



ASP.NET MVC BASICS: VALIDATION

2017

VALIDATION BASICS

```
// ~/Models/Appointment.cs
public class Appointment
{
    public string ClientName { get; set; }

    [DataType(DataType.Date)]
    public DateTime Date { get; set; }

    public bool TermsAccepted { get; set; }
}
```

VALIDATION BASICS

```
<!-- ~/Views/Appointment/Appointment.cshtml -->
@model MvcApp.Models.Appointment
@{
    ViewBag.Title = "Make A Booking";
}
<h4>Book an Appointment</h4>
@using (Html.BeginForm())
{
    <p>Your name: @Html.EditorFor(m => m.ClientName)</p>
    <p>Appointment Date: @Html.EditorFor(m => m.Date)</p>
    <p>@Html.EditorFor(m => m.TermsAccepted) I accept the terms & conditions</p>
    <input type="submit" value="Make Booking" />
}
```

VALIDATION BASICS

```
// ~/Controllers/AppointmentController.cs
public class AppointmentController : Controller
{
    private IRepository<Appointment> _repository;

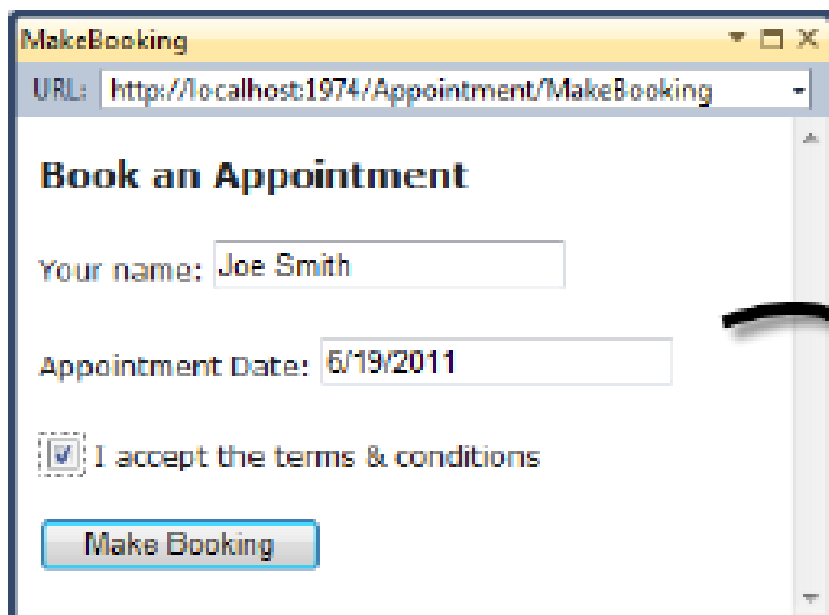
    public AppointmentController(IRepository<Appointment> repository) {
        _repository = repository;
    }

    public ActionResult MakeBooking() {
        return View(new Appointment { Date = DateTime.UtcNow });
    }

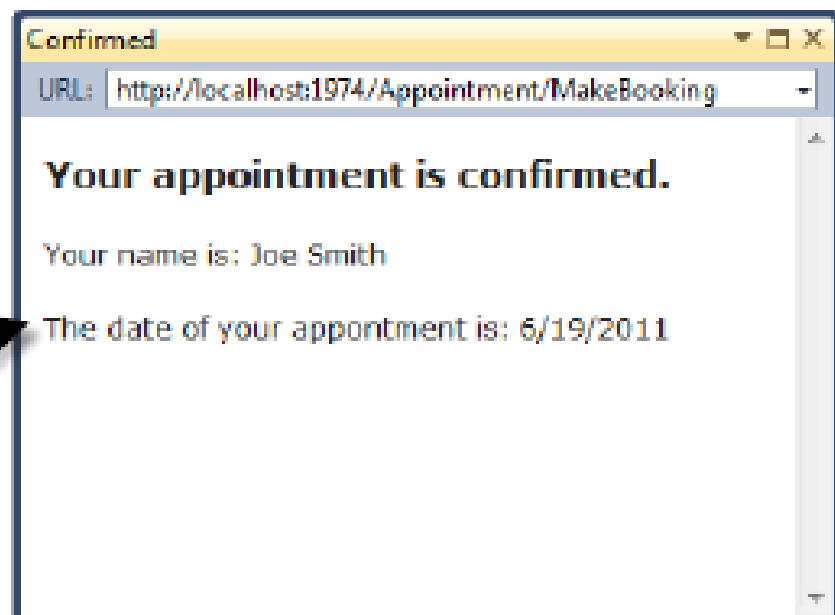
    [HttpPost]
    public ActionResult MakeBooking(Appointment appointment){
        _repository.SaveAppointment(appointment);
        return View("Completed", appt);
    }
}
```

VALIDATION BASICS

После передачи данных о встрече в хранилище контроллер визуализирует представление **Completed.cshtml**, которое обеспечивает отклик пользователю.



