

Creating Web Sites with Django and Bing Maps

Overview

In this hands-on lab we'll describe how to create a website with Django. Windows Azure Web Sites provide limited free hosting and rapid deployment – and now you can use Django! Besides, you can also create your own website with Bing Map!

Objectives

In this hands-on lab, you will learn how to:

- Create a new Web Site in Windows Azure by using Django.
- Add a new page to show your current location by Bing map.
- using FTP to manage those websites.

Prerequisites

The following is required to complete this hands-on lab:

- A Windows Azure subscription - [sign up for a free trial](#)
- Python 2.7 and Django 1.4 - You can either get these on your own or you can quickly and easily install these by using the Windows Installer link on <http://www.windowsazure.com/en-us/develop/python/>.

Exercises

This hands-on lab includes the following exercises:

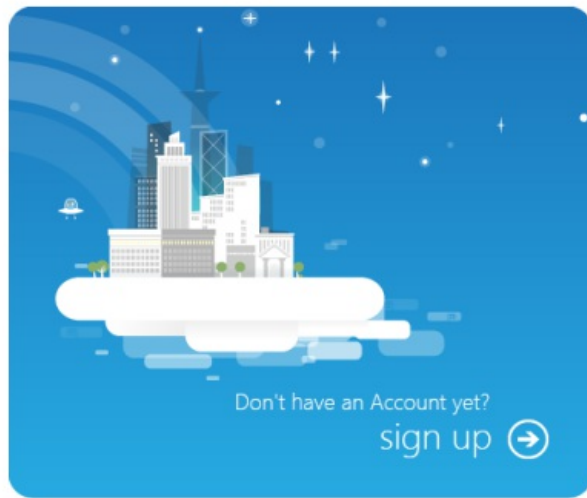
1. [Exercise 1: Creating a Windows Azure Web Site with Django](#)
2. [Exercise 2: Get your current location from Bing Map on your website.](#)

Estimated time to complete this lab: **45** minutes.

Excercise 1: Creating a Windows Azure Web Site with Django

During this exercise you will create a new web site of Django.

1. Go to the [Windows Azure Management Portal](#) and sign in using the Microsoft credentials associated with your subscription.



sign in

Microsoft account [What's this?](#)

☐ Keep me signed in

Can't access your account?
Sign in with a single-use code

Don't have a Microsoft account? [Sign up now](#)

Log on to Windows Azure Management Portal

- Click **New** on the command bar.

Windows Azure

PREVIEW

[ALL ITEMS](#)

WEB SITES

VIRTUAL MACHINES
8

CLOUD SERVICES
12

SQL DATABASES
12

STORAGE
5

NETWORKS
0

web sites PREVIEW

NAME	STATUS	SUBSCRIPTION	LOCATION	URL	
testwebapp	→ Running	3-Month Free Trial	North Europe		
testwebapp2	✓ Running	3-Month Free Trial	North Europe		

