

TRY

NETMUG

The background is a dark blue gradient with a subtle checkerboard pattern. It features several pixel art clouds in various shades of blue and white, and a few small white stars scattered across the top half.

Level 2

# Getting User Input

Level 2 – Section 1

# Getting User Input

