

PRÍKLADY VYUŽITIA JAVASCRIPT KNIŽNÍC V OBLASTI TVORBY TEMATICKÝCH MÁP NA INTERNETE

Michal RYBÁR, Vladimír BAČÍK

Examples of the use of JavaScript libraries for the creation of thematic maps on the Internet

Abstract: The map presentation of data in the Internet is very popular, not only in applications for experts, but also for lay users. Various formats and technologies give to web designers a variety of options, how to bring data closer to end users in the form of map outputs. We focused in our paper on the practical use of selected JavaScript libraries, which functionality can be exploited just in the distribution of maps on the Internet. Complex embrace of existing libraries in one paper is not possible, therefore our choice is focused on following libraries: amMaps, jVectorMap, jQuery Mapael and Kartograph. Basic specification of a library is supplemented by practical examples with an appropriate description of the source code, where its understanding is necessary for the application of the relevant library. Attributes appearing in these maps were stored in the database (MySQL). From these data was generated a relevant JSON file, while the notation of this file must be compatible with the appropriate JavaScript library. The paper is intended mainly to applicants who want to publish thematic interactive maps with their own maps, without the use of commercial tools and API services (such as Yahoo Maps, Bing Maps, OpenStreetMap, Google Maps, etc.). Individual above mentioned libraries could be used in all cases, when data visualization with spatial expression on the Internet is necessary.

Keywords: Thematic Mapping, Web Mapping, Scalable Vector Graphics, JavaScript Object Notation, Database, JavaScript

Úvod

Predložený príspevok sa zameriava na možnosti distribúcie a vizualizácie najmä priestorových dát v prostredí internetu. Hlavným cieľom bolo porovnanie funkcionality vybraných JavaScript knižníc používaných pri aplikáciách typu Web Mapping. Vzhľadom na aplikačný charakter príspevku, boli jednotlivé charakteristiky knižníc doplnené o viaceré príklady, vrátane použitého zdrojového kódu, ktorého pochopenie je kľúčovým faktorom úspešnej implementácie týchto knižníc. V príspevku sú zhrnuté základné teoretické poznatky z oblasti „web mapping“ technológií a spôsobu prezentácie priestorových dát v prostredí internetu. Príspevok je určený najmä záujemcom, ktorí chcú publikovať tematické interaktívne mapy s vlastnou podkladovou vrstvou (mapou), bez použitia komerčných nástrojov a bez použitia rozhrania pre programovanie aplikácií (Application Programming Interface - API) iných služieb (napríklad Yahoo Maps, Bing Maps, OpenStreetMap, Google Maps a pod.). Na tomto mieste treba podotknúť, že znalosť programovacích jazykov (JavaScript, PHP) a databázových systémov je nutnosťou. Bez potrebnej znalosti týchto komunikačných prostriedkov nie je možná ich plnohodnotná implementácia. Jednotlivé dokumentácie sú však dostupné a záujemca o tvorbu mapových výstupov ich môže plnohodnotne využívať pri tvorbe vlastných internetových aplikácií, ktorých hlavným zameraním je distribúcia priestorových dát v podobe tematických máp.

Nové digitálne technológie zavádzané do kartografie a naopak zavádzanie kartografie do informačných technológií prakticky ukončili klasickú (ručnú) kartografickú tvorbu. Súčasný proces tematického mapovania a tvorby máp je dnes dynamickejší, pružnejší, interaktívnejší. Jedinečné

Mgr. Michal RYBÁR, Univerzita Komenského v Bratislave, Prírodovedecká fakulta, Centrálné výpočtové stredisko, Mlynská dolina, Ilkovičova 6, 842 15 Bratislava, e-mail: michal.rybar@fns.uniba.sk

Mgr. Vladimír BAČÍK, PhD., Univerzita Komenského v Bratislave, Prírodovedecká fakulta, Katedra humánnej geografie a demogeografie, Mlynská dolina, Ilkovičova 6, 842 15 Bratislava, e-mail: bacik@fns.uniba.sk

vlastnosti multimédií a virtuálnej reality dodali digitálnym mapám novú hĺbku a poskytli tak nové možnosti kartografii ako vede, ako aj používateľom a širokej verejnosti (Voženílek, 2007). Treba si však uvedomiť, že rozvoj širokého spektra odborov využívajúcich kartografickú vizualizáciu a dostupnosť internetu, má zásadný vplyv na zapojenie širokej verejnosti nielen do využívania, ale aj do tvorby máp. Laická verejnosť má tak možnosť ľahko získať rôzne priestorové dáta, podkladové mapy, návody pre tvorbu a softvér slúžiaci k vytvoreniu máp. Internet zároveň slúži ako médium pre prezentáciu a distribúciu máp. Okrem iných, aj tieto skutočnosti majú za následok takzvanú „laicizáciu kartografie“, ktorú Čerba (2011, s. 31) definuje nasledovne: Laicizácia kartografie je stav, keď kartografické nástroje (softvér, webové služby) nepoužívajú len odborníci v oblasti kartografie, ale tiež nepoučení používatelia, ktorým chýbajú základné informácie, vrátane definícií.

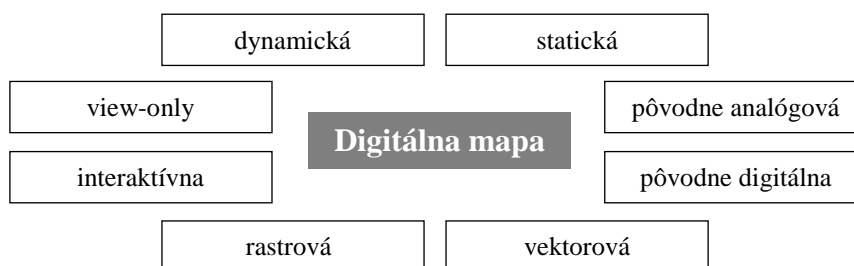
1. Publikácia máp v prostredí Internetu

Zdieľanie máp prostredníctvom internetu sa datuje zhruba od roku 1993, keď bol vytvorený prvý graficky prehliadač „Mosaic“. Najprv sa internetové prostredie využívalo k publikovaniu jednoduchých statických máp, pretože neexistoval žiadny špeciálny formát pre popis kartografickej informácie. Veľkým problémom bola nízka prenosová rýchlosť a malá pamäťová kapacita osobných počítačov (personal computer – PC) v porovnaní s pomerne veľkou veľkosťou mapových súborov v rastrovom formáte (Voženílek, 2007).

Najjednoduchšou metódou publikovania na internete, je vloženie obrazu (mapy) priamo do webovej stránky. Toto je možné realizovať viacerými spôsobmi, pričom najjednoduchší je použitie HTML kódu (HyperText Markup Language). Takáto prezentácia je súčasťou webstránky, nie je možné ju zväčšovať, zmenšovať, prípadne upravovať, alebo s ňou inak pracovať. Rastrový obrázok sa publikuje väčšinou vo formáte JPG (JPEG - Joint Photographic Experts Group) alebo PNG (Portable Network Graphics) a vektorový obrázok vo formáte SWF (ShockWave Flash), prípadne SVG (Scalable Vector Graphics), ktorý má v súčasnosti podporu vo väčšine bežných internetových prehliadačov bez potreby externého pluginu. Tieto uvedené metódy využívajú značkovací jazyk HTML. Na strane používateľa vyžadujú len internetový prehliadač a na strane servera webový server (Cajthaml, 2006).

