IdeploymentService – restartovani aplikace kdyz je vydana nova verze

[‎4/‎11/‎2018 10:17 AM] Filip Čálek:

ted jsem si jeste vzpomnel na jednu vec, ktera s tim nesouvisi

ale asi ji tam chci

v mr frameworku mame IDeploymentService. coz je udelatko, ktery v pravidelnym intervalu kontroluje, zda nebyla vydana nova verze aplikace

[‎4/‎11/‎2018 10:18 AM] Peter Hlavenka:

ano ??

[‎4/‎11/‎2018 10:19 AM] Filip Čálek:

pokud ano, uzivatel je informovan a po nejakym case dojde k restartu aplikace. delame to proto, ze uzivatele jsou trubky a jsou schopny mit aplikace pusteny klidne mesice bez restartu

takhle je donutime, aby si vzdy stahli verzi novou

[‎4/‎11/‎2018 10:21 AM] Peter Hlavenka:

ok podivam se na to

[‎4/‎11/‎2018 10:21 AM] Filip Čálek:

staci jen zaregistrovat do containeru a v boostraperu zavolat init metodu. o zbytek se postara trida sama

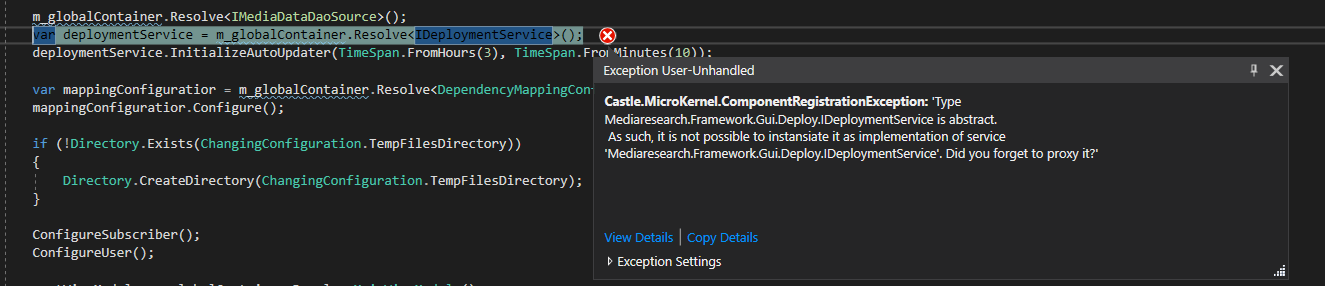
[‎4/‎11/‎2018 10:22 AM] Peter Hlavenka:

tim lip :)

container.Register(Component.For<IDeploymentService>().ImplementedBy<IDeploymentService>().LifestyleSingleton());

var deploymentService = m\_globalContainer.Resolve<IDeploymentService>();

deploymentService.InitializeAutoUpdater(TimeSpan.FromMinutes(15), TimeSpan.FromMinutes(5));



**Reseni :** rozhrani implemented by trida

container.Register(Component.For<IDeploymentService>().ImplementedBy<DeploymentService>().LifestyleSingleton());

[‎4/‎11/‎2018 10:29 AM] Peter Hlavenka:

Jaky checkInterval mam nastavit ?

[‎4/‎11/‎2018 10:31 AM] Filip Čálek:

v SA mam napr. 15 minut check a 5 minut shutdown

a dej toto prosim do configu, at se to da lehce menit

<castle>

<properties>

<mediaDataDbAlias>mediaDataDB</mediaDataDbAlias>

<mediaDataConnectionString>Data Source=stoupa; Initial Catalog=mediadata3DT; Integrated Security=SSPI</mediaDataConnectionString>

<serverTimeZone>Central Europe Standard Time</serverTimeZone>

<iconPath>/MIR.Media.Changing2.Gui;component/ChangingIcons/Changing.png</iconPath>

<info>Default</info>

<deploymentServiceCheckTimeSpan>15</deploymentServiceCheckTimeSpan>

<deplymentServiceShutdownTimeSpan>5</deplymentServiceShutdownTimeSpan>

</properties>

<components>

<component id="MediaDataDbConfiguration"

type="MIR.Media.Changing2.Shell.Installers.MediaDataDbConfiguration, MIR.Media.Changing2.Shell">

<parameters>

<mediaDataDbAlias>#{mediaDataDbAlias}</mediaDataDbAlias>

<mediaDataConnectionString>#{mediaDataConnectionString}</mediaDataConnectionString>

<serverTimeZone>#{serverTimeZone}</serverTimeZone>

</parameters>

</component>

<component id="ChangingConfiguration"

type="MIR.Media.Changing2.Shell.Installers.ChangingConfiguration, MIR.Media.Changing2.Shell"

service="MIR.Media.Changing2.Shell.Installers.IChangingConfiguration, MIR.Media.Changing2.Shell">

