**Walidacja danych**

Zp\_9\_20

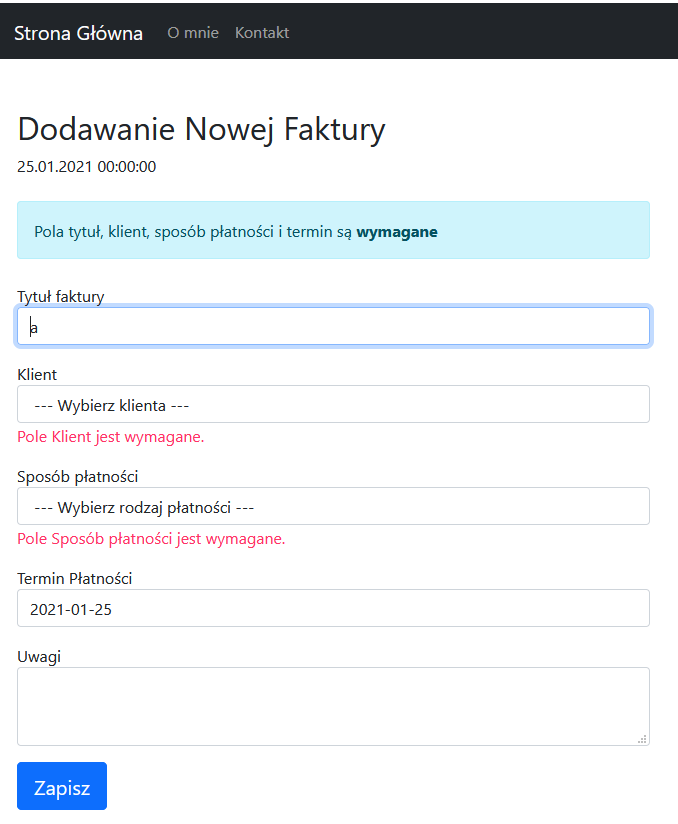
W kontrolerze walidujemy każdą akcję typu post

|  |
| --- |
| public ActionResult Invoice(Invoice invoice)  {  var userId = User.Identity.GetUserId();  invoice.UserId = userId;  if (!ModelState.IsValid)  {  var vm = PrepareInvoiceVm(invoice, userId);  return View("Invoice", vm);  }  if (invoice.Id == 0)  \_invoiceRepository.Add(invoice);  else  \_invoiceRepository.Update(invoice);  return RedirectToAction("Index");  } |

Jeżeli dane są ok, to pomijamy tę sekcję w przeciwnym razie przekierowujemy z powrotem do tego samego widoku. Ale powinniśmy przekazać do tego widoku te dane, które zostały wpisane przez użytkownika. Jak walidować wiemy już na podstawie samych modeli.  
Działa to tak, że w przypadku nie wpisania kompletu danych strona się w tle przeładowuje, nie dając przy tym żadnego komunikatu.

|  |
| --- |
| <div class="form-group mb-3">  @Html.LabelFor(x => x.Invoice.Title)  @Html.TextBoxFor(x => x.Invoice.Title,  new { @class = "form-control", autofocus = "autofocus" })  @Html.ValidationMessageFor(x => x.Invoice.Title)  </div> |

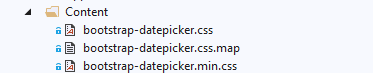
Ctrl + k + d



Aby zmienić ten kolor uruchamiamy narzędzia developerskie w przeglądarce F12.  
Przechodzimy na zakładkę inspektor obiektów, klikamy na interesujący nas element na stronie.  
W okienku obok zostaną wyświetlone style powiązane z tym elementem, między innymi wartość oraz nazwa pliku css – w tym przypadku site.css



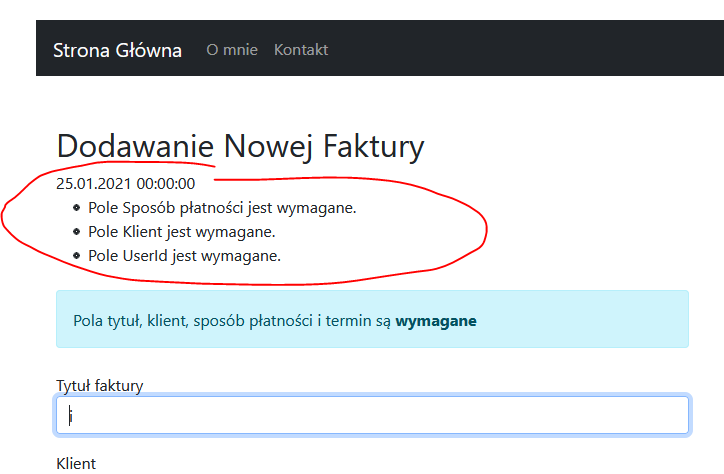
Ten plik odnajdziemy w katalogu content naszego projektu.



|  |
| --- |
| .field-validation-error {  color: #ff3366;  } |

Aby wyświetlić wszystkie błędy walidacji w jednym miejscu używamy helpera: Html.ValidationSummary

|  |
| --- |
| <div class="row">  <div class="col-lg-6">  @using (Html.BeginForm("Invoice", "Home"))  {  @Html.ValidationSummary()  if (Model.Invoice.Id == 0)  {  <p class="alert alert-info mb-4 mt-4">Pola tytuł… i termin są   <strong>wymagane</strong> </p>  }  @Html.HiddenFor(x => x.Invoice.Id) |



Aby spersonalizować treść informacji o błędzie

|  |
| --- |
| namespace InvoiceManager.Models.Domains  {  public class Invoice  {  public Invoice()  {  InvoicePositions = new Collection<InvoicePossition>();  }  public int Id { get; set; }  [Required]  [Display(Name = "Tytuł faktury")] |

Możemy dodać parameter do atrybutu [Required]

|  |
| --- |
| public int Id { get; set; }  [Required(ErrorMessage ="Pole Tytuł jest wymagane !!!")]  [Display(Name = "Tytuł faktury")] |