Autor Čerba (2006) sa vo svojej práci „Mapy na internete“ podrobnejšie venuje digitálnym (internetovým) mapám, ktoré prehľadne rozdeľuje podľa niekoľkých hľadísk – podľa miery ovplyvňovania zobrazenia mapy používateľom, podľa typu zdrojových dát, podľa možností použitia mapy (mapy jednoúčelové a viacúčelové), podľa spôsobu vizualizácie, a tiež podľa dynamických vlastností mapy. Toto delenie rozširuje klasické delenie máp, ktoré poznáme z kartografie – delenie máp podľa mierky, podľa účelu, podľa rozsahu zobrazovaného územia, podľa obsahu mapy, podľa spôsobu vzniku a pod. Ak si ako parameter klasifikácie máp zoberieme mieru ovplyvnenia používateľom, tak rozlišujeme mapy: statické a interaktívne. Statická mapa je určená len na čítanie (zobrazenie) – view-only. Opakom statických máp, sú interaktívne mapy, ktoré disponujú možnosťami komunikácie aplikácie (mapy) a používateľa, ktorý si môže pomocou ovládacích prvkov mapou pohybovať, zväčšovať či zmenšovať a pod. Existujú rôzne úrovne používateľskej interaktivity. Všeobecne interaktivitu zaisťuje technika „klikacích“ máp (Clicable Maps, technika citlivých plôch – HTML tag <area>), softvérový modul, plug-in (väčšinou zaisťuje jednoduché operácie s mapou – približovanie, oddiaľovanie, posúvanie a pod.), alebo rôzne skriptovacie jazyky. Inou možnosťou je klasifikácia podľa typu zdrojových dát, z ktorých bola mapa generovaná. Tu rozoznávame pôvodne analógové mapy, ktoré boli digitalizované. Obzvlášť staré mapy sa na internete prezentujú vo forme digitalizovaných originálov alebo kópií. Príkladom môžu byť mapy I., II. a III. vojenského mapovania, Müllerove mapy a mapy stabilného katastra, ktoré sú k dispozícii na serveri Laboratória geoinformatiky Univerzity Jana Evangelisty Purkyně (<http://oldmaps.geola.b.cz>). V tomto prípade sa jedná o digitalizované mapy obohatené o interaktívnu zložku. Tieto mapy využívajú HTML tag <area>. Pri výbere konkrétneho územia sa zobrazí mapa celého zvoleného územia, ktorá je rozdelená na jednotlivé polia, po kliknutí na vybrané pole (územie) sa zobrazí podrobná digitalizovaná stará mapa vybraného územia, ktorá obsahuje základné ovládacie prvky (pohyb) a prvky zoom (približovanie – vzdľavovanie). Ďalším spôsobom klasifikácie je delenie máp na statické a dynamické. Zatiaľ čo statické mapy zobrazujú stále rovnaké dáta (aj keď s nimi môžeme vykonávať rôzne operácie), v dynamických mapách dochádza k zmene zobrazovaných

dát (mapa sa automaticky generuje). Variantom dynamickej mapy môže byť animácia. Statické mapy sa môžu použiť na jednoduchú prezentáciu orientačných plánov, alebo ako hypertextové odkazy na dynamické mapové aplikácie. Jednou z metód tvorby statických máp, je aj technika citlivých plôch, keď jednotlivé segmenty mapy umožňujú odkazovať na iné súbory alebo webové stránky. Medzi najjednoduchšie dynamické mapy patria napríklad mapy vo formáte animovaného súboru GIF (rastrový formát). Animácie ponúkajú aj niektoré vektorové grafické formáty, SVG alebo SWF. Zložitejšie dynamické mapy sú publikované v rôznych video formátoch – AVI, MPEG, QuickTime a pod. Celé delenie je schematicky znázornené na obr. 1.

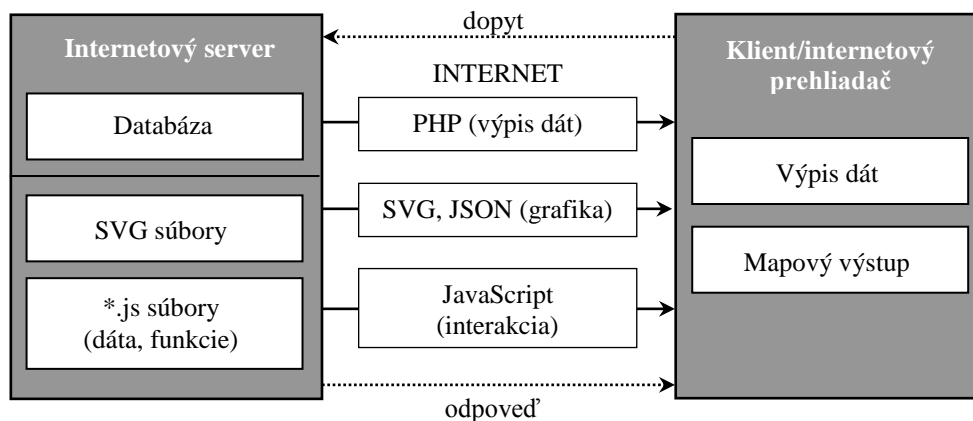


Obr. 1 Klasifikácia digitálnych máp (Čerba, 2006)

2. JavaScript knižnice využívané v oblasti internetového mapovania

Univerzálny postup pri vytváraní interaktívnych tematických máp, ktoré boli vytvorené v tomto príspevku, by sa dali zhrnúť do nasledovných krokov. Základná schéma fungovania je zobrazená na obr. 2, pričom priestorové a atribútové dáta sú uložené v samostatných prostrediach (súboroch). Priestorové dáta sú zapísané vo formáte SVG, ktorý je prekonvertovaný do kompatibilného JavaScript Object Notation (JSON) zápisu pre danú knižnicu. Atribútové dáta sú uložené v relačnej databáze MySQL, z ktorej sú vyberané a následne je z nich generovaný textový zápis v požadovanom tvare, napríklad JSON. Podrobné naštudovanie dokumentácie jednotlivých knižníc je nevyhnutným predpokladom úspešnej realizácie mapového výstupu v podobe mapy umiestnenej na vlastných webových stránkach. Z hľadiska metodologického boli pri spracovaní dát využité štandardné postupy (import pripravených CSV súborov do databázy MySQL pomocou rozhrania phpMyAdmin). Tvorba samotnej ukážky tematickej mapy sa skladala z viacerých častí. Základom bola príprava mapového výstupu (vo formáte SVG), zápis HTML súboru s potrebnými zápismi jednotlivých nastavení zobrazovanej mapy, zápis potrebných knižníc a dát v požadovanom formáte. Samotný výber z MySQL databázy a generovanie JSON súborov prebiehalo pomocou skriptovacieho jazyka PHP.

Počet knižníc určených pre tvorbu mapových výstupov v prostredí Internetu je veľmi veľký a ich úplné obsiahnutie v jednom príspevku je nemožné. Do výsledného výberu boli zahrnuté nasledovné knižnice: jQuery Mapael, jVectorMap, Kartograph, amMaps. Všetky uvedené knižnice sú napísané v programovacom jazyku JavaScript. Hlavným kritériom výberu bola možnosť pracovať s vlastnými podkladovými mapami, bez použitia API iných služieb (Yahoo Maps, Bing Maps, OpenStreetMap, Google Maps a pod.). Ďalším kritériom bol dátum aktualizácie danej knižnice, ktorý bol stanovený na rok 2014. Na tomto mieste je vhodné upozorniť na absenciu knižnice D3.js, ktorá je v súčasnosti jednou z najdynamickejších využívaných vizualizačných knižníc v prostredí Internetu. Pracuje s formátom TopoJSON, ktorého popis nie je predmetom predloženého príspevku. Niektoré základné špecifikácie tohto formátu možno nájsť v Bačík (2015). Základný prehľad použitých JavaScript knižníc je v tab. 1.



Obr. 2 Schematické znázornenie fungovania použitých JavaScript knižníc (Bačík, 2008)

2.1 jQuery Mapael

Knižnica jQuery je založená na Raphael.js¹ knižnici, ktorá umožňuje zobraziť dynamické tematické vektorové mapy. jQuery využíva natívne technológie integrované do prehliadačov: JavaScript CSS, HTML, SVG. Podporuje všetky moderné prehliadače a Internet Explorer (IE) 9+. Pre staršie verzie IE, je nutné načítať jQuery 1.11.x a Raphael.js 2.1.2. Autori knižnice dávajú na svojich stránkach, podobne ako iní tvorcovia, k dispozícii viaceré vytvorené mapy, ktoré je možné okamžite využívať v aplikáciách (v tomto prípade sú to mapy štátov a regiónov: Argentína, Belgicko, Francúzsko, Nemecko, Škandinávia, Holandsko, Thajsko, Ukrajina, Spojené kráľovstvo, Wales, Európska únia). Základom jednotlivých knižníc je API referencia² (verzia 1.1.0). Všetky nastavenia sú k dispozícii ako argumenty funkcie \$.fn.mapael. Mnohé z nich majú definované predvolené hodnoty v premenných \$.fn.mapael.defaultOptions a \$.fn.mapael.legendDefaultOptions v súbore „jquery.mapael.js“. Prostredníctvom týchto nastavení sa definuje názov mapy, aká mapa sa bude zobrazovať, povoliť funkciu „zoom“, ktorá približuje a odďaľuje mapu pomocou tlačidiel alebo pomocou kolieska na myši. Veľmi dôležitou možnosťou formátu SVG je zobrazenie bodov (markers) pomocou zemepisných súradníc (šírka a dĺžka), alebo pravouhlých súradníc (x, y). Ďalej je to zakrivenie elementov (línie) medzi dvoma bodmi a nastavenia rôznych atribútov. Súčasťou API je aj možnosť definovať legendu, farbu pozadia, farebné spektrum kartogramu, farby kartogramu, farbu zobrazených bodov, farbu elementov a mnoho ďalších nastavení.

