

KAPITOLA 10

Práce s API pro mapy

Na začátku byla služba MapQuest (www.mapquest.com) stránkou, která dovolovala uživateli vyhledat mapy a získat itinerář cesty po Spojených státech. Během éry dot-com se MapQuest stala neuvěřitelně populární, takže se dostala i na burzu. MapQuest následně upoutala pozornost America Online, která tuto službu nakonec v roce 2000 získala do svého vlastnictví. Je pravda, že i ostatní vyvinuli konkurenční stránky s mapami (nejpozoruhodnější z nich jsou od Yahoo! a Microsoftu), ale pro mapy a plánování cest stále zůstala nejpopulárnější MapQuest. Během několika následujících let prošly stránky s mapami koloběhem změn, ovšem v základu zůstaly pořád stejné.

Když v roce 2004 přišly Google Maps (maps.google.com, později local.google.com), přinesly skutečně revoluční rozhraní pro tradiční webové mapové systémy. Místo klasické interakce typu "klikni a čekej", kterou pro otáčení a přibližování map používala MapQuest a další, Google Maps použil pro stahování dalších informací (včetně přiblížení mapy) komunikaci založenou na Ajaxu, což znamenalo, že stránka se nikdy nemusela načítat znovu. A navíc tu byla schopnost chytnout a táhnout mapu, která poskytla uživateli opravdu mimořádný zážitek v online světě s mapami.

Zdokonalování Google Maps vzbudilo zájem o online práci s mapami, a také o možnosti, které pro tyto potřeby nabízí Ajax. Společnosti Yahoo!, Microsoft a dokonce i MapQuest byly donuceny aktualizovat své mapové projekty takovým způsobem, aby mohly lépe konkurovat mapám od Google. Dosáhly toho – jak jinak – s použitím Ajaxu, a s použitím různých vzorů pro uživatelská rozhraní.

S nástupem mnoha různých vývojových trendů se vývojáři samozřejmě inspirovali novým rozhraním, které mapy od Google používaly (a také ostatními ajaxovými aplikacemi). Mnoho webových vývojářů pak dokázalo prostřednictvím reverzního inženýrství vložit Google Maps do svých vlastních stránek, čímž v podstatě všem ukázali ohromné možnosti Ajaxu. Ačkoliv to nebylo nijak na škodu, tato skutečnost otevřela společnosti Google oči, což vedlo ke zveřejnění API Google Maps. Společnosti Yahoo!, Microsoft a MapQuest pak samozřejmě následovaly tohoto příkladu (stejně jako předtím), takže i ony poskytly ajaxové API pro práci s mapami, což zaplavilo svět mnoha různými technologiemi pro vložení map do webových stránek.

Vzestup mashups

Blízký příbuzný různých typům API pro mapy je tzv. koncept mashups. Mashups jsou webové aplikace, které na jednom místě kombinují informace z mnoha zdrojů tak, aby vám poskytly nové uživatelské prožitky. Tyto informace přitom nejsou umístěny v jediném zdroji – je tomu právě naopak. Informace se získávají z mnoha různých zdrojů, které zveřejňují užitečné informace např. prostřednictvím webových služeb, RSS atd. Mashups tradičně kombinují tyto informace s mapou.

Služba Chicago Crime (www.chicagocrime.org) je obecně považována za první mashup, který zkombinoval informace o kriminalitě v Chicagu s mapou vygenerovanou pomocí API Google Maps. Je zajímavou skutečností, že vývojáři této služby integrovali Google Maps do své aplikace dlouho před tím, než bylo dostupné oficiální API od Google. Postupem času (a za použití neustále se rozvíjejícího API Google Maps) se Chicago Crime rozrostlo do rozsáhlé aplikace, která pokrývá téměř všechny aspekty zločinu v oblasti Chicaga s rozdělením typu zločinu na pouliční, okresní a mnohé další.

Dalším populární projektem ve sféře mashups byl Housing Maps (www.housingmaps.com), který kombinuje seznamy ubytování z Craig's List (www.craigslist.org) s mapou vygenerovanou pomocí Google Maps. Mapa je používána pro zobrazení lokace, kde jsou navíc umístěny různé potřebné informace, jako třeba adresy či fotografie dostupných ubytovacích kapacit.

