

Pokročilý Leaflet

Cvičení 9

**Webová kartografie –
úvod**

Podzim 2024

Filip Leitner

Naučíme se:

- zobrazovat v mapě **externí data** – GeoJSON
- načítat externí data – **asynchronní JavaScript**
- používat v Leafletu **pluginy**
- zobrazovat **velké objemy dat** v Leafletu – clustering, vlastní ikony

Zápis funkce

Běžný zápis:

```
function (data) {  
  return data.user;  
}  
  
function (a, b) {  
  return a + b;  
}
```

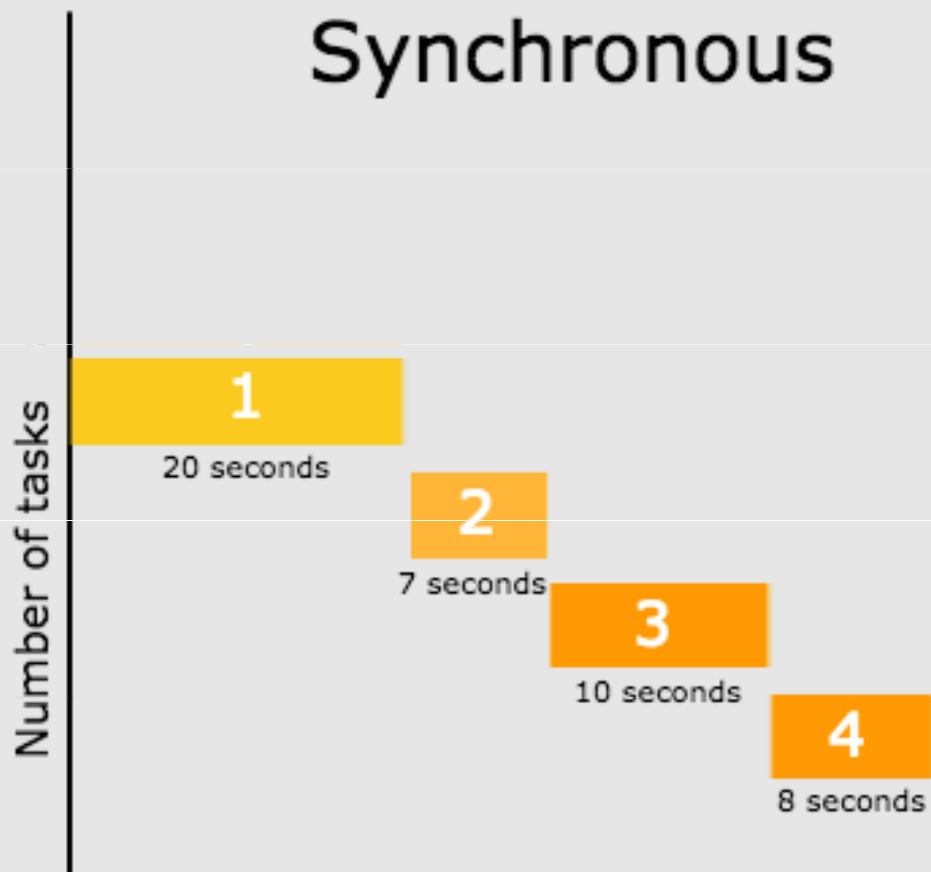
ES6+:

```
(data) => {  
  return data.user;  
}  
  
(a, b) => {  
  return a + b;  
}  
  
// nebo jednoduše:  
data => data.user;  
  
(a, b) => a + b;
```

- pokud má funkce pouze jeden argument, můžeme vypustit závorky
- pokud vrátíme pouze jednoduchý výraz, můžeme vypustit závorky i klíčové slovo

