Подходы к разработке web-приложений

Подходы к разработке web-приложений

Все подходы к разработке web-приложений могут быть разделены на 3 большие категории:

1. Подходы, основанные на программировании или скриптах: внешние программы или скрипты; расширения web-сервера.

(Plugin’ы к nginx, Lua скрипты, и т.д.)

2. Подходы, основанные на использовании шаблонов web-страниц, включающих вставки кода скриптов и специальных серверных тэгов. (скрипты типа .php, .pl для отдельных handler’ов)

3. Объектные среды (web-frameworks).

Image-Filter

Crop, resize, rotate, and perform other transformations on GIF, JPEG, and PNG images, with the Image-Filter dynamic module supported by NGINX, Inc.

Installation Instructions

1. Install the Image-Filter module.

   For Amazon Linux, CentOS, Oracle Linux, and RHEL:

   ```
   $ yum install nginx-plus-module-image-filter
   ```

   For Debian and Ubuntu:

   ```
   $ apt-get install nginx-plus-module-image-filter
   ```

   For SLES:

   ```
   $ zypper install nginx-plus-module-image-filter
   ```

   For Alpine:

   ```
   $ apk add nginx-plus-module-image-filter
   ```

2. Put the `load_module` directive in the top-level (“`main`”) context of NGINX Plus configuration file, **nginx.conf**:

   ```
   load_module modules/ngx_http_image_filter_module.so;
   ```

3. Perform additional configuration as required by the module.

4. Reload NGINX Plus to enable the module:

   ```
   $ nginx -t && nginx -s reload
   ```

Change language: Russian
Submit a Pull Request Report a Bug

Работа с формами

Одно из главнейших достоинств PHP - то, как он работает с формами HTML. Здесь основным является то, что каждый элемент формы автоматически становится доступным вашим программам на PHP. Для подробной информации об использовании форм в PHP читайте раздел Переменные из внешних источников. Вот пример формы HTML:

Пример #1 Простейшая форма HTML

```
<form action="action.php" method="post">
 <p>Ваше имя: <input type="text" name="name" /></p>
 <p>Ваш возраст: <input type="text" name="age" /></p>
 <p><input type="submit" /></p>
</form>
```

В этой форме нет ничего особенного. Это обычная форма HTML без каких-либо специальных тегов. Когда пользователь заполнит форму и нажмёт кнопку отправки, будет вызвана страница `action.php`. В этом файле может быть что-то вроде:

Пример #2 Выводим данные формы

```
Здравствуйте, <?php echo htmlspecialchars($_POST['name']); ?>.
Вам <?php echo (int)$_POST['age']; ?> лет.
```

Пример вывода данной программы:

```
Здравствуйте, Сергей. Вам 30 лет.
```

Если не принимать во внимание куски кода с htmlspecialchars() и (int), принцип работы данного кода должен быть прост и понятен. htmlspecialchars() обеспечивает правильную кодировку "особых" HTML-символов так, чтобы вредоносный HTML или Javascript не был вставлен на вашу страницу. Поле age, о котором нам известно, что оно должно быть число, мы можем просто преобразовать в int, что

Программные подходы

В данном подходе web-приложением (динамическим ресурсом, связанным с URL-адресом) является внешняя программа, составленная на некотором универсальном языке программирования высокого уровня, или скрипт, составленный с помощью скриптового языка, выполнение которого производится также с помощью внешней программы – интерпретатора скриптов (script engine).

Основной проблемой с программным подходом к разработке web- приложений является их ориентация на написание кода. Разметка HTML и другие конструкции форматирования встраиваются в логику работы программы с помощью операторов вывода.

Программные подходы

Это ограничивает возможности web-дизайнеров вносить свой вклад в оформление создаваемой приложением страницы. Web-дизайнер может разрабатывать макет страницы, а программист должен затем преобразовать его в код и связать со скриптом или программой. Для изменения практически любого элемента формируемой страницы требуется вмешательство программиста, касается ли это изменения логики работы программы, либо изменения оформления и расположения элементов страницы.

Внешние программы

Простейший способ динамически формировать web-страницы в ответ на HTTP-запрос заключается в том, чтобы передать работу по решению требуемой задачи и формированию HTML-страницы внешней программе, которая должна получать переданные в HTTP-запросе входные параметры и сформировать выходную страницу на языке HTML.

Первой широко используемой, независимой от типа web-сервера программной технологией создания и выполнения web-приложений была технология Common Gateway Interface (CGI, общий шлюзовой интерфейс). Она определяла набор правил, которым должна следовать программа, чтобы она могла выполняться на разных HTTP-серверах и операционных системах.

Внешние программы

В соответствии с CGI-технологией при поступлении в web-сервер HTTP-запроса, который включает ссылку не на статическую страницу, а на CGI-программу, создается новый процесс, в котором запускается требуемая прикладная программа.

Технология CGI задает способ передачи такой программе параметров, входящих в состав HTTP-запроса. Передача входных данных может выполняться либо с помощью фиксированного набора переменных среды (environment variables), которые могут создаваться одной программой и использоваться другими программами, либо через входные данные функции, с которой начинается работа программы (функция main()), а результаты работы программы (HTML-страница) возвращаются с помощью стандартного потока вывода STDOUT.