Praktický príklad použitia tejto knižnice ukážeme na vlastnom príklade zobrazujúcom mieru evidovanej nezamestnanosti v okresoch SR (január 2015)³. Atribúty potrebné pre vykreslenie tematickej mapy boli uložené v databáze MySQL, pričom samotné dopytovanie na konkrétne dáta prebieha pomocou štandardnej SQL syntaxe. Hlavným identifikátorom prepojenia tabuliek bol kód okresu, na základe ktorého bola vytvorená relácia medzi tabuľkami (geo_kod_okres, geo_nezamestnanost). Výsledný SVG súbor bol vytvorený v prostredí programu ArcGIS, po príslušnej konverzii Křovákovho zobrazenia do Mercatorovho zobrazenia. Knižnica umožňuje pracovať s mapou v ktoromkoľvek kartografickom zobrazení, ale v prípade definovania (zobrazovania) bodov (markerov) pomocou zemepisnej šírky a dĺžky, je potrebné v kóde definovať transformáciu medzi zemepisnými a pravouhlými súradnicami uviesť správne prepočet do x, y pravouhlých súradníc.

¹ Raphael.js je JavaScript knižnica, ktorá zjednodušuje prácu s vektorovou grafikou na internete (<http://dmitrybaranovskiy.github.io/raphael/>).

² Kompletná dokumentácia API sa nachádza na stránke: <http://www.vincentbroute.fr/mapael/>.

³ Príklad je dostupný na adrese: http://geo.treska.eu/mapael/slovakia_nezamestnanost.php.

Výsledné zobrazenie mapy (obr. 3) spočíva v načítaní príslušných atribútov uložených v databáze a vizualizácie mapového výstupu vo formáte JSON. Výber atribútov a prepojenie tabuliek je zabezpečené základným príkazom SELECT (riadok 02 v uvedenom PHP kóde). Následne prebehne výpočet evidovanej miery nezamestnanosti (riadky 09 až 11). Z vyberaných dát sa vytvára JSON zápis, z ktorého knižnica jQueryMapael vygeneruje a vykreslí výslednú mapu.

Tab. 1 Prehľad použitých JavaScript knižníc a ich základné vlastnosti

	jQuery Mapael 1.1.0	jVectorMap 2.0.4	amMaps 3.19.0	Kartograph 0.8.7
Domovská adresa	nevelo.fr/mapael	jvectormap.com	amcharts.com	kartograph.org
Dátum poslednej aktualizácie	31.8.2015	11.8.2015	9.2.2016	15.5.2014 (vývoj bol zastavený)
Autor	Vincent Brouté	Kirill Lebedev	Antanas Marcelionis a kol.	Gregor Aisch
Kompatibilita	všetky moderné prehliadače a Internet Explorer 9+	všetky moderné prehliadače a Internet Explorer 6+	všetky moderné prehliadače, Internet Explorer 9+ a mobilné zariadenia (iOS a Android 4+)	nebola uvedená
Pridávanie vlastných máp	Áno (pomocou konvertora)	Áno (pomocou konvertora)	Áno (pomocou konvertora)	Áno (priama podpora SVG)
„Zoom“	Áno	Áno	Áno	Nie
Zobrazenie bodov pomocou zemepisných súradníc	Áno	Áno	Áno	Áno
Interaktivita	Áno	Áno	Áno	Áno ⁴
Kreslenie línií	Áno	Nie	Áno	Áno
Kartodiagramy	Áno	Áno	Áno	Áno
Zmena kartografického zobrazenia	Nie	Nie	Áno	Nie
Pridávanie vlastných obrázkov (značiek)	Nie	Áno	Áno	Nie
Legenda	Áno	Áno	Áno	Áno
Mierka	Nie	Nie	Nie	Nie
Nástroje exportu a tlače	Nie	Nie	Nie	Áno

⁴ V nami uvedenom príklade sa nepodarilo ukázkou uviesť do interaktívnej podoby

PHP kód výberu dát z databázy a následného generovania JSON:

```

01 <?php
02 $result = $mysqli->query("SELECT geo_nezamestnanost.ea_obyv,
    geo_nezamestnanost.nezamest_obyv, geo_nezamestnanost.id, geo_kod_okres.nazov,
    geo_kod_okres.skratka FROM geo_nezamestnanost, geo_kod_okres WHERE
    geo_nezamestnanost.id = geo_kod_okres.id"); {
03     while ($row = $result->fetch object()) {
04         $id = $row->id;
05         $ea_obyv = $row->ea_obyv;
06         $nezamest_obyv = $row->nezamest_obyv;
07         $nazov = $row->nazov;
08         $skratka = $row->skratka;
09         $nezamestnanost = $nezamest_obyv / $ea_obyv;
10         $nezamestnanost = round($nezamestnanost, 4); //zaokruhlovanie
11         $nezamestnanost_per = $nezamestnanost * 100;
12         echo "\t\t\"" . $id . "\": {\n";
13         echo "\t\t\t\"value\": \"" . $nezamestnanost . "\",\n";
14         echo "\t\t\t\"tooltip\": {\n";
15         echo "\t\t\t\t\"content\": \"<span style=\\\"font-weight:bold;\\\">" .
    $nazov . "</span><br \\/><br \\/>Nezamestnanost: " . $nezamest-
    nanost_per . "%\n";
16         echo "\t\t\t\t,\n";
17         echo "\t\t\t\t\"text\": {\t\t\t\t\"content\": \"" . $skratka . "\", \"attrs\"
    : {\t\t\t\t\t\"font-weight\" : \"bold\", \"font-size\" : 10, \"fill\" :
    \"#000\"}},\n";
18         echo "\t\t\t\t,\n";
19     }
20 }??>

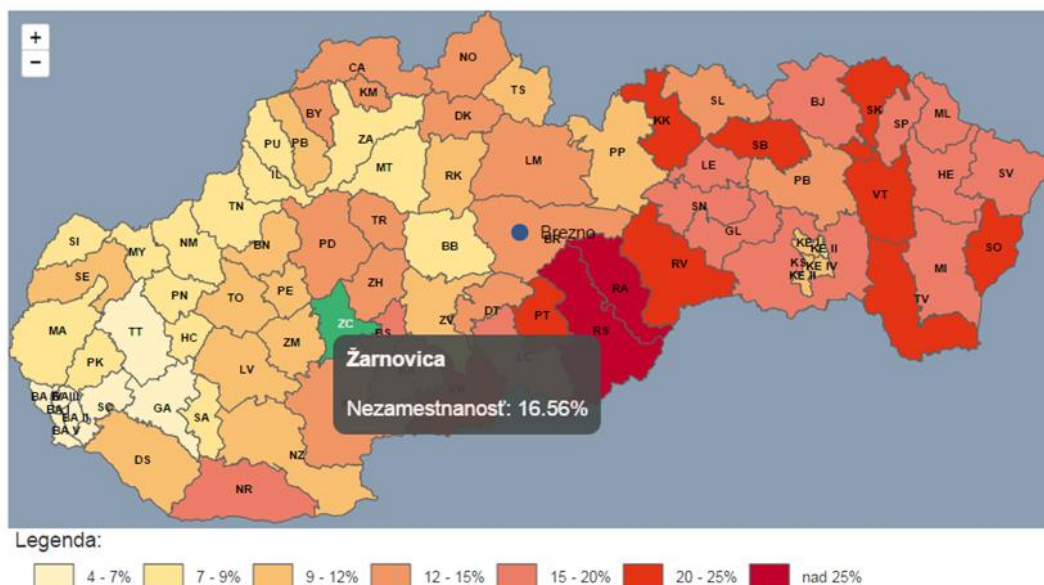
```

Ukážka vygenerovaného JSON súboru:

```

01 areas: {
02     "101": {
03         "value": "0.0499",
04         "tooltip": {
05             "co-
06             tent": "<span style=\"font-weight:bold;\">Bratislava I</span><br
    \\/><br \\/>Nezamestnanost: 4.99%"
07         },
08         "text": { "content": "BA I", "attrs" : {"font-
09         weight" : "bold", "font-size" : 10, "fill" : "#000"}},
10     },
11     "102": {
12         "value": "0.0663",
13         "tooltip": {
14             "co-
15             tent": "<span style=\"font-weight:bold;\">Bratislava II</span><br
    \\/><br \\/>Nezamestnanost: 6.63%"
16         },
17         "text": {"content": "BA II", "attrs" : {"font-weight" : "bold", "font-
18         size" : 10, "fill" : "#000"}},
19     },
20 }