+

NEW

BROWSE

STOP

UPLOAD

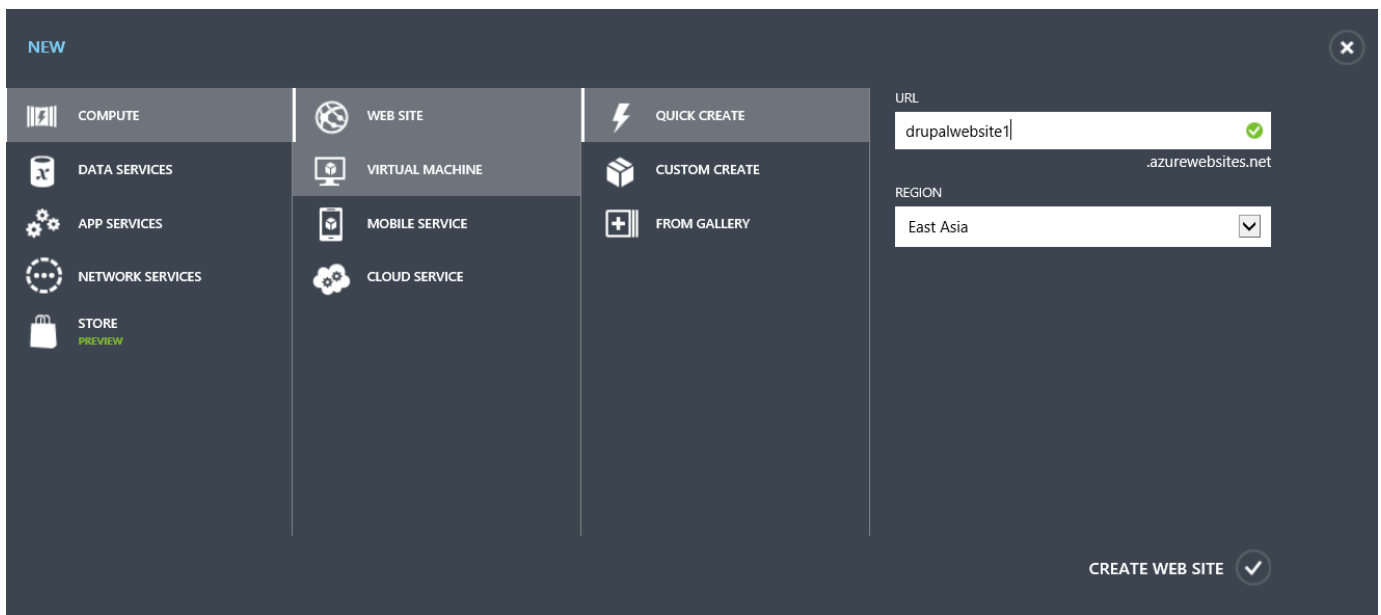
DELETE

WEBMATRIX

?

Creating a new Web Site

- Click **Web Site** and then **Quick Create**.



Quick Creating a new web site

The site will be quickly set up. Then we want to create a Django website.

4. Create a Django Website

From here we're ready to setup the enlistment with the web site. We'll need to do a few things:

1. Include the Django library and other libraries that we'll be using to run the web site.
2. Include the Django application code.

First, we'll include the Django library. To do this we'll create a new directory called site-packages and copy our installed version of Django there with these commands:

```
CommandPrompt mkdir site-packages cd site-packages xcopy /s C:\Python27\lib\site-packages\* .
```

This copies all the libraries located in site-packages, including Django. If there are libraries that are not used by your web site, feel free to remove them.

```
C:\Windows\System32\cmd.exe

C:\PythonWebSite>mkdir site-packages
C:\PythonWebSite>cd site-packages
C:\PythonWebSite\site-packages>xcopy /s c:\Python27\lib\site-packages\* .
C:\Python27\lib\site-packages\django\template\loader_tags.pyc
C:\Python27\lib\site-packages\django\template\response.py
C:\Python27\lib\site-packages\django\template\response.pyc
C:\Python27\lib\site-packages\django\template\smartif.py
C:\Python27\lib\site-packages\django\template\smartif.pyc
C:\Python27\lib\site-packages\django\template\__init__.py
C:\Python27\lib\site-packages\django\template\__init__.pyc
C:\Python27\lib\site-packages\django\template\loaders\app_directories.py
C:\Python27\lib\site-packages\django\template\loaders\app_directories.pyc
C:\Python27\lib\site-packages\django\template\loaders\cached.py
C:\Python27\lib\site-packages\django\template\loaders\cached.pyc
C:\Python27\lib\site-packages\django\template\loaders\eggs.py
C:\Python27\lib\site-packages\django\template\loaders\eggs.pyc
C:\Python27\lib\site-packages\django\template\loaders\filesystem.py
C:\Python27\lib\site-packages\django\template\loaders\filesystem.pyc
C:\Python27\lib\site-packages\django\template\loaders\__init__.py
C:\Python27\lib\site-packages\django\template\loaders\__init__.pyc
C:\Python27\lib\site-packages\django\templatetags\cache.py
C:\Python27\lib\site-packages\django\templatetags\cache.pyc
```