Пример простых CGI-программ

Пример программы на C

```
#include "stdafx.h"
#include "cgi1.h"
#include <stdlib.h>
int main(int argc, TCHAR* argv[ ], TCHAR* envp[ ]) {
printf("Content-Type: text/html\n\n"); printf("<HTML>\n");
printf("<HEAD> <TITLE>Пример CGI-программы </TITLE></HEAD>\n");
printf("<BODY>"\n); printf("Пример CGI-программы на C\n");
int i=0;
while(envp[i]) {
   // печать параметров переданных программе в массиве envp
   printf("<p>значение параметра %s </p>",envp[i]);
   i++;
}
printf("</BODY>\n"); printf("</HTML>\n);
return 0;
}
```

Пример скрипта на Perl

```
#! /usr/local/bin/perl

sub ReadFormFields { . . . }
sub PrintFormFields {
    my $fieldsRef = shift;
    my $key, $value;
    print "Content-Type: text/html\n\n" ;
    print "<html>\n<head><title>hello</title></head>\n" ;
    print "<body>\n" ;
    foreach $key (keys(%$fieldsRef) ) {
        $value = $$fieldsRef{ $key} ; print " <h3>$key: $value</h3>\n";
    }
    print "</body>\n</html>\n";
}
&ReadFormFields(\%fields) ;
&PrintFormFields(\%fields) ;
exit 0;
```

Внешние программы

Технология CGI позволяет использовать любой язык программирования, который может работать со стандартными устройствами ввода/вывода. Кроме этого, CGI-программы можно писать с использованием скриптовых языков, которые называются "CGI-скриптами". Примерами скриптовых CGI-языков являются, например, Perl, Python или Tel. При использовании скрипта web-ссрвср вызывает на выполнение внешнюю программу – интерпретатор скриптов (script engine), которой передаются данные HTTP-запроса и имя файла, в котором содержится запрашиваемый пользователем скрипт. А затем данная программа выполняет указанный скрипт и возвращает серверу сформированную HTML-страницу.

Недостатки технологии CGI

- Основной проблемой является производительность: для каждого HTTP-запроса к CG-программе web-сервер запускает новый процесс, который заканчивает работу только после завершения программы. Работа по созданию и завершению процессов является достаточно трудоемкой, что может очень быстро понизить производительность системы; кроме этого, различные активные процессы начинают конкурировать за системные ресурсы, такие как оперативная память.
- Для составления и отладки CGI-программ разработчик должен обладать достаточно большим опытом программирования на одном из языков, на котором можно программировать CGI-программы.
- В CGI-программах программный код и код разметки полностью перемешаны. Дизайнер должен знать программирование, чтобы менять структуру web-страниц.

Fast CGI

Попыткой объединить переносимость CGI-приложсний с эффективностью является технология FastCGI. Данная технология основывается на простой идее: вместо необходимости каждый раз запускать новый процесс для обработки CGI-скрипта FastCGI позволяет не закрывать процессы, связанные с CGI-скриптами, после окончания обработки, а использовать их для обработки новых запросов к CGI-программам. А это означает, что не требуется постоянно запускать и удалять новые процессы, т. к. один и тот же процесс может использоваться многократно для обработки запросов. Такие процессы могут инициализироваться только один раз при их создании.

Модули сервера, которые выполняют функциональность FastCGI, взаимодействуют с HTTP-сервером с помощью своих собственных API. Эти API стараются скрыть детали реализации и конфигурирования от FastCGI-приложений, но разработчики все равно должны знать особенности реализации технологии FastCGI, т. к. модули различных типов серверов несовместимы между собой.

Интерфейс ISAPI

Для web-сервера Microsoft IIS (Internet Information Server) был разработан специальный программный интерфейс – ISAPI, позволяющий создавать приложения, расширяющие стандартные возможности данного web-сервера. ISAPI представляет собой библиотеку функций, с помощью которой программисты могут создавать web-приложения в виде DLL-модулей (динамически подключаемых библиотек), формирующих HTML- страницы. Такие web-приложсния работают намного быстрее обычных CGI-программ, т. к. они более тесно интегрированы в web-сервер.

Интерфейс ISAPI

ISAPI-расширения могут связываться с вызовом файлов, имеющих специальные расширения, либо с файлами, содержащимися в заданных каталогах или во всем web-сайте. *ISAPI-фильтры* используются для изменения или совершенствования функциональности IIS-сервера. Обычно они обрабатывают (фильтруют) каждый поступающий HTTP-запрос. Фильтры могут применяться для анализа и модификации исходящих HTTP-ответов.

ISAPI-приложения могут разрабатываться с помощью разных языков, в основном это .NET, ASP, Delphi, C++

Интерфейс Java Servlet API

Другой широко используемой технологией расширения архитектуры web-сервера является прикладной интерфейс Java Servlet API, который связывает web-сервер с виртуальной машиной Java Virtual Machine (JVM). Виртуальная машина JVM поддерживает выполнение специальной Java-программы (контейнер сервлетов), которая отвечает за управление данными сеанса работы и выполнение Java-сервлетов.

Сервлеты – это специальные классы на языке Java, которые имеют доступ к информации из HTTP-запросов. Они формируют HTTP-ответы, которые возвращаются браузерам.

Интерфейс Java Servlet API

В отличие от ISAPI-расширений технология Servlet API является переносимой между разными web-серверами, операционными системами и компьютерными платформами. Сервлеты выполняются одинаково в любой среде, которая предоставляет совместимый с ними контейнер сервлетов. Технология Servlet API используется большим количеством разработчиков и поддерживается многими известными web-серверами.
Чаще всего таким сервером выступает Apache Tomca

