```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using JerryWebMVC.Models;
using Paging;
using System.Web.UI.WebControls;

namespace JerryWebMVC.Controllers
{
    public class WebServiceController : Controller
    {
        //
        // GET: /WebService/

        public ActionResult Index()
        {
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "";

            return View();
        }

        public ActionResult Dilbert()
        {
            ViewData["Title"] = "daily dilbert";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "dilbert";

            return View();
        }

        [HttpGet]
        public ActionResult Amazon()
        {
            ViewData["Title"] = "amazon book search";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "amazon";

            AmazonModel model = new AmazonModel();
            return View(model);
        }

        [HttpPost]
        public ActionResult Amazon(AmazonModel pmodel)
        {
            ViewData["Title"] = "amazon book search";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "amazon";
            AmazonModel newmodel = new AmazonModel();
            newmodel.BookSearch(pmodel.Title, pmodel.Author, pmodel.KeyWords, "");
            TempData["title"] = pmodel.Title;
            TempData["author"] = pmodel.Author;
            TempData["keywords"] = pmodel.KeyWords;
            var books = newmodel.Books.AsQueryable();

            ViewData["books"] = books.ToPagedList(0, 5);

            return View(newmodel);
        }


        public ActionResult PageAmazon(int page)
        {
            string title;
            string author;
```

```
            string keywords;

            title = TempData["title"]==null ? string.Empty : TempData["title"].ToString();
            author = TempData["author"] == null ? string.Empty : TempData["author"].
    ToString();
            keywords = TempData["keywords"] == null ? string.Empty : TempData["keywords"].
    ToString();

            ViewData["Title"] = "amazon book search";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "amazon";
            AmazonModel newmodel = new AmazonModel();
            newmodel.BookSearch(title, author, keywords, "");
            newmodel.Author = author;
            newmodel.Title = title;
            newmodel.KeyWords = keywords;
            TempData["title"] = title;
            TempData["author"] = author;
            TempData["keywords"] = keywords;
            var books = newmodel.Books.AsQueryable();



            ViewData["books"] = books.ToPagedList(page, 5);
            return View("Amazon", newmodel);

        }

        public ActionResult Weather()
        {
            ViewData["Title"] = "weather by zip code";
            WeatherModel model = new WeatherModel();
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "weather";
            return View(model);
        }

        [HttpPost]
        public ActionResult Weather(WeatherModel model)
        {
            ViewData["Title"] = "weather by zip code";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "weather";

            if (!String.IsNullOrEmpty(model.ZipCode))
            {
                model.GetWeather(model.ZipCode);
                if (String.IsNullOrEmpty(model.LocalWeather.City))
                {
                    model.ErrMsg = "Zip Code Note Found";
                }
            }
            return View(model);
        }


        [ChildActionOnly]
        public ActionResult LocalWeather(string zipcode)
        {
            WeatherModel model = new WeatherModel();
            model.GetWeather(zipcode);
            model.GetForecast(zipcode);

            if (string.IsNullOrEmpty(model.LocalWeather.City))
            {
                return null;
            }
```

```
            else
            {
                return View(model);
            }
        } // LocalWeather

        [HttpGet]
        public ActionResult PeriodicTable()
        {
            ViewData["Title"] = "periodic table";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "period";
            ElementModel model = new ElementModel();
            ListItem list = new ListItem();

            ViewData["Elements"] = new SelectList(model.Elements);

            return View(model);
        }

        [HttpPost]
        public ActionResult PeriodicTable(ElementModel model)
        {
            ViewData["Title"] = "periodic table";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "period";

            return View(model);

        }

        [ChildActionOnly]
        public ActionResult ElementData(string psymbol)
        {
            if (String.IsNullOrEmpty(psymbol))
            {
                return null;
            }
            else
            {
                ElementModel model = new ElementModel();
                model.GetElement(psymbol);
                return View(model);
            }

        }

        public ActionResult Smoothie()
        {
            ViewData["Title"] = "smoothie store";
            ViewData["PageRef"] = "wcf";
            ViewData["SubPageRef"] = "smoothie";
            SmoothieModel model = new SmoothieModel();

            return View(model);

        }


    }
}
```