```



Obr. 3 Výsledná mapa evidovanej nezamestnanosti vytvorená pomocou knižnice jQuery Mapael (Výsledok je dostupný na adrese: http://geo.treska.eu/mapael/slovakia_nezamestnanost.php)

2.2 jVectorMap

Tak ako predchádzajúca knižnica, aj táto používa natívne technológie prehliadača, typu JavaScript CSS, HTML, SVG. Na stránke tvorca knižnice je uvedená jej kompatibilita so všetkými bežne používanými internetovými prehliadačmi (Opera, Chrome, Mozilla, Safari) vrátane Internet Exploreru od verzie 6. Autor ponúka viacero máp: mapu sveta, svetových regiónov, štátov a miest (zoznam vytvorených máp je dostupný na stránke knižnice). Každá mapa je vytvorená v dvoch kartografických zobrazeniach: Millerovom a Mercatorovom. Ku všetkým mapám je uvedený aj zoznam kódov (identifikátorov), pomocou ktorých sú pomenované jednotlivé objekty (štáty, územia atď.). Všetky mapy sú vytvorené z údajov licencovaných pod slobodnými licenciami, takže je možné ich použiť bez poplatku. Pri tvorbe vlastných podkladových máp má tvorca dve základné možnosti. Jednou z nich je možnosť konverzie súborov z formátu Shapefile (SHP) do formátu JSON pomocou skriptu napísaného v programovacom jazyku Python⁵. Ďalšia možnosť je konverzia mapy vo formáte SVG pomocou on-line nástroja „SVG to jVectorMap converter“⁶. Aj v tomto prípade, podobne ako pri všetkých použitých knižniciach, tvorí základ práce s dátami API referencie⁷ (verzia 2.0.4). Požadované parametre sa zapisujú do súboru s JavaScript kódom. Medzi základné parametre patria: názov volanej mapy, farba pozadia mapy, rôzne parametre pre „zoom“, nastavenie štýlu regiónov, popisov a značiek (východiskový stav, ak sa nachádza kurzor myši nad regiónom, ak je región vybraný), začiatočnú polohu a mierku mapy, rôzne udalosti (JavaScript), ktoré sa vykonajú po určitej akcii (napríklad kliknutí na objekt). Ďalšou dôležitou časťou API je opis prevodu medzi zemepisnými a pravouhlými súradnicami v jednom, alebo medzi dvoma súradnicovými systémami. Autor dáva k dispozícii zobrazovacie vzťahy (rovnice) medzi nasledujúcimi kartografickými zobrazeniami: Albersove rovnakoplošné kužeľové (Albers), Lambertovo konformné kužeľové, Merkatorovo a Millerovo zobrazenie. Ďalšie rovnice sa dajú doprogramovať do súboru s názvom „jquery-jvectormap.js“.

⁵ jVectorMap converter: <https://github.com/bjornnd/jvectormap/tree/master/converter>

⁶ SVG to jVectorMap converter: <http://svgto.jvectormap.com/>

⁷ Podrobne je API referencie opísaná na stránke: <http://jvectormap.com/documentation/>

Rozhodli sme sa pri praktickom príklade využiť mapu Venezuely, ktorá bola dostupná na stránke knižnice. Aj v tomto prípade boli dáta (získané z viacerých zdrojov) organizované v databáze. V použitej podkladovej mape sú jednotlivé štáty Venezuely identifikované pomocou kódovania ISO 3166-2. Ďalším použitým stĺpcom je kód OCEI⁸, ide o kód ktorý používa Národný štatistický inštitút Venezuely. Každý štát má pridelené číslo od 1 po 25, reťazením číselných kódov ďalších jednotiek (municipalít, departmentov, farností) sa získa jedinečný kód pre každé obývané miesto vo Venezuele. Tieto dáta boli získané zo stránky GeoHive⁹ a Administrative Divisions of Countries (Statoids)¹⁰. Dáta o podiele obyvateľov žijúcich v mestách boli získané zo sčítania ľudu 2011 Národného štatistického inštitútu (Instituto Nacional de Estadística)¹¹. Každá federálna entita (štát) je identifikovaná pomocou kódu OCEI. Takto získané dáta boli vložené do databázy štandardným spôsobom (importom CSV súborov do databázy MySQL). Výsledný JSON súbor s dátami sa generuje pomocou PHP kódu výberom z MySQL databázy. SQL dopyt prepojí tabuľky cez identifikátory (OCEI kód) a vyberú sa len potrebné atribúty (názov štátu, ISO kód, hlavné mesto štátu, miera urbanizácie). Vybrané dáta boli následne spracované do potrebného tvaru JSON. Fázy potrebné pre zobrazenie výslednej mapy sú zobrazené v nasledujúcich kódoch zápisu, pričom podoba výslednej mapy je zobrazená pod uvedeným kódom na obr. 4.

PHP kód výberu dát z databázy a následného generovania JSON:

```

01 <?php
02 if ($result = $mysqli->query("SELECT geo_venezuela_state.State,
    geo_venezuela_state.ISO, geo_venezuela_state.Capital,
    geo_venezuela_urban_pop.Urban_population
03 FROM geo_venezuela_state, geo_venezuela_urban_pop
04 WHERE geo_venezuela_state.OCEI = geo_venezuela_urban_pop.OCEI")) {
05     echo "\t\"states\": [\n";
06     while ($row = $result->fetch_object()) {
07         $state       = $row->State;
08         $iso         = $row->ISO;
09         $capital     = $row->Capital;
10         $urban_population = $row->Urban_population;
11         echo "\t\t{\n";
12         echo "\t\t\t\"sname\": \"" . $state . "\",\n";
13             "\t\t\t\"ccode\": \"" . $iso . "\",\n";
14             "\t\t\t\"urban_population\": \"" . $urban_population .
15             "\t\t\t\"capital\": \"" . $capital . "\"\n";
16         echo "\t\t},\n";
17     }
18     echo "]\n";
19 }
20 ?>

```

Ukážka vygenerovaného JSON súboru:

```

01 var dataS = {
02     "states": [
03         {
04             "sname": "Distrito Capital",
05             "ccode": "VE-A",
06             "urban_population": "99.94",
07             "capital": "Caracas"
08         },
09         {
10             "sname": "Amazonas",

```

⁸ OCEI - Oficina Central de Estadística e Informática (Národný štatistický inštitút)

⁹ Global Population Statistics (GeoHive): <http://www.geohive.com/cntry/venezuela.aspx>

¹⁰ Administrative Divisions of Countries (Statoids): <http://www.statoids.com/uve.html>

¹¹ Instituto Nacional de Estadística (INE): <http://www.redatam.ine.gob.ve/>


```

11     "ccode": "VE-Z",
12     "urban_population": "60.29",
13     "capital": "Puerto Ayacucho"
14   },
15   {
16     "sname": "Anzoátegui",
17     "ccode": "VE-B",
18     "urban_population": "91.57",
19     "capital": "Barcelona"
20   },
21   ...
22 ]
23 };

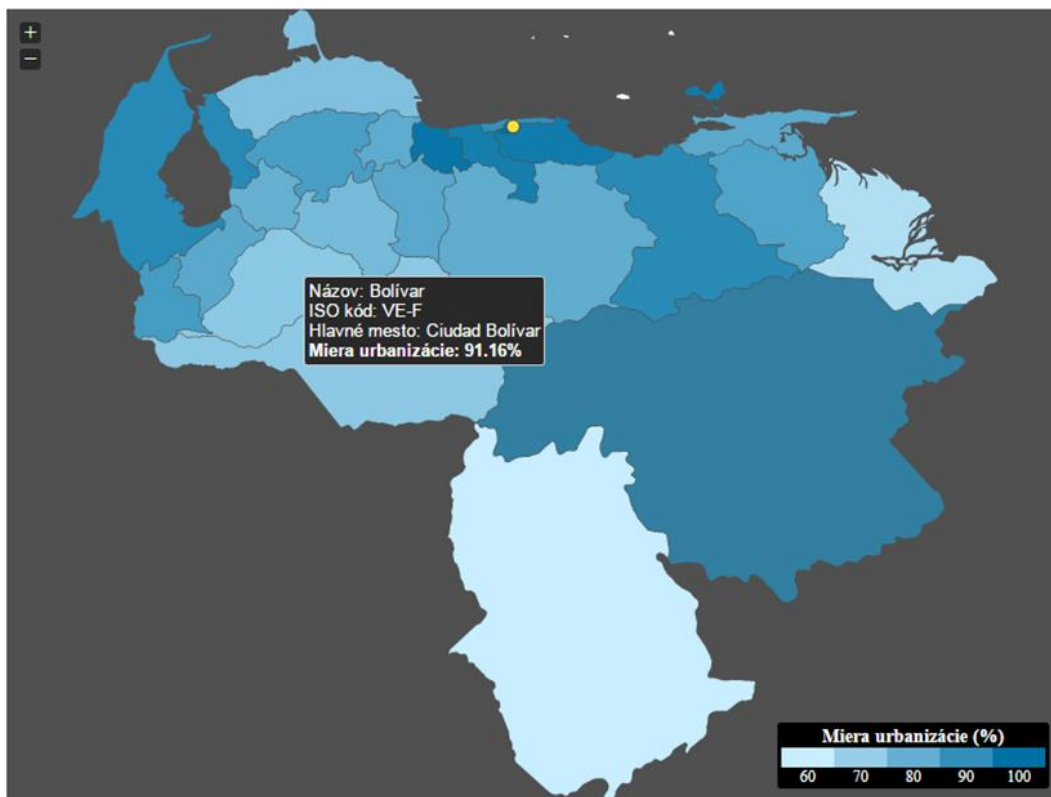
```

