

Obsah

Zadanie	2
Kontaktný formulár	2
Google Maps.....	9

Zadanie

Teraz je tvojou úlohou zvládnuť postupne konkrétne postupy a implementácie štandardných funkcií, ktoré sa bežne na jednotlivé weby implementujú.

Kontaktný formulár

Vytvor kontaktný formulár podobný ako nájdeš na <http://www.dufeksoft.com/sk/kontakt>

Napíšte nám



Meno

E-mail

Telefón

Sem napíšte správu

Šup s tým k nám

Vyplnenie a odoslanie kontaktného formulára by malo odoslať e-mail majiteľovi stránky a aj človeku, ktorý kontaktný formulár vyplnil.

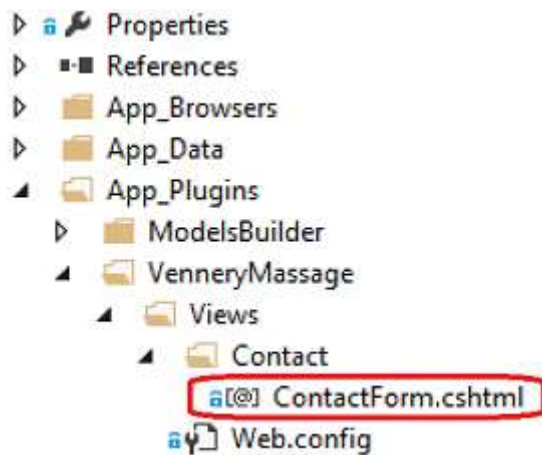
Kontaktný formulár by sa mal vkladať do rozhrania CMS Umbraco ako Macro, takže si nezabudni naštudovať ako sa používa macro v CMS Umbraco. Je to naozaj užitočný nástroj.

Ďalšou vecou je, aby samotné View pre kontaktný formulár bolo vytvorené tak, že bude ako doplnok, ktorý sa bude nachádzať v adresári App_Plugins. Na to sa používa `SurfaceController`, takže aj o ňom si naštuduj ako sa používa.

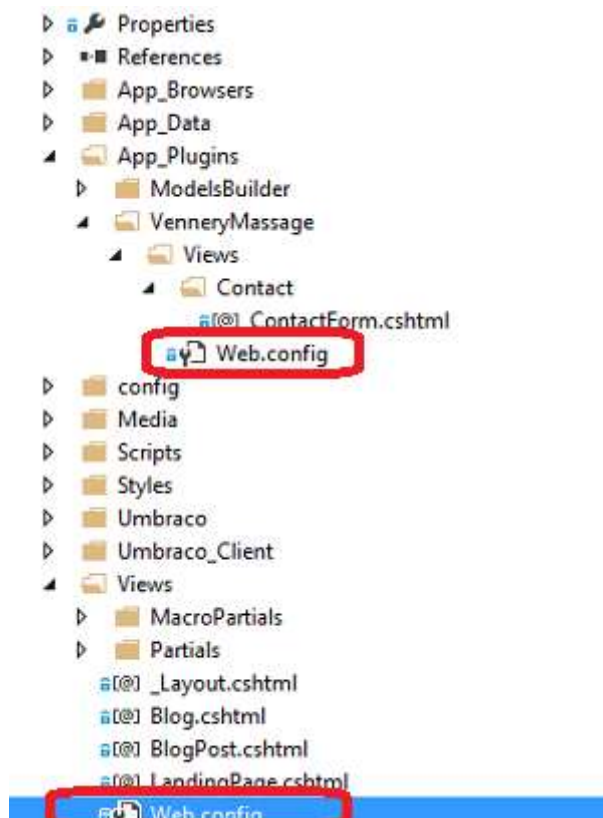
Tu je ukážka kódu ako to môže byť v zdrojových kódach zapísané

```
[PluginController("VenneryMessage")]  
public class ContactController : SurfaceController  
{  
    // GET: ContactFormSurface  
    public ActionResult Index()  
    {  
        return View("ContactForm", new ContactModel());  
    }  
}
```

A tu je ukážka umiestnenia View súboru



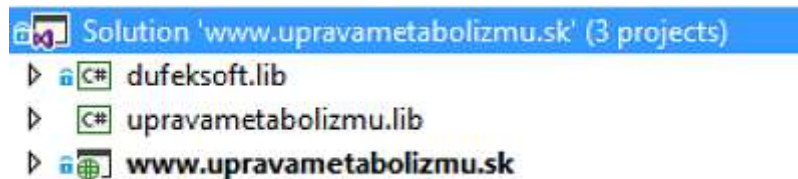
Neprehliadni súbor Web.config, ktorý je na obrázku tiež zobrazený a nachádza sa v adresári App_Plugins/VenneryMessage/Views. Tento súbor je veľmi dôležitý. Ak tam nebude, nebudú sa View pre Plugin zobrazovať a pokus o ich zobrazenie skončí s chybou. Prekopíruj tam súbor, ktorý sa nachádza v roote webu v adresári Views ako to zobrazuje nasledovný obrázok.



