

Vývoj Internetových Aplikací

HTML 5 API

Ing. Michal Radecký, Ph.D.

www.cs.vsb.cz/radecky

<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>

<http://html5slides-1117.appspot.com>

http://www.w3schools.com/html/html5_intro.asp

<http://www.zdrojak.cz/serialy/webdesigneruv-pruvodce-po-html5/>

<https://web.dev/learn>



Storage

- Náhrada za Cookies
 - data se neposílají v každém requestu
 - možnost ukládání většího množství dat
 - přístup pouze autorem dat
 - napojení událostí
- Princip uložení – dvojice klíč/hodnota (string)
- **LocalStorage** – data uložená bez omezené doby platnosti
- **SessionStorage** – data s platností pro jedno sezení
- Přístup pomocí rozhraní nebo indexů

```
if(typeof(Storage)!=="undefined")
{
  // Yes!
}
else
{
  // Sorry! No web storage support..
}
```

Web database

- Web SQL Database
 - API pro ukládání dat na straně klienta databázovým/relačním způsobem (SQL)
 - Aktuálně není dále rozvíjeno jako standard v rámci HTML5
 - Metody: *openDatabase*, *db.transaction*, *tr.executeSQL*
- IndexedDB
 - Řešení pro ukládání většího množství strukturovaných dat
 - Rychlé hledání s využitím indexování
 - Synchronní i asynchronní přístup
 - Objektově a transakčně orientovaná, používá dvojici klíč/hodnota(objekt)
 - API pomocí objektu **indexedDB**

Off-line aplikace

- Off-line běh webových stránek s využitím cachování
- Snižování nároků na rychlost a přenesená data
- Využívá tzv. Cache Manifest (text/cache-manifest)
 - Samostatný soubor s definicí cachovacích pravidel
 - CACHE – cachuje uvedené soubory při jejich načtení pro další použití
 - NETWORK – uvedené soubory nebudou nikdy cachovány
 - FALLBACK – náhrada za necachované soubory
- Aktualizace
 - Vyčištěním cache
 - Programově
 - Změnou manifestu
- Zastaralé, možnost využití Service Workerů

Web Workers

- Implementace „vláken“ v prostředí webové stránky, provádění algoritmů na pozadí bez ovlivnění interakce s uživatelem
- Využívá se externích JS souborů spouštěných jako tzv. `WebWorker` – synchronní přístup
- Primární je objekt **Worker**
- Worker pracuje na globální úrovni, s hlavním skriptem komunikuje pomocí zpráv (*postmessage* – *onmessage*)
- Nemá přístup k objektům *window*, *document*, *parent*
- *Worklet* – nezávislý přístup k renderovacímu vláknu pro náročnější multimediální operace
- *ServiceWorker* – nezávislé zpracování kódu s komplexním životním cyklem, primární je role proxy mezi aplikací a sítí

Web Sockets

- Vyspělé rozhraní pro obousměrnou asynchronní komunikaci (klient-server) bez nutnosti „čekat“ na zprávu od serveru (události)
- Nutná podpora na obou stranách (HTTP, TCP)
- Možnost využití v kombinaci s WebWorkers
- Hlavní objekt je **WebSocket**
- Implementace událostí *onopen*, *onmessage*, *onclose* a metody *send*
- Alternativy: Server Sent Event, MQTT, WebRTC, ...
<https://ably.com/topic/websocket-alternatives>

Drag & Drop

- Podpora jedné ze základních uživatelských funkcionalit
- Je možné přesouvat jakýkoliv obsahový objekt/prvek/element – **draggable=„true“**
- Implementace událostí *ondragstart*, *ondrop*, *ondragover*
- Práce s objektem události *dataTransfer.SetData* (*GetData*)

Drag-In (File API)

- Možnost přesunutí objektu z lokálního počítače do prostoru webové stránky
- Vychází z principu Drag & Drop, kdy je nutné zachytit událost *ondrop* na odpovídajícím elementu
- Přístup k přenášeným souborům pomocí *DataTransfer.files* (obdobně Input typu „file“)
- File API poskytuje objekty **File**, **FileList**, **Blob**, **FileReader**, **URL**
- File API slouží k práci se soubory přímo uvnitř stránky s možností přístupu k jejich obsahu (textově, binárně, Base64)

FileSystem API

- Rozšiřuje možnosti File API o zápis souborů (**BlobBuilder, FileWriter**) a jejich organizaci (**DirectoryReader, FileEntry/DirectoryEntry, LocalFileSystem**)
- Využívá tzv. virtuální souborový systém v rámci vnitřního prostředí stránky (sandbox) – přístup pomocí metody *requestFileSystem*
- Vhodné především pro binární data (dočasně i perzistentně) – upload souborů, mezi-úložiště multimediálních dat, editace souborů, off-line práce

