background-image:url(
```

9tbS8iIHhtbG5zOnN0UmVmPSJodHRwOi8vb3VhbnMuYWRvYmUuY29tL3hhcC8xLjAvc1R5cGUvUmVzb3VyYVSZlWYjIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iIHhtcE1NOK9yaWdpbFsRG9jdW1lbnRJRd0ieG1wLmRpZDo4NDA1RDQxQUE3MjA2ODExODhDNke3NjAxQjM4MUZDNSIgeG1wT06RG9jdW1lbnRJRd0ieG1wLmRpZDpDOUM5NjE4Q0REODYxMUUxQjVDNEE4QjczREVCRENENCIGeG1wT06SW5zdGFuY2VJRd0ieG1wLm1pZDpDOUM5NjE4QkREODYxMUUxQjVDNEE4QjczREVCRENENCIGeG1wONyZWFO0b3JUB29sPSJBZG9iZSBQaG90b3Nob3AgQ1M1IE1hY2ludG9zaCI+IDx4bXBNTTpEZXXJpdmVkrJvbSBzdFJlZjppbnN0YW5jZU1EPSJ4bXAuaWlkOjk4MjBGQzc5QjkyMDY4MTE4OEM2QjcwQTAWNDE1NQ1IiBzdFJlZjpkb2N1bWVudE1EPSJ4bXAuZGlkOjg0MDVENDFBQTcyMDY4MTE4OEM2QTc2MDFCMzgxRM1Ii8+IDwvcmlRlc2NyaXB0aW9uPiA8L3JkZjpsREY+IDwveDp4bXBtZXRhPiA8P3hwYWNrZXQgZ5kPSJyIj8+14Nt4AAAAWxJREFUeNpi/P//P8NAAsZRB4w6YDA7QAuItwCxAILm/gPiBUBcQokD2IG4BjFyfCUG9QRQgMVBcuA2IvYkGNC47NRwQHOPrIDYl5iHACK8w4apLNWID4KxKKEHLCVzDgnBugC8V5sjB2gCCNcxzIEfvQHcFCoaEqQBwOxIZALaflkCIQ8wDxGagadSTlOlBHOAHxa0iKAeYCKP4AxEuR+PgwfBPBeK//8kDm2FmkRMCBS9B4jNKAi5f5REwSwky/8C8XUgfoVsKBowQSsTLgJxCiLTkhYFFtAgBKktBIBIqLrMFLQXwBiEWR5UkMgAYjvQovbezjU9AOxIXAboImDfO4CxG/wlySEgDEBY0FADIGViLGcnDSwhDl2EAVEJ8H4i/YJFnISICKsgsOkVEaEwhdqF49MFDqIqi2it0DMCsQ7sVSPRJshTWxEly3IaWAuECcDsOFngIVSItGW8WjDhh1ALEAIAAA+H41QW+KcxcAAAAASUVORK5CYII=); }

[role="toolbar"] .pack-icon-move {background-image:

url(); }

[role="toolbar"] .pack-icon-delete {background-image:

url(

```
html, body {width:100%; height:100%;}
html, body, #gmap {margin:0; padding:0; font-family:sans-serif;}
#gmap {position: absolute; top:0; left:0; width:100%; bottom:66px;}
#geoButton {position:absolute; bottom:15px; right:15px; width:40px;
height:40px;}
#status {position:absolute; top:5px; right:5px; width:132px; height:130px;
padding-left:5px; font-size:9pt; line-height:150%; color:white; font-
weight:bold; background-color:black; overflow:hidden; opacity:0.7;}
```

 [app.js](#) JavaScript (9,34 kByte) 22.03.2020 22:34

```

/*jslint indent: 2, white: true, browser: true, devel: true */
/*global navigator,google,Blob,URL */

'use strict';

function App() {
    this.body = null; // html body element
    this.gmap = null; // html div wrapper for google map
    this.stat = null; // html div for status output
    this.btnMark = null; // btml button for start / stop tracking
    this.btnCenter = null; // html button for start / stop center actual
    position
    this.wakeLock = null; // geolocation wakelock
    this.watchID = null; // geolocation watch id
    this.counter = 0; // geolocation watch position counter
    this.map = null; // google map
    this.marker = null; // google marker
    this.poly = null; // google maps polyline to display track
    this.lastPos = {}; // last position to detect changes
    this.marks = []; // marks cache send to map
    this.track = []; // full track
    this.gps = null; // xml-File
}

App.prototype = {
    // start tracking
    start: function () {
        function formatNum (s, n) {
            s = String(s);

```