The screenshot shows a web browser window titled "MakeBooking". The address bar displays the URL "http://localhost:1974/Appointment/MakeBooking". The page content includes a heading "Book an Appointment", a text input field for "Your name:" containing "Joe Smith", a text input field for "Appointment Date:" containing "6/19/2011", a checkbox labeled "I accept the terms & conditions" which is checked, and a "Make Booking" button at the bottom.



The screenshot shows a web browser window titled "Confirmed". The address bar displays the URL "http://localhost:1974/Appointment/MakeBooking". The page content includes a heading "Your appointment is confirmed.", followed by the text "Your name is: Joe Smith" and "The date of your appointment is: 6/19/2011". A curved arrow points from the "MakeBooking" form to this "Confirmed" page, indicating the transition after a successful booking.

A wide-angle photograph of a busy London street, likely Regent Street, during the 'golden hour' of sunset. The sun is low in the sky, creating a strong, warm glow and long shadows across the pavement. A large, diverse crowd of pedestrians is walking in both directions. On the left, a grand white stone building with classical architectural features like columns and arched windows is visible. A Union Jack flag flies from a balcony. A 'superdry' store is located on the ground floor. A red traffic light is visible at the intersection. In the background, a building is under construction, covered in scaffolding. On the right, a Starbucks logo is visible on a building facade. A large, semi-transparent blue rectangle with the word 'VALIDATION' in white, bold, sans-serif capital letters is overlaid across the middle of the image.

VALIDATION

EXPLICIT VALIDATION

```
[HttpPost]
public ViewResult MakeBooking(Appointment appointment)
{
    if (string.IsNullOrEmpty(appointment.ClientName))
    {
        // Создает новую ошибку и привязывает ее к конкретному свойству модели
        ModelState.AddModelError("ClientName", "Please enter your name");
    }

    if (ModelState.IsValidField("Date") && DateTime.Now > appointment.Date)
    {
        ModelState.AddModelError("Date", "Please enter a date in the future");
    }

    ... // other code goes here
}
```

EXPLICIT VALIDATION

Проверка свойства `ClientName`:

```
if (string.IsNullOrEmpty(appointment.ClientName)) {  
    ModelState.AddModelError("ClientName", "Please enter your name");  
}
```

Пользователь должен указать значение для этого свойства, поэтому мы проверяем его с помощью `string.IsNullOrEmpty()`.

Если значение не получено, мы используем `ModelState.AddModelError()` для указания свойства, вызвавшего проблему ("`ClientName`"), и сообщения, которое будет отображаться.

EXPLICIT VALIDATION

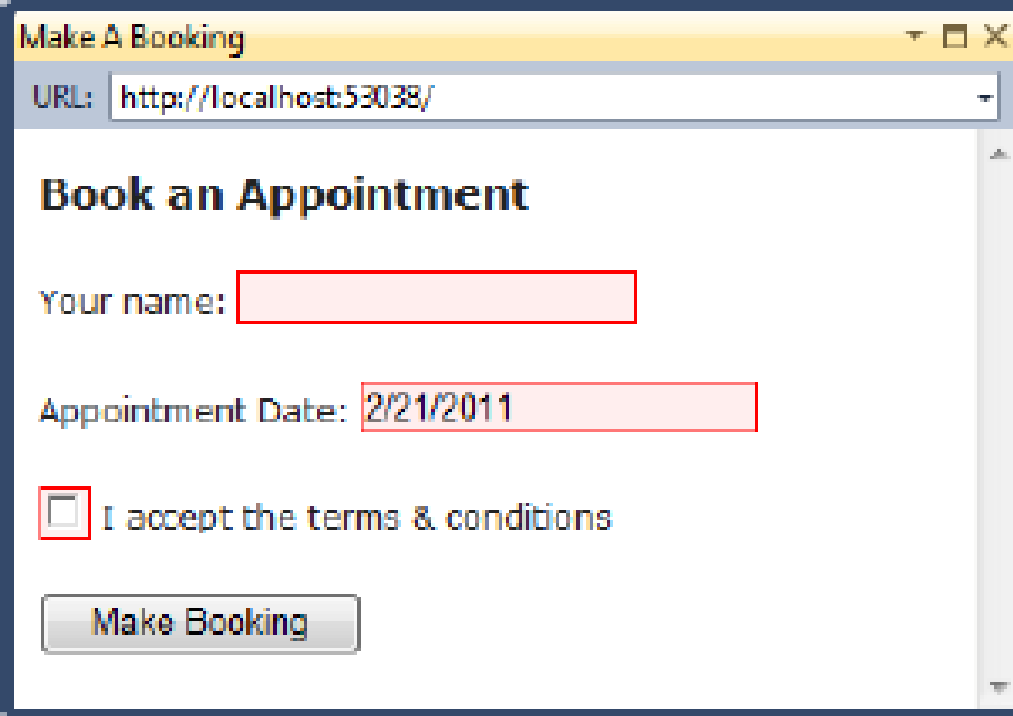
После проверки достоверности всех свойств в объекте модели мы читаем свойство `ModelState.IsValid`, чтобы выяснить, возникали ли ошибки.