Ukážka JavaScript kódu (definovanie parametrov mapy):

```

01 var dataS = {
02   <?php include ('venezuela_data.php'); ?>
03 };
04 var stateData = [];
05 $.each(dataS.states, function() {
06   stateData[this.ccode] = this.urban_population;
07 });
08 $(function() {
09   $('#map').vectorMap({
10     map: 've_mil_en',
11     markerStyle: {
12       initial: {
13         fill: '#F8E23B',
14         stroke: '#383f47'
15       }
16     },
17     markers: [{
18       latLng: [10.5, -66.91667],
19       name: 'Hlavné mesto: Caracas'
20     }, ],
21     series: {
22       regions: [{
23         values: stateData, //load the data
24         scale: ['#C8EEFF', '#0071A4'],
25         legend: {
26           horizontal: true,
27           title: 'Miera urbanizácie (%)'
28         }
29       }],
30     },
31     onRegionTipShow: function(e, el, code) {
32       var state = $.grep(dataS.states, function(obj, index) {
33         return obj.ccode == code;
34       })[0]; //snag the first one
35       //only if selected country was found in dataS
36       if (state != undefined) {
37         el.html(
38           "Názov: " + state.sname +
39           "<br/>ISO kód: " + state.ccode +
40           "<br/>Hlavné mesto: " + state.capital +
41           "<br/><b>Miera urbanizácie: " + state.urban_population +
42           "%</b>"
43         );
44       }
45     });
46 });

```



Obr. 4 Výsledná mapa podielu obyvateľov žijúcich v mestách vytvorená pomocou knižnice jVectorMap (Výsledok je dostupný na adrese: <http://geo.treska.eu/jvectormap/venezuela.php>)

2.3 amMaps

Knižnica amMaps je určená okrem tvorby mapových výstupov aj na vizualizáciu dát v podobe grafov. Je rozdelená do troch častí, z toho sú dve určené pre vytváranie grafov (Charts v3 a Stock Chart v3) a tretia časť Interactive Javascript Maps je venovaná tvorbe a publikovaniu webových interaktívnych máp. Knižnica je založená na technológiách HTML5, CSS, JavaScript a SVG, pričom kompatibilita je zabezpečená vo všetkých moderných prehliadačoch (Firefox, Chrome, Safari, Opera a Internet Explorer od verzie 9). Navyše je zabezpečená kompatibilita aj pre mobilné zariadenia s operačným systémom iOS (iPad, iPhone, iPod Touch) a Android (od verzie 4), čo možno považovať za veľmi pozitívnu charakteristiku vzhľadom na neustály nárast podielu pripojených na internet pomocou mobilných zariadení. Ako jediná knižnica zo všetkých spomenutých v tomto príspevku, má jedinečné funkcie, ako je WordPress plugin, funkcia uloženia mapy do rastrových aj vektorových formátov a priamu tlač. K dispozícii sú mapy takmer všetkých väčších štátov sveta, so základným administratívnym členením¹². Väčšinou sú k dispozícii v dvoch úrovniach generalizácie. Pridávanie vlastných máp je pomerne jednoduché. Dôležitým momentom je použitie Mercatorovho zobrazenia, keďže transformácia medzi zemepisnými a pravouhlými súradnicami funguje správne len pri použití tohto zobrazenia. Na vytváranie vlastných máp je k dispozícii on-line na-

¹² Autor ponúka veľké množstvo máp, ktoré sú dostupné vo formáte JSON na adrese: <https://github.com/amcharts/ammap3/tree/master/ammap/maps/js> alebo vo formáte SVG na <https://www.amcharts.com/svg-maps/>.

stroj¹³, ktorý zabezpečuje prevod SVG do potrebného formátu JSON. Celý postup tvorby vlastnej podkladovej mapy je popísaný v podrobnom návode¹⁴.

Požadované parametre sa zapisujú do JavaScript kódu (API verzia 3.19.0). API ponúka možnosť upraviť vzhľad a funkcie mapy, nápovedy (tooltipov), legendy, definovanie bodov a línií pomocou zemepisných súradníc. Z kartografických zobrazení je podporované len Mercatorovo zobrazenie. Vo verzii 3.19.0 pribudla funkcia zmeny kartografického zobrazenia „za chodu“, pričom zdrojová mapa musí byť v Mercatorovom zobrazení. Mapu je možné transformovať do nasledujúcich zobrazení: Winkelovo III, Eckertovo III, Eckertovo V, Eckertovo VI, Millerovo a ekvidištančné valcové (Equirectangular). Ako jediná knižnica ponúka prehľadovú mapu, ktorá zobrazuje aktuálnu úroveň priblíženia a pozíciu v mape. Dokumentácia k API sa nachádza na webovej stránke projektu: <http://docs.amcharts.com/3/javascriptmaps>. Celkovo je API spracované podrobne a na veľmi vysokej úrovni.

V prípade tvorby tematických máp pre územie Slovenska je možné využiť mapu na úrovni krajov, ktorá je dostupná na stránkach knižnice. Ak chce autor spracovať iné hierarchické úrovne, je nutné využiť štandardné postupy konverzie mapového výstupu do SVG a jeho následné transformovanie do formátu JSON. V tomto prípade odporúčame použiť konvertor pre amMaps.

Príklad použitia tejto knižnice je ilustrovaný na príklade odletov z bratislavského letiska M. R. Štefánika – Airport Bratislava, a. s. (BTS) (obr. 5). Dáta potrebné pre získanie relevantných dát boli spracované na základe stránky OpenFlights.org a stránky bratislavského letiska M. R. Štefánika. Dáta boli uložené v dvoch databázových tabuľkách „geo_airports“ (táto obsahovala okrem príslušných kódov potrebných pre prepojenie aj zemepisné súradnice jednotlivých letísk) a „geo_bts_departures“. Tieto dve tabuľky sú prepojené pomocou trojmiestneho kódu IATA¹⁵. Ukážka prepojenia tabuliek je v nasledujúcom výpise zdrojového kódu:

```
01 | SELECT * FROM geo_bts_departures, geo_airports WHERE geo_bts_departures.IATA  
   | = "geo_airports.IATA_FAA"
```

Vygenerovaný zápis dát pomocou PHP skriptu má nasledovný tvar: 3

```
01 | lines: [  
02 |   {  
03 |     latitudes: [48.1702, 35.7581],  
04 |     longitudes: [17.2127, 10.7547],  
05 |     customData: "<strong>Číslo letu:</strong> 6D6489<br />  
   |                 Monastir <- Bratislava",  
06 |   }, {  
07 |     latitudes: [48.1702, 52.4539],  
08 |     longitudes: [17.2127, -1.748],  
09 |     customData: "<strong>Číslo letu:</strong> FR734<br />  
   |                 Birmingham <- Bratislava",  
10 |   }, {  
11 |     latitudes: [48.1702, 53.3336],  
12 |     longitudes: [17.2127, -2.8497],  
13 |     customData: "<strong>Číslo letu:</strong> FR4023<br />  
   |                 Liverpool <- Bratislava",  
14 |   }, ... ],  
15 | images: [  
16 |   {  
17 |     svgPath: targetSVG,  
18 |     latitude: 35.7581,  
19 |     longitude: 10.7547,
```

