

00a69f943b ▼

...

xunit.frameworks.autofac / src / Xunit.Frameworks.Autofac / **AutoFacRegistrationExtensions.cs** /

<> Jump to ▼



dabide Renamed to xunit.frameworks.autofac

History

1 contributor

44 lines (38 sloc) | 2.02 KB

...

```
1 using System;
2 using Autofac.Builder;
3
4 // ReSharper disable once CheckNamespace
5 namespace Xunit.Frameworks.Autofac {
6     public static class AutoFacXUnitRegistrationExtensions
7     {
8         /// <summary>
9         ///     Registers a component so all dependant components will resolve the same shared ins
10        ///     lifetime scope.
11        /// </summary>
12        public static IRegistrationBuilder<TLimit, TActivatorData, TStyle> InstancePerTest<TLimit,
13            this IRegistrationBuilder<TLimit, TActivatorData, TStyle> registration)
14        {
15            if (registration == null) throw new ArgumentNullException(nameof(registration));
16
17            return registration.InstancePerMatchingLifetimeScope(AutofacTestScopes.Test);
18        }
19
20        /// <summary>
21        ///     Registers a component so all dependant components will resolve the same shared ins
22        ///     lifetime scope.
23        /// </summary>
24        public static IRegistrationBuilder<TLimit, TActivatorData, TStyle> InstancePerTestClass<TL
25            this IRegistrationBuilder<TLimit, TActivatorData, TStyle> registration)
26        {
27            if (registration == null) throw new ArgumentNullException(nameof(registration));
```

```
        return registration.InstancePerMatchingLifetimeScope(AutofacTestScopes.TestClass);
    }

    /// <summary>
    ///     Registers a component so all dependant components will resolve the same shared ins
    ///     lifetime scope.
    /// </summary>
    public static IRegistrationBuilder<TLimit, TActivatorData, TStyle> InstancePerTestCollecti
        this IRegistrationBuilder<TLimit, TActivatorData, TStyle> registration)
    {
        if (registration == null) throw new ArgumentNullException(nameof(registration));

        return registration.InstancePerMatchingLifetimeScope(AutofacTestScopes.TestCollection)
    }
}
}
```