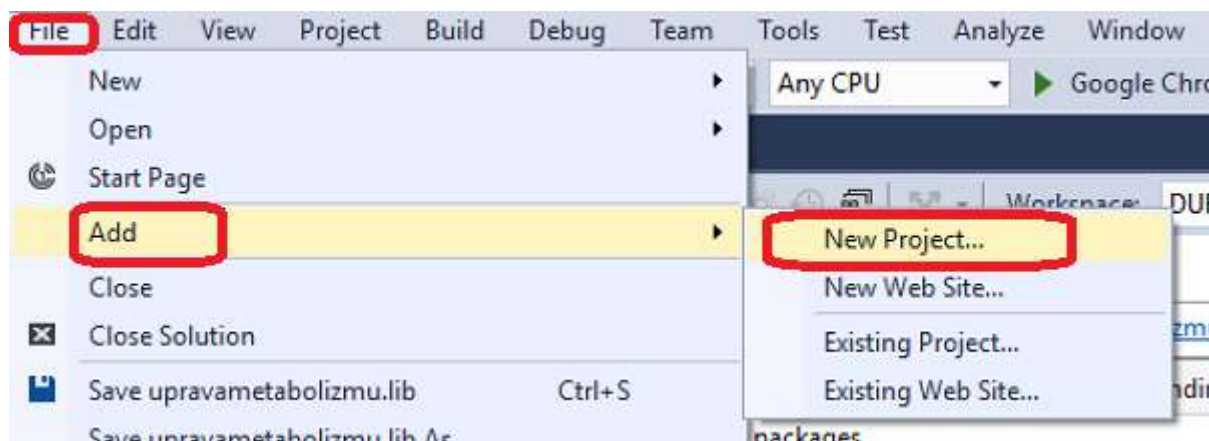
Ak nebudeš mať k dispozícii na odosielanie e-mailov svoj e-mailový server, napíš mi a vytvorím pre teba prístup k nášmu.

Pri tvorbe webových stránok je dôležité zdrojové kódy usporiadať rozumne, aby sa ľahko udržiavali a boli znovu použiteľné aj pri tvorbe ďalších webových stránok. Preto urob príslušný Model pre kontaktný formulár aj `SurfaceController` v samostatnom projekte.

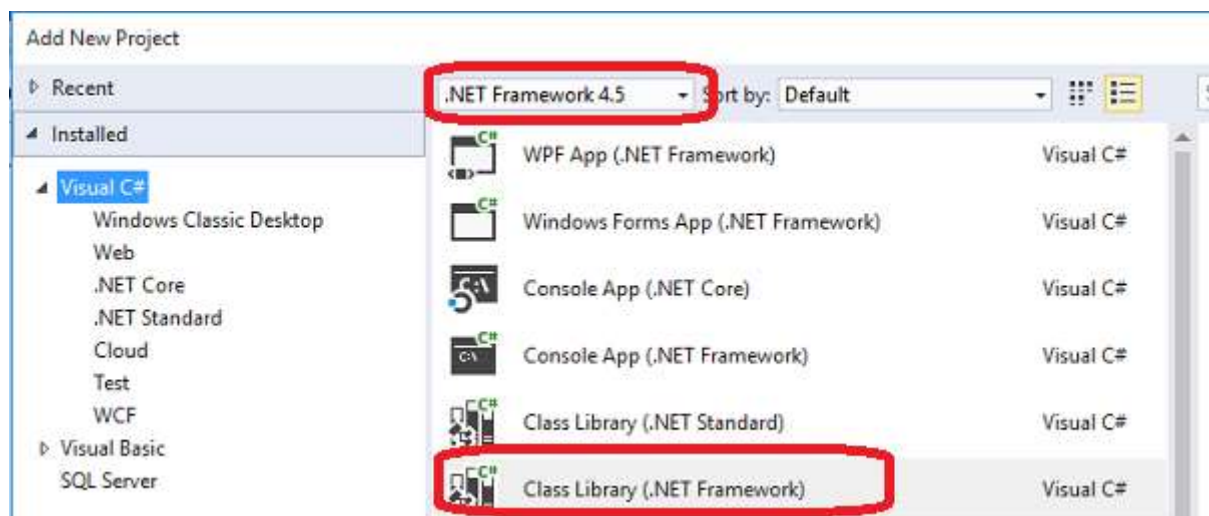
Solution pre konkrétny webový projekt môže potom pozostávať z viacerých projektov ako to zobrazuje nasledovný obrázok.