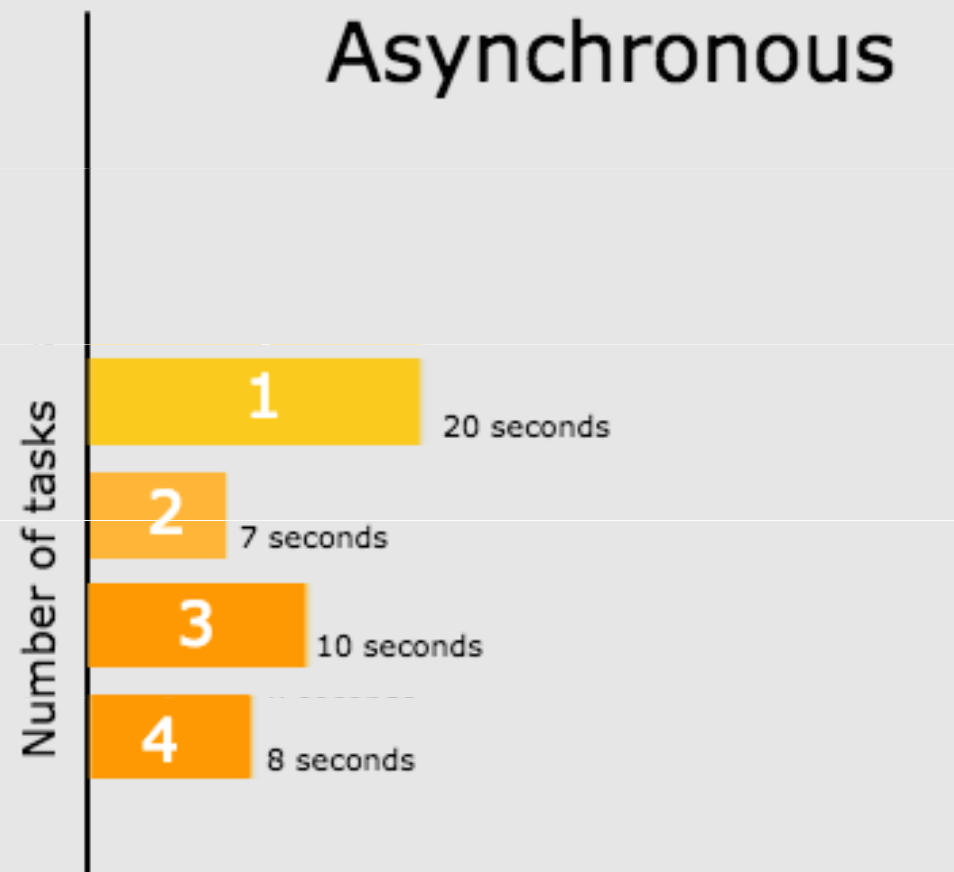
Sync vs Async

Synchronous



Total time taken by the tasks.
45 seconds

Asynchronous



Total time taken by the tasks.
20 seconds

HTTP dotazy

- AJAX – Asynchronous JavaScript And
- XML umožňují **dynamicky načítat data**
- na HTTP dotazy se vztahuje řada bezpečnostních limitů:
 - Stránky načtené přes HTTPS by měly dotazovat další zdroje **opět přes HTTPS**, ne HTTP
 - <https://portswigger.net/web-security/cors>
 - https://en.wikipedia.org/wiki/Cross-site_scripting
- skript není zpracován lineárně, ale části kódu se spustí **až po načtení** externího zdroje

XMLHttpRequest

Dříve (**nebudeme používat**):

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState === 4 && this.status === 200) {
        console.log(JSON.parse(this.response));
    }
};
xhttp.open("GET", "http://example.com/data.json", true);
xhttp.send();
```

// jednodušeji pomocí jQuery:

```
$.get("http://example.com/data.json", function (data) {
    console.log(data);
});
```

Fetch API

Budeme používat (ES6+):

```
fetch("http://example.com/data.json")  
  .then(response => response.json())  
  .then(data => console.log(data));
```

- Výraz v metodě `.then()` je **zpracován asynchronně**, po získání odpovědi.
- <http://bit.ly/medium-fetch-api>
<https://css-tricks.com/using-fetch/>

```
const getData = async () => {  
  const response = await fetch("http://example.com/data.json");  
  const data = await response.json();  
  console.log(data)  
}  
getData()
```

Špinavý trik

- Pokud pracujete lokálně a nepoužíváte webserver, `fetch("./lokalni_data.json")` vám **nebude fungovat** :-)
- buďto nahrajete data na server (např. <https://bit.ly/vozejkmap-geojson>) – ve výsledném cvičení použijete GeoJSON z WFS našeho Geoserveru

špinavý trik:

- v souboru `data.json` vložíte před JSON objekt `const DATA`
- `= data.json` nejlépe uložte jako `data.js`
`data.js` načtete ve stránce jako skript:
- `<script src="data.js">`
data se načítají lineárně! ⇒ může se **výrazně prodloužit načítání stránky**

Špinavý trik

index.html