Это свойство возвращает `false`, если вы регистрировали какие-то проблемы или если с проблемами столкнулся `model binder`:

```
if (ModelState.IsValid)
{
    repository.SaveAppointment(appointment);
    return View("Completed", appointment);
}
else
{
    return View();
}
```

EXPLICIT VALIDATION

Это обеспечивает установку красной рамки и розового фона для любого элемента, содержащего ошибочные данные:

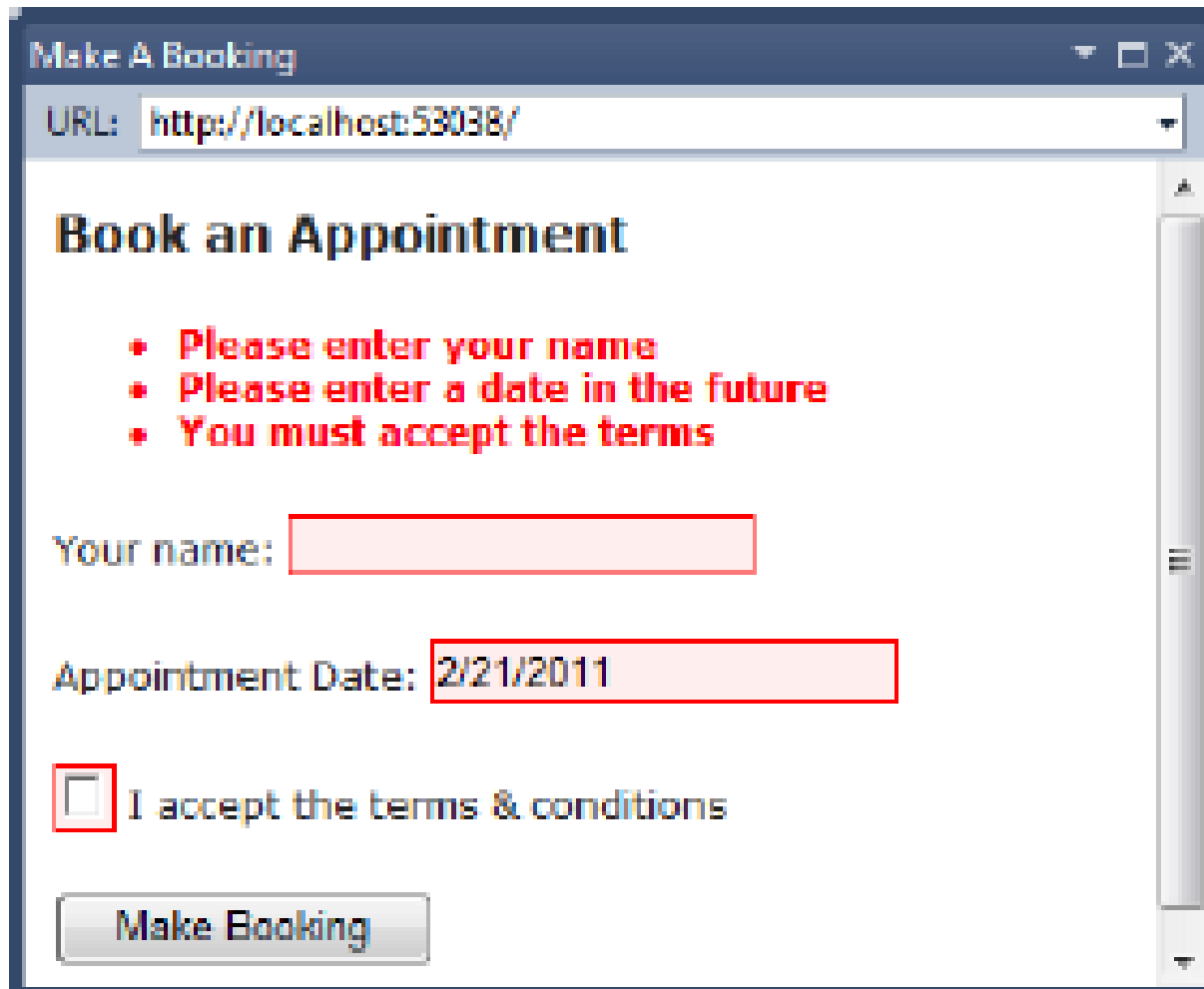


The screenshot shows a web browser window titled "Make A Booking" with the URL "http://localhost:53038/". The page content is titled "Book an Appointment". It contains three input fields: "Your name:", "Appointment Date:", and a checkbox labeled "I accept the terms & conditions". The "Your name:" and "Appointment Date:" fields are highlighted with a red border and a light pink background, indicating they contain invalid data. The checkbox is also highlighted with a red border. A "Make Booking" button is located at the bottom of the form.

VALIDATION MESSAGES

```
@model MvcApp.Models.Appointment
@{
    ViewBag.Title = "Make A Booking";
}
<h4>Book an Appointment</h4>
@using (Html.BeginForm())
{
    @Html.ValidationSummary()
    <p>Your name: @Html.EditorFor(m => m.ClientName)</p>
    <p>Appointment Date: @Html.EditorFor(m => m.Date)</p>
    <p>@Html.EditorFor(m => m.TermsAccepted) I accept the terms & conditions</p>
    <input type="submit" value="Make Booking" />
}
```

VALIDATION MESSAGES



The screenshot shows a web browser window titled "Make A Booking" with the URL "http://localhost:53038/". The page content is titled "Book an Appointment" and lists three validation requirements in red text:

- Please enter your name
- Please enter a date in the future
- You must accept the terms

Below the list, the form fields are as follows:

- "Your name:" followed by an empty text input field with a red border.