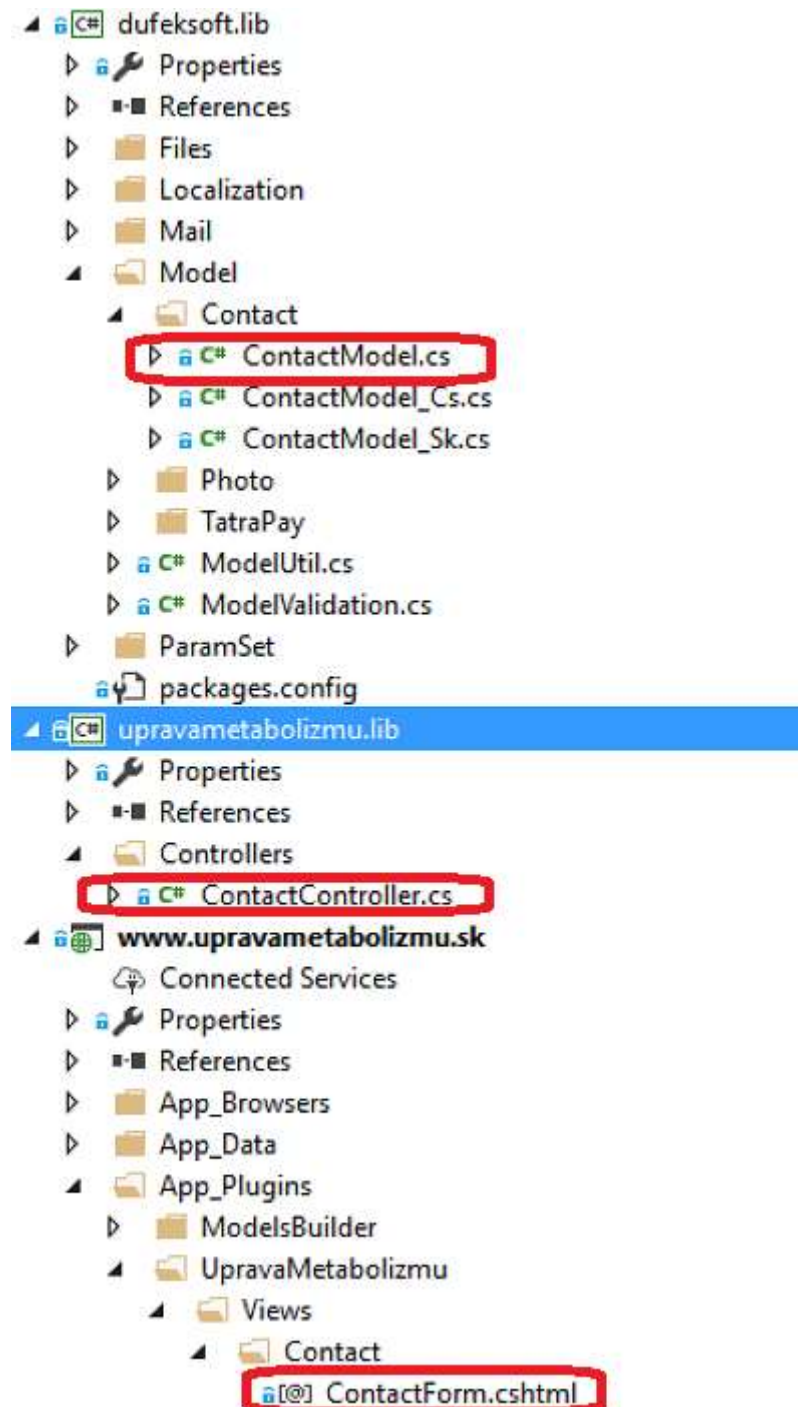
Pridanie nového projektu urobíš nasledovne.



A vyber typ projektu podľa nasledovného obrázka.

