

# PUBLICLOOP (v0.4.0)

Vývoj skvělé, neuvěřitelné, úžasné a spektakulární aplikace pro vyhledávání spojení mezi bodem A a B ve veřejném prostoru krok za krokem.

## Navigace na stránce Mapa

Na stránku {publicloop\_root}/app/frondend/map.html si přidáme responzivní navigační pruh. Pro jednoduchost zvolíme komponentu Navbar z frameworku Bootstrap (<https://getbootstrap.com/docs/5.3/components/navbar/>).

```
<nav class="navbar navbar-dark bg-dark fixed-top">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">PUBLICLOOP</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="offcanvas"
      data-bs-target="#offcanvasDarkNavbar" aria-controls="offcanvasDarkNavbar"
      aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="offcanvas offcanvas-end text-bg-dark" tabindex="-1"
      id="offcanvasDarkNavbar" aria-labelledby="offcanvasDarkNavbarLabel">
      <div class="offcanvas-header">
        <h5 class="offcanvas-title" id="offcanvasDarkNavbarLabel">PUBLICLOOP</h5>
        <button type="button" class="btn-close btn-close-white"
          data-bs-dismiss="offcanvas" aria-label="Close"></button>
      </div>
      <div class="offcanvas-body">
        <ul class="navbar-nav justify-content-end flex-grow-1 pe-3">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="#"
              id="signOut">Sign out</a>
          </li>
        </ul>
      </div>
    </div>
  </div>
</nav>
```

## Práce s mapovým podkladem OpenLayer

Pro práci s mapovými podklady použijeme knihovnu OpenLayer <https://openlayers.org/>. Nejdříve si přidáme příslušnou knihovnu do hlavičky souboru {publicloop\_root}/app/frondend/map.html.

```
<head>
  ...
```

```
<script src="https://cdn.jsdelivr.net/npm/ol@v9.0.0/dist/ol.js"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/ol@v9.0.0/ol.css">
...
</head>
```

Dále si do souboru {publicloop\_root}/app/frondend/map.html vložíme element map, do kterého budeme vykreslovat mapu.

```
<div id="map"></div>
```

Element map si nastylujeme.

```
<style>
  html,
  body {
    margin: 0;
    height: 100%;
  }
  #map {
    position: absolute;
    top: 56px;
    bottom: 0;
    width: 100%;
  }
</style>
```

Dále si přidáme skript, který nám zajistí zobrazení mapy.

```
<script type="text/javascript">
  var lat = 15.767525;
  var lng = 50.033570;
  var zoom = 13;

  var view = new ol.View({
    center: ol.proj.fromLonLat([lat, lng]),
    zoom: zoom
  });

  var layers = [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ];

  var map = new ol.Map({
    target: 'map',
    layers: layers,
    view: view
```

```
});  
</script>
```

Stáhneme si mapovou značku z [https://www.iconfinder.com/icons/299087/marker\\_map\\_icon](https://www.iconfinder.com/icons/299087/marker_map_icon) a uložíme si ji do adresáře {publicloop\_root}/app/frondend/images/marker.png. Značku si zobrazíme na mapě na pozici naší fakulty [50.033570, 15.767525].

```
<script type="text/javascript">  
  //markers  
  var iconStyle = new ol.style.Style({  
    image: new ol.style.Icon({  
      anchor: [0.5, 1],  
      src: 'images/marker.png'  
    })  
  });  
  
  var labelStyle = new ol.style.Style({  
    text: new ol.style.Text({  
      font: '16px Montserrat,sans-serif',  
      overflow: true,  
      fill: new ol.style.Fill({  
        color: '#000'  
      }),  
      stroke: new ol.style.Stroke({  
        color: '#fff',  
        width: 3  
      }),  
      offsetY: -60  
    })  
  });  
  
  var style = [iconStyle, labelStyle];  
  
  //marker  
  var markers = new ol.layer.Vector({  
    source: new ol.source.Vector(),  
    style: function(feature) {  
      labelStyle.getText().setText(feature.get('name'));  
      return style;  
    },  
    minZoom: 9,  
  });  
  
  markers.setZIndex( 1004 );  
  map.addLayer(markers);  
  
  var lat_marker = 15.767525;  
  var lng_marker = 50.033570;  
  
  var marker = new ol.Feature({  
    geometry: new ol.geom.Point(ol.proj.fromLonLat([ lat_marker, lng_marker])),  
    name: 'FEI',  
  });
```

```
markers.getSource().addFeature(marker);
</script>
```

