**ASP.NET的页输出缓存**

2015-08-31 15:45:36[7](http://www.th7.cn/Program/net/201508/549848.shtml)人点击

整理了一下MSDN中关于ASP.NET缓存机制的论述。

通常，应用程序可以将那些频繁访问的数据，以及那些需要大量处理时间来创建的数据存储在内存中，从而提高性能。例如，如果应用程序使用复杂的逻辑来处理大量数据，然后再将数据作为用户频繁访问的报表返回，避免在用户每次请求数据时重新创建报表可以提高效率。同样，如果应用程序包含一个处理复杂数据但不需要经常更新的页，则在每次请求时服务器都重新创建该页会使工作效率低下。

在这些情况下，为了帮助您提高应用程序的性能，ASP.NET 使用两种基本的缓存机制来提供缓存功能。第一种机制是应用程序缓存，它允许您缓存所生成的数据，如 DataSet 或自定义报表业务对象。第二种机制是页输出缓存，它保存页处理输出，并在用户再次请求该页时，重用所保存的输出，而不是再次处理该页。应用程序缓存

应用程序缓存提供了一种编程方式，可通过键/值对将任意数据存储在内存中。使用应用程序缓存与使用应用程序状态类似。但是，与应用程序状态不同的是，应用程序缓存中的数据是易失的，即数据并不是在整个应用程序生命周期中都存储在内存中。使用应用程序缓存的优点是由 ASP.NET 管理缓存，它会在项过期、无效、或内存不足时移除缓存中的项。还可以配置应用程序缓存，以便在移除项时通知应用程序。有关更多信息，请参见缓存应用程序数据。

使用应用程序缓存的模式是，确定在访问某一项时该项是否存在于缓存中，如果存在，则使用。如果该项不存在，则可以重新创建该项，然后将其放回缓存中。这一模式可确保缓存中始终有最新的数据。

有关更多信息，请参见如何：检索缓存项的值。

页输出缓存在内存中存储处理后的 ASP.NET 页的内容。这一机制允许 ASP.NET 向客户端发送页响应，而不必再次经过页处理生命周期。页输出缓存对于那些不经常更改，但需要大量处理才能创建的页特别有用。例如，如果创建大通信量的网页来显示不需要频繁更新的数据，页输出缓存则可以极大地提高该页的性能。可以分别为每个页配置页缓存，也可以在 Web.config 文件中创建缓存配置文件。利用缓存配置文件，只定义一次缓存设置就可以在多个页中使用这些设置。

页输出缓存提供了两种页缓存模型：整页缓存和部分页缓存。整页缓存允许将页的全部内容保存在内存中，并用于完成客户端请求。部分页缓存允许缓存页的部分内容，其他部分则为动态内容。有关更多信息，请参见ASP.NET 页。

部分页缓存可采用两种工作方式：控件缓存和缓存后替换。控件缓存有时也称为分段缓存，这种方式允许将信息包含在一个用户控件内，然后将该用户控件标记为可缓存的，以此来缓存页输出的部分内容。这一方式可缓存页中的特定内容，并不缓存整个页，因此每次都需重新创建整个页。例如，如果要创建一个显示大量动态内容（如股票信息）的页，其中有些部分为静态内容（如每周总结），这时可以将静态部分放在用户控件中，并允许缓存这些内容。

缓存后替换与控件缓存正好相反。这种方式缓存整个页，但页中的各段都是动态的。例如，如果要创建一个在规定时间段内为静态的页，则可以将整个页设置为进行缓存。如果向页添加一个显示用户名的 Label 控件，则对于每次页刷新和每个用户而言，Label 的内容都将保持不变，始终显示缓存该页之前请求该页的用户的姓名。但是，使用缓存后替换机制，可以将页配置为进行缓存，但将页的个别部分标记为不可缓存。在此情况下，可以向不可缓存部分添加 Label 控件，这样将为每个用户和每次页请求动态创建这些控件。有关更多信息，请参见缓存 ASP.NET 页的某些部分。