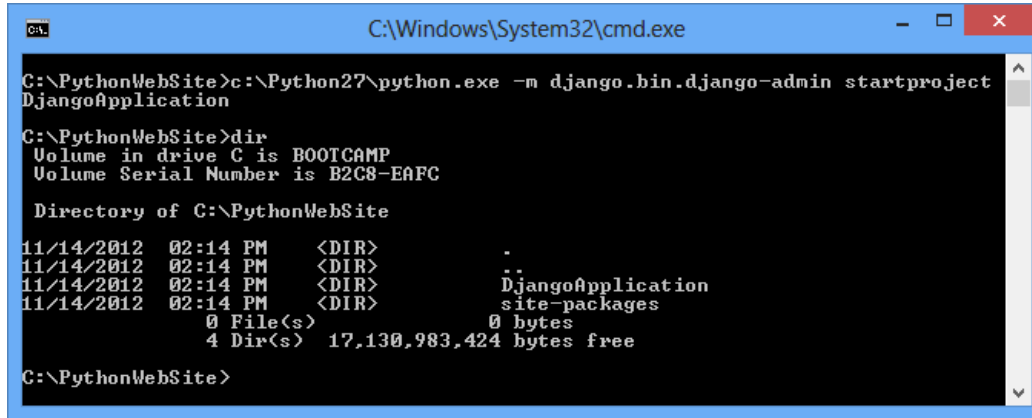
Copy Django Libraries

Next we'll create our initial Django application. You can do this just as you'd create any other Django application from the command line or you can use [Python Tools for Visual Studio](#) to create the project. We'll show you the first option here.

5. Create Django Application

To create the new project from the command line you just need to run this command which will create the Django application into the DjangoApplication folder:

```
CommandPrompt C:\Python27\python.exe -m django.bin.django-admin startproject DjangoApplication
```



```
C:\Windows\System32\cmd.exe

C:\PythonWebSite>c:\Python27\python.exe -m django.bin.django-admin startproject
DjangoApplication

C:\PythonWebSite>dir
Volume in drive C is BOOTCAMP
Volume Serial Number is B2C8-EAFC

Directory of C:\PythonWebSite

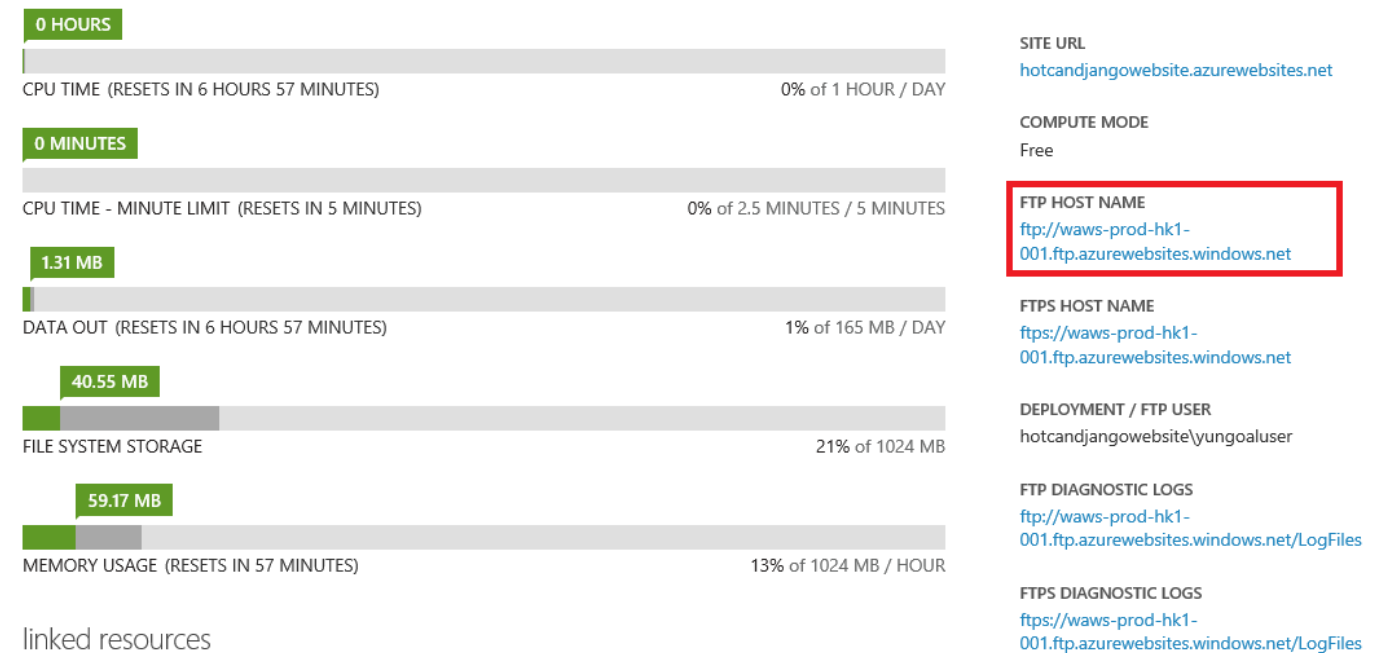
11/14/2012  02:14 PM    <DIR>          .
11/14/2012  02:14 PM    <DIR>          ..
11/14/2012  02:14 PM    <DIR>          DjangoApplication
11/14/2012  02:14 PM    <DIR>          site-packages
               0 File(s)              0 bytes
               4 Dir(s)  17,130,983,424 bytes free

C:\PythonWebSite>
```