¹³ SVG to JSON converter: <http://extra.ammap.com/parser/>

¹⁴ Podrobný návod na tvorbu vlastnej mapy pre amMap je dostupný na stránke: <http://www.amcharts.com/tutorials/creating-custom-maps-for-javascript-ammap/>

¹⁵ IATA Codes: <https://www.iata.org/services/pages/codes.aspx>

```

20     title: "Monastir",
21     customData: "<strong>Štát:</strong> Tunisia<br /><strong>Letisko:</strong> Habib Bour-guiba Intl<br /><strong>Odlet z BTS:</strong> 9:55<br /><strong>Číslo letu:</strong> 6D6489<br /><strong>Spoločnosť:</strong> Travel Service Airlines Slovakia<br />",
22     scale: 0.7
23 }, {
24     svgPath: targetSVG,
25     latitude: 52.4539,
26     longitude: -1.748,
27     title: "Birmingham",
28     customData: "<strong>Štát:</strong> United Kingdom<br /><strong>Letisko:</strong> Birmingham<br /><strong>Odlet z BTS:</strong> 22:05<br /><strong>Číslo letu:</strong> FR734<br /><strong>Spoločnosť:</strong> Ryanair<br />",
29     scale: 0.7
30 }, {
31     svgPath: targetSVG,
32     latitude: 53.3336,
33     longitude: -2.8497,
34     title: "Liverpool",
35     customData: "<strong>Štát:</strong> United Kingdom<br /><strong>Letisko:</strong> Liverpo-ol<br /><strong>Odlet z BTS:</strong> 21:10<br /><strong>Číslo letu:</strong> FR4023<br /><strong>Spoločnosť:</strong> Ryanair<br />",
36     scale: 0.7}...]

```

Vlastnosti výslednej mapy sú definované pomocou nasledovného JavaScript kódu.

```

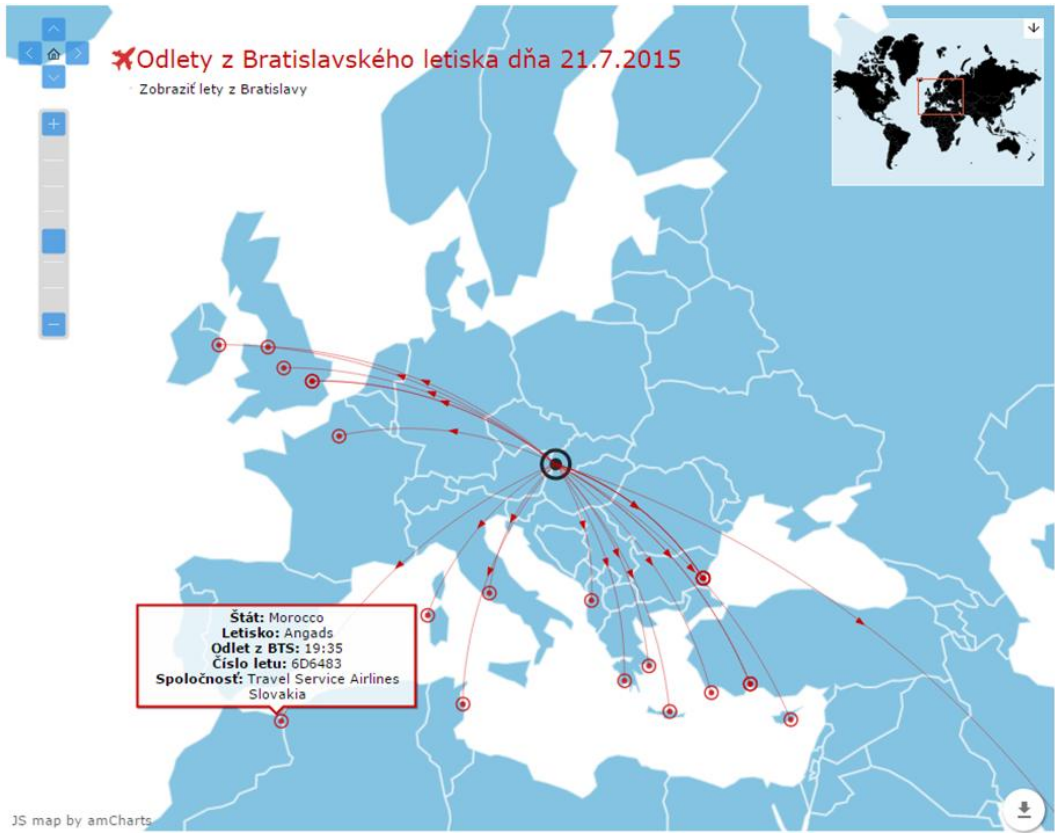
01     var map = AmCharts.makeChart("chartdiv", {
02         type: "map",
03         "theme": "light",
04         dataProvider: {
05             linkToObject: "bratislava",
06             map: "worldLow",
07             zoomLevel: 3.5,
08             zoomLongitude: 17.2127,
09             zoomLatitude: 48.1702,
10         },
11         areasSettings: {
12             unlistedAreasColor: "#FFCC00",
13             unlistedAreasAlpha: 0.9,
14         },
15         imagesSettings: {
16             color: "#CC0000",
17             rolloverColor: "#CC0000",
18             selectedColor: "#000000",
19             balloonText: "[[customData]]"
20         },
21         linesSettings: {
22             arc: -0.7,
23             arrow: "middle",
24             color: "#CC0000",
25             alpha: 0.4,
26             arrowAlpha: 1,
27             arrowSize: 4,
28             balloonText: "[[customData]]"
29         },
30         backgroundZoomsToTop: true,
31         linesAboveImages: true,
32         "smallMap": {},
33         "export": {

```

```

34     "enabled": true,
35     "menu": [ {
36         "class": "export-main",
37         "menu": [ {
38             "label": "Exportovať",
39             "menu": [ "PNG", "JPG", "SVG", "PDF" ]
40         } ]
41     } ]
42 },
43 });

```



Obr. 5 Výsledná mapa odletov z Bratislavského letiska M. R. Štefánika vytvorená pomocou knižnice am-Maps (Výsledok je dostupný na adrese: http://geo.treska.eu/ammap/odlety_bts.php)

2.4 Kartograph

V príspevku predstavíme aj knižnicu Kartograph. Na stránkach knižnice sa však možno dozvedieť, že autor prestal túto knižnicu vyvíjať a sám odporúča používať knižnicu D3.js¹⁶. Kartograph.js je JavaScript knižnica založená na Raphael.js, CSS, HTML a SVG. Kompatibilita s prehliadačmi nie je uvedená. Knižnica ako jediná vie pracovať s mapami priamo vo formáte SVG.

¹⁶ Data-Driven Documents (D3.js): <http://d3js.org/>

Navyše autor ponúka možnosť použiť konvertor `kartograph.py`¹⁷, ktorý ponúka možnosť konverzie mapy z formátu SHP do formátu SVG. Tu treba upozorniť na fakt, že aj tento konvertor prestal autor vyvíjať a odporúča použiť on-line nástroj `Mapshaper`¹⁸, ktorý slúži na editáciu SHP, GeoJSON, TopoJSON a niekoľkých ďalších dátových formátov. Pochopiteľne je možné za účelom konverzie využiť aj niektorý zo štandardne používaných DesktopGIS programov. Autor na svojich stránkach neponúka možnosť priameho použitia niektorých vopred vytvorených máp určených na priame použitie.

V prípade použitia API referencie (verzia 0.8.7) sa jednotlivé parametre zapisujú do JavaScript kódu. API ponúka možnosť upraviť štýl mapy: farbu pozadia, hraníc, areálov, farebnú škálu zobrazovaných údajov. V ďalšej časti sa API venuje rôznym „udalostiam“ myši (mouse events). Ide o definovanie akcií, ktoré sa vykonajú pri rôznych udalostiach (kliknutie, dvojité kliknutie, ukázanie kurzorom na areál). Nastavenie nápovedy (tooltipov) však podlieha načítaniu ďalšej knižnice `jquery.qtip.js`. API ponúka aj možnosť zobrazenia bodových znakov na mape, ktoré sú lokalizované v súradnicovom systéme WGS-84. Popis zaujímavej funkcie vyplnenia areálov bodkovou vzorkou žiaľ API neobsahuje. Výslednú kvalitu API dokumentácie tejto knižnice možno hodnotiť ako málo podrobnú a nezrozumiteľnú.

