

ASP.NET MVC use with Autofac

1. Add Autofac and Autofac.Mvc5 References from NuGet

2. Create ContainerBuilder and register components in application startup

A. Add one line code in Application_Start() function of global.asax file.

```
DependencyConfig.RegisterDependencyResolvers();
```

B. Add the DependencyConfig class in the folder App_Start:

```
public static class DependencyConfig
{
    public static IContainer RegisterDependencyResolvers()
    {
        ContainerBuilder builder = new ContainerBuilder();
        RegisterDependencyMappingDefaults(builder);
        RegisterDependencyMappingOverrides(builder);
        IContainer container = builder.Build();
        // Set Up MVC Dependency Resolver
        DependencyResolver.SetResolver(new AutofacDependencyResolver(container));
        // Set Up WebAPI Resolver
        GlobalConfiguration.Configuration.DependencyResolver = new
AutofacWebApiDependencyResolver(container);
        return container;
    }

    private static void RegisterDependencyMappingDefaults(ContainerBuilder builder)
    {
        Assembly coreAssembly = Assembly.GetAssembly(typeof(IStateManager));
        Assembly webAssembly = Assembly.GetAssembly(typeof(MvcApplication));

        builder.RegisterAssemblyTypes(coreAssembly).AsImplementedInterfaces().InstancePerRequest();
        builder.RegisterAssemblyTypes(webAssembly).AsImplementedInterfaces().InstancePerRequest();

        builder.RegisterControllers(webAssembly);
        builder.RegisterModule(new AutofacWebTypesModule());
    }

    private static void RegisterDependencyMappingOverrides(ContainerBuilder builder)
    {
        // builder.RegisterType<WebSettingManager>().AsImplementedInterfaces().SingleInstance();
        builder.RegisterType<SearchDataService>().As<ISearchDataService>();
    }
}
```

3. Add Interface

```
public interface ISearchDataService
{
    string CombineConfirmedCasesDailyCountURL(string postcode);
}
```

4. Add Service code

```

public class SearchDataService : ISearchDataService
{
    public string CombineConfirmedCasesDailyCountURL(string postcode)
    {
        string part1 = ConfigurationManager.AppSettings["ConfirmedCasesByPostcode1"];
        string part2 = ConfigurationManager.AppSettings["ConfirmedCasesByPostcode2"];
        return part1 + " " + postcode + " " + part2;
    }
}

```

5. Call service in controller

```

public class HomeController : Controller
{
    private readonly ISearchDataService _searchDataService;

    public HomeController(ISearchDataService iSearchDataService)
    {
        // Injected from Autofac
        _searchDataService = iSearchDataService;
    }

    [HttpPost]
    public ActionResult Index(RetrieveDataViewModel model)
    {
        if (!ModelState.IsValid)
            return View(model);

        string webAddress = _searchDataService.CombineConfirmedCasesDailyCountURL(model.Postcode);
        model.DailyCountResult = _searchDataService.GetCasesDailyCountList(webAddress);

        return View(model);
    }
}

```

Reference:

<https://dependencyinjector.wordpress.com/2014/12/19/setting-up-dependency-injection-in-asp-net-mvc-with-autofac/>
<https://www.codeproject.com/Articles/808894/loC-in-ASP-NET-MVC-using-Autofac>