Best practice to initialize web tier and connect to MapGuide Enterprise.

Published date: 2011-01-14
ID: DD2011011401

Applies to:
Autodesk MapGuide® Enterprise 2011
Autodesk MapGuide® Enterprise 2010
Autodesk MapGuide Enterprise 2009

Issue
What is the generate process of creating a MapGuide based web application? How can I get started?

I notice that the MapGuide initialization should occur in every page needing it; it should probably be done as a class and not as a function in each of the pages. I have tried to implement this but so far without success.
Solution
In MapGuide development, web tier initialization and creating connection to MapGuide server are the most common used tasks in MapGuide web application developing. It is a good idea to impalement this task into a utility class to make it reusable. In this devnote, we will give you a general introduction how to creating a custom WebGIS application based on MapGuide, and how to implement and use this utility class.

Please see the code snippet below, it is well documented, should be easy to understand.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using OSGeo.MapGuide;

public class UtilityClass
{
public UtilityClass()
{
}

    // The mapguide connection site object
    MgSiteConnection siteConnection;

    public MgSiteConnection GetSiteConnection()
    {
        return siteConnection;
    }

    // Initialize the web tier, "webconfig.ini" file is supposed to be copied to current folder of web application
    // from C:\Program Files\Autodesk\MapGuideEnterprise2011\WebServerExtensions\www\webconfig.ini
    public void InitializeWebTier(HttpRequest Request)
    {
        string realPath = Request.ServerVariables["APPL_PHYSICAL_PATH"];
        String configPath = realPath + "webconfig.ini";
        MapGuideApi.MgInitializeWebTier(configPath);
    }
```

```
// connect to server with specifed session string
    public void ConnectToServer(String sessionID)
    {
        MgUserInformation userInfo = new MgUserInformation(sessionID);
        siteConnection = new MgSiteConnection();
        siteConnection.Open(userInfo);
    }

    //Connect to MapGuide server with specifed username/pwd
    public void ConnectToServer()
    {
        MgUserInformation userInfo = new MgUserInformation("Administrator", "admin");
        siteConnection = new MgSiteConnection();
        siteConnection.Open(userInfo);
    }

}
```

Now let's look at how to use this utility class. To create a MapGuide based web application; there are 5 steps to embed a MapGuide viewer into a webpage:

- Copy all dlls from "C:\Program Files\Autodesk\MapGuideEnterprise2011\WebServerExtensions\www\mapviewernet\bin" to "YourWebApp\Bin";
- Copy "webconfig.ini" from "C:\Program Files\Autodesk\MapGuideEnterprise2011\WebServerExtensions\www\webconfig.ini" to current folder of your web application;
- Initialize web tier and connect to MapGuide server with specified username and password;
- Get MgSite object and generate session string;
- Pass the session string and resource identifier of web layout to MapGuide viewer.

Please see the code snippet below:

```
<%@ Page Language="C#" AutoEventWireup="true"  CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<frameset border="0" framespacing="0" rows="0,*">
  <frame />
  <frame id="viewerFrame" src="http://localhost/mapguide2011/mapviewernet/ajaxviewer.aspx?SESSION=<%= sessionId %>&WEBLAYOUT=<%= webLayout %>"  />
</frameset>
</html>
```

Code behind:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using OSGeo.MapGuide;

public partial class _Default : System.Web.UI.Page
{
    //Pay attention to these variables should be marked as "public",
    //so that it can be accessed by <%= sessionId %> and <%= webLayout %>
    public string sessionId, webLayout;

    protected void Page_Load(object sender, EventArgs e)
    {
        UtilityClass utility = new UtilityClass();
        //Initlize web tier, passing the request context to search for the path of webconfig.ini file
        utility.InitializeWebTier(Request);
        //This is the first time to connect to MapGuide, we can connect with Administrator/admin, which is specifed in
utiliy class implementaion
        utility.ConnectToServer();

        MgSiteConnection connection = utility.GetSiteConnection();
        //Get the MgSite object and create session string
        sessionId = connection.GetSite().CreateSession();
        //Save the session string to ASP.NET session for feature use
        Session["MySession"] = sessionId;

        webLayout = "Library://Sampele/Sheboygon/Layouts/weblayout.WebLayout";

    }
}
```

In other custom web pages, you should initialize web tier and connect to MapGuide Server with the session string, which is created in first page. The following code snippets demonstrate how to list all MapGuide layers in a web page. In MapGuide Studio, you need to create an “Invoke URL” command referring to this page, and add the command into web layout (taskbar menu, toolbar or context menu). Code snippet goes as below:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using OSGeo.MapGuide;
using System.Collections;
using System.Text;
```

```
public partial class MapInforrmation : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      UtilityClass util = new UtilityClass();
// Initialize web tier using the webconfig.ini file, which is located at current folder of web application
      util.InitializeWebTier(HttpContext.Current.Request);

      //Get MapGuide session string from ASPNET SESSION
       string sessionId = Session["MySession"].ToString();

      //Connect to MapGuide Server with this MapGuide session string
      util.ConnectToServer(sessionId);
      MgSiteConnection siteConn = util.GetSiteConnection();

      MgResourceService resSvc = siteConn.CreateService(MgServiceType.ResourceService) as
MgResourceService;

      //Output all the layers of map
      MgMap map = new MgMap();
      if (Request["mapname"] != null)
      {
         string mapName = Request["mapname"].ToString();
         map.Open(resSvc, mapName);

         Response.Write("Current Map's SRS is : " + map.MapSRS + "<br><br>");
         Response.Write("maps layer count is :" + map.GetLayers().Count.ToString() + "<br><hr>");

         for (int i = 0; i < map.GetLayers().Count; i++)
         {
            Response.Write(map.GetLayers()[i].Name + "<br>");
         }
      }

      Response.Write("<hr>");

   }
}
```

Note: you need do some minor changes if you are not using MapGuide Enterprise 2011