- "Appointment Date:" followed by a text input field containing "2/21/2011" with a red border.
- A checkbox (unchecked) with a red border, followed by the text "I accept the terms & conditions".

At the bottom of the form is a "Make Booking" button.

VALIDATION MESSAGES

```
<div class="validation-summary-errors" data-valmsg-summary="true">
  <ul>
    <li>Please enter your name</li>
    <li>Please enter a date in the future</li>
    <li>You must accept the terms</li>
  </ul>
</div>
```

Сообщения об ошибках выражаются в виде списка, содержащегося в элементе `<div>`, к которому применен стиль `class="validation-summary-errors"`. Этот стиль определен в файле `~/Content/Site.css` и при необходимости может быть изменен.

VALIDATION MESSAGES

Метод	Описание
@Html.ValidationSummary()	Генерирует область сводки с сообщениями об ошибках
@Html.ValidationSummary(bool)	Если параметр bool равен true , отображаются только сообщения об ошибках уровня модели.
@Html.ValidationSummary(string)	Отображает сообщение переданное в параметре string перед сводкой об ошибках проверки достоверности
@Html.ValidationSummary(bool, string)	Работает как два предыдущих метода вместе.

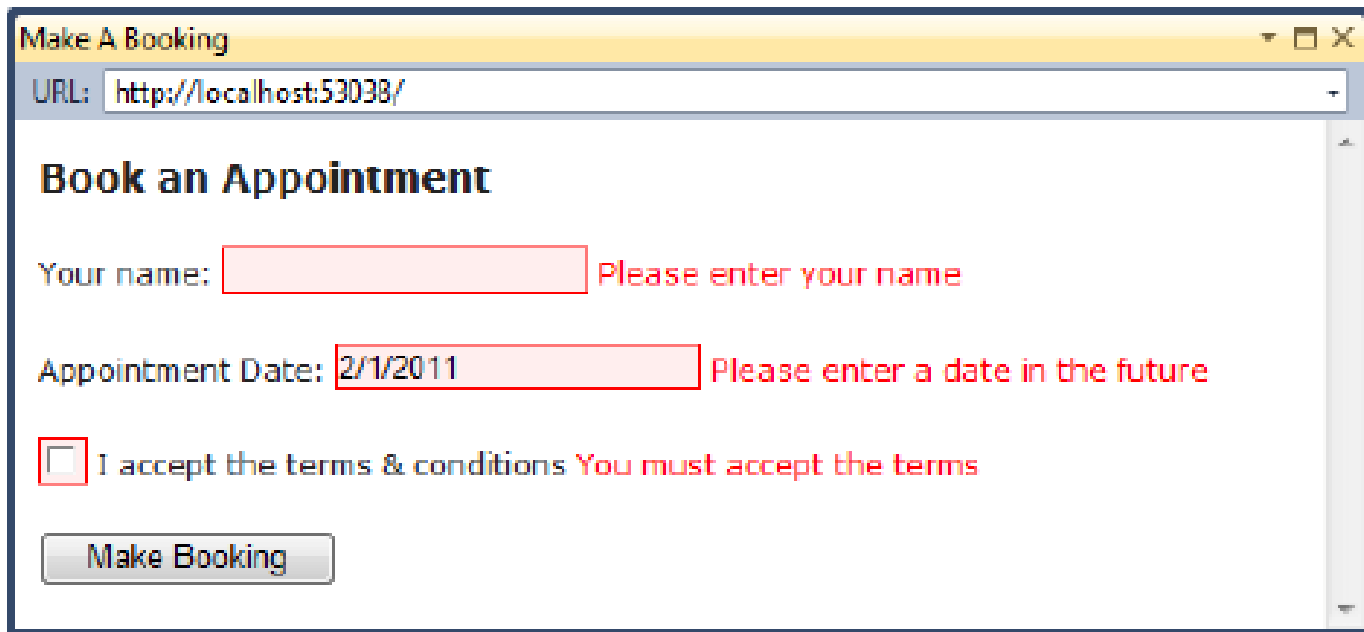
VALIDATION MESSAGES

```
@model MvcApp.Models.Appointment
@{ ViewBag.Title = "Make A Booking"; }

<h4>Book an Appointment</h4>
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <p>
        Your name: @Html.EditorFor(m => m.ClientName)
        @Html.ValidationMessageFor(m => m.ClientName)
    </p>
    <p>
        Appointment Date: @Html.EditorFor(m => m.Date)
        @Html.ValidationMessageFor(m => m.Date)
    </p>
    <p>
        @Html.EditorFor(m => m.TermsAccepted) I accept the terms & conditions
        @Html.ValidationMessageFor(m => m.TermsAccepted)
    </p>
    <input type="submit" value="Make Booking" />
}
```