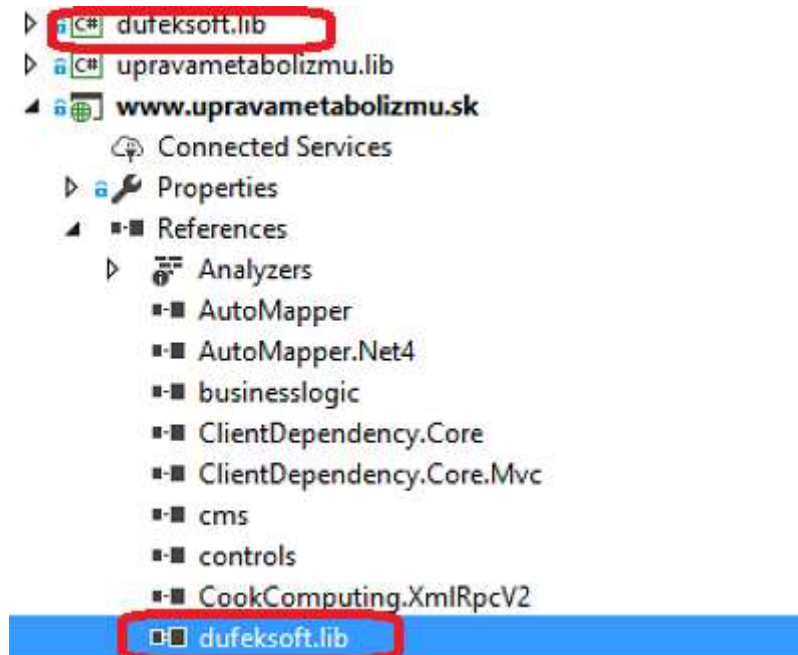


Podľa potreby môžeš potom jednotlivé prvky celého riešenia rozdeliť medzi rozličné projekty ako to ilustruje nasledovný obrázok pre prípad kontaktného formulára.

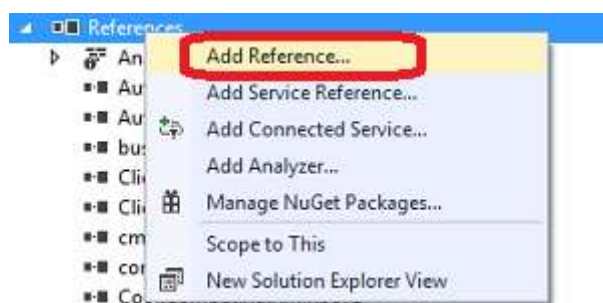


Ak sa v niektorom projekte odvolávaš na prvky z iného projektu, musíš na daný projekt vytvoriť referenciu, aby to vedel systém medzi sebou prepojiť.

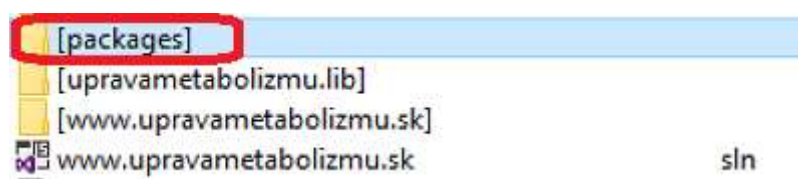
Príklad je na nasledovnom obrázku. Aby projekt `www.upravametabolizmu.sk` poznal prvky z projektu `dufeksoft.lib`, tak je potrebné na projekt `dufeksoft.lib` vytvoriť referenciu v projekte `www.upravametabolizmu.sk`.



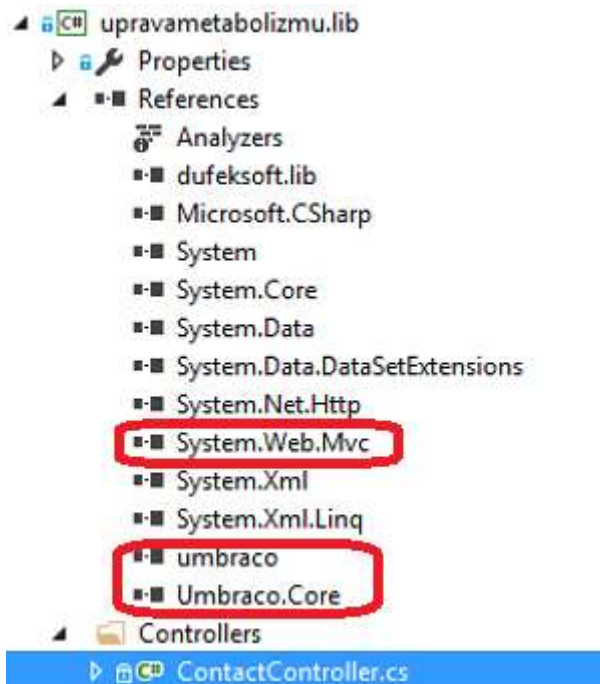
To urobíš funkciou **Add Reference** cez kontextové menu po kliknutí pravým tlačidlom myši na **References**.



