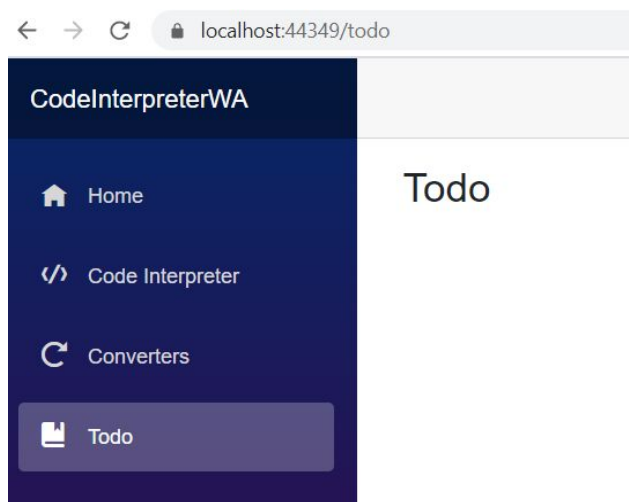
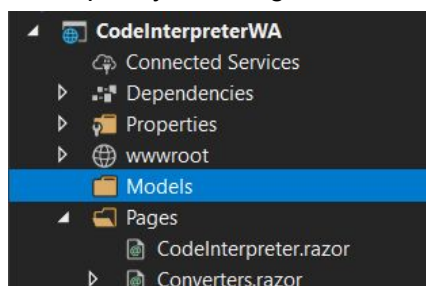


1. Proszę dodać nowy komponent o nazwie **Todo**. Komponent ten będzie pozwalał na tworzenie oraz zarządzanie listą zadań do zrobienia.
2. Następnie proszę zdefiniować routing dla nowego komponentu. **@page "/todo"**.
3. Podobnie jak w poprzednich krokach proszę dodać nowy komponent *Todo* do panelu nawigacji. Korzystając z dostępnych darmowych ikon na stronie <https://useiconic.com/open> proszę wybrać jedną i przypisać ją do nowo utworzonej nawigacji.
4. Proszę skompilować i uruchomić aplikację. W moim przypadku wygląda to tak:



5. Kolejnym krokiem jest dodanie folderu o nazwie **Models**, w tym miejscu jak sama nazwa wskazuje będziemy przechowywali różnego rodzaju modele w naszej aplikacji. W kroku **35** utworzyliśmy model dla przeliczników, ale dla uproszczenia umieściliśmy go bezpośrednio w komponencie. Nie jest to dobra praktyka dlatego od teraz nasze modele będziemy przetrzymywali w folderze *Models*.



6. Do folderu *Models* proszę dodać nową klasę o nazwie **TodoItemModel** która będzie reprezentować pojedynczy element z naszej listy zadań do zrobienia.

```
namespace CodeInterpreterWA.Models
{
    0 references
    public class TodoItemModel
    {
        0 references | 0 exceptions
        public string Title { get; set; }
        0 references | 0 exceptions
        public bool IsDone { get; set; }
    }
}
```

7. Następnym krokiem jest wykorzystanie pliku **_Imports.razor**. Plik ten pozwala na implementację dyrektyw zapisanych w tym pliku do wszystkich komponentów. Proszę dodać dyrektywę **@using** wskazującą nowy folder z modelami. Od tego momentu każdy z komponentów powinien mieć dostęp do wszystkich modeli z tego folderu.

```
@using CodeInterpreterWA.Models
```

8. Teraz proszę powrócić do komponentu *Todo* i w sekcji **@code** proszę dodać listę typu *TodoItemModel* o nazwie **todos**. Do listy proszę dodać dwa domyślnie zadania

```
private IList<TodoItemModel> todos = new List<TodoItemModel>
{
    new TodoItemModel
    {
        IsDone = false,
        Title = "Nauczyć się Blazor",
    },
    new TodoItemModel
    {
        IsDone = true,
        Title = "Napisać przelicznik $ na PLN",
    },
};
```

9. Kolejnym krokiem jest wyświetlenie zadań do zrobienia. W tym celu proszę dodać następujący kod do komponentu *Todo*.

```
<h3>Todo</h3>

<ul class="list-group">
  @foreach (var todo in todos)
  {
    <li class="list-group-item">@todo.Title</li>
  }
</ul>
```

10. Proszę sprawdzić czy aplikacja działa poprawnie.

Todo

Nauczyć się Blazor

Napisać przelicznik \$ na PLN

11. Musimy umożliwić użytkownikowi dodawanie nowych zadań. W tym celu proszę dodać przycisk oraz input. Akcja kliknięcia przycisku powinna uruchomić metodę **Add** którą dodasz w następnym kroku. Aby pobrać wartość wpisaną przez użytkownika należy powiązać input ze zmienną o nazwie **todo**. Zmienną utworzysz w następnym kroku.

```
<h3>Todo</h3>

<div class="col-md-4 mb-3">
  <input class="form-control" placeholder="Todo item" @bind="todo" />
  <button class="btn btn-primary mt-1" @onclick="Add">Add</button>
</div>
```

12. W tym kroku proszę utworzyć prywatną zmienną typu string o nazwie **todo**, oraz metodę o nazwie **Add**, której zadaniem jest dodanie nowego zadania do listy oraz wyczyszczenie zawartości zmiennej **todo**.

```
private string todo;  
  
private void Add()  
{  
    if (!string.IsNullOrEmpty(todo))  
    {  
        todos.Add(new TodoItemModel { Title = todo });  
        // clear todoInput  
        todo = string.Empty;  
    }  
}
```

13. Teraz aby umożliwić użytkownikowi zaznaczenie które zadania zostały już zrobione proszę zmienić kod wyświetlający zadania w następujący sposób:

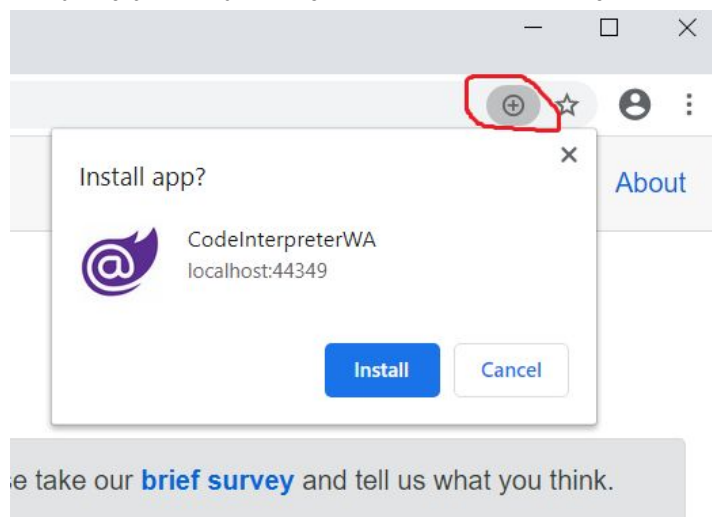
```
<ul class="list-group">  
    @foreach (var todo in todos)  
    {  
        <li class="list-group-item">  
            @todo.Title  
            <div class="float-right custom-control custom-checkbox">  
                <input type="checkbox" class="form-check-input" id="doneCheckBox" @bind="todo.IsDone">  
                <label class="form-check-label" for="doneCheckBox">Done</label>  
            </div>  
        </li>  
    }  
</ul>
```

14. Proszę skompilować i uruchomić aplikację oraz sprawdzić czy działa poprawnie.

Todo

Todo item	
Add	
Nauczyć się Blazor	<input type="checkbox"/> Done
Napisać przelicznik \$ na PLN	<input checked="" type="checkbox"/> Done
Dodane zadanie.	<input type="checkbox"/> Done

15. Ponieważ podczas tworzenia projektu zaznaczyliśmy opcję PWA teraz możemy zainstalować naszą aplikację i korzystać z niej jak z natywnej aplikacji systemu. Proszę zainstalować naszą aplikację korzystając z przycisku jak na obrazku poniżej



16. Proszę uruchomić aplikację i sprawdzić czy działa poprawnie.