VALIDATION MESSAGES

Вспомогательный метод `@Html.ValidationMessageFor()` отображает сообщения об ошибках проверки достоверности для одиночного свойства модели:



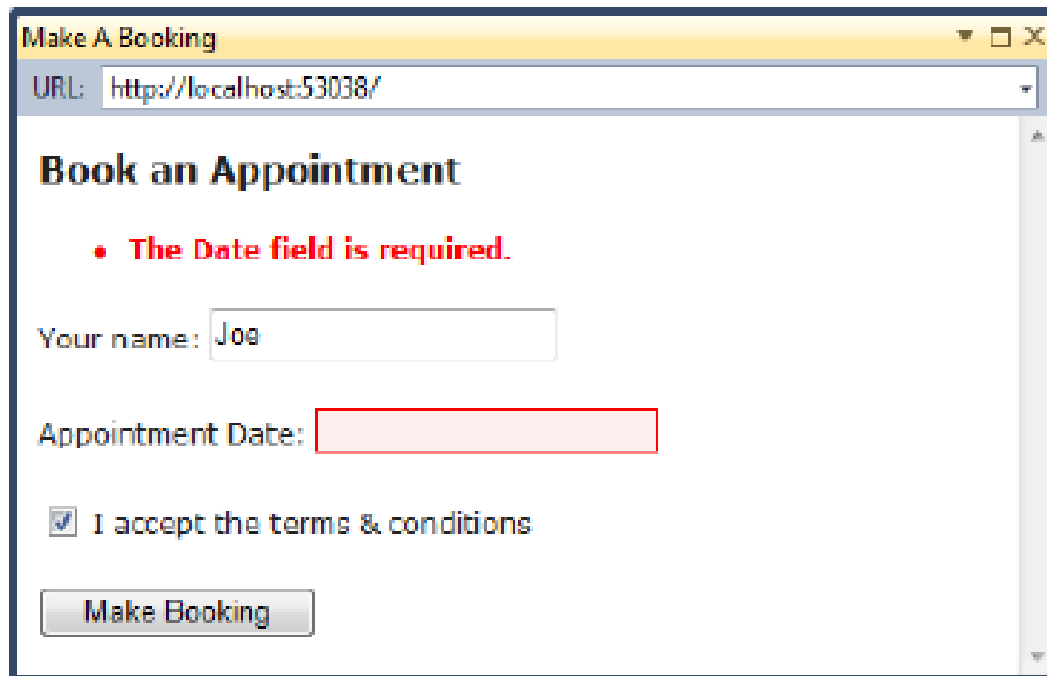
The screenshot shows a web browser window titled "Make A Booking" with the URL "http://localhost:53038/". The page content is titled "Book an Appointment". It contains three input fields, each with a red border and a red error message to its right:

- "Your name:" followed by an empty text box and the message "Please enter your name".
- "Appointment Date:" followed by a text box containing "2/1/2011" and the message "Please enter a date in the future".
- A checkbox that is unchecked, followed by the text "I accept the terms & conditions" and the message "You must accept the terms".

