页面内容通常是JSON格式，并使用客户端模板/视图呈现。改善启动体验的一种技术是预渲染主页。预呈现技术是将主页和一些重要页面创建为静态HTML文件。1对主页和这些重要页面的初始请求会立即在屏幕上获得HTML内容。当应用程序代码在后台下载和解析时，用户可以查看页面。

尽管某些网络爬虫（例如Google的Googlebot [2]）也可以对客户端呈现的应用程序进行爬网和编制索引，但它需要更多的资源，因此被推迟了。Google使用两波索引来处理这些类型的应用程序：.1。在第一波中，HTML和CSS将几乎立即被爬网和建立索引。.2。在第二波浪潮中，Google将重新呈现和索引JavaScript生成的数据，这可能需要几个小时到一周以上的时间[3]。其他Web爬网程序根本无法处理客户端呈现的应用程序。相反，服务器渲染的应用程序可以立即由Google编制索引，并且也可以由其他搜索引擎轻松处理。为了解决这个问题，Google建议使用动态渲染[4]。

动态渲染（或预渲染）是将客户端渲染的内容发送给用户，将服务器渲染的内容发送给搜索引擎和其他爬网程序的原理[2，4]。内容由无头浏览器3呈现并缓存，而不是向爬虫提供一个空的网页。由于网页输出是在功能齐全的浏览器中模拟的，因此动态呈现适用于所有类型的SPA，而与基础框架或系统无关。结果，可以将动态渲染委派给托管服务提供商-这是直接受支持的，例如由Netlify [5]。

预渲染技术是有限的，并且无法缩放。当应用程序具有许多页面和路线时，或者当它们具有动态内容时（例如，页面内容基于用户权限时），无法将动态内容预呈现为HTML文件。更好的方法是所谓的同构JavaScript方法或通用JavaScript，它可以在客户端和服务器端呈现相同的应用程序代码。AppRun是同构的/通用的。AppRun组件可以使用AppRun服务器端视图引擎在客户端和服务器端进行渲染（请参见图9-2）。

我们将开发一个示例AppRun SSR / SPA。该应用程序的页面呈现在服务器端。当应用程序启动时，主页是服务器渲染的全页面HTML。它在导航栏和主页上具有相同的时间戳（请参见图9-3）。应用程序开始重新加载主页后，我们可以单击“主页”菜单。屏幕已部分更新，以具有与导航栏中的时间戳不同的时间戳。从服务器返回的JSON数据是虚拟DOM（请参见图9-4）。

MVVM最重要的部分是ViewModel，它是view的抽象表达，但没有引用View。

MVVM设计模型可以将业务和逻辑分开，提高代码的可重用性。同时，它使程序易于开发，测试和维护[1]。开发人员和页面设计人员可以顺利合作。它可以提高系统开发效率。对于具有相同业务逻辑但UI要求不同的企业，开发人员只需要修改页面，而需要修改MVVM模型中MES的逻辑。因此，由于二次开发，MVVM设计模式可以大大减少代码量。与传统的MVC设计模型相比，MVVM设计模型清楚地将页面和页面逻辑分开。然后通过XAML和WPF的数据绑定技术连接页面和逻辑。因此，MVVM设计模型对于MES开发具有优势，并减少了二次开发的代码量。

MVVM设计模型是MVC设计模型的扩展。它优化了WPF核心属性，例如绑定，数据模板，命令和行为。在MVVM设计模型中，视图层通过数据绑定和命令绑定与视图模型层进行通信。视图模型层从模型层读取数据，并将数据转换为视图层以供显示[2]。视图数据绑定视图模型模型命令通知UI当前逻辑业务逻辑和数据图1 MVVM类图MVVM设计模型可分为几层，如图1.1所示：模型层，视图模型层，视图层。A.模型层包含真实数据和信息的模型层会费，记录所有业务对象和对象之间的关系。主要实现业务信息的二次打包。根据面向对象的规则，MES的业务信息被打包到可以通过视图模型读取的业务对象中。例如过程对象，工作站对象，人员对象，计划对象等。在整个MES中，模型层是系统业务的核心，并且是所有业务的抽象描述。模型层减少了业务层和视图模型层的互连。当业务信息发生变化时，可以通过更新模型层来维护和更新MES。B.视图层视图层是最经常与客户通信的层，它向客户显示所有信息。类似于MVC的视图层，用户可以通过视图层中的通信页面获得良好的体验。在MVVM设计模型中，视图层仅用于开发前景页面。如窗框，用户控件，控件样式。没有业务和视图逻辑。因此，只有那些熟悉XAML语言的人才能设计View层。查看页面和逻辑可以同时开发。它极大地提高了系统开发的效率。UI设计者可以通过简单的绘图视图层来设计初始系统模型，以符合MES调查中的用户要求。同时，初始系统模型可以用作系统的最终用户界面。视图层是最常修改的层。因此，视图层在MVVM设计模型中是分开的。此设计模型减少了UI修改的影响。