Praktické využitie knižnice je dokumentované, na príklade podielu ľudí žijúcich v chudobe v jednotlivých štátoch USA v roku 2015 (obr. 6). Údaje o miere chudoby v štátoch USA boli získané zo stránky „Information Please“, ktorá sa odvoláva na zdroj U.S. Census Bureau. Tabuľka obsahovala údaje o miere chudoby a názov štátu. Mapa USA spracovaná autorom, nie je z geografického hľadiska správna pretože, na nej chýbajú štáty Havaj a Aljaška. Pre identifikáciu jednotlivých štátov slúži atribút `data-id="FIPS kód v tvare USxy"`.

Generovanie JSON súboru s dátami je riešené pomocou nasledovného PHP skriptu.

```

01 <?php
02 if ($result = $mysqli->query("SELECT * FROM `geo_usa_poverty_rate`")) {
03     $i = 1;
04     $numResults = $result->num_rows;
05     $counter = 0;
06     while ($row = $result->fetch_object()) {
07         $i++;
08         $country_code = $row->FIPS;
09         $poverty_rate = $row->poverty rate 2010;
10         //posledny riadok DB (nezapise ciarku na konci)
11         if (++$counter == $numResults) {
12             //prida do ID pre cisla 1 - 9 na zaciatok 0 (01, 02 ...)
13             if ($i <= 8) {
14                 echo "\t\"US0\" . $country_code . "\":{\\"poverty\":" .
15                     $poverty_rate . ", \"id\":"\"US0\" . $country_code . "\"}\n";
16             } else {
17                 echo "\t\"US\" . $country_code . "\":{\\"poverty\":" .
18                     $poverty_rate . ", \"id\":"\"US\" . $country_code . "\"}\n";
19             }
20             //vsetky riadky okrem posledneho
21             else {
22                 //prida do ID pre cisla 1 - 9 na zaciatok 0 (01, 02 ...)
23                 if ($i <= 8) {
24                     echo "\t\"US0\" . $country_code . "\":{\\"poverty\":" .
25                         $poverty_rate . ", \"id\":"\"US0\" . $country code . "\"},\n";
26                 } else {
27                     echo "\t\"US\" . $country_code . "\":{\\"poverty\":" .
28                         $poverty_rate . ", \"id\":"\"US\" . $country_code . "\"},\n";
29                 }
30             }
31         }
32     }
33 }

```

¹⁷ `Kartograph.py`: <https://github.com/kartograph/kartograph.py/>

¹⁸ `Mapshaper`: <http://www.mapshaper.org/>

```

28     }
29   }
30   ?>

```

Vygenerovaný dátový súbor JSON má nasledujúcu podobu:

```

01 {
02   "US01":{"poverty":18.1, "id":"US01"},
03   "US02":{"poverty":9.6, "id":"US02"},
04   ...
05   "US56":{"poverty":11.0, "id":"US56"}
06 }

```

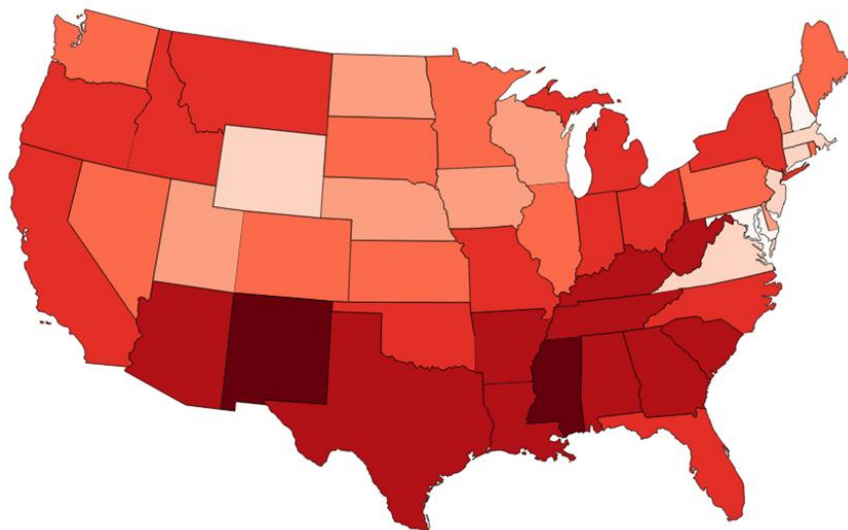
Zápis parametrov mapy je nasledovný:

```

01 $(function() {
02   var map,
03       colorscale,
04       poverty_map= {},
05       w = $('#map').parent().width();
06   // initialize qtip tooltip class
07   $.fn.qtip.defaults.style.classes = 'ui-tooltip-bootstrap';
08   $.fn.qtip.defaults.style.def = false;
09   $.getJSON('usa_data.php', function(poverty_map) {
10     $.get('maps/usa.svg', function(svg) {
11       $.each(['Reds'], function(i, brewer) {
12         var div = $('<div />').addClass('mapnail');
13         $('#map').append(div);
14         var div = $('<div />').addClass('mapnail');
15         $('#map').append(div);
16         var map = kartograph.map(div, w * 0.6, w/3),
17             color = chroma.scale(brewer).domain(poverty_map, 7,
18               'linear', 'poverty');
18         map.setMap(svg);
19         map.addLayer('usa', {
20           styles: {
21             'stroke-width': 0.7,
22             fill: function(d) {
23               return color(poverty_map[d.id].poverty); },
23             stro-ke: function(d) {
24               return color(poverty_map[d.id].poverty).darker(); }
24           },
25         });
26       });
27     });
28   });
29 });

```

Práca s touto knižnicou je pomerne náročná. Od používateľa si vyžaduje rozsiahle znalosti programovania v JavaScripte. Ďalšou nevýhodou tejto knižnice je používanie ďalších knižníc, ktoré pridávajú funkcionality (napr. vyššie spomínaná knižnica jquery.qtip.js, knižnica Chroma.js, ktorá vytvára farebné spektrum tematickej mapy). Knižnici chýbajú základné funkcie, ako sú napríklad „zoom“, možnosť zobrazit' legendu atď. Absentujúca legenda je kartografický nedostatok, ktorý výrazne znižuje možnosti použitia tejto knižnice v oblasti tematického mapovania.



Obr. 6 Výsledná mapa podielu ľudí žijúcich v chudobe v štátoch USA vytvorená pomocou knižnice Kartograph (Výsledok je dostupný na adrese: http://geo.treska.eu/kartograph/usa_chudoba.php)

Záver

Cieľom príspevku bol pokus o zhodnotenie a opísanie možností distribúcie priestorových dát v prostredí internetu pomocou vybraných JavaScript knižníc. Jednotlivé Web Mappingové knižnice, ktoré sme v krátkosti opísali v príspevku, boli testované na konkrétnych príkladoch a tvorbe tematických máp (s reálnymi dátami). Najzásadnejšou podmienkou výberu knižníc, bola možnosť práce s vlastnou podkladovou mapou v požadovanom formáte danej knižnice, pričom postup jej vytvorenia mala každá knižnica opísaná na svojej domovskej adrese. Všetky testované knižnice sú relatívne často aktualizované, až na Kartograph, kde autor ukončil vývoj a vyzýva používať knižnicu D3.js. Naopak, najčastejšie aktualizovanou knižnicou bola amMaps. Kompatibilita všetkých knižníc bola na dobrej úrovni. jVectorMap je dokonca kompatibilný s verziou Internet Exploreru 6, ktorý bol vydaný v roku 2001 a poslednýkrát aktualizovaný v roku 2008. V dnešnom dynamickom svete mobilných technológií je dôležitá aj optimalizácia internetových stránok pre tento segment, ktorú zabezpečovala jediná knižnica amMaps. V prípade Kartographu kompatibilita nebola uvedená, pri testovaní sme sa však nestretli s problémami pri zobrazovaní. Jedným z pozitívnych momentov využitia týchto knižníc, bola podpora zobrazovania bodov pomocou zemepisných súradníc s následnou možnosťou vytvárať jednoduché bodovo lokalizované diagramy. Z hľadiska posudzovania výslednej mapy ako kartografického diela, možno za najväčší nedostatok použitých knižníc považovať absenciu implementácie grafickej mierky v jednotlivých API. Knižnice jQuery Mapael a jVectorMap majú veľmi podobnú funkcionálnosť, práca s nimi je pomerne jednoduchá a efektívna. Ponúkajú síce len základné funkcie, avšak vďaka tomu je ich dokumentácia veľmi prehľadná a ľahko zrozumiteľná. Pri hodnotení práce s knižnicou a vytváraní ukážok, môžeme ako najťažkopádnejšiu označiť knižnicu Kartograph. Jedným z hlavných dôvodov bola nedostatočná dokumentácia. Pri vytváraní ukážky sa nám nepodarilo vytvoriť funkčnú interaktívnu mapu, v ktorej sa navyše nezobrazovala legenda. Naopak, za najvydarenejšiu knižnicu možno z nášho pohľadu považovať knižnicu amMaps. Práca s ňou je veľmi prehľadná a pohodlná, k čomu vo výraznej miere prispieva veľmi dobre napísaná dokumentácia, ako aj množstvo ukážkových príkladov, na ktorých je možné pochopiť základné vlastnosti a funkcionálnosť tejto knižnice. Ako jediná knižnica ponúka nástroj exportu a tlače, čo znamená, že zobrazovanú mapu je možné uložiť v predefinovaných formátoch (JPG, PNG, PDF a SVG). Ďalšou najnovšie pridanou vlastnosťou a z kartografického hľadiska veľmi zaujímavou, je prepínanie medzi rôznymi kartografickými zobrazeniami. Au-