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FormServlet extends HttpServlet {
public void doGet( HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
  response.setContentType("text/html" ) ;
  PrintWriter out = response.getWriter();
  out.println("<html>\n<head><title>hello</title></head>" ) ;
  out.println("<body>");
  Enumeration e = request.getParameterNames() ;
  while (e. hasMoreElements() ) {
      String name = (String) e.nextElement() ;
      String value = request.getParameter(name) ;
      out.println(" <h3>" + name + &: & + value + " </ h3>" ) ;
  }
  out.println( "</body>\n</html>" ) ;
}
public void doPost(HttpServletRequest request, HttpServletResponse response) throws
IOException, ServletException {
  doGet( request, response);
}}
```

Подходы на основе шаблонов

Подходы, основанные на *шаблонах* (template approaches – шаблонные подходы), используют в качестве адресуемых объектов (имеющих URL-адрес) не программы или скрипты, а *"шаблоны"*. По существу шаблонами являются HTML-файлы с дополнительными "тэгами" (серверными, используемыми только на стороне сервера), которые задают методы включения динамически формируемого контента. Таким образом, файл шаблона содержит HTML-код, который описывает общую структуру страницы, и дополнительные ***серверные тэги***, размещенные таким образом, чтобы формируемое с их помощью содержание страницы имело требуемый вид.

Подходы на основе шаблонов

В конце 90-х гг. многие компании разработали свои собственные технологии обработки шаблонов на стороне web-сервера, включающие скрипты. Компания Netscape предложила технологию LiveWire (которая развилась в язык Server-Side JavaScript), а другие компании разработали такие технологии, как NetDynamics, Dynamo и Cold Fusion (из этих технологий до настоящего времени используется только Cold Fusion).

В настоящее время к наиболее распространенным технологиям разработки web-приложений на основе шаблонов относятся следующие: *Server-Side Includes* (*SSI*), *Cold Fusion*, *Active Server Pages (ASP)* и *Java Server Pages (JSP).*

Технология SSI

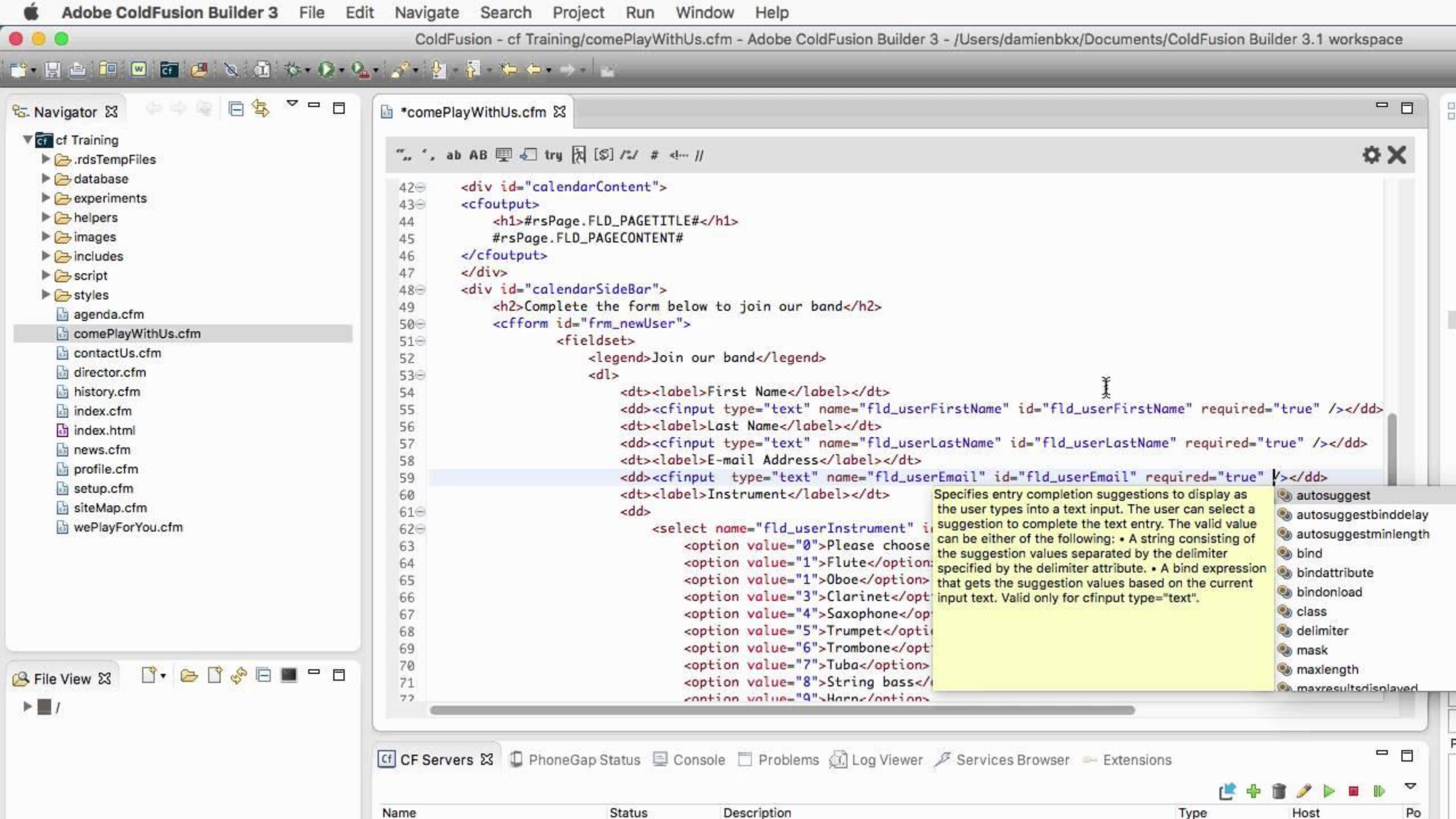
Технология вставок на стороне сервера – Server Side Includes (SSI) – является старой технологией, которая появилась почти одновременно с технологией CGI. SSI предоставляет возможность вставки дополнительных файлов (или результатов выполнения CGI-скриптов) в HTML- страницу. Вставка в шаблон SSI-инструкций выполняется с помощью следующего формата:

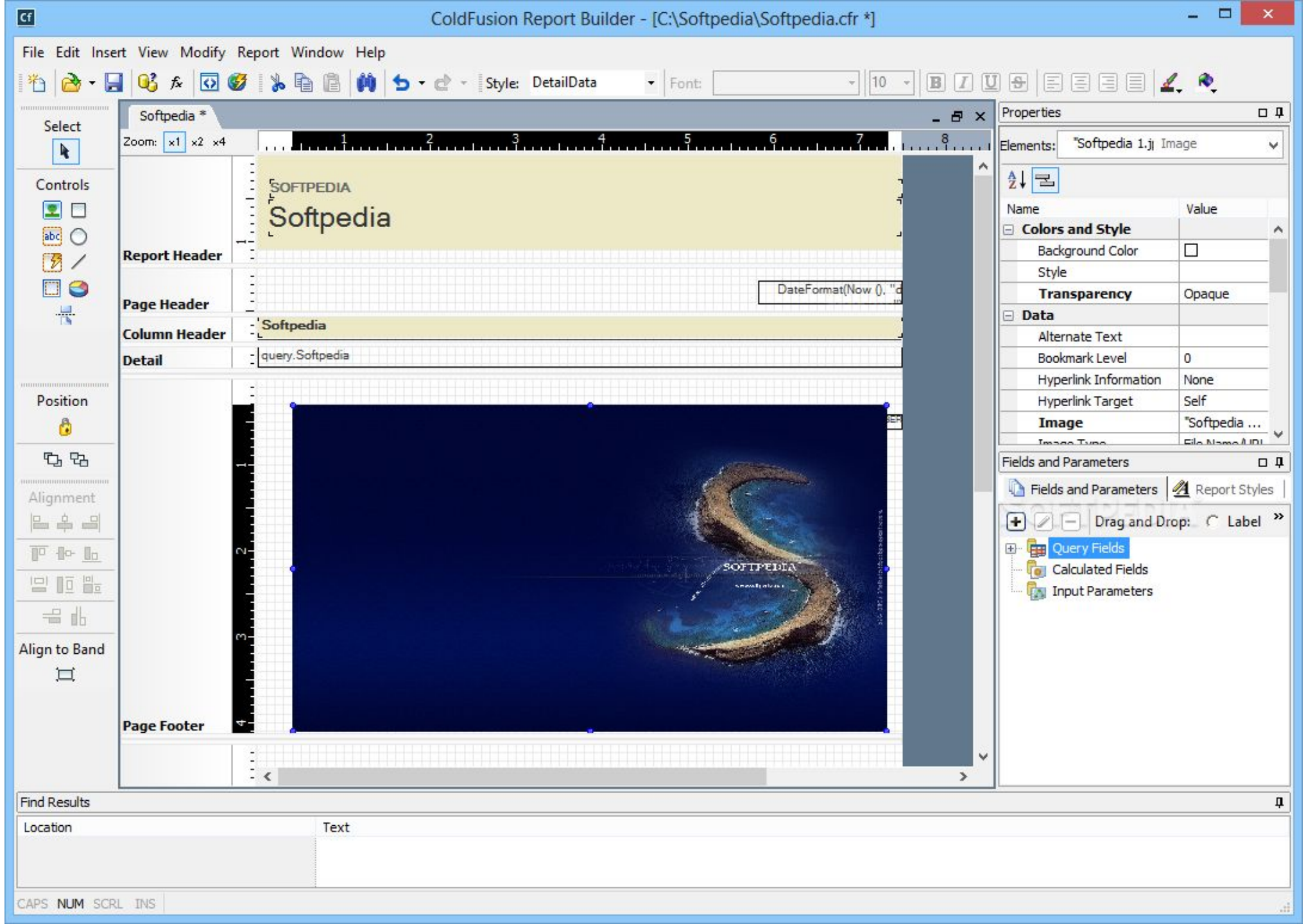
<!-- ins

