1. 新建console项目
2. Nuget添加topshelf，提示net版本不对，根据官方git文档，调整net版本为Topshelf uses the .NET Framework v4.5.2.，安装成功
3. 余下按<https://www.cnblogs.com/yanglang/p/7199913.html>操作

# 配置运行

没错，整个程序已经开发完了，接下来，只需要简单配置一下，即可以当服务来使用了。安装很方便：

安装：TopshelfDemo.exe install

启动：TopshelfDemo.exe start

卸载：TopshelfDemo.exe uninstall

安装成功后，接下来，我们就可以看到服务里多了一个服务：

# 扩展说明

## Topshelf Configuration 简单配置

官方文档，对HostFactory 里面的参数做了详细的说明：<http://docs.topshelf-project.com/en/latest/configuration/config_api.html> ，下面只对一些常用的方法进行简单的解释：

我们将上面的程序代码改一下：

HostFactory.Run(x => //1

{

x.Service<TownCrier>(s => //2

{

s.ConstructUsing(name => new TownCrier()); //配置一个完全定制的服务,对Topshelf没有依赖关系。常用的方式。  
　　　　　　　　　　　　//the start and stop methods for the service  
                    s.WhenStarted(tc => tc.Start()); //4

s.WhenStopped(tc => tc.Stop()); //5

});

x.RunAsLocalSystem(); // 服务使用NETWORK\_SERVICE内置帐户运行。身份标识，有好几种方式，如：x.RunAs("username", "password"); x.RunAsPrompt(); x.RunAsNetworkService(); 等

x.SetDescription("Sample Topshelf Host服务的描述"); //安装服务后，服务的描述

x.SetDisplayName("Stuff显示名称"); //显示名称

x.SetServiceName("Stuff服务名称"); //服务名称

});

重装安装运行后：

 通过上面，相信大家都很清楚 SetDescription、SetDisplayName、SetServiceName区别。不再细说。

## Service Configuration 服务配置

Topself的服务一般有主要有两种使用模式。

### 一、简单模式。继承ServiceControl接口，实现该接口即可。

实例：

namespace TopshelfDemo

{

public class TownCrier : ServiceControl

{

private Timer \_timer = null;

readonly ILog \_log = LogManager.GetLogger(typeof(TownCrier));

public TownCrier()

{

\_timer = new Timer(1000) { AutoReset = true };

\_timer.Elapsed += (sender, eventArgs) => \_log.Info(DateTime.Now);

}

public bool Start(HostControl hostControl)

{

\_log.Info("TopshelfDemo is Started");

\_timer.Start();

return true;

}

public bool Stop(HostControl hostControl)

{

throw new NotImplementedException();

}

}

class Program

{

public static void Main(string[] args)

{

var logCfg = new FileInfo(AppDomain.CurrentDomain.BaseDirectory + "log4net.config");

XmlConfigurator.ConfigureAndWatch(logCfg);

HostFactory.Run(x =>

{

x.Service<TownCrier>();

x.RunAsLocalSystem();

x.SetDescription("Sample Topshelf Host服务的描述");

x.SetDisplayName("Stuff显示名称");

x.SetServiceName("Stuff服务名称");

});

}

}

}

### 二、常用模式。

实例：

namespace TopshelfDemo

{

public class TownCrier

{

private Timer \_timer = null;

readonly ILog \_log = LogManager.GetLogger(

typeof(TownCrier));

public TownCrier()

{

\_timer = new Timer(1000) { AutoReset = true };

\_timer.Elapsed += (sender, eventArgs) => \_log.Info(DateTime.Now);

}

public void Start(){ \_timer.Start();}

public void Stop() { \_timer.Stop(); }

}

class Program

{

public static void Main(string[] args)

{

var logCfg = new FileInfo(AppDomain.CurrentDomain.BaseDirectory + "log4net.config");

XmlConfigurator.ConfigureAndWatch(logCfg);

HostFactory.Run(x =>

{

x.Service<TownCrier>(s =>

{

s.ConstructUsing(name => new TownCrier());

s.WhenStarted(tc => tc.Start());

s.WhenStopped(tc => tc.Stop());

});

x.RunAsLocalSystem();

x.SetDescription("Sample Topshelf Host服务的描述");

x.SetDisplayName("Stuff显示名称");

x.SetServiceName("Stuff服务名称");

});

}

}

}