

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 5

ASP.NET MVC

2024-11-05

Liczba punktów do zdobycia: **10/42**

Zestaw ważny do: 2024-11-19

*Uwaga! Dwa pierwsze zadania dotyczą podsystemu MVC w obu dostępnych wersjach, .NET Framework i .NET Core. Podczas rozwiązywania zwrócić uwagę na podobieństwa i różnice. Po-
cząwszy od zadania trzeciego, zadania rozwiązywać w jednej, wybranej przez siebie wersji MVC.*

1. **(1p)** Zadanie z paskiem zadań (2.4) powtórzyć w MVC ASP.NET Framework. Do utworzenia pól tekstowych dla poszczególnych zadań oraz do odebrania na serwerze danych od użytkownika wykorzystać wiązanie (binding) do tablicy (a nie do 10 pól o różnych nazwach). Zwrócić uwagę na prawidłową walidację danych.
2. **(1p)** Zadanie z paskiem zadań (2.4) powtórzyć w MVC ASP.NET.Core. Do utworzenia pól tekstowych dla poszczególnych zadań oraz do odebrania na serwerze danych od użytkownika wykorzystać wiązanie do tablicy. Zwrócić uwagę na prawidłową walidację danych.
3. **(1p)** Zadanie z przesyłaniem plików/danych w obie strony (3.7) powtórzyć w ASP.NET MVC. Zarówno do odebrania strumienia danych w kontrolerze jak i do zwrócenia zawartości do użytkownika użyć elementów infrastruktury MVC.
4. **(1p)** Pokazać przykłady użycia kilku typowych rozszerzeń, np.:

- `Html.ActionLink`
- `Html.BeginForm`
- `Html.CheckBox[For]`
- `Html.DropDownList[For]`
- `Html.Password[For]`
- `Html.RadioButton[For]`
- `Html.TextBox[For]`
- `Html.TextArea[For]`

Do czego służy `Html.Raw`?

5. **(6p) (dodawanie/edycja)** Przygotować aplikację która pozwala przeglądać, dodawać, edytować i usuwać dane z wybranej kategorii (lista studentów, zbiór książek, itp.). Formalnie:
 - dane powinny pochodzić z bazy danych (SqlServer/Postgresql) - tu zwrócić uwagę na przedstawiony na wykładzie sposób inicjowania kontekstu do danych

- w .NET Framework na przykład przez kontroler bazowy udostępniający obiekt połączenia
- w .NET Core przez rejestrację metody tworzącej obiekt połączenia lub repozytorium (`AddScoped<...>`)
- dane powinny być pokazywane w postaci tabelki - tworzyć ją w widoku, na podstawie danych
- każdy rekord w tabeli powinien, oprócz zawartości poszczególnych atrybutów, pozwalać użyć odnośnika *Edycja* i *Usunięcie*
 - użycie odnośnika *Edycja* powinno przekierowywać żądanie do osobnego widoku w którym odbywa się edycja z możliwością zapisania lub anulowania
 - użycie odnośnika *Usunięcie* powinno przekierowywać żądanie do osobnego widoku w którym jest zapytanie o potwierdzenie usunięcia
- pod/nad tabelą powinien być odnośnik *Dodawanie*, przekierowujący do osobnego widoku pozwalającego na dodanie nowego elementu, z możliwością zapisania lub anulowania

Do tworzenia linków w kodzie użyć metody rozszerzającej, dostępnej w kodzie kontrolera przez **this.Url.Action**, pozwalającej przekazać parametr do kontrolera/akcji/identyfikatora elementu który podlega edycji (adresy mają postać np. `/student/edycja/123` lub `/student/usuwanie/123`).

Zagadnienie dodatkowe (niepunktowane): prosty sposób prezentacji tabeli renderowanej na podstawie danych pomija dwa ważne, typowe elementy - sortowanie i stronicowanie. Jak w prezentowanym podejściu można rozwiązać oba? Jakiego rodzaju mechanizmu użyć żeby zapamiętać wybory użytkownika (sortuję "po nazwisku", oglądam stronę numer 17)?

Zagadnienie dodatkowe (niepunktowane): jak zabezpieczyć taki sposób przekazywania parametrów przed możliwością zbyt łatwej modyfikacji adresu w pasku adresowym przez użytkownika? Czy i kiedy brak ochrony parametru może być niepożądany? Jakiego rodzaju zagrożenie wnosi?

Wiktor Zychla