Referencie môžeš vytvárať na projekty v rámci rovnakého Solution, ale aj na iné knižnice, ktoré máš u seba na disku. Takto sa referencujú napríklad knižnice Umbraco, ktoré ti NuGet nainštaloval do adresára Packages.



Aby si mohol vytvoriť funkčný samostatný projekt, ktorý bude obsahovať `SurfaceController` musíš doplniť manuálne do nového projektu minimálne referencie na knižnice podľa nasledovného obrázka.

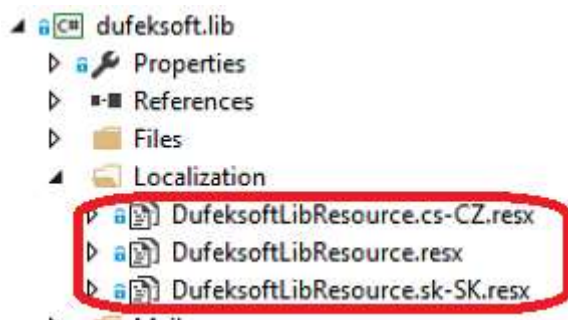


System.Web.Mvc je referencia na súbor
packages\Microsoft.AspNet.Mvc.5.2.3\lib\net45\System.Web.Mvc.dll

umbraco je referencia na súbor
packages\UmbracoCms.Core.7.6.1\lib\net45\umbraco.xml

Umbraco.Core je referencia na súbor
packages\UmbracoCms.Core.7.6.1\lib\net45\Umbraco.Core.xml

A aby toho ešte nebolo málo, tak ešte posledná požiadavka. Model pre kontaktný formulár urob viacjazyčný. Využiješ k tomu systém lokalizácie, ktorý má .NET a Visual Studio k dispozícii. Texty pre jednotlivé jazyky sa uložia do špecializovaných resource súborov podobne ako to znázorňuje nasledovný obrázok.



Model pre kontaktný formulár potom môže byť nasledovný.

```
public class ContactModel
{
    static ResourceManager _rm = null;

    public static ResourceManager RM
    {
        get
        {
            if (_rm == null)
            {
                _rm = new ResourceManager(typeof(dufeksoft.lib.Localization.DufeksoftLibResource));
            }

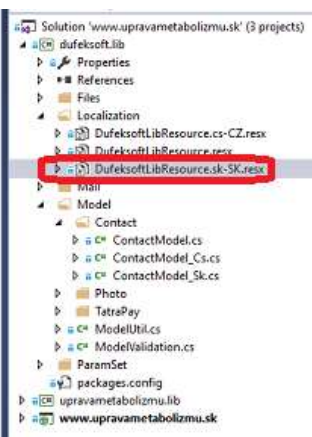
            return _rm;
        }
    }

    /// <summary>
    /// Name
    /// </summary>
    [Required(ErrorMessageResourceType = typeof(DufeksoftLibResource), ErrorMessageName = "RequiredName")]
    [Display(ResourceType = typeof(DufeksoftLibResource), Name = "LabelName")]
    public string Name { get; set; }
}
```

Ako vidieť na obrázku vyššie, tak pre jednotlivé popisy atribútov modelu a chybové hlásenia sa použijú identifikátory pre resource stringy.

Editovaný resource súbor pre slovenský jazyk potom bude nasledovný.