```
<html>
<head><title>hello</title></head>
<body>
  <! -- #exec cgi http://mysite.org/cgi-bin/zip-ssi.cgi -->
</body>
</html>
```

Технология Cold Fusion

Другой достаточно популярной технологией, основанной на шаблонах, является технология Cold Fusion, разработанная компанией Adobe. Пример шаблона, описанного на основе данной технологии, показан на рис. 3.6. В данном шаблоне используются специальные тэги, внешне очень похожие на HTML-гэги, но начинающиеся с приставки "CF". Такие тэги не передаются браузерам, а обрабатываются средой выполнения на стороне web-сервера.


Adobe ColdFusion Builder 3 File Edit Navigate Search Project Run Window Help
ColdFusion - cf Training/comePlayWithUs.cfm - Adobe ColdFusion Builder 3 - /Users/damienbkx/Documents/ColdFusion Builder 3.1 workspace
Navigator
cf Training
.rdsTempFiles
database
experiments
helpers
images
includes
script
styles
agenda.cfm
comePlayWithUs.cfm
contactUs.cfm
director.cfm
history.cfm
index.cfm
index.html
news.cfm
profile.cfm
setup.cfm
siteMap.cfm
wePlayForYou.cfm
File View
*comePlayWithUs.cfm
42
43 <cfoutput>
44 <h1>#rsPage.FLD_PAGETITLE#</h1>
45 #rsPage.FLD_PAGECONTENT#
46 </cfoutput>
47
48
49 <h2>Complete the form below to join our band</h2>
50 <cfform id="frm_newUser">
51 <fieldset>
52 <legend>Join our band</legend>
53 <dl>
54 <dt><label>First Name</label></dt>
55 <dd><cfinput type="text" name="fld_userFirstName" id="fld_userFirstName" required="true" /></dd>
56 <dt><label>Last Name</label></dt>
57 <dd><cfinput type="text" name="fld_userLastName" id="fld_userLastName" required="true" /></dd>
58 <dt><label>E-mail Address</label></dt>
59 <dd><cfinput type="text" name="fld_userEmail" id="fld_userEmail" required="true" /></dd>
60 <dt><label>Instrument</label></dt>
61 <dd>
62 <select name="fld_userInstrument"
63 <option value="0">Please choose
64 <option value="1">Flute</option
65 <option value="1">Oboe</option>
66 <option value="3">Clarinet</opt
67 <option value="4">Saxophone</op
68 <option value="5">Trumpet</opti
69 <option value="6">Trombone</opt
70 <option value="7">Tuba</option>
71 <option value="8">String bass</
Specifies entry completion suggestions to display as the user types into a text input. The user can select a suggestion to complete the text entry. The valid value can be either of the following: • A string consisting of the suggestion values separated by the delimiter specified by the delimiter attribute. • A bind expression that gets the suggestion values based on the current input text. Valid only for cfinput type="text".
autosuggest
autosuggestbinddelay
autosuggestminlength
bind
bindattribute
bindonload
class
delimiter
mask
maxlength
CF Servers
PhoneGap Status
Console
Problems
Log Viewer
Services Browser
Extensions
Name
Status
Description
Type
Host


ColdFusion Report Builder - [C:\Softpedia\Softpedia.cfr *]
File Edit Insert View Modify Report Window Help
Style: DetailData
Font:
Select
Controls
Position
Alignment
Align to Band
Softpedia *
Zoom: x1 x2 x4
SOFTPEDIA
Softpedia
Report Header
Page Header
DateFormat(Now (), "d
Column Header
Softpedia
Detail
query.Softpedia
Page Footer
Properties
Elements: "Softpedia 1.jp Image
Name
Value
Colors and Style
Background Color
Style
Transparency
Opaque
Data
Alternate Text
Bookmark Level
0
Hyperlink Information
None
Hyperlink Target
Self
Image
"Softpedia ...
Fields and Parameters
Fields and Parameters
Report Styles
Drag and Drop:
Label
Query Fields
Calculated Fields
Input Parameters
Find Results
Location
Text
CAPS NUM SCRL INS

Технология Cold Fusion

Преимущество данного подхода заключается в том, что такой шаблон может создаваться и поддерживаться дизайнером страницы, который имеет базовые знания языка HTML и web-графики, но не имеет опыта программирования. Специальные тэги, которые являются "расширением" HTML, в некоторой степени похожи на инструкции (тэги) SSI тем, что дизайнеры web-страниц, имеющие небольшой опыт работы, могут быстро научиться использовать их.

Технология PHP Hypertext Preprocessor

Технология "PHP Hypertext Preprocessor" (хотя первоначально она соответствовала "Personal Home Page"), или просто PHP, позволяет разработчикам встраивать программный код в шаблоны с помощью языка, сходного с языком скриптов Perl. Ниже приведен пример фрагмента PHP-шаблоиа:

```
<b>
<?php if ($xyz >= 3 ) { print $myHeading; }
  else {
?>
  DEFAULT HEADING
<?php } ?>
</b>
```

Технология PHP Hypertext Preprocessor

Из данного примера понятно, что текст, встроенный в блоки вида <?php ...?>, обрабатывается PHP-процессором, а текст, стоящий вне таких блоков, обрабатывается как аргумент, переданный операторам print.

(Такой же подход используется и в технологиях JavaServer Pages и ASP.Net Web Forms.)

Технология Active Server Pages

Компания Microsoft разработала технологию ASP (Active Server Pages), которая объединила возможности создания шаблонов, включающих скрипты, с доступом к наборам OLE- и COM-объектов, имеющихся в операционной системе Windows, в т. ч. и к ODBC-источникам данных. Данная технология, объединенная с бесплатным web-сервером Internet Information Server (IIS), быстро стала популярной среди программистов, использующих Visual Basic, которые оценили возможность использования в шаблонах языка VBScript. Как и PHP-шаблоны, ASP-страницы могут включать блоки скриптов

Технология Active Server Pages

В отличие от технологии PHP, ASP не связан с одним конкретным скриптовым языком. В ASP в качестве стандартного языка используется язык Visual Basic Scripting Edition (VBScript), но может использоваться и язык JavaScript.

В ASP-шаблоны (как и в PHP-шаблоны) могут включаться блоки, выделенные с помощью тэгов <% ... %>, которые содержат код скрипта, выполняемый интерпретатором ASP-шаблонов, при формировании ответа. HTML-разметка, которая находится вне таких блоков, рассматривается как исходный HTML-код и просто переписывается в формируемую HTML- страницу. Кроме этого, в начало шаблона могут добавляться директивы страницы, которые информируют систему обработки об используемом скриптовом языке.

Технология Active Server Pages

```
<% @LANGUAGE = VBScript %>
<% Set conn = Server.CreateObject("ADODB. Connection")
conn.open("Data Source=mydata; User ID=myname;Password=******")
Set results = Server.CreateObject("ADODB.RecordSet")
Set results.ActiveConnection = conn
query = "SELECT X, Y, Z FROM TABLE1 WHERE X > 23"
results.Open query %>
<html> <head><title>Active Server Page Example</title></head>
<body id="c09-body-0003 " bgcolor="#ffffff" >
<table>
   <tr>
      <td align="center">X</td> <td align="center">Y</td> <td align="center">Z</td>
   </tr>
<%
   While Not results. EOF
      Response.Write "<tr>" Response.Write "<td>" &results(" X") & " </td>"
      Response.Write "<td>" &results(" Y") & " </td>"
      Response.Write "<td>" &results(" Z") & " </td>" Response.Write "</tr>"
   Wend
%>
</table>
</body>
</html>
```

X	Y	Z
A	12	.005
G	15	.078
J	10	.125
M	27	.351

Технология Active Server Pages

Тот факт, что технология ASP входила в состав web-сервера Microsoft IIS, сделал её очень привлекательной для использования разработчиками, работающими в ОС Windows. В связи с популярностью технологии ASP она была реализована и на других платформах, помимо Microsoft IIS. Основным преимуществом данной технологии является ускорение разработки и установки относительно простых web-приложений.

Технология Java Server Pages

Технология JSP была ответом компании Sun на популярность технологии Microsoft ASP.

В первой строке примера показана директива страницы <%@ page ...> (интересно обратить внимание на ее сходство с директивой web-формы ASP.Net), в которой указывается на импортирование классов из пакета java.io. В следующей строке выполняется объявление переменной. Код на языке Java выделяется (также как и в PHP и ASP) специальными последовательностями символов <% ... %>.