Abyste mohli vytvořit mashup s mapou, musíte mít přístup k informacím o poloze. Většina takových informací je reprezentována fyzickou adresou ulice. Tyto adresy musí být asociovány ke specifickým pozicím na mapě, což je možné udělat prostřednictvím techniky geokódování.

Geokódování

Geokódování je proces, během kterého jsou informace spojovány s konkrétními geografickými body ve světě. Tyto body jsou identifikovány pomocí stupňů v zeměpisné šířce a délce, které si možná pamatujete ze základní školy jako severní-j jižní a východní-západní míry (v tomto pořadí). Možná vás překvapí, že většina API pro práci s mapami nezná pozici adres jako takových – znají pouze pozici bodu, který je dán zeměpisnou šířkou a délkou. Všechny adresy musejí být převedeny na množinu bodů, a teprve pak mohou být tyto body zobrazeny na mapě.

Všechny API pro práci s mapami vyžadují použití desetinného čísla pro vložení hodnoty zeměpisné šířky i délky. To je pravděpodobně odlišné od toho, co jste se naučili ve škole, kde se zeměpisná šířka a délka udávala pomocí stupňů, minut a sekund. Pokud máte pozici v tomto formátu, budete muset použít nějaký převaděč pro získání hodnot v desetinném formátu, které pak můžete následně použít pro dané API.

Stránky pro geokódování

Většina zemí poskytuje geokódovanou informaci o terénu. Ve Spojených státech například U.S. Census Bureau geokóduje téměř všechny dálnice a pozemní silnice v zemi. Tato data jsou k dispozici veřejnosti a mohou být přístupna skrze systém Tiger (Topologically Integrated Geographic Encoding and Referencing system, www.census.gov/geo/www/tiger). Procházení všech těchto informací je ovšem obtížným procesem, protože pouze pro Spojené státy mají tato data více než 20 GB. S novým zájmem o mapy a mashups se na webu objevilo mnoho nových služeb, které poskytují snadnější přístup k těmto informacím.

- **Geocoder.us** (www.geocoder.us). Tato webová stránka umí vrátit zeměpisnou šířku a délku jakékoliv adresy ve Spojených státech. Jednoduše jděte na stránku a vložte adresu. Vracená informace poskytuje oba formáty pozice: stupně/minuty/sekundy i desetinný formát.
- **Travel GIS** (www.travelgis.com/geocode). Tato stránka nabízí geokódovanou informaci pro 24 zemí prostřednictvím velmi jednoduchého rozhraní. Adresy jsou vráceny pouze v desetinném formátu.
- **WorldKit GeoCoder** (<http://brainoff.com/geocoder>). Je to jednoduchá webová stránka, kde můžete vložit adresu a získat zeměpisnou šířku a délku v desetinném formátu společně s některými dalšími informacemi. Tato stránka vám poskytne mapu světa a přesně určí každou pozici, kterou jste do mapy vložili pomocí červené tečky. Mapu si můžete přiblížit či oddálit, přičemž stačí kliknout do mapy pro získání požadovaných zeměpisných údajů.

Služby pro geokódování

Ačkoli jsou stránky poskytující geokódované informace velmi užitečné, jsou pouze minimem pro vytvoření samotného mashupu. Většina mashup totiž vyžaduje dynamické vyhledávání geokódovaných informací, protože uživatel je s aplikací v přímé interakci. V tomto případě vám pomůže několik webových geokódovacích služeb, které nabízejí vyhledávání adres v reálném čase:

- **Yahoo! Maps Geocoding Service** (<http://developer.yahoo.com/maps/rest/V1/geocode.html>). Tato služba vrací XML kód, který obsahuje zeměpisnou šířku a délku, adresu ulice, města, státu a zip kód dané adresy. Tato služba je určena pro nekomerční účely, takže jste omezeni na 5 000 vyhledávání za den. Pro získání Yahoo! Application ID se navíc musíte registrovat na http://api.search.yahoo.com/webservices/register_application.
- **Google Maps Geocoding Service** (www.google.com/apis/maps/documentation/#Geocoding_Examples). Toto API může být nastaveno tak, aby vracelo data v XML, KML (Google's Keyhole Markup Language), CSV nebo JSON, přičemž vrací všechny informace o dané adrese – její souřadnice a plnou informaci o adrese (země, zip kód atd.). Stejně jako v případě Yahoo!, je i tato služba určena pro nekomerční použití. Zde je ovšem limit 50 000 hledání na den. Před použitím Google Maps Geocoding Service se musíte zaregistrovat na www.google.com/apis/maps/signup.html, abyste získali klíč k API. Toto API v neposlední řadě poskytuje i javascriptový přístup pro geokódování informací.