At the bottom of the form is a button labeled "Make Booking".

ADDITIONAL VALIDATION

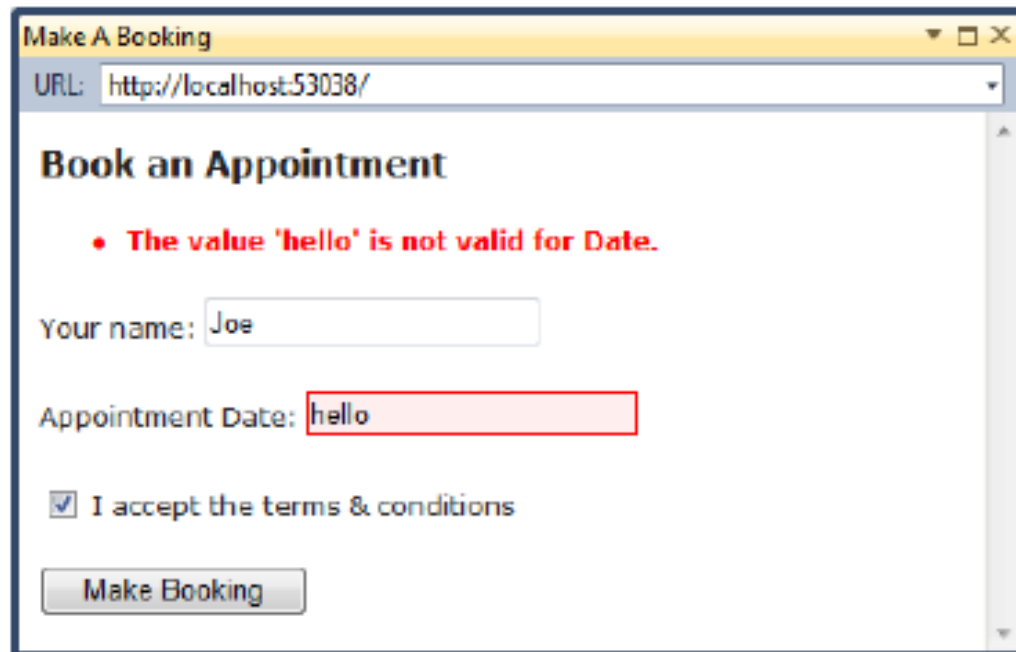
- **Model binder** по умолчанию выполняет валидацию значений как часть процесса биндинга.
- Вот что произойдет, если очистить поле **Appointment Date** и отправить форму:



The screenshot shows a web browser window with the title "Make A Booking". The address bar displays "URL: http://localhost:53038/". The main content area has the heading "Book an Appointment". Below the heading, there is a red error message: "• The Date field is required." The form contains two input fields: "Your name:" with the value "Joe" and "Appointment Date:" which is empty and highlighted with a red border. Below the date field, there is a checkbox labeled "I accept the terms & conditions" which is checked. At the bottom of the form is a button labeled "Make Booking".

ADDITIONAL VALIDATION

- Если значение не было задано, отображается сообщение показанное выше.
- Если задано, но не может быть успешно преобразовано к типу свойства модели, отображается другое сообщение:



The screenshot shows a web browser window titled "Make A Booking". The address bar displays "URL: http://localhost:53038/". The main content area has the heading "Book an Appointment". Below the heading, there is a red error message: "• The value 'hello' is not valid for Date." The form contains two input fields: "Your name:" with the value "Joe" and "Appointment Date:" with the value "hello". The "Appointment Date" field is highlighted with a red border. Below the input fields, there is a checked checkbox labeled "I accept the terms & conditions" and a "Make Booking" button.


```
[HttpPost]
public ViewResult MakeBooking(Appointment appt)
{
    if (ModelState.IsValid)
    {
        repository.SaveAppointment(appt);
        return View("Completed", appt);
    }
    else
    {
        return View();
    }
}
```

- Любые ошибки, зарегистрированные в **model binder**'е, будут доступны в **ModelState** внутри **action**.

METADATA VALIDATION

```
public class Appointment
{
    [Required]
    public string ClientName { get; set; }

    [DataType(DataType.Date)]
    [Required(ErrorMessage = "Please enter a date")]
    public DateTime Date { get; set; }

    [Range(typeof(bool), "true", "true",
        ErrorMessage = "You must accept the terms")]
    public bool TermsAccepted { get; set; }
}
```

METADATA VALIDATION

Атрибут	Описание
[Compare("PropertyName")]	Два свойства должны иметь одинаковые значения.
[Range(10, 20)]	Значение не должно выходить за пределы, указанные минимальным и максимальным значениями.
[RegularExpression("template")]	Строковое значение должно соответствовать указанному шаблону регулярного выражения.
[Required]	Значение не должно быть пустым.
[StringLength(10)] [StringLength(10, MinimumLength = 2)]	Строковое значение не должно быть длиннее или короче указанной максимальной длины.