```
<%@ page import="java.io.*" %>
<%! private CustomObject myObject; %>
<h1>My Heading</h1>
<%
   for(int i = 0; i < myObj ect. getCount(); i++) { %>
      <p>Item #<%= i %> is ' <%= myObject.getItem(i) %>' . </p>
<% } %>
```

Технология Java Server Pages

Как и технология PHP, выполнение JSP-страниц реализуется с помощью препроцессора, который преобразует (транслирует) их в исходный код сервлета. HTML-разметка, которая стоит вне выделенных блоков, транслируется в операторы print языка Java (как показано на рис. 3.9).

Технология JSP развивалась, и со временем к ней были добавлены такие новые возможности, как библиотеки JSP-тэгов
(JSP taglib). Библиотека тэгов *taglib* – это набор специальных (серверных) JSP-тэгов, которые не передаются в HTTP-ответе браузеру, а используются при обработке JSP-страницы в контейнере сервлетов на стороне web-сервера.

Технология Java Server Pages

Фактически каждый специальный тэг – это некоторая функциональность, для реализации которой в противном случае потребовалось бы включать некоторый встроенный блок, содержащий Java-код. Например, двумя наиболее часто используемыми тэгами являются: <jsp:useBean> и <jsp:getProperty>. Префикс "jsp:" говорит о том, что это не HTML-тэг, а специальный (серверный) тэг, который будет использоваться на стороне сервера. Тэг <jsp:useBean> позволяет разработчикам встраивать в JSP-страницу JavaBean-объекты (созданные и наполненные приложением в ходе сеанса работы пользователя). К ним можно получить доступ и изменить их значения с помощью тэгов <jsp:getProperty> и <jsp:setProperty>.

Подходы на основе объектных сред

Обычные скриптовые технологии на стороне сервера используют различные объекты, но не позволяют разрабатывать и использовать собственные классы и создавать на их основе объекты. В связи с этим дальнейшее развитие web-технологий было связано с созданием специальных объектно ориентированных технологий разработки web- приложений. Использование данных технологий позволяет сделать разработку web-приложений более сходной с разработкой обычного объектно ориентированного программного обеспечения.

Подходы на основе объектных сред

Объектные среды (фреймворки) представляют собой следующий уровень совершенствования разработки web-приложений.