API Google Maps

Když se poprvé objevily Google Maps, byly obětí mnoha hackerů. Tito lidé byli totiž fascinováni touto novou generací ajaxových aplikací, která uměla věci, jež předtím nikdo neviděl. Vývojáři z celého světa chtěli vědět, jak tato aplikace pracuje a jak by to mohli využít pro sebe. Ačkoliv tito hackeři nakonec neudělali společnosti Google žádné škody, otevřelo to oči lidem v Mountain View v Kalifornii, takže nějakou chvíli poté bylo zveřejněno API Google Maps a nabídnuto veřejnosti.

Jak API pracuje?

API Google Maps je jedno z nejzajímavějších využití Ajaxu v tom ohledu, že nepotřebuje používat XHR nebo plovoucí rámce (alespoň ne nezbytně). API místo toho používá dynamické vlastnosti obrázků pro získání nových informací ze serveru na požádání. Ačkoli nepoužívá obrázkovou techniku popsanou v kapitole 2, pracuje se stejnou základní myšlenkou – zdroje obrázků se mohou v kteroukoliv chvíli změnit. API Google Maps používá tuto funkčnost pro vytvoření iluze posunu jednoho velkého obrázku, i když ve skutečnosti je to tak, že se načítají pouze malé části celého obrázku, čímž je vytvořen dojem posunu obrázku mnohem většího.

Prvotní pohled na mapu je rozdělen do několika obrázků, které jsou umístěny vedle sebe, čímž vyvolávají dojem jednoho velkého obrázku. Když je mapa poprvé načtena, API určuje, kolik těchto obrázků je nezbytných pro kompletní vyplnění kontejneru mapy. Obrázky jsou uspořádány v mřížce, která přesahuje okraje kontejneru mapy. Pokud je mapa uživatelem přiblížena, každý z obrázků má nastavenou novou URL adresu, což načte nové obrázky do HTML prvků ``. Tím je vytvořena iluze asynchronního přiblížení – ve skutečnosti je ovšem nový obraz položen na starý, což je nahrazovací technika, která existuje už od konce devadesátých let.

Když je mapa uživatelem posunuta do stránky, díky zajímavě použitému JavaScriptu to vypadá, jako by mapa byla nekonečným obrázkem. Obrázky se skutečně pohybují tak, jak uživatel táhne myši, ale jakmile obrázky zmizí z viditelného prostoru mapy, jsou odstraněny a umístěny na jiný konec mapy. Například obrázky, které zmizí z pravé strany mapy, jsou umístěny mimo viditelnou oblast levé strany mapy a obrázky, jež zmizí ze spodní části mapy, jsou umístěny mimo viditelnou oblast horní strany mapy. A právě tímto je vytvořena iluze, kdy uživatel si myslí, že posunuje jeden velký obrázek. Joel Webber, jeden z prvních vývojářů, který se snažil zjistit, jak Google Maps fungují, přirovnal tuto techniku k výstavbě železniční trati, kdy se vezme jeden kus z konce a přemístí se na začátek – obrázky nejsou vytvářeny, a ani rušeny. Jsou pouze přemísťovány dokola.

Na pozadí sice existuje ještě několik dalších činností, které zajišťují různé dodatečné funkce pro mapu, ale většina práce je prováděna samotnými obrázky.

Začínáme

Pro začátek potřebujete mít nějaký Google účet (například pro přístup k Gmailu). Pokud ho ještě nemáte, jděte na adresu `www.google.com/accounts`, kde získáte informace, jak se zaregistrovat.

Dalším krokem je zajít na adresu `www.google.com/apis/maps/signup.html` a získat klíč k API. K tomu musíte zadat URL adresu, která se bude odkazovat na umístění, kde bude API použito. Toto umístění je konkrétním adresářem na vašem serveru, takže adresy `www.mydomain.com/maps1` a `www.mydomain.com/maps2` budou vyžadovat dva samostatné klíče.