Create Django Application

Then you will upload all files to the Website you created by Ftp.

- Go to the Windows Azure portal dashboard and copy the *FTP HOSTNAME*



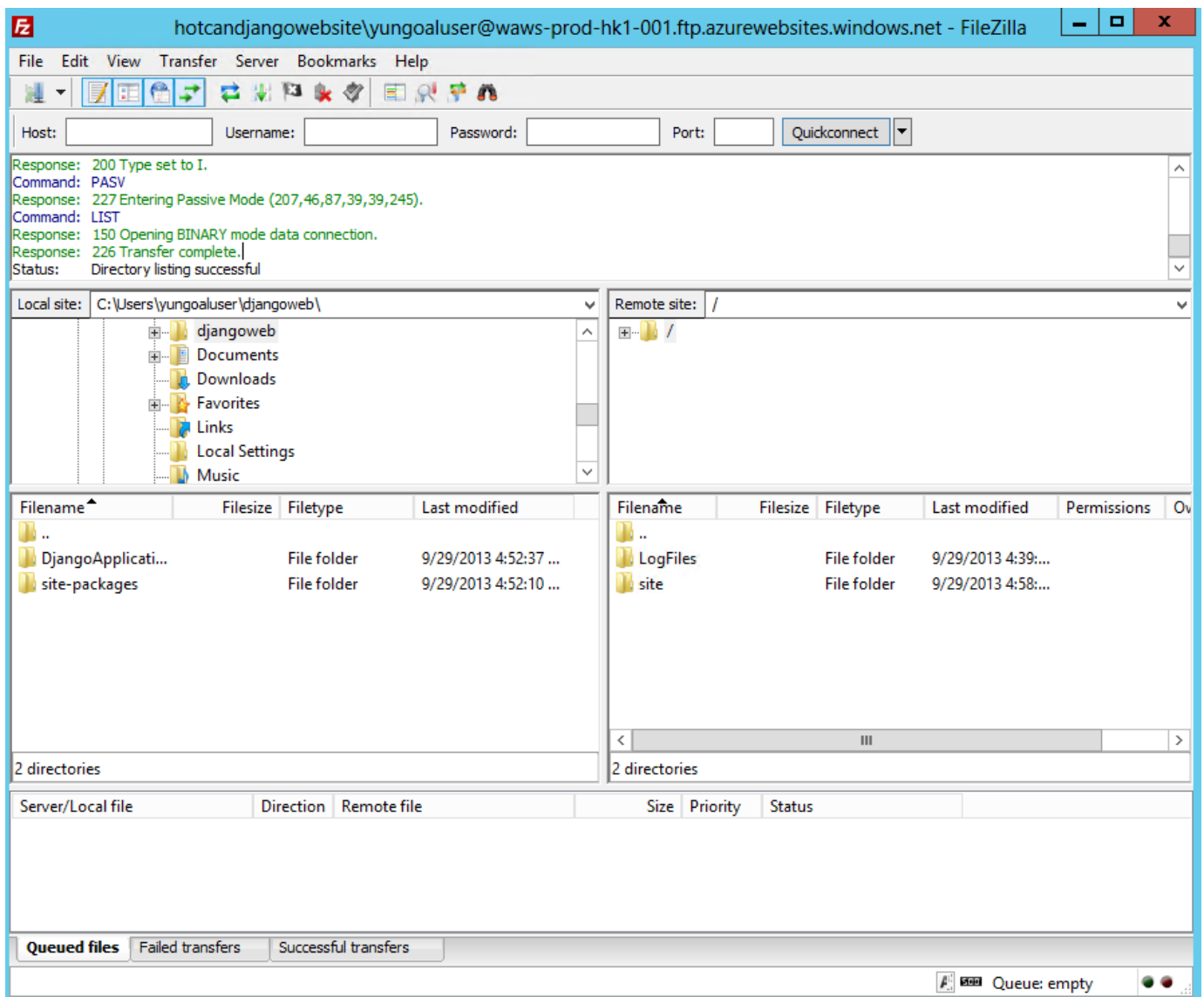
linked resources

Obtaining the FTP deployment hostname

- Connect to the FTP publishing service by FileZilla. You can download and install FileZilla to manage all your folders. FileZilla is a free ftp solution. The client version can be downloaded from [here](#).

Provide the **Host Name**, **User Name** and **Password** of your deployment credentials. Make sure that the **User Name** is prefixed by the **Web Site** name (e.g. `djangoSample\admin`)

Note: Replace the first `waws-prod-blu-001.ftp.azurewebsites.windows.net` value with the one obtained from the portal. Also remove the `ftp://` prefix as depicted above.



Use FileZilla

Note: Deployment credentials are other than the Live ID associated with your Windows Azure subscription and are valid for use with all Windows Azure web sites associated with your