Geolokalizace

- Možnost získání GPS (latitude, longitude, altitude, accuracy, speed, timestamp) souřadnic polohy uživatele
- Podmíněno povolením ze strany uživatele a zabezpečeným připojením
- Závislé na technických možnostech zařízení, příp. lokalizace pomocí internetového připojení
- API přístupné v objektu **navigator.geolocation**
- Metody *getCurrentPosition* a *watchPosition*

```
if ("geolocation" in navigator)
{
  /* geolocation is available */
}
else
{
  /* geolocation IS NOT available */
}
```

Přístup k hardware

- Orientace zařízení a poloha v prostoru
- Přístup ke kameře a mikrofonu
- Hlasový vstup
- Doteková gesta
- Fullscreen režim
- Tisk
- Autentifikace
- NFC, Bluetooth
- <https://web.dev/articles/devices-introduction>

Grafika

- **Bitmapová grafika** – využití Canvas elementu (musí mít uzavírací element)
 - Nad elementem se vytváří tzv. kontext, metodou *getContext(„2d“)*
 - Kontext pak poskytuje API pro kreslení, kreslí se postupně
 - Animace pomocí *setTimeout* a *setInterval*. Ideálně pomocí *requestAnimationFrame* – využívá standardní animační smyčku
- **Vektorová grafika** – využití SVG
 - Modifikace DOM – specifického XML
 - Možnost vazby vizuálních elementů na události v JS
- **3D grafika** – využití WebGL
 - Kontext „webgl“
 - Vnitřní API pro kreslení vychází z OpenGL
 - WebVR, WebAR, WebXR

Vlastní data- atributy

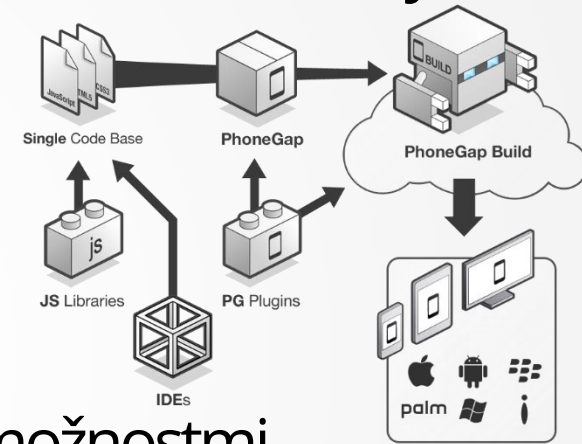
- Možnost ukládání specifických, aplikačně orientovaných, dat, pro které není odpovídající standardní možnost
- Použití prefixu **data-*** (prohlížeč tyto atributy ignoruje)
- Přístup pomocí vlastnosti *dataset* daného elementu
- Vhodné pro ukládání pracovních či stavových hodnot ve vazbě k elementům, konfiguračních hodnot, pro analýzu, apod.

Progressive Web Applications (PWA)

- Webová aplikace (HTML, CSS, JS), která je doplněná/rozšířená o funkce, které dovolují používat aplikaci jako mobilní (nativní).
 1. **Progresivní** – nerozlišují, v jakém prostředí pracuje uživatel (tenký klient)
 2. **Responzivní** – přizpůsobení se různým zařízením s různými zobrazovacími možnostmi
 3. **Nezávislé na připojení** – možnost práce (byť omezené) i v režimu offline
 4. **App-Like** – uživatelský zážitek je blízký nebo roven použití nativní aplikace
 5. **Aktuální** – proces aktualizace pomocí Service workers, a to nejen dat, ale i samotné aplikace
 6. **Bezpečné** – pouze na HTTPS protokolu
 7. **Dohledatelné** – jsou dohledatelné a indexovatelné (díky manifestu)
 8. **Instalovatelné** – není nutné stahování z aplikačních „stores“, stačí znát adresu a „instalační“ proces je zajištěn samotnou aplikací (ikona na ploše, apod.)
 9. **Dostupné** – snadno se sdílí prostřednictvím adresy URL, nevyžadují komplexní instalaci
- <https://web.dev/pwa-checklist/>
- <https://www.vzhurudolu.cz/prirucka/pwa>
- <https://www.rascasone.com/cs/blog/progresivni-webova-aplikace-vyhody>

Mobilní aplikace

- Jeden z důsledků HTML 5 technologií je implementace na mobilních zařízeních, a to nejen ve formě mobilních webových stránek, ale plnohodnotných aplikací
- Základem je „webová aplikace“ HTML5+JS+CSS doplněná o funkcionality specifického API. Výsledkem pak nativní crossplatform aplikace – webový prohlížeč s rozšířenými možnostmi.
- Využívá se abstrakční vrstvy, která zajistí rozhraní mezi web. aplikací a funkcionalitou na HW a OS úrovni
- Camera, Geolocation, Compass, Contacts, Media, Accelerometer, Network, Notification, Storage, Filesystem



Další prvky

- Webmanifest
- WebAssembly – instrukční binární formát programu a virtuální stroj