A photograph of a busy London street, likely Regent Street, during the "golden hour" of sunset. The sun is low in the sky, creating a strong, warm glow and long shadows on the pavement. On the left, a grand white stone building with classical architectural features like columns and arched windows is visible. A Union Jack flag flies from a balcony. A red traffic light is in the foreground. A "Superdry" store is visible on the ground floor. In the background, a building is under construction, covered in scaffolding. On the right, a Starbucks logo is visible on a building facade. A large, semi-transparent blue banner with white text is superimposed across the middle of the image.

CLIENT SIDE VALIDATION

CLIENT SIDE VALIDATION SETTINGS

Валидация на стороне клиента управляется двумя параметрами в файле *Web.config*, как показано ниже:

```
<configuration>
  <appSettings>
    <add key="ClientValidationEnabled" value="true"/>
    <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
  </appSettings>
  ...
</configuration>
```

Чтобы проверка на стороне клиента работала, оба параметра иметь значение *true*. При создании нового проекта MVC среда Visual Studio добавляет эти параметры и устанавливает их в *true* по умолчанию.

CLIENT SIDE VALIDATION SETTINGS

В качестве альтернативы управлять этими параметрами можно программно в *Global.asax*:

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();

    HtmlHelper.ClientValidationEnabled = true;
    HtmlHelper.UnobtrusiveJavaScriptEnabled = true;

    RegisterGlobalFilters(GlobalFilters.Filters);
    RegisterRoutes(RouteTable.Routes);
}
```

CLIENT SIDE VALIDATION SETTINGS

```
// ~/App_Start/BundleConfig.cs
public static void RegisterBundles(BundleCollection bundles)
{
    bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
        "~/Scripts/jquery-{version}.js"));

    bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
        "~/Scripts/jquery.validate*"));
}

// ~/Views/Shared/Layout.cshtml
...
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/jqueryval")
...
```

CLIENT SIDE VALIDATION

```
public class Appointment
{
    [Required]
    [StringLength(10, MinimumLength = 3)]
    public string ClientName { get; set; }

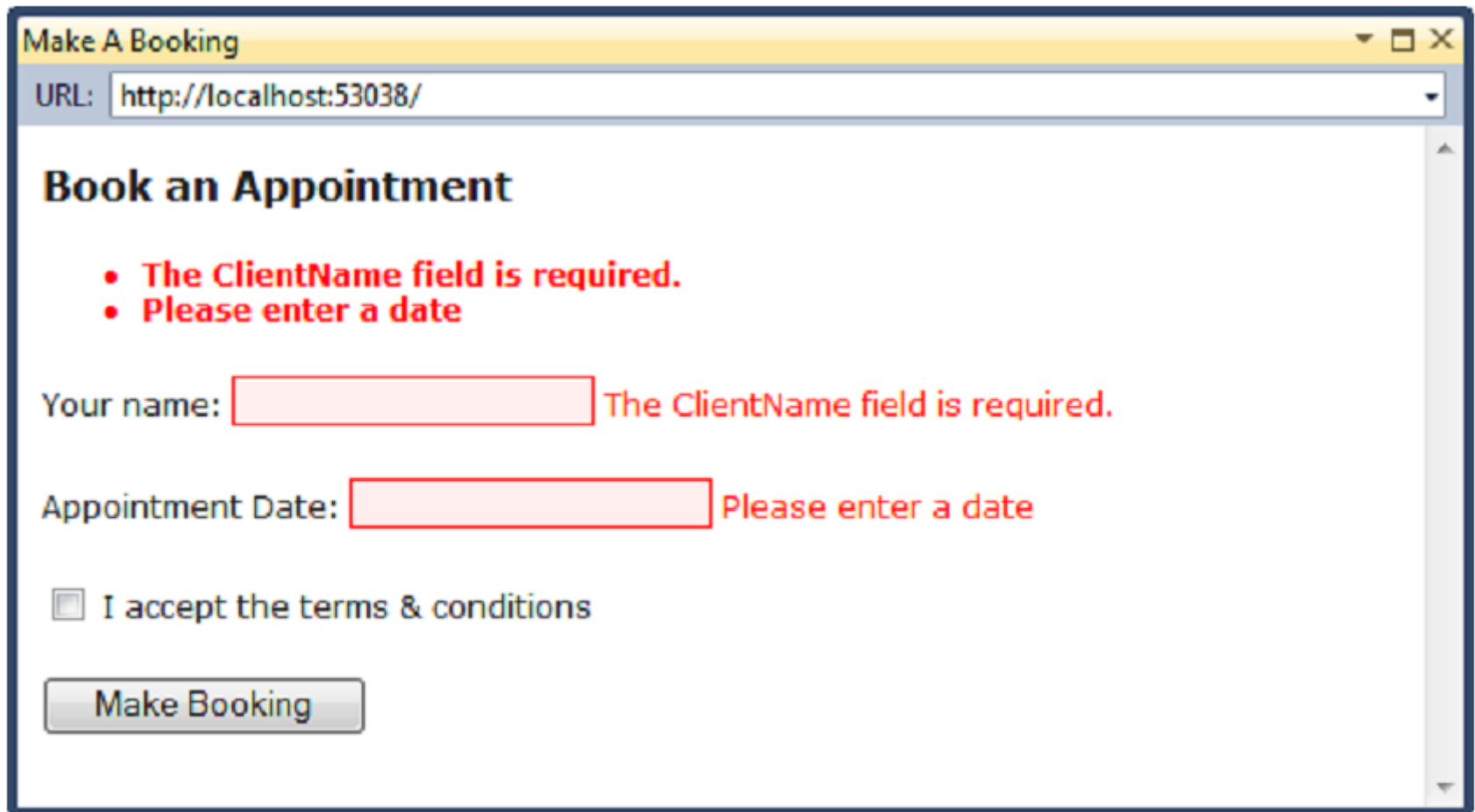
    [DataType(DataType.Date)]
    [Required(ErrorMessage = "Please enter a date")]
    public DateTime Date { get; set; }