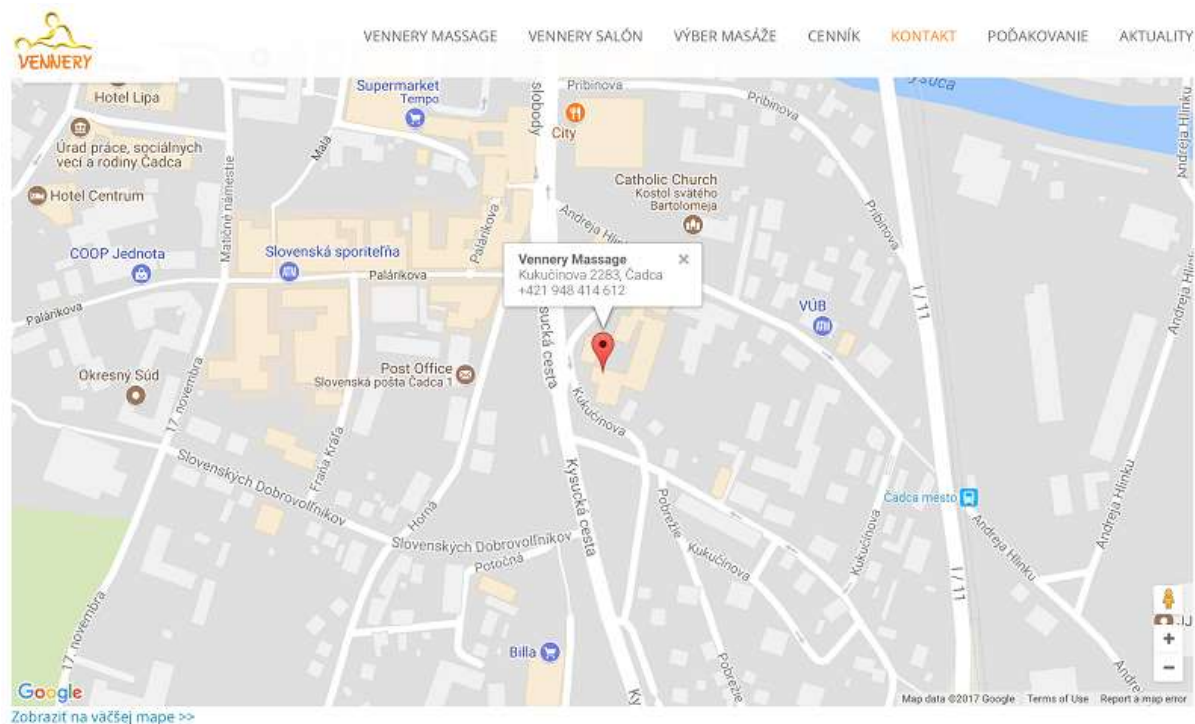
Name	Value	Comment
ContactRequestCultureId	_Sk	
ContactRequestSubject	Vaša správa: (0)	
ContactRequestTemplate	ContactSendSuccess_Sk	
InvalidEmail	Neplatná e-mailová adresa.	
LabelEmail	E-mail	
LabelErrorInfo	Odoslanie správy bolo neúspešné. Prosím opravte chyby a skúste	
LabelHeader	Napište nám správu	
LabelMsgSent	Vaša správa bola úspešne odoslaná.	
LabelMsgText	Sem napíšte správu	
LabelName	Meno	
LabelSeeYouSoon	Čoskoro sa ozveme.	
LabelSend	Odoslať	
LabelSubject	Predmet	
LabelThankYou	Ďakujeme, že ste nás kontaktovali.	
RequiredEmail	E-mailová adresa musí byť zadaná	
RequiredMsgText	Text správy musí byť zadaný	
RequiredName	Meno musí byť zadané	
RequiredSubject	Predmet musí byť zadaný	



Výhoda tohto riešenia je v tom, že ak potom daný model použiješ vo vytvorenom Umbraco webe, ktorý je vytvorený vo viacerých jazykoch, tak aj preklady cez resource stringy sa vždy použijú tie správne, ktoré sú priradené pre daný jazyk.


















Google Maps

Ďalšou úlohou je vytvorenie náhľadu na Google Maps pre konkrétnu adresu a miesto ako je to napríklad na webe <http://www.vennerymassage.sk/kontakt/>. Ukážka je na nasledovnom obrázku.



Vkladanie kódu pre zobrazenie mapy vyrieš taktiež pomocou Macra. Oproti kontaktnému formuláru je úloha trochu jednoduchšia, keďže nebudeš potrebovať vytvárať `SurfaceController`.

Výsledné View pre Macro bude umiestnené v adresári Views/MacroPartials, kde sa automaticky vytvára View pre Macro. Ukážka je na nasledovnom obrázku.

- ▷  Properties
- ▷  References
- ▷  App_Browsers
- ▷  App_Data
- ▷  App_Plugins
- ▷  config
- ▷  Media
- ▷  Scripts
- ▷  Styles
- ▷  Umbraco
- ▷  Umbraco_Client
- ▲  Views
 - ▲  MacroPartials
 -  BlogPostPromo.cshtml
 -  ContactForm.cshtml
 -  **GoogleMap.cshtml**
 -  YouTubeVideo.cshtml