# Toolkit5.5从.net core2.2升级到.net core3.1需要注意的问题

## 为什么要升级

微软在其文档中说明了.net core各版本的支持周期，如下：

.NET Core 3.1 是未来三年包含来自 Microsoft 的支持的 LTS 版本。 强烈建议将应用移到 .NET Core 3.1。 其他主要版本的当前生命周期如下所示：

| **长期支持** | |
| --- | --- |
| **Release** | **说明** |
| .NET Core 3.0 | 生命周期终结于 2020 年 3 月 3 日。 |
| .NET Core 2.2 | 生命周期终结于 2019 年 12 月 23 日。 |
| .NET Core 2.1 | 生命周期终结于 2021 年 8 月 21 日。 |

根据这个说明，为了维持Toolkit5.5的长期稳定性，因此将从.net core2.2升级到.net core3.1。

为了方便区分以前的版本，Toolkit5.5的版本号将升级到Toolkit5.6。此外，一些采用.net standard2.0版本的库也将同步升级到.net standard2.1。

## 升级的基本情况

升级过程比较简单，基本上都是修改相关的版本，比如从net standard2.0变成net standard2.1，net core2.2变成net core3.1，同时使用update-package更新依赖包的版本。然后编译一下即可。

有变化的几个库主要是Razor，WebApp以及启动的ToolkitWeb项目。

## Razor的变化

Razor是Toolkit系列库中变化最大的，主要原因是，其依赖的Razor库升级后，部分API起了变化。通过分析最新的RazorLight代码，更新了相关的代码。

在配置上，启动项目需要在csproj中增加：

<PreserveCompilationReferences>true</PreserveCompilationReferences>

否则，在运行razor时会出错。

## WebApp的变化

由于IHostingEnvironment过期，因此WebApp中使用IWebHostEnvironment来替代原先的IHostingEnvironment。

Request.Body必须异步访问。而原先可以使用同步访问来获取数据，在.net core3.1下，只能使用异步方式进行访问。否则程序就会报错。

## 启动ToolkitWeb项目的变化

首先是项目文件中必须追加：

<PreserveCompilationReferences>true</PreserveCompilationReferences>

其次，IHostingEnvironment修改为IWebHostEnvironment，因此Startup类需要进行相关调整。