subscription. If you don't know your deployment credentials you can easily reset them using the management portal. Open the web site **Dashboard** page and click the **Reset deployment credentials** link. Provide a new password and click Ok.

New user name and password

Git and FTP cannot use your Windows account to authenticate, so this dialog lets you specify a user name and password that can be used when using those technologies.

This user name and password can be used to deploy to any web site in your subscription. You do not need to set credentials for every web site that you create.

USER NAME

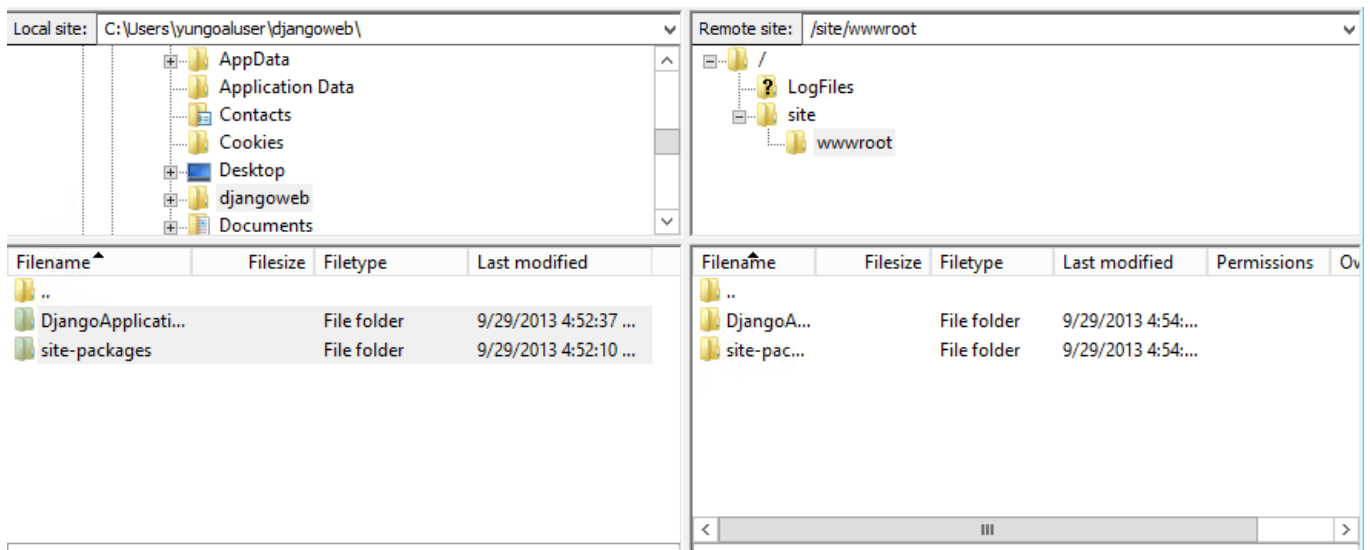
NEW PASSWORD

CONFIRM PASSWORD



Entering the username and password

- Click **Quick Connect** and Upload all files using the FileZilla. Navigate to your local Django site on the left and navigate to site\wwwroot on the remote. Then upload the **DjangoApplication** and **site-packages** folder to remote. And you also need to remove the pre-created file **hostingstart.html**



