

PUBLICLOOP (v0.1.0)

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

Vývojový stack

V prvním kroku si sestavíme vývojovou platformu, na které vývoj aplikace postavíme. Celý stack bude postaven s ohledem na skutečnost, že na učebnách jsou PC s OS Windows s omezenými právy zápisu. Platformu postavíme co možná nejvíce portable.

V konfiguraci jednotlivých částí platformy budeme používat proměnnou {publicloop_root}, která bude označovat cestu souborového systému k výchozímu bodu platformy (např. "/temp/publicloop/").

Webový server Apache

Binárku webového serveru Apache si stáhneme z <https://www.apachelounge.com/download/>. Archiv rozbalíme do adresáře {publicloop_root}/apache. Upravíme konfigurační soubor {publicloop_root}/apache/conf/httpd.conf.

```
Define SRVROOT "/temp/publicloop/apache"
ProxyPass /backend http://localhost:3000/

DocumentRoot "${SRVROOT}/../app/frondend"
<Directory "${SRVROOT}/../app/frondend">
    ...
</Directory>

LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Vytvoříme si adresáře {publicloop_root}/app/frondend a {publicloop_root}/app/backend, které budou sloužit k uložení zdrojových kódů projektu.

Nyní si můžeme otestovat funkčnost webového serveru, který si spustíme z příkazového řádku.

```
{publicloop_root}/apache/bin/httpd.exe
```

Ukončit webový server můžeme pomocí klávesové zkratky Ctrl + C, případně otevřeme novou konzoli a běžící proces ukončíme příkazem taskkill.

```
taskkill /f /im "httpd.exe"
```

Pro spuštění webového serveru si vytvoříme dávku {publicloop_root}/start-apache.bat.

```
@echo off
echo Apache is trying to start - please wait ...

apache\bin\httpd

if errorlevel 0 goto finish
if errorlevel 1 goto error
goto finish

:error
echo.
echo Apache could not be started
pause

:finish
pause
```

Pro zastavení webového serveru si vytvoříme dávku {publicloop_root}/stop-apache.bat.

```
@echo off
taskkill /F /IM "httpd.exe"
```

Databázový server MariaDB

Binárku databázového serveru MariaDB si stáhneme z <https://mariadb.org/download/>. Archiv rozbalíme do adresáře {publicloop_root}/mariadb. Otevřeme si konzoli a z příkazové řádky provedeme inicializaci DB

```
{publicloop_root}/mariadb/bin/mysql_install_db.exe --datadir=..\data
```

Otevřeme konzoli s příkazovým řádkem a spustíme databázový server.

```
{publicloop_root}/mariadb/bin/mysqld.exe --console
```

Ukončit databázový server můžeme pomocí klávesové zkratky Ctrl + C, případně otevřeme novou konzoli a běžící proces ukončíme pomocí příkazu taskkill.

```
taskkill /f /im "mysqld.exe"
```

Pro spuštění databázového serveru si vytvoříme dávku {publicloop_root}/start-mariadb.bat.

```
@echo off
echo MariaDb is trying to start - please wait ...

mariadb\bin\mysqld --no-defaults

if errorlevel 0 goto finish
if errorlevel 1 goto error
goto finish

:error
echo.
echo MariaDb could not be started
pause

:finish
pause
```

Pro zastavení databázového serveru si vytvoříme dávku {publicloop_root}/stop-mariadb.bat.

```
@echo off
taskkill /F /IM "mysqld.exe"
```

Runtime JavaScript prostředí node.js

Binárku node.js si stáhneme z <https://nodejs.org/en/download>. Provedeme instalaci node.js spolu se správcem balíčků npm. Ověříme si instalaci verze node.js.

```
{publicloop_root}/node -v
```

Ověříme si instalaci správce balíčků npm.

```
{publicloop_root}/npm -v
```

IDE Webstorm

Binárku si stáhneme z <https://www.jetbrains.com/webstorm/> (Free 30-day trial).

Studenti UPCE se přihlásí do účtu JetBrains na adrese <https://account.jetbrains.com/login> (s univerzitním emailen). V účtu si mohou stáhnout instalaci WebStormu včetně přidělené roční licence.

Provedeme instalaci. Po instalaci si nástroj spustíme a rozšíříme o zásuvný modul [Database Tool and SQL for Webstorm](#) (File->Settings->Plugins).

Data modeler MySQL Workbench

Binárku si stáhneme z <https://dev.mysql.com/downloads/workbench/>. Otevřeme konzoli s příkazovým řádkem a provedeme instalaci.

```
msiexec /a mysql-workbench-community-8.0.36-winx64.msi /qb TARGETDIR=H:\
```

Nástroj spustíme souborem MySQLWorkbench.exe.

Projekt Publicloop

Ve WebStormu si založíme nový projekt, který umístíme do adresáře {publicloop_root}/app. V projektu již by měly být vytvořené dva adresáře:

- {publicloop_root}/app/frondend
- {publicloop_root}/app/backend

Fronddend

V adresáři publicloop_root}/app/frondend si vytvoříme výchozí publikační soubor {publicloop_root}/app/frondend/index.html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>PublicLoop</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/styles.css">
</head>
<body>
  <h1>Publicloop</h1>
  <script src="js/app.js"></script>
</body>
</html>
```

Dále si vytvoříme dva prázdné soubory:

- {publicloop_root}/app/frondend/css/styles.css
- {publicloop_root}/app/frondend/js/app.css

Pro aplikaci si vytvoříme logo. Kreativité se meze nekladou :-). Logo si uložíme do adresáře {publicloop_root}/app/frondend/images/publicloop_logo.svg.