tori tejto knižnice pravidelne pridávajú nové funkcie a vylepšenia, dokonca ponúkajú ďalšie dve knižnice na tvorbu grafov, vďaka čomu sa AmMaps a AmCharts (táto pozostáva z dvoch základných knižníc – „Charts“ a „Stock Chart“) stávajú veľmi silným nástrojom na vizualizáciu priestorových a nepriestorových dát v prostredí internetu. Na webovej stránke projektu amCharts je uvedené, že tieto knižnice používajú aj nadnárodné korporácie ako sú napr.: Cisco, IBM, eBay, Microsoft, Amazon, NASA, Samsung, Siemens, Audi, SAP a mnoho ďalších. Aj táto skutočnosť hovorí nepriamo o vhodnosti použitia tohto nástroja pri tvorbe on-line máp v prostredí internetu. Výsledné hodnotenie knižníc sa opiera o vlastné skúsenosti autorov a ich zručnosti pri využívaní podobných nástrojov. Každý tvorca podobne zameraných aplikácií môže mať preferencie odlišné, avšak celková funkcionálna a kompatibilita hrajú nesmierne dôležitú úlohu v ktoromkoľvek prípade nasadenia týchto nástrojov do prostredia internetu.

Príspevok vznikol vďaka projektu APVV-0018-12 „Humánno-geografické a demografické interakcie, uzly a kontradikcie v časopriestorovej sieti“.

Literatúra

- BAČÍK, V. (2008). Vizualizácia geodát v prostredí internetu - teoretické východiská a praktické riešenia na vybranom príklade. In *Zborník prác CEVASTATU, Bratislava*, INFOSTAT, 16 s.
- BAČÍK, V. (2015). Možnosti formátu TopoJSON a knižnice D3 pri vizualizácii geodát v prostredí internetu. *Geographia Cassoviensis*, 9, 1, Košice (Ústav geografie, Prírodovedecká fakulta UPJŠ), s. 5-16.
- CAJTHAML, J. (2006). Historické mapy v prostredí mapového serveru. In: *Geoinformatics FCE CTU*. Praha, s. 95-100 [cit. 2015-04-18]. Dostupné na: <http://projekty.geolab.cz/gacr/a/files/cajt_geoctu_06.pdf>
- ČERBA, O. (2011). Analýza definíc pojmu mapa [online]. *Kartografické listy*, roč. 19, s. 31-37. [cit. 2015-06-18]. Dostupné na: <<http://gis.fns.uniba.sk/kartografickelisty/archiv/KL19/3.pdf>>
- ČERBA, O. (2006). Mapy na internete. Plzeň, Západočeská univerzita v Plzni. [cit. 2015-06-22]. 10 s. Dostupné na: <https://www.academia.edu/267921/Mapy_Na_Internetu>
- VOŽENÍLEK, V. (2007). Agenda súčasnej počítačovej kartografie. In *GIS Ostrava 2007*, Sekce 7: Vizualizace prostorových dat, 8 s. [cit. 2015-06-20]. Dostupné na: <http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2007/sbornik/Referaty/Sekce7/Vozenilek.pdf>

S u m m a r y

Examples of the use of JavaScript libraries for the creation of thematic maps on the Internet

The paper focuses on options in the distribution and visualization mainly of spatial data in the Internet environment. The key objective was to explain certain basic properties of selected JavaScript libraries for the purpose of distribution of thematic maps on the Internet. The paper is intended mainly for individuals who wish to publish thematic interactive maps with their own underlying layer (map), without any use of commercial tools and without any use of other API services (such as Yahoo Maps, Bing Maps, OpenStreetMap, Google Maps, etc.). Here it should be noted that the knowledge of programming languages (JavaScript, PHP) and database systems is a must. Without the necessary knowledge of these means of communication, their full valued implementation is not possible. However, a particular documentation is available, and persons interested in the development of map outputs can fully enjoy their benefits in the creation of their own applications.

The number of libraries designated for the creation of map outputs on the Internet is very high and it is impossible to cover them fully in the single paper. The following libraries were included in the final selection: jQuery Mapael, jVectorMap, Kartograph, amCharts. All of them are written in the JavaScript programming language. Here we should mention the absence of the D3.js library, which is currently one of the most dynamic visualization libraries used on the Internet. It works with the TopoJSON format and description of which is not the subject of this paper.

The selected libraries have many common and also different elements. Map visualization as such is in the form of JSON files containing the required parameters. In the context of arrangement of necessary attributes, all examples used the MySQL database, whereas the PHP language was used for communication with the database. Working with the library is limited to a certain extent by the availability of documentation (API) as well as by the knowledge of JavaScript by the author of the application.

All tested libraries are updated relatively frequently, except of Kartograph where the author himself recommends to use the D3.js library. The compatibility of particular libraries with Internet browsers is on a very good level; JVectroMap is even compatible with the Internet Explorer 6 version, which was released in 2001 and last updated in 2008. In the present dynamic world of mobile technologies, also the website optimisation is important for this segment, which was ensured by the only library, AmMaps. For Kartograph the compatibility was not specified, however, in the testing we did not encounter any problems in the displaying. In the context of assessment of the final map as a cartographic work, we can consider the absence of a graphic scale for particular API as the biggest deficit of libraries. JQuery Mapael and JVectorMap libraries have very similar functionality, the work with them is relatively simple and effective. They offer only basic functions, but thanks to it their documentation is very transparent and easily understandable. In assessing the work with the library and developing of examples, we can mark Kartograph as the most cumbersome library. One of the main reasons was insufficient documentation. On the contrary, in our opinion, the best prepared library is the AmMaps library. It is very transparent and comfortable to work with it, to a significant extent thanks to very precise documentation as well as many demonstration examples on which it is possible to understand the basic properties and functionality of this library. As the only library, it offers export and printing tools, which means that the map being displayed can be saved in predefined formats (JPG, PNG, PDF and SVG). Another newest attribute and a very interesting one in the cartographic context, is switching between various map projections. The authors of this library add new functions and improvements on a regular basis, and even offer two other libraries for the creation of graphs, thanks to which amMaps and amCharts become a very powerful tools for the visualization of spatial and nonspatial data on the Internet. The final evaluation of libraries is based on own experiences of authors and on their skills in the application of similar tools. Every author of similar applications can have different preferences, but the overall functionality and compatibility play a very important role in any deployment of these tools on the Internet.

Fig. 1 Classification of digital maps (Čerba, 2006)

Fig. 2 Schematic illustration of functioning of used JavaScript libraries (Bačík, 2008)

Fig. 3 The resulting map of registered unemployment rate generated by the use of jQuery Mapael library (The result is available at: http://geo.treska.eu/mapael/slovakia_nezamestnanost.php)

Fig. 4 The resulting map of urbanisation rate generated by the use of JVectorMap library (The result is available at: <http://geo.treska.eu/jvectormap/venezuela.php>)

Fig. 5 The resulting map of departures from Bratislava Airport generated by the use of AmMaps library (The result is available at: http://geo.treska.eu/ammap/odlety_bts.php)

Fig. 6 The resulting map of people living in poverty generated by the use of Kartograph library (The result is available at: http://geo.treska.eu/kartograph/usa_chudoba.php)

Tab. 1 Overview of used JavaScript libraries and their basic features

Prijaté do redakcie: 10.decembra 2015

Zaradené do tlače: december 2015