<parameters>

<iconPath>#{iconPath}</iconPath>

<info>#{info}</info>

<deploymentServiceCheckTimeSpan>#{deploymentServiceCheckTimeSpan}</deploymentServiceCheckTimeSpan>

<deplymentServiceShutdownTimeSpan>#{deplymentServiceShutdownTimeSpan}</deplymentServiceShutdownTimeSpan>

</parameters>

</component>

</components>

</castle>

namespace MIR.Media.Changing2.Shell.Installers

{

public interface IChangingConfiguration

{

string Info { get; }

string IconPath { get; }

TimeSpan DeploymentServiceCheckTimeSpan { get; }

TimeSpan DeplymentServiceShutdownTimeSpan { get; }

}

}

**Poznamka: Pokud dame** DeploymentServiceCheckTimeSpan **setter set; nebudeme muset tuto propertu inicializovat z konstruktoru . (Pod poznamkou) Konstruktor vubec nemusi v rozhrani nebo tride (napr. trida Properties) existovat**

public class ChangingConfiguration : IChangingConfiguration

{

public ChangingConfiguration(string info, string iconPath, TimeSpan deploymentServiceCheckTimeSpan, TimeSpan deplymentServiceShutdownTimeSpan)

{

Info = info;

IconPath = iconPath;

DeploymentServiceCheckTimeSpan = deploymentServiceCheckTimeSpan;

DeplymentServiceShutdownTimeSpan = deplymentServiceShutdownTimeSpan;

}

public static string TempFilesDirectory { get; } = Path.Combine(Path.GetTempPath(), "Changing");

public string Info { get; }

public string IconPath { get; }

}

**Bootstrapper:**

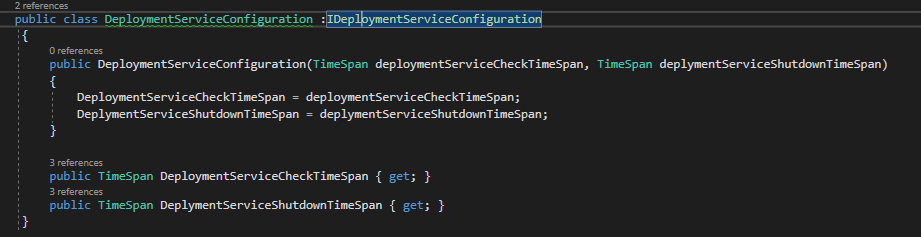
var deploymentService = m\_globalContainer.Resolve<IDeploymentService>();

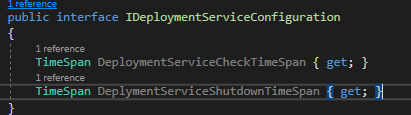
var checkTimeSpan = changingConfiguration.DeploymentServiceCheckTimeSpan;

var shutDownTimeSpan = changingConfiguration.DeplymentServiceShutdownTimeSpan;

deploymentService.InitializeAutoUpdater(checkTimeSpan, shutDownTimeSpan);

**V cenikovadle jsem musel vytvorit vlastni rozhrani :**

****



<component id="IDeploymentServiceConfiguration" type="MIR.Pricing.DeploymentService.DeploymentServiceConfiguration, MIR.Pricing" service="MIR.Pricing.DeploymentService.IDeploymentServiceConfiguration, MIR.Pricing">

<parameters>

<deploymentServiceCheckTimeSpan>#{deploymentServiceCheckTimeSpan}</deploymentServiceCheckTimeSpan>

<deplymentServiceShutdownTimeSpan>#{deplymentServiceShutdownTimeSpan}</deplymentServiceShutdownTimeSpan>

</parameters>

</component>

Do properties nahore:

<deploymentServiceCheckTimeSpan>15</deploymentServiceCheckTimeSpan>

<deplymentServiceShutdownTimeSpan>5</deplymentServiceShutdownTimeSpan>

Bootstrapper:

var deploymentService = Container.Container.Current.Resolve<IDeploymentService>();

var deploymentConfiguration = Container.Container.Current.Resolve<IDeploymentServiceConfiguration>();

var checkTimeSpan = deploymentConfiguration.DeploymentServiceCheckTimeSpan;

var shutDownTimeSpan = deploymentConfiguration.DeplymentServiceShutdownTimeSpan;

deploymentService.InitializeAutoUpdater(checkTimeSpan, shutDownTimeSpan);

**Snazil jsem se registraci cpat do spatneho installeru a psalo mi Could not resolve because of dependency na moji properte. Spravny installer je CommonInstaller:**

****

Hotovo