Jako logo aplikace lze použít volně dostupnou ikonu, například https://www.iconfinder.com/icons/7225807/arrow_infinity_icon.

Pro aplikaci si vytvoříme favicon a umístíme do adresáře {publicloop_root}/app/frondend.

K vytvoření faviconu můžeme použít například nástroj <https://favicon.io/>.

Pro tvorbu frondendu aplikace využijeme framework Bootstrap, který si stáhneme z <https://getbootstrap.com/docs/5.3/getting-started/download/>. Do adresáře {publicloop_root}/app/frondend/css si nahrajeme soubor bootstrap.css a do adresáře {publicloop_root}/app/frondend/js si nahrajeme soubor bootstrap.js.

Do souboru {publicloop_root}/app/frondend/index.html přidáme podporu Bootstrapu.

```
...  
<link rel="stylesheet" href="css/bootstrap.css">  
...  
<script src="js/bootstrap.js"></script>  
...
```

V aplikaci si vytvoříme 3 nové html soubory:

- {publicloop_root}/app/frondend/signin.html
- {publicloop_root}/app/frondend/signup.html
- {publicloop_root}/app/frondend/map.html

Všechny soubory budou mít stejný zdrojový kód jako soubor {publicloop_root}/app/frondend/index.html. Pouze v elementu title změníme název titulků:

- Sign in ({publicloop_root}/app/frondend/signin.html)
- Sign up ({publicloop_root}/app/frondend/signup.html)
- Map ({publicloop_root}/app/frondend/map.html)

Upravíme zdrojové kódy stránek s využitím příkladů Bootstrapu:

- <https://getbootstrap.com/docs/5.3/examples/cover/> - {publicloop_root}/app/frondend/index.html
- <https://getbootstrap.com/docs/5.3/examples/heroies/> (blok Vertically centered hero sign-up form) - {publicloop_root}/app/frondend/signin.html, {publicloop_root}/app/frondend/signup.html

Konkrétní úpravy viz příložené zdrojové kódy.

Backend

V adresáři {publicloop_root}/app/backend si spustíme příkazovou řádku a z ní vytvoříme node.js projekt.

```
npm init
```

V jednotlivých krocích inicializace projektu doplňujeme požadovaná data. Na konci inicializace projektu se vytvoří soubor package.json.

Soubor package.json můžeme dle potřeby manuálně upravovat (viz <https://docs.npmjs.com/cli/v10/configuring-npm/package-json>).

V adresáři {publicloop_root}/app/backend si vytvoříme soubor {publicloop_root}/app/backend/index.js, který bude představovat endpoint pro náš backend aplikace.

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello from PublicLoop (index.js)');
});
server.listen(port, hostname, () => {
  console.log(`PublicLoop running at http://${hostname}:${port}/`);
});
```

Backend projektu spustíme z příkazové řádky.

```
{publicloop_root}/app/backend/node index
```

Backend projektu zastavíme klávesovou zkratkou Ctrl + C nebo použijeme příkaz taskkill.

```
{publicloop_root}/app/backend/taskkill /f /im "node.exe"
```

Pro spuštění backendu si vytvoříme dávku {publicloop_root}/start-backend-app.bat.

```
@echo off
echo PublicLoop is trying to start - please wait ...

node app\backend\index.js

if errorlevel 0 goto finish
if errorlevel 1 goto error
goto finish

:error
echo.
echo PublicLoop could not be started
pause
```

```
:finish  
pause
```

Pro zastavení backendu si vytvoříme dávku {publicloop_root}/stop-backend-app.bat.

```
@echo off  
taskkill /F /IM "node.exe"
```

Pro otestování funkčnosti si do souboru {publicloop_root}/app/frondend/map.html přidáme následující zdrojový kód.

```
<script>  
  const myHeaders = new Headers();  
  myHeaders.append("Accept", "plain/text");  
  const myOptions = {  
    method: "GET",  
    headers: myHeaders  
  };  
  const myRequest = new Request("http://localhost/backend", myOptions);  
  fetch(myRequest)  
    .then((response) => {  
      if (!response.ok) {  
        throw new Error(`HTTP error, status = ${response.status}`);  
      }  
      return response.text();  
    })  
    .then((text) => {  
      var p = document.createElement("p");  
      p.appendChild(document.createTextNode(`${text}`));  
      document.body.innerHTML = "";  
      document.body.appendChild(p);  
    })  
    .catch((error) => {  
      var p = document.createElement("p");  
      p.appendChild(document.createTextNode(`Error: ${error.message}`));  
      document.body.innerHTML = "";  
      document.body.appendChild(p);  
    })  
</script>
```

Alternativně můžeme na backendu použít framework Express (<https://expressjs.com/>). Nejdříve si framework do projektu přidáme.

```
npm install express
```

Poté můžeme přepsat zdrojový kód souboru {publicloop_root}/app/backend/index.js s použitím frameworku Express.

```
const express = require('express');
const hostname = '127.0.0.1';
const port = 3000;

const app = express();
app.get('/',(req,res)=>res.send('Hello from PublicLoop (index.js)'))
app.listen(3000, ()=>console.log(`PublicLoop running at http://${hostname}:${port}/`))
```