```
...  
  <script src="vozejkmap.js"></script>  
  <script src="script.js"></script>  
</body>  
</html>
```

vozejkmap.js

```
const DATA = { ... } // objekt obsahuje celý GeoJSON objekt
```

script.js

```
console.log(DATA); // funguje!
```

L.geoJSON

<https://leafletjs.com/examples/geojson/>

```
const VOZEJK_URL =  
  'https://gist.githubusercontent.com/SLeitgeb/f136a1d4d28c2  
  f9ebdfe035bc3027b6d/raw/87b3bea8252bea3438fe7dfa93  
  7b79dcb83f0bea/vozejkmap.geojson';  
fetch(VOZEJK_URL)  
  .then(response => response.json())  
  .then(data => {  
    const VOZEJK = L.geoJSON(data)  
      .addTo(MAP);  
  });
```

Dále je dobré:

- přidat popis bodu pomocí `.bindPopup()`
- přidat vrstvu do překryvných vrstev pomocí `.addOverlay()`

L.geoJSON

<https://leafletjs.com/examples/geojson/>

```
const CITIES = 'https://ahocevar.com/geoserver/wfs?' +  
  'service=WFS&version=2.0.0&request=GetFeature&' +  
  'typename=ne:ne_10m_populated_places&' +  
  'outputFormat=application/json&srsname=EPSG:4326';  
fetch(CITIES)  
  .then(response => response.json())  
  .then(data => {  
    const VOZEJK = L.geoJSON(data)  
    .addTo(MAP);  
  });
```

Dále je dobré:

- přidat popis bodu pomocí `.bindPopup()`
- přidat vrstvu do překryvných vrstev pomocí `.addOverlay()`

Leaflet Pluginy

<https://leafletjs.com/plugins.html>

Přidáváme jako jakékoliv jiné zdroje, např.:

index.html

```
<html>
<head>
  <link rel="stylesheet" href="https://.../MarkerCluster.css">
  <link rel="stylesheet"
    href="https://.../MarkerCluster.Default.css">
</head>
<body>
  <script src="https://.../leaflet.markercluster.js"></script>
  <script src="script.js"></script>
</body>
</html>
```

L.markerCluster

```
const CLUSTER = L.markerClusterGroup()  
  .bindPopup(layer => layer.feature.properties.description)  
  .addTo(MAP);  
  
CLUSTER.addLayer(SOME_LAYER);
```

Komentované příklady

<https://gist.github.com/FilipLeitner/361d19d4df340ecce67b759b7a46efb7>

Úkol

- dokončete svoji webovou stránku včetně připomínek
 - využití flex/grid na rozložení elementů = směr + pozice. Nestačí display:flex + margin auto na child element
 - Minimálně 2 varianty v pozicování výsledného layoutu => aby jednotlivé podstránky nebyly jen kopie s jiným textem a obrázky
 - minimálně jedna podstránka (layout) upravovaná s využitím media query (změna css pravidla na základě šířky obrazovky - @media screen and)
- stránka bude obsahovat min. textový a obrazový obsah (responzivní a podle pokynu upravené obrázky + 1 upraveny pomocí srcset +sizes) k tématu a vkusně použitou interaktivní mapu v Leafletu
- použijte vámi připravená data prostřednictvím Geoserveru
 - geoJSON vektorových dat získaný pomocí WFS služby
 - WMS vrstva (volitelně)

Úkol

- Odevzdajte výsledok ktorý, nebudem musieť upravovať = správne cesty k súborom apod.
- validé HTML
- Prejdite si predchádzajúce komentáre a zapracujte ich
- odevzdat do 22. 12.
- bodované (max 20 b.)
- odevzdávat v archivu (web.zip)

Jak dál?

Přečtěte si víc a

- **procvičujte**

- <https://exercism.io/tracks/javascript>

- <https://javascript.info/>

- e-booky **zdarma**

- <https://github.com/getify/You-Dont-Know-JS>

- Vladimir Agafonkin: <https://github.com/mourner>

- <https://bost.ocks.org/mike/> <https://medium.com/>

Ptejte se

kdykoli

v

kdekoli

v

jakkoliv

co nejdřív...