Uploading all files

- Web Site Configuration

We need to configure the web site to know about our Django project and to use the wfastcgi handler. To do this we can click on the Configure tab along the top of the screen where we'll want to scroll down to the bottom half of the page which contains app settings and handler mappings.

All of the settings that are set here will turn into environment variables during the actual request. This means that we can use this to configure the DJANGOSETTINGSMODULE environment variable as well as PYTHONPATH and WSGI_HANDLER. If your application has other configuration values you could assign these here and pick them up out of the environment. Sometimes you'll want to set something which is a path to a file in your web site, for example we'll want to do this for PYTHONPATH. When running as a Windows Azure web site your web site will live in "D:\home\site\wwwroot\" so you can use that in any location where you need a full path to a file on disk.

For setting up a Django application we need to set three environment variables. The first is `DJANGO_SETTINGS_MODULE` which provides the module name of the Django application which will be used to configure everything. The second is the `PYTHONPATH` environment variable so that we can find the package which the settings module lives in. The third is `WSGI_HANDLER`. It's a module/package name, followed by the attribute in the module to be used; for example `mypackage.mymodule.handler`. Add parentheses to indicate that the attribute should be called. So for these variables we will set them up as:

```
DJANGO_SETTINGS_MODULE = DjangoApplication.settings
```

```
PYTHONPATH = D:\home\site\wwwroot\DjangoApplication;D:\home\site\wwwroot\site-packages
```

```
WSGI_HANDLER = django.core.handlers.wsgi.WSGIHandler()
```

app settings



DJANGO_SETTINGS_MODULE	DjangoApplication.settings
PYTHONPATH	D:\home\site\wwwroot\DjangoApplication;D:\home\site\wwwroot\si
WSGI_HANDLER	django.core.handlers.wsgi.WSGIHandler()
KEY	VALUE

Django App Settings

Then we need to configure our handler mapping. For this we register the handler for all extensions, using the path to the Python interpreter and the path to the `wfastcgi.py` script:

```
EXTENSION = *
```

```
SCRIPT_PROCESSOR_PATH = D:\python27\python.exe
```

```
ADDITIONAL_ARGUMENTS = D:\python27\scripts\wfastcgi.py
```

handler mappings



*	D:\python27\python.exe	D:\home\site\wwwroot\wfastcgi.py
EXTENSION	SCRIPT PROCESSOR PATH	ADDITIONAL ARGUMENTS (OPTIONAL)

Django Handler Mapping

- Finally we can go back to the Dashboard, and go down to the SITE URL on the left hand side and click on the link and we'll open our new Django site:

File Edit View Favorites Tools Help

It worked!

Congratulations on your first Django-powered page.

Of course, you haven't actually done any work yet. Here's what to do next:

- If you plan to use a database, edit the `DATABASES` setting in `DjangoApplication/settings.py`.
- Start your first app by running `python manage.py startapp [appname]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

Django Website

Exercise 2 - Get your current location from Bing Map

Next let's create a very simple html page and show your current location in your website. You can use Bing Map.

You will see you current location on your website.



