

УДК 004

**ЦЕЛЕСООБРАЗНОСТЬ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ ASP.NET
И ЕЕ ПЛАТФОРМЫ WEBAPI ПРИ РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЙ****С.Н. МАЛАШОНОК,***(Представлено: Ю.Н. КРАВЧЕНКО)*

Рассматривается технология разработки ASP.NET, которая является частью платформы NetFramework и предназначена для написания клиент-серверных приложений и веб-сервисов, и платформы WebAPI, которая является частью ASP.NET. Анализируются целесообразность и преимущества ее использования.

ASP.NET представляет собой технологию, направленную на создание серверных веб-приложений и веб-сервисов различной сложности. Она содержит большое количество готовых элементов управления и различных технологий, начиная от технологии, предоставляющий доступ к базам данных и заканчивая технологией создания распределенных приложений.

Технология развивалась из более ранней версии технологии MicrosoftASP. Технология ASP имела ограничения по своим возможностям. Код программы вставлялся прямо в HTML-страницу в виде специальных тегов, что было неудобно и создавало путаницу.

Пример такой HTML-страницы представлен на рисунке 1.

```
<%@ Language=VBScript %>
  <% Option Explicit%>
  <HTML>
  <HEAD>
  <META HTTP-EQUIV="Content-Type" content="text/html">
  <TITLE>
  Hello ASP World!
  <TITLE>
  </HEAD>
  <BODY>
  <P><%
  Dim i;
  for i=1 to 5
  Response.Write("<FONT SIZE=" & i)
  Response.Write(">Этот код генерирует ASP!</FONT>")
  next i
  %>;</P>
  </BODY>
  </HTML>
```

Рисунок 1. – Пример файла ASP

Microsoft полностью перестроила ASP.NET, основываясь на CommonLanguageRuntime (CLR), которая является основой всех приложений Microsoft .NET. Разработчики могут писать код для ASP.NET, используя практически любые языки программирования, входящие в комплект .NET Framework (C#, Visual Basic.NET и JScript.NET) [1]. Как и все приложения .NET, приложения ASP.NET всегда компилируются. На первом этапе компиляции код компилируется в промежуточный язык IL, который на втором этапе компилируется в код на низкоуровневом машинном языке. IL-код создается только один раз и генерируется снова только при изменении исходного кода, а файлы, содержащие машинный код, кэшируются в системном каталоге. Благодаря этому механизму ASP.NET приложения не нуждаются в выполнении компиляции при каждом запросе веб-страницы.

В отличие от ASP, ASP.NET является объектно-ориентированной технологией, которая не только предоставляет коду доступ ко всем объектам .NETFramework, но и позволяет использовать все концепции объектно-ориентированного программирования. Технология позволяет создавать повторно используемые классы, использовать интерфейсы, расширять классы с помощью механизма наследования.

ASP.NET предоставляет возможность кэшировать ответы, сгенерированные страницей ASP.NET. Кэширование позволяет обработать запросы к одной и той же страницы с помощью кэша, при этом код, с помощью которого была создана страница, не будет выполняться повторно.

В ASP.NET3.5 появилась технология LanguageIntegratedQuery(LINQ), которая представляет собой набор расширений для языков C# и VisualBasic. Она позволяет писать на C# или VisualBasic код, способный манипулировать находящимися в памяти данными во многом в той же манере, что и запросы к базе данных. Самой главной разновидностью LINQ является LINQ to Objects, которая позволяет брать коллекцию объектов и выполнять запрос, извлекающий определенную информацию из некоторых этих объектов [2].

С выходом ASP.NET4.5 появился новый инструмент для создания веб-служб – WebAPI, который позволяет создавать HTTP-службы, доступные для широкого круга клиентов, включая браузеры и мобильные устройства.

Упрощение создания веб-служб является неотъемлемой особенностью Web API. Оно представляет собой значительное улучшение по сравнению с другими технологиями построения веб-служб, которые компания Microsoft предлагала на протяжении последнего десятилетия [3].

С помощью WebApi можно создать библиотеку классов, которая будет обрабатывать HTTP-запросы. Данная библиотека может быть расположена в исполняющей среде, благодаря чему к ней можно будет обращаться по HTTP. Открытые методы контроллеров являются конечными точками HTTP, которые могут возвращать данные в виде двоичных данных, JavaScriptObjectNotation (JSON), eXtensibleMarkupLanguage (XML) или обычного текста. Регулируемые правила маршрутизации позволяют задать форму UniformResourceLocator (URL), что значительно упрощает передачу данные в методы контроллера. Пример класса контроллера с методом представлен на рисунке 2.

```
public class ValuesController : ApiController
{
    // GET api/values
    public IEnumerable<string> Get()
    {
        return new string[] { "value1", "value2" };
    }

    // GET api/values/5
    public string Get(int id)
    {
        return "value";
    }
}
```

Рисунок 2. – Метод контроллера

Таким образом, к основным преимуществам ASP.NET относятся:

- компиляция кода;
- использование объектно-ориентированного программирования;
- кэширование страниц;
- большое количество стандартных объектов, предоставляемых платформой .NetFramework;
- удобное создание веб-сервисов.

ЛИТЕРАТУРА

1. Принципы ASP.NET [Электронный ресурс]. – Режим доступа: https://studopedia.ru/2_30490_printsipi-ASPNET.html. – Дата доступа: 16.09.2017.
2. Мак-Дональд, М. MicrosoftASP.NET 4 с примерами на C# 2010 для профессионалов / Мэтью Мак-Дональд, Адам Фримен, Марио Шпуста. – 4-е изд. ; под ред. Ю.Н. Артеменко. – М. : И.Д. Вильямс, 2010. – 1424 с.
3. Одностраничные приложения [Электронный ресурс]. – Режим доступа https://professorweb.ru/my/ASP.NET/mvc/level8/8_1.php. – Дата доступа: 21.09.2017.