```

    while (s.length < n) {
        s = "0" + s;
    }
    return s;
}

function formatDeg(n) {
    var deg = Math.floor(n), minutes, seconds, cents;
    n = (n - Math.floor(n)) * 60;
    minutes = Math.floor(n);
    n = (n - Math.floor(n)) * 60;
    seconds = Math.floor(n);
    n = (n - Math.floor(n)) * 100;
    cents = Math.floor(n);
    return deg + "°" + formatNum(minutes, 2) + "'" + formatNum(seconds, 2) +
"." + formatNum(cents, 2) + '"';
}

var self = this;
this.btnMark.className = "pack-icon-mark active";
this.stat.innerHTML = 'get it ...';
this.poly.setMap(this.map);
this.lastPos = {'lat': 0, 'lon': 0};
this.wakeLock = navigator.requestWakeLock('gps');
this.watchID = navigator.geolocation.watchPosition(
    function (pos) {
        var lat, lon, alt, speed, posMark, latlng, path;
        ++self.counter;
        lat = pos.coords.latitude;
        lon = pos.coords.longitude;
        alt = pos.coords.altitude;
        speed = pos.coords.speed;
        if (((self.lastPos.lat !== lat) || (self.lastPos.lon !== lon)) &&
(pos.coords.accuracy < 32)) {
            self.marks.push({'lat': lat, 'lon': lon});
            self.lastPos = {'lat': lat, 'lon': lon, 'alt': alt, 'speed': speed,
'accuracy': pos.coords.accuracy};
            self.track.push({'lat': lat, 'lon': lon, 'alt': alt, 'ts':
pos.timestamp});
        }
        if (!document.hidden && (self.marks.length > 0)) {
            path = self.poly.getPath();
            while (self.marks.length > 0) {
                posMark = self.marks.shift();
                lat = posMark.lat;
                lon = posMark.lon;
                latlng = new google.maps.LatLng(lat, lon);
                path.push(latlng);
            }
            if (latlng) {
                self.marker.setPosition(latlng);
                self.marker.setVisible(true);
                if (self.btnCenter.className === "pack-icon-move active") {

```

```

        self.map.panTo(latlng);
    }
}

self.stat.innerHTML = "lat: " + ((self.lastPos.lat === null) ? "no
lat" : formatDeg(Math.abs(self.lastPos.lat)) + (self.lastPos.lat < 0 ? "S" :
"N")) + "<br />" +
                                "lon: " + ((self.lastPos.lon === null) ? "no lon"
: formatDeg(Math.abs(self.lastPos.lon)) + (self.lastPos.lon < 0 ? "W" : "E"))
+ "<br />" +
                                // Referenzelipsoid WGS84 + Quasigeoid height
(47.5)
                                "alt: " + ((self.lastPos.alt === null) ? "no alt"
: Math.round(self.lastPos.alt - 47.5) + "m NN") + "<br />" +
                                "speed: " + ((self.lastPos.speed !== null && !
isNaN(self.lastPos.speed)) ? (self.lastPos.speed * 3.6).toFixed(0) + "km/h" :
"no speed") + "<br />" +
                                "accuracy: ±" + self.lastPos.accuracy + "m" + "
<br />" +
                                "count: " + self.track.length + "/" +
self.counter;
    },
    function (error) {
        //self.btnMark.className = "pack-icon-mark";
        ++self.counter;
        self.stat.innerHTML = 'error' + '<br />' + 'try to get it ...';
    },
    {
        enableHighAccuracy: true,
        maximumAge: 3000,
        timeout: 3000
    }
);
navigator.vibrate(200);
},
// stop tracking
stop: function () {
    this.btnMark.className = "pack-icon-mark";
    navigator.geolocation.clearWatch(this.watchID);
    this.wakeLock.unlock();
    this.marker.setVisible(false);
},
// clear tracking history
clear: function () {
    var path = this.poly.getPath();
    path.clear();
    this.track = [];
    this.marks = [];
    this.counter = 0;
},
// save track

```