You current location

1. Create a new website according to the steps in [Exercise 1](#).

NEW

COMPUTE

DATA SERVICES

APP SERVICES

NETWORK SERVICES

STORE
PREVIEW

WEB SITE

VIRTUAL MACHINE

MOBILE SERVICE

CLOUD SERVICE

QUICK CREATE

CUSTOM CREATE

FROM GALLERY

URL

hotcanwebsitelocation

.azurewebsites.net


REGION

East Asia

CREATE WEB SITE

Create a new website

2. Visit the Bing Maps Portal.



Maps Account Center

[Bing Maps Account Center Help](#)

Resources

- [Bing Maps Platform](#)
- [Bing Maps APIs](#)
- [Bing Maps Forums](#)
- [Bing Maps Blog](#)
- [Bing Maps AJAX Control 7.0 ISDK](#)

Contact Us

Account issues
mpnet@microsoft.com

Licensing inquiries
maplic@microsoft.com

[Bing Maps APIs Terms of Use](#)
[Bing Maps Platform APIs' Terms of Use \(TOU\)](#)
[Bing Maps API Terms of Use for Pre-release Windows 8 Metro Style Apps \(Beta\)](#)

Welcome to the Bing Maps Account Center

Your Bing Maps Account lets you create Bing Maps Keys to use with the Bing Maps APIs. You can also create and manage data sources that contain entities with a location. [Learn more](#)

[Sign In](#)

[New User](#)
You will need a Microsoft account (Formerly known as Windows Live ID)

Bing Map Portal

you can use your Microsoft account to login. If you don't have any Microsoft account, you can click **New User** to register one.

1. After login into the portal, just click **Create or view keys** to create your own key. Input your application name, url, key type and application type. You can just select **Basic** key and choose your correct application type. Please ensure the **Application URL** is the same as the website you have created in the last steps.

[Bing Maps Account Center Help](#)
[Announcements](#)

My Account

- [Update or view account details](#)
- [Create or view keys](#)
- [View my Bing Maps API usage](#)

Data Sources

- [Upload data to a data source](#)
- [Manage my data sources](#)
- [View data source information](#)

Resources

- [Bing Maps Platform](#)
- [Bing Maps APIs](#)
- [Bing Maps Forums](#)
- [Bing Maps Blog](#)
- [Bing Maps AJAX Control 7.0 ISDK](#)

Contact Us

Account issues
mpnet@microsoft.com

Licensing inquiries
maplic@microsoft.com

[Bing Maps APIs Terms of Use](#)
[Bing Maps Platform APIs' Terms of Use \(TOU\)](#)
[Bing Maps API Terms of Use for Pre-release Windows 8 Metro Style Apps \(Beta\)](#)

My keys

You are able to create two keys (Trial or Basic) for most application types and one additional Windows Store app Trial key under this account. Trial keys expire after 90 days and cannot be converted to Basic keys. You cannot delete keys or generate more than 3 keys from this account. If you intend to create a non-trial application, make one of the keys a Basic key and read the governing [TOU](#) that defines usage limits.

Create key

* **Application name**

Application URL

* **Key type**

Basic


[What's This](#)

* **Application type**

Education

* **Enter the characters you see**

[Try a new image](#)



NK8&6

[Submit](#)

Click [here](#) to view/download complete list of keys.

Red keys are expired trial keys.

Orange keys expire in less than 30 days.

2. You will get your key after you submit your information. Please **Save** the key and you will use it in your html page.

mylocation

http://myweb2.azurewebsites.net

Update

Basic / Education

Created Date: 09/29/2013 Expiration Date: None

Bing Map Key

3. Open the file in **Sources\location.html** in text editor, replace the **[ApplicationKey]** with your own application key.

```
map = new Microsoft.Maps.Map(document.getElementById('myMap'), { credentials: '[ApplicationKey]' });
```

Change Application Key

4. Use FileZilla to upload the file **location.html** to **site/wwwroot** folder.

Filename	Filesize	Filetype	Last modified	Filename	Filesize	Filetype	Last modified
..				..			
DjangoApplicati...		File folder	29/09/2013 23:25:38	hostingstart.html	65,954	HTML Doc...	30/09/2013 02:...
site-packages		File folder	29/09/2013 23:24:51	location.html	1,227	HTML Doc...	30/09/2013 02:...
location.html	1,227	HTML Docum...	30/09/2013 02:41:38				

Upload the location file

5. Browse the new website [http://\[yourwebsite\].azurewebsites.net/location.html](http://[yourwebsite].azurewebsites.net/location.html).
6. You will get your current location by clicking "GetCurrentLocation" button.



You current location

Summary

By completing this hands-on lab you learned the following:

- Creating a Windows Azure Web Site with Django.
- Get your current location from Bing Map on your website.