Вместо объединения разметки и логики в единый модуль, объектные среды поддерживают принцип отделения содержания от представления. Модули, ответственные за создание контента, отделяются от модулей, которые показывают это содержание в конкретном формате.

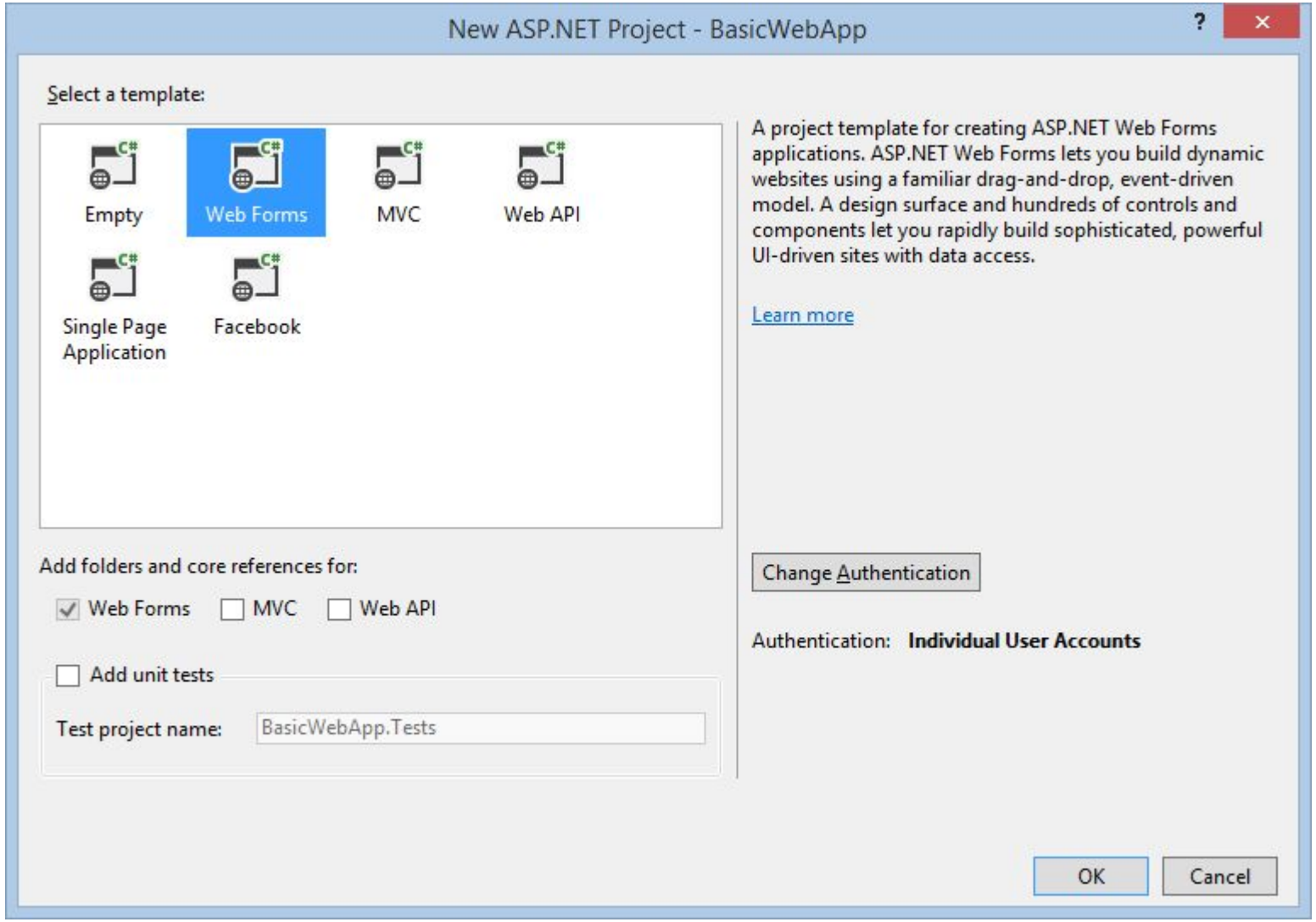
Подходы на основе объектных сред

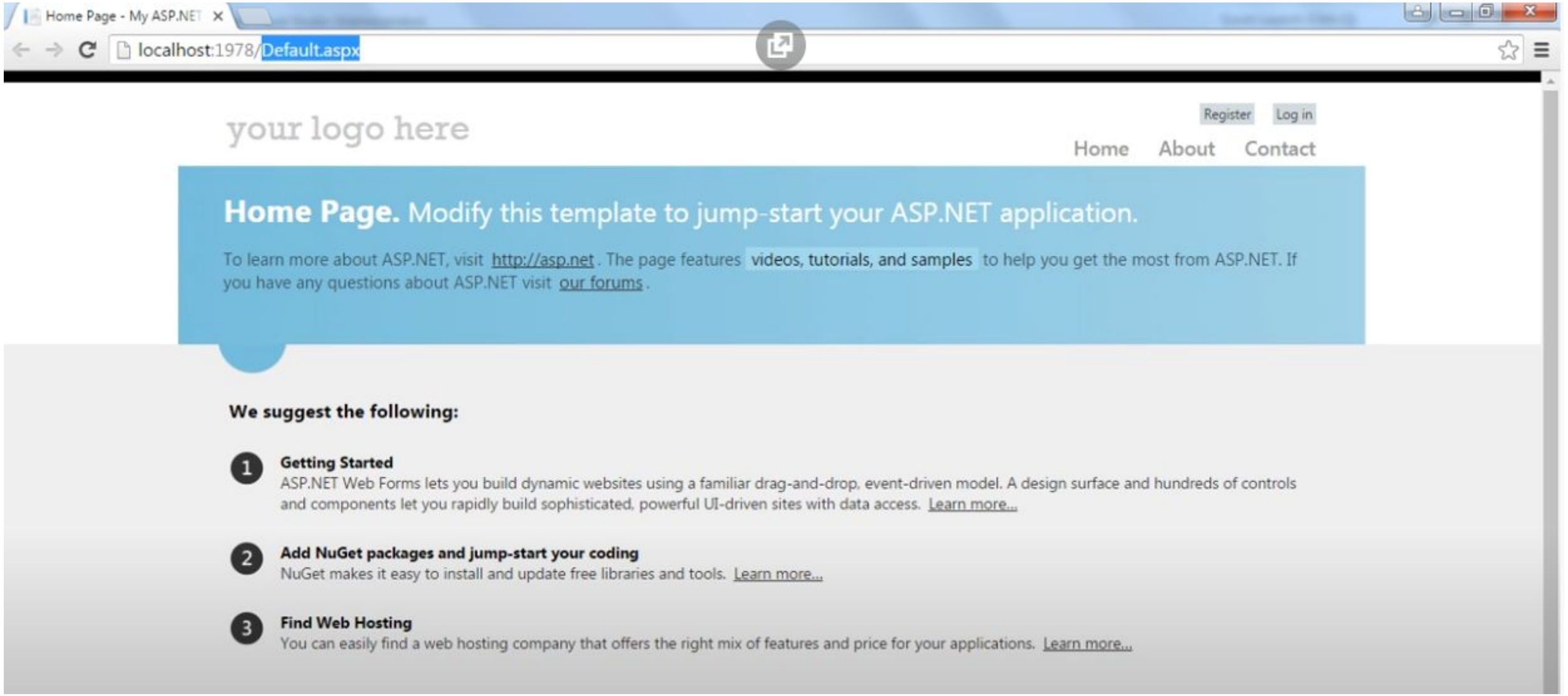
В настоящее время есть два подхода к созданию объектно ориентированных web-приложений:

- подходы, основанные на наборе специальных web-страниц (web-форм), связанных с описаниями классов, объекты которых будут создаваться и использоваться при их вызове (например, технология ASP.Net Web Forms; технология Java Server Pages);
- подходы, основанные на использовании наборов классов, соответствующих шаблону Model-View-Controller (MVC) (например, технологии на основе языка Java – Spring и технология компании Microsoft – ASP.Net MVC).

Объектный подход на основе форм

Подход на основе web-форм является дальнейшим развитием скриптовых серверных технологий. В данном подходе в HTML-документы добавляются специальные тэги, обрабатываемые на стороне сервера. Кроме этого, можно описывать и использовать собственные тэги в виде классов на универсальных языках программирования (Java, C#, Visual-Basic и т. п.), создавать на стороне сервера объектной модели web-приложения, аналогичные объектной модели локального приложения.


New ASP.NET Project - BasicWebApp
Select a template:
Empty
Web Forms
MVC
Web API
Single Page Application
Facebook
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.
Learn more
Add folders and core references for:
Web Forms
MVC
Web API
Change Authentication
Authentication: Individual User Accounts
Add unit tests
Test project name:
BasicWebApp.Tests
OK
Cancel

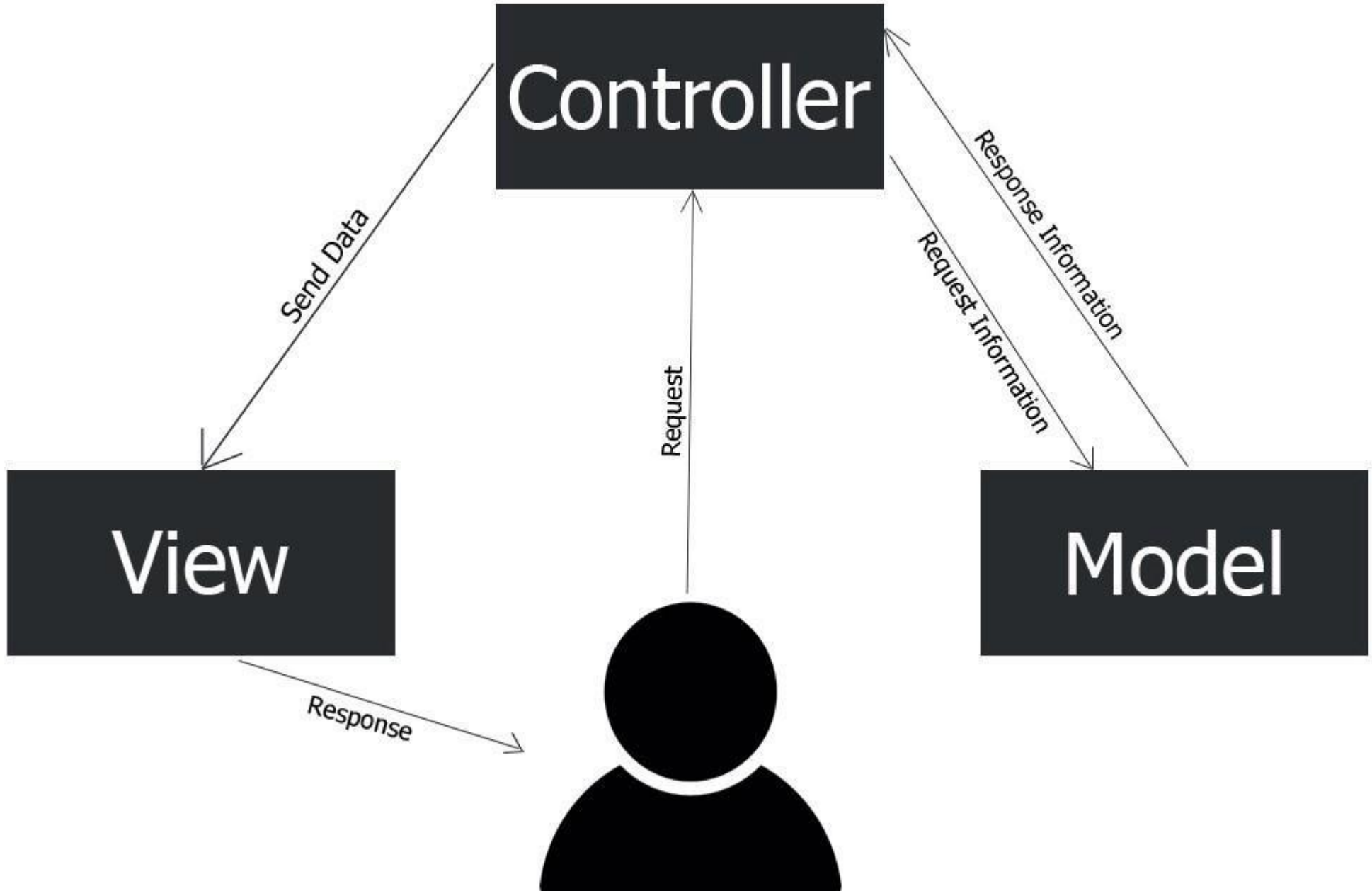

Home Page - My ASP.NET
localhost:1978/Default.aspx
your logo here
Register
Log in
Home
About
Contact
Home Page. Modify this template to jump-start your ASP.NET application.
To learn more about ASP.NET, visit http://asp.net. The page features videos, tutorials, and samples to help you get the most from ASP.NET. If you have any questions about ASP.NET visit our forums.
We suggest the following:
1
Getting Started
ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access. Learn more...
2
Add NuGet packages and jump-start your coding
NuGet makes it easy to install and update free libraries and tools. Learn more...
3
Find Web Hosting