```

save: function () {
    var self = this, dateStr = new Date().toISOString(), gpx = [], blob,
    sdcard, request;
    gpx.push('<?xml version="1.0" encoding="UTF-8" standalone="no" ?>\n');
    gpx.push('<gpx xmlns="http://www.topografix.com/GPX/1/1" version="1.1"
creator="www.gocher.me">\n');
    gpx.push('<metadata><link href="http://www.gocher.me">
<text>gocher.me</text></link>\n');
    gpx.push('<time>' + dateStr + '</time></metadata>\n');
    gpx.push('<trk>\n');
    gpx.push('  <trkseg>\n');
    this.track.forEach(
        function (pos, i) {
            gpx.push('    <trkpt lat="' + pos.lat + '" lon="' + pos.lon + '">' +
                ((pos.alt !== undefined) ? ('<ele>' + pos.alt + '</ele>')
: '') +
                '<time>' + new Date(pos.ts).toISOString() + '</time>' +
                '</trkpt>\n');
        });
    gpx.push('  </trkseg>\n');
    gpx.push('</trk>\n');
    gpx.push('</gpx>\n');
    blob = new Blob(gpx, {'type': 'application/gpx+xml'});
    sdcard = navigator.getDeviceStorage("sdcard");
    request = sdcard.addNamed(blob, 'tracks/' + dateStr.replace('/', '-
').replace(':', '-') + '.gpx');
    request.onsuccess = function () {
        var name = this.result;
        alert('File "' + name + '" successfully wrote on the sdcard storage
area');
    };
    // An error typically occur if a file with the same name already exist
    request.onerror = function () {
        if (this.error.name === 'SecurityError') {
            // fallback download
            blob.name = dateStr.replace('/', '-').replace(':', '-') + '.gpx';
            self.stat.appendChild(document.createElement('br'));
            var elem = document.createElement('a'),
                gpxUrl = URL.createObjectURL(blob);
            elem.setAttribute('href', gpxUrl);
            elem.setAttribute('download', blob.name);
            self.stat.appendChild(elem);
            elem.appendChild(document.createTextNode('download'));
        } else {
            alert('Unable to write the file: ' + this.error.name);
        }
    };
},
// display toolbar
addToolbar: function () {
    function addButton(className, caption, ul) {

```



```

    var li, btn;
    li = document.createElement('li');
    ul.appendChild(li);
    btn = document.createElement('button');
    li.appendChild(btn);
    btn.className = className;
    btn.appendChild(document.createTextNode(caption));
    return btn;
}

var self = this, toolbar, ul, btnDelete, btnShare;
toolbar = document.createElement('div');
toolbar.setAttribute('role', "toolbar");
this.body.appendChild(toolbar);
ul = document.createElement('ul');
toolbar.appendChild(ul);
btnDelete = addButton("pack-icon-delete", "Delete", ul);
btnDelete.onclick = function () {
    self.clear();
};
ul = document.createElement('ul');
toolbar.appendChild(ul);
this.btnMark = addButton("pack-icon-mark", "Mark", ul);
this.btnMark.onclick = function () {
    if (self.btnMark.className === "pack-icon-mark") {
        self.start();
    } else {
        self.stop();
    }
};
this.btnCenter = addButton("pack-icon-move", "Move", ul);
this.btnCenter.onclick = function () {
    if (self.btnCenter.className === "pack-icon-move") {
        self.btnCenter.className = "pack-icon-move active";
    } else {
        self.btnCenter.className = "pack-icon-move";
    }
};
btnShare = addButton("pack-icon-share", "Share", ul);
btnShare.onclick = function () {
    self.save();
};
},
// initialize app
init: function () {
    this.body = document.getElementsByTagName('body')[0];
    this.gmap = document.createElement('div');
    this.gmap.setAttribute('id', 'gmap');
    this.body.appendChild(this.gmap);

    this.stat = document.createElement("div");
    this.stat.setAttribute('id', 'status');

```

```

this.body.appendChild(this.stat);
this.addToolbar();
// Google Maps
this.map = new google.maps.Map(
    this.gmap, {
        zoom: 17,
        zoomControl: false,
        streetViewControl: false,
        scrollwheel: false,
        mapTypeControl: true,
        keyboardShortcuts: false,
        mapMaker: false,
        noClear: true,
        overviewMapControl: false,
        rotateControl: false,
        disableDefaultUI: true,
        center: new google.maps.LatLng(51.528710, 6.289250),
        mapTypeId: google.maps.MapTypeId.TERRAIN
    }
);
var symbol = {
    path: 'M0 0 a4 4 0 1 1 0 0.0001 z',
    fillColor: 'red',
    fillOpacity: 0.6,
    scale: 1,
    strokeColor: 'black',
    strokeWeight: 1
};
this.marker = new google.maps.Marker({
    position: new google.maps.LatLng(0, 0),
    map: this.map,
    icon: symbol
});
this.poly = new google.maps.Polyline({
    strokeColor: '#FF0000',
    strokeOpacity: 0.5,
    strokeWeight: 3
});
this.start();
}
};

window.addEventListener('DOMContentLoaded', function() {
    var app = new App();
    app.init();
});

```