Přidáme si funkcionalitu, která nám zajistí vykreslení aktuální polohy. K tomu je nutné, aby klient povolil ve svém webovém prohlížeči API určování polohy. Mapová značka aktuální poloha je modrá tečka.

```
<script>
  //current point
  var geolocation = new ol.Geolocation({
    trackingOptions: {
      enableHighAccuracy: true
    },
    projection: view.getProjection()
  });

  geolocation.setTracking(true);

  // update the HTML page when the position changes.
  geolocation.on('change', function() {
    //console.log(geolocation.getPosition());
  });

  geolocation.on('error', function(error) {
    //console.log(error.message);
  });

  var accuracyFeature = new ol.Feature();
  accuracyFeature.setStyle(new ol.style.Style({
  }));

  geolocation.on('change:accuracyGeometry', function() {
    accuracyFeature.setGeometry(geolocation.getAccuracyGeometry());
  });

  var positionFeature = new ol.Feature();
  positionFeature.setStyle(new ol.style.Style({
    image: new ol.style.Circle({
      radius: 6,
      fill: new ol.style.Fill({
        color: '#3399CC'
      }),
      stroke: new ol.style.Stroke({
        color: '#fff',
        width: 2
      })
    })
  }));

  geolocation.on('change:position', function() {
    var coordinates = geolocation.getPosition();
    positionFeature.setGeometry(coordinates ?
      new ol.geom.Point(coordinates) : null);
  });
</script>
```

```

    new ol.layer.Vector({
      map: map,
      source: new ol.source.Vector({
        features: [accuracyFeature, positionFeature]
      })
    });
  </script>

```

## Plánování trasy

Do menu si přidáme tlačítko pro zadání plánované trasy.

```

...
<li class="nav-item">
  <a class="nav-link active" aria-current="page" href="#" id="planJourney">Plan journey</a>
</li>
...

```

Pro tlačítko Plan journey přidáme akci, která zobrazí modální okno.

```

<script>
  document.getElementById("planJourney").addEventListener('click', function (event) {
    event.preventDefault();

    document.querySelector("#offcanvasDarkNavbar > div.offcanvas-header > button")
      .dispatchEvent(new Event('click'));

    let planModal = new bootstrap.Modal(document.getElementById('planJourneyModal'), {});
    planModal.show();
  })
</script>

```

Do zdrojového kódu si přidáme modální okno frameworku Bootstrap

<https://getbootstrap.com/docs/5.3/components/modal/>.

```

<div class="modal fade" id="planJourneyModal" tabindex="-1" aria-labelledby="exampleModalLabel"
  aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">Plan Journey</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
        </button>
      </div>
      <div class="modal-body">
        ...
      </div>
    </div>
  </div>
</div>

```

```

        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-primary" id="findJourney">Find</button>
        </div>
    </div>
</div>
</div>

```

Do body modálního okna si přidáme formulář s položkami:

- From
- To
- Depart at
- Arrive at

```

<form>
<div class="form-floating mb-3">
    <input type="text" class="form-control form-control-sm" id="fromInput"
        placeholder="Find a place ...">
    <label for="fromInput">From</label>
</div>
<div class="form-floating mb-3">
    <input type="text" class="form-control form-control-sm" id="toInput"
        placeholder="Find a place ...">
    <label for="toInput">To</label>
</div>

<ul class="nav nav-tabs" id="myTab" role="tablist">
    <li class="nav-item" role="presentation">
        <button class="nav-link active" id="depart-tab" data-bs-toggle="tab"
            data-bs-target="#depart-tab-pane" type="button" role="tab"
            aria-controls="depart-tab-pane" aria-selected="true">Depart at</button>
    </li>
    <li class="nav-item" role="presentation">
        <button class="nav-link" id="arrive-tab" data-bs-toggle="tab"
            data-bs-target="#arrive-tab-pane" type="button" role="tab"
            aria-controls="arrive-tab-pane" aria-selected="false">Arrive at</button>
    </li>
</ul>
<div class="tab-content" id="myTabContent">
    <div class="tab-pane fade show active pt-3" id="depart-tab-pane"
        role="tabpanel" aria-labelledby="depart-tab" tabindex="0">
        <div class="form-floating mb-3">
            <input type="datetime-local" class="form-control form-control-sm"
                id="dateDepartInput" placeholder="...">
            <label for="dateDepartInput">Date</label>
        </div>
    </div>
    <div class="tab-pane fade" id="arrive-tab-pane" role="tabpanel"
        aria-labelledby="arrive-tab" tabindex="0">
        <div class="tab-pane fade show active pt-3" id="arrive-tab-pane"
            role="tabpanel" aria-labelledby="arrive-tab" tabindex="0">
            <div class="form-floating mb-3">

```

```

        <input type="datetime-local" class="form-control form-control-sm"
            id="dateArriveInput" placeholder="...">
        <label for="dateArriveInput">Date</label>
    </div>
</div>
</div>
</div>
</form>

```

Položky dateDepartInput a dateArriveInput vyplníme aktuálním časem.

```

<script>
    function setCurrentDatetime(id_element) {
        var now = new Date();
        now.setMinutes(now.getMinutes() - now.getTimezoneOffset());
        document.getElementById(id_element).value = now.toISOString().slice(0,16);
    }

    setCurrentDatetime('dateDepartInput');
    setCurrentDatetime('dateArriveInput');
</script>

```

Jako poslední si přidáme akci pro tlačítko find.

1. zavřít modální okno
2. zobrazit loader
3. načíst hodnoty formuláře a zaslat je na /journey/find
4. po obdržení odpovědi skryt loader a zobrazit výsledek

```

<script>
    document.getElementById("findJourney").addEventListener('click', function (event) {
        event.preventDefault();

        bootstrap.Modal.getInstance(document.getElementById('planJourneyModal')).hide();

        document.getElementById("loader").style.display = 'block';

        let from = document.getElementById("fromInput").value;
        let to = document.getElementById("toInput").value;
        let depart = document.getElementById("dateDepartInput").value;
        let arrive = document.getElementById("dateArriveInput").value;

        let plan_data = {
            from: from,
            to: to,
            depart: depart,
            arrive: arrive,
            active: "depart"
        }

        if (document.querySelector("#myTab li button.active").id == "arrive-tab") {

```

```

    plan_data['active'] = "arrive";
  }

  fetch('http://localhost/backend/journey/find', {
    method: "POST",
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify(plan_data)
  })
    .then((response) => {
      if (!response.ok) {
        throw new Error(`HTTP error, status = ${response.status}`);
      }
      return response.text();
    })
    .then((text) => {
      document.getElementById("loader").style.display = 'none';
      //TODO
      console.log(text);
    })
    .catch(error => {
      //TODO handle error
    })
  })
</script>

```

Na backendu přidáme v souboru {publicloop\_root}/app/backend/index.js routu /journey/find.

```

case '/journey/find':
  res.statusCode = 200;
  res.setHeader('Content-Type', 'application/json');
  res.end(JSON.stringify({journey: "TODO"}));
break;

```