You can easily find a web hosting company that offers the right mix of features and price for your applications. Learn more...

Подход на основе архитектурного шаблона MVC

В соответствии с архитектурным шаблоном MVC все приложение делится на три логических компонента:

- *Модель* (Model),
- *Представление* (View)
- *Контроллер* (Controller).

Model-View-Controller



Подход на основе архитектурного шаблона MVC

Модель (Model) – это набор классов, реализующих всю бизнес-логику web-приложения. Эти классы отвечают за обработку данных (сущностей), размещение их в БД, чтение из БД, а также за взаимодействие между самими объектами, составляющими такие данные.

Подход на основе архитектурного шаблона MVC

Представление (View) – набор классов и шаблонов, отвечающих за интерфейс взаимодействия с пользователями (User Interface, UI). Обычно они формируют HTML-страницы, показывающие пользователям данные из модели. На основе данных модели пользователям представляется возможность их просматривать и редактировать.

Подход на основе архитектурного шаблона MVC

Контроллер (Controller) – это связующее звено между первыми двумя компонентами. Классы данного компонента получают данные, содержащиеся в запросе к серверу (например, значения, полученные из отправленной формы) и передают их в *Модель* для обработки и сохранения. После этого *Контроллер* выбирает, каким способом показать их клиенту с помощью использования некоторого *Представления,* и передает ему данные для формирования HTML-ответа.

Подход на основе архитектурного шаблона MVC

Такое разделение web-приложения на части упрощает структуру приложения за счет более строго разделения его уровней. Логика пользовательского интерфейса располагается в представлении, логика ввода-вывода в контроллере, а бизнес-логика – в модели. Достигается полное отделение логики работы приложения от представления данных. Разработчик получает полный контроль над формируемым HTML- документом. Облегчается задача выполнения тестирования приложения.

Подход на основе архитектурного шаблона MVC

Примерами технологий разработки на основе MVC являются:

- Spring Framework (основанная на языке Java);
- технология ASP.Net MVC, входящая в состав набора технологий ASP.Net платформы .Net Framework;
- технология Ruby on Rails (Ruby – язык программирования, а Rails – фреймворк, использующий данный язык).