    public bool TermsAccepted { get; set; }
}
```

CLIENT SIDE VALIDATION

```
@using (Html.BeginForm()) {  
    @Html.ValidationSummary()  
    <p>  
        Your name: @Html.EditorFor(m => m.ClientName)  
        @Html.ValidationMessageFor(m => m.ClientName)  
    </p>  
    <p>  
        Appointment Date: @Html.EditorFor(m => m.Date)  
        @Html.ValidationMessageFor(m => m.Date)  
    </p>  
    <p>  
        @Html.EditorFor(m => m.TermsAccepted) I accept the terms & conditions  
        @Html.ValidationMessageFor(m => m.TermsAccepted)  
    </p>  
    <input type="submit" value="Make Booking" />  
}
```

CLIENT SIDE VALIDATION



The screenshot shows a web browser window with the title "Make A Booking". The address bar displays "URL: http://localhost:53038/". The main content area is titled "Book an Appointment" and contains two red error messages:

- The ClientName field is required.
- Please enter a date

Below the errors, there are two input fields:

Your name: The ClientName field is required.

Appointment Date: Please enter a date

Below the input fields, there is a checkbox labeled "I accept the terms & conditions".

At the bottom, there is a button labeled "Make Booking".

CLIENT SIDE VALIDATION

The screenshot shows a web browser window with the title "Make A Booking". The address bar displays "URL: http://localhost:53038/". The main content area is titled "Book an Appointment" and contains two red bullet points indicating validation errors:

- The ClientName field is required.
- Please enter a date

Below the errors, the form fields are shown with red borders and error messages:

Your name: The field ClientName must be a string with a minimum length of 3 and a maximum length of 10.

Appointment Date: Please enter a date

☐ I accept the terms & conditions

CLIENT SIDE VALIDATION

The screenshot shows a web browser window with the title 'Make A Booking'. The address bar displays 'URL: http://localhost:53038/'. The main content area is titled 'Book an Appointment' and contains two red error messages:

- The ClientName field is required.
- Please enter a date

Below the errors, there is a form with the following elements:

- A label 'Your name:' followed by a text input field containing the text 'Joe'.
- A label 'Appointment Date:' followed by an empty date input field with a red border, and the text 'Please enter a date' to its right.
- A checkbox labeled 'I accept the terms & conditions'.
- A 'Make Booking' button.

CLIENT SIDE VALIDATION

The screenshot shows a web browser window with the title "Make A Booking". The address bar displays the URL "http://localhost:53038/". The main heading of the page is "Book an Appointment". Below the heading, there is a red error message: "• Please enter a date". The form contains two input fields: "Your name:" with the value "Joe" and "Appointment Date:" which is empty and highlighted with a red border. To the right of the empty date field is another red error message: "Please enter a date". Below the date field is a checkbox labeled "I accept the terms & conditions", which is currently unchecked. At the bottom of the form is a blue button labeled "Make Booking".

Make A Booking

URL: http://localhost:53038/

Book an Appointment

- Please enter a date

Your name:

Appointment Date: Please enter a date

☐ I accept the terms & conditions

ANY QUESTIONS?



**THANKS FOR YOUR
ATTENTION!**