Google Maps API má některá významná omezení, kterých byste si měli být vědomi:

- API je pouze pro nekomerční účely. K získání komerční licence musíte kontaktovat přímo společnost Google.
- Stránka používající API Google Maps nemá žádné omezení ohledně počtu zobrazení této stránky, ovšem pokud předpokládáte více než půl milionu zobrazení za den, měli byste kontaktovat společnost Google pro získání komerční licence.
- Není dovoleno vkládat reklamy, které se objeví uvnitř zobrazovacího pole mapy.
- Vaše stránka musí vždy používat nejnovější verzi API. Společnost Google obecně dává uživatelům měsíc na přechod na nově vydanou verzi.

A dále potřebujete jeden soubor JavaScriptu, který je nutný k tomu, abyste mohli začít používat API Google Maps. Na rozdíl od jiných API zde není možné stahovat soubory lokálně. Místo toho musíte zpřístupnit soubor, který je umístěný na serveru Google Maps. Tento soubor musí obsahovat verzi a váš klíč v následujícím formátu:

`http://maps.google.com/maps?file=api&v={version}&key={your-key}`

Například – pokud je nejnovější verze 2, měli byste mít ve stránce obsažen následující kód:

```
<script type="text/javascript"
src="http://maps.google.com/maps?file=api&v=2&key={your key}"></script>
```

Jakmile je tento soubor vložen do stránky, můžete začít vytvářet svou aplikaci.

Základy Google Maps

Hlavní objekt v API Google Maps se jmenuje `GMap2`. Konstruktor přijímá jeden argument, kterým je prvek, jenž by měl obsahovat mapu. Je doporučováno, aby tento obalový prvek byl prvek `<div>` – tak se zajistí nejlepší kompatibilita a rozšiřitelnost. Tento prvek `<div>` může být oštylován obvyklým způsobem, ovšem jako minimum musí mít nastavenou výšku a šířku. Objekt `GMap2` je dost inteligentní na to, aby mohl pracovat uvnitř stylů nastavených pro obalující prvek `<div>`, takže nebudete muset přistupovat ke kompromisům ohledně designu vaší stránky kvůli zobrazení mapy. Pro vytvoření mapy za použití prvku `<div>` s identifikátorem `divMap` použijte následující kód:

```
var oMap = new GMap2(document.getElementById("divMap"));
```

Jakmile je vytvořen objekt mapy, musíte inicializovat pohled na danou oblast. To je provedeno zavoláním metody `setCenter()`, která přijímá dva argumenty – bod udávající zeměpisnou šířku/výšku a úroveň přiblížení. První argument musí být objekt `GLatLng` (který je vytvořen předáním

zeměpisné šířky a délky v desetinném formátu). Druhý argument je úroveň přiblížení, kde 0 je úplné oddálení, přičemž jakékoliv číslo větší než 0 odkrývá více detailů na mapě. Například následující kód soustřeďuje mapu na Spojené státy tak, aby byla vidět celá země:

```
var oMap = new GMap2(document.getElementById("divMap"));
oMap.setCenter(new GLatLng(32, -92), 3);
```

Existují i prohlížeče, které nemusí API Google Maps podporovat, takže před tvorbou nového objektu GMap2 je nejlepší provést kontrolu za použití funkce GBrowserIsCompatible():

```
if (GBrowserIsCompatible()) {
var oMap = new GMap2(document.getElementById("divMap"));
oMap.setCenter(new GLatLng(32, -92), 3);
}
```

Tyto čtyři řádky kódu jsou vším, co potřebujete k umístění jednoduché mapy na stránku (podívejte se na obrázek 10-1).



Obrázek 10-1. Mapa USA.

Mapa, která je zobrazena pomocí tohoto kódu, je velmi jednoduchá a hodně omezená. Zatímco mapa Spojených států je dobře viditelná, nabízí velmi málo možností pro interakci s uživatelem. Je sice možné pohybovat zobrazovacím polem mapy, ale to je tak zhruba všechno, co nám tato mapa nabízí. Takže přidejme do mapy nějaké ovládací prvky, které zvýší její schopnosti.

Ovládání

Oficiální rozhraní Google Maps na <http://maps.google.com> nabízí velké množství možností, jak může uživatel manipulovat s mapou. Každá z těchto možností je řízena jiným ovládacím prvkem. API Google Maps poskytuje mnoho standardních ovládacích prvků, které mohou být použity buď pro implementaci plného rozhraní Google Maps, nebo pouze pro vybrané součásti, jež jsou nezbytné pro vaše záměry:

- `GLatLngControl`. Obvyklý ovládací prvek pro posun a přiblížení, který je zobrazen na <http://maps.google.com>.
- `GSmallMapControl`. Zmenšená verze předchozího ovládacího prvku. Obsahuje pouze ikony plus/minus a ovládání směru (ovšem bez posuvníku pro přiblížení).
- `GSmallZoomControl`. Zobrazí posuvník přiblížení, ale bez ovladačů pro směr.
- `GScaleControl`. Stupnice udávající jednotky v mílích a kilometrech.
- `GOverviewMapControl`. Oddálený pohled na mapu se zvýrazněným aktuálním zobrazovacím polem.
- `GMapTypeControl`. Ovládací prvek pro výběr typu mapy (klasická mapa, satelitní mapa, hybridní mapa).

Jeden nebo více z těchto ovládacích prvků mohou být přidány do mapy metodou `addControl()`. Každý z těchto ovládacích prvků může být vytvořen bez parametrů a vložen do této metody:

```
oMap.addControl(new GSmallMapControl());
```

Ačkoli toto se provádí nejčastěji po vytvoření objektu `GMap2`, ovládací prvek může být přidán v kteroukoliv dobu. Ovládací prvek může být odstraněn s použitím metody `removeControl()`, pokud tedy na něj máte referenci:

```
var oControl = new GSmallMapControl();  
oMap.addControl(oControl);  
// dále nějaký úkol  
oMap.removeControl(oControl);
```

První tři ovládací prvky, `GLatLngControl`, `GSmallMapControl` a `GSmallZoomControl`, by neměly být použity dohromady, protože všechny zabírají stejné místo na mapě (horní levý roh). `GMapTypeControl` může být bezpečně použit s libovolným, protože zabírá horní pravý roh.

Pokud chcete, aby vaše mapa měla ovládací prvky hned od počátku, měli byste je přidat ihned po vytvoření objektu `GMap2`, ale současně před tím, než zavoláte `setCenter()`, například:

```
if (GBrowserIsCompatible()) {  
    var oMap = new GMap2(document.getElementById("divMap"));  
    oMap.addControl(new GSmallMapControl());  
    oMap.addControl(new GMapTypeControl());  
}
```



```
oMap.setCenter(new GLatLng(32, -92), 3);
}
```

Přidání těchto ovládacích prvků přináší mapu jako na obrázku 10-2.



Obrázek 10-2. Vylepšená mapa s ovládacími prvky.

Posouvání mapy

Použitím několika metod objektu `GMap2` je možné dynamicky ovládat pohled na mapu, která byla jednou načtena. Ačkoli můžete uživateli zpřístupnit různé ovládací prvky mapy pro přibližování a pohyb mapy, někdy může být nezbytné ovládat mapu odděleně. Všechna činnost, která může být vykonána prostřednictvím ovládacích prvků, může být vykonána i přímo – zavoláním metody JavaScriptu pro dané chování.

Metoda `setCenter()` byla použita dříve pro inicializaci zobrazení mapy, ale může být také kdykoliv použita znovu pro zacílení mapy na konkrétní bod. Nové zacílení proběhne okamžitě a bez animace. Pro hladší přechod k novému bodu mapy existuje několik následujících metod:

- `panBy (vzdálenost)`. Specifikuje vzdálenost (jako `GSize`), o kterou by se mapa měla posunout.
- `panDirection(x, y)`. Specifikuje směr, kterým by se mapa měla posunout. Argument `x` by měl být -1 pro pohyb vlevo, 0 aby se mapa nepohybovala, nebo 1 pro pohyb vpravo. Argument `y` by měl být -1 pro pohyb nahoru, 0 aby se mapa nepohybovala, nebo 1 pro pohyb směrem dolů.
- `panTo(center)`. Specifikuje objekt `GLatLng`, který by měl být novým středem mapy. Mapa provede animovaný přesun na tuto pozici (stejně jako `setCenter()` s výjimkou animace).

Tyto metody mohou být použity kdykoliv pro pohyb mapou na novou pozici, například:

```
oMap.panBy(new GSize(20,20)); //Posuň mapu o 20 pixelů v obou směrech  
oMap.panDirection(1, 0); //Posuň mapu doprava  
oMap.panTo(new GLatLng(50, -80)); //Posuň mapu na danou pozici
```

Informační okna (bubliny)

Informační okna poskytují dodatečné informace o bodu na mapě. Na stránkách Google Maps jsou tato okna použita pro poskytnutí adresy o bodu na mapě, ale mohou být použita i pro více účelů. Informační okna vypadají jako dialogové bubliny z kreslených komiksů – zaoblená bílá bublina, která ukazuje na dané místo na mapě (podívejte se na obrázek 10-3).



Obrázek 10-3. Informační okno na mapě.

Základy informačních oken

Informační okno může být pomocí metody `openInfoWindow()` objektu `GMap2` otevřeno kdykoliv. Tato metoda přijímá tři argumenty: objekt `GLatLng` udávající, kde by mělo být informační okno zobrazeno, uzel DOM poskytující obsah informačního okna a nepovinně konfigurační objekt.

Pro otevření jednoduchého informačního okna ve středu mapy použijte následující kód:

```
oMap.openInfoWindow(oMap.getCenter(),  
    document.createTextNode("Center of the map!"));
```

Metoda `getCenter()` objektu `GMap2` vrací objekt `GLatLng` pro střed mapy, což zajistí, že informační okno bude v tomto případě směřovat přesně ke středu. Ačkoliv toto informační okno zobrazuje

pouze text, je pořád nutné vložit jako druhý argument uzel DOM, takže textový uzel je vytvořen se zprávou k zobrazení.

Následuje druhá metoda, `openInfoWindowHtml()`, která dovoluje, aby jako tělo informačního okna byl vložen HTML řetězec místo uzlu DOM. Tato metoda přijímá tři stejné argumenty (bod, na který má být ukázáno, obsah okna a nepovinný konfigurační objekt). Metoda je volána takto:

```
oMap.openInfoWindowHtml(oMap.getCenter(), "<em>Center of the map!</em>");
```

Tento příklad otevře informační okno s textem zvýrazněným pomocí kurzívy (za předpokladu, že zde nejsou použity styly CSS, které by přepisovaly výchozí zobrazení prvku ``). Takto je možné vytvořit text za běhu a zobrazit jej v informačním okně bez potřeby vytvářet objekty DOM.

Konfigurační volby

Třetím argumentem výše zmíněných metod je konfigurační objekt pro informační okno. Tento objekt obsahuje navíc jednu z následujících vlastností:

- `maxWidth`. Maximální dovolená šířka informačního okna v pixelech.
- `onCloseFn`. Funkce, která se má zavolat po zavření informačního okna.
- `onOpenFn`. Funkce, která se má zavolat po otevření informačního okna.

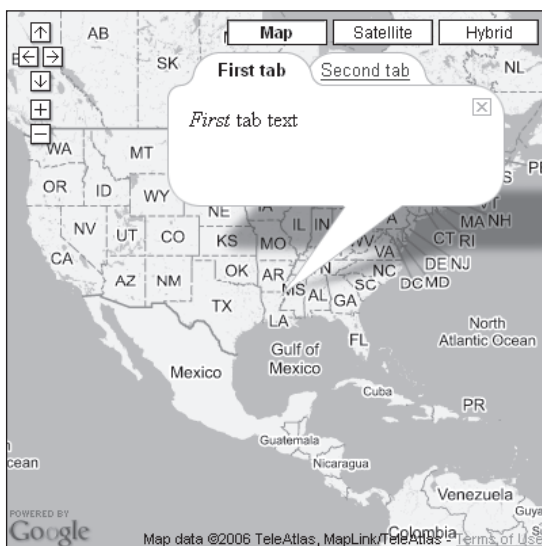
Tento konfigurační objekt může být zahrnut přímo v konstruktoru, například:

```
oMap.openInfoWindowHtml(oMap.getCenter(), "<em>Center of the map!</em>",  
    { onCloseFn: function() { alert("Closed") } });
```

Když spustíte tento kód, je zobrazena výstraha poté, co uživatel klikne na tlačítko pro zavření informačního okna. Obecně řečeno – nastavení vlastnosti `onCloseFn` je nejužitečnější z dostupných možností, protože poskytuje možnost zachytit jinak nevypátratelné události. Vlastnost `MaxWidth` může být nahrazena prostřednictvím CSS a vlastnost `onOpenFn` může být jednoduše napodobena zavoláním náležité funkce ihned po zavolání `openInfoWindow()`, nebo `openInfoWindowHtml()`, protože obě jsou synchronní operace.

Informační okna se záložkami

Novým vylepšením API Google Maps verze 2 je informační okno se záložkami, které může být použito pro zobrazení více informací o konkrétním bodě na mapě, aniž by to zabralo nějaký velký prostor navíc (viz obrázek 10-4).



Obrázek 10-4. Informační okno se záložkami.

Stejně jako obyčejná informační okna může být i verze se záložkami vytvořena dvěma metodami: `openInfoWindowTabs()` a `openInfoWindowTabsHtml()`. Obě tyto metody přijímají tři argumenty: objekt `GlatLng` specifikuji bod, na který má být ukázáno, pole objektu `GinfoWindowTab` reprezentující záložky a nepovinný konfigurační objekt. Rozdíl mezi těmito dvěma metodami má co dělat s daty dostupnými uvnitř každého objektu `GinfoWindowTab`. Když používáte `openInfoWindowTabs()`, každý objekt `GinfoWindowTab` musí být vytvořen za použití řetězce pro titulek záložky a uzlu DOM pro obsah záložky. Metoda `openInfoWindowTabsHtml()` očekává, že každý `GinfoWindowTab` bude vytvořen za použití řetězce pro titulek záložky a řetězce pro jeho obsah, který může obsahovat i HTML kód. Tento kód vytvoří informační okno se dvěma záložkami:

```
var aTabs = [
  new GinfoWindowTab("First tab", document.createTextNode("First tab text")),
  new GinfoWindowTab("Second tab", document.createTextNode("Second tab text"))
];
oMap.openInfoWindowTabs(oMap.getCenter(), aTabs);
```

První část tohoto kódu vytváří pole obsahující dva objekty `GinfoWindowTab`, jejichž obsahem jsou dva textové uzly. Toto pole je pak předáno jako druhý argument `openInfoWindowTabs()` pro zobrazení informačního okna. Pro zobrazení naformátovaného HTML textu místo prostého textu použijte metodu `openInfoWindowTabs()` a připojte obsah záložky jako řetězec:

```
var aTabs = [
  new GinfoWindowTab("First tab", "<em>First</em> tab text"),
  new GinfoWindowTab("Second tab", "<em>Second</em> tab text")
];
oMap.openInfoWindowTabsHtml(oMap.getCenter(), aTabs);
```

Tento kód vytváří výsledek, který jste viděli na obrázku 10-4. Všimněte si, že pouze tři řádky se změnilo – dva řádky definující objekty `GInfoWindowTab` a jeden řádek s voláním metody.

Konfigurační objekt může obsahovat stejné možnosti jako konfigurační objekt, který byl použit s obvyčejnými informačními okny, ale také další vlastnost `selectedTab`. Hodnotou této vlastnosti je celé číslo udávající záložku, která by měla být vybrána, když je informační okno zobrazeno ve výchozím stavu – standardní hodnota je 0, což vybírá první záložku. Abychom vybrali druhou záložku, následující kód přidává konfigurační objekt s vlastností `selectedTab` nastavenou na 1:

```
var aTabs = [
new GInfoWindowTab("First tab", "<em>First</em> tab text"),
new GInfoWindowTab("Second tab", "<em>Second</em> tab text")
];
oMap.openInfoWindowTabsHtml(oMap.getCenter(), aTabs, { selectedTab: 1 });
```

Zvětšeniny mapy

Zvětšenina mapy je speciální typ informačního okna, které ukazuje přiblížený pohled na konkrétní bod na mapě. Obsahem tohoto informačního okna je menší verze hlavní mapy, která je doplněna o tlačítka pro změnu typu mapy a pro ovládání přiblížení (viz obrázek 10-5).



Obrázek 10-5. Zvětšenina mapy.

Metoda `showMapBlowup()` je použita pro otevření informačního okna se zvětšeninou mapy. Tato metoda přijímá dva argumenty: objekt `GLatLng`, který specifikuje bod, na nějž má být ukázáno a nepovinný konfigurační objekt. Například pro zobrazení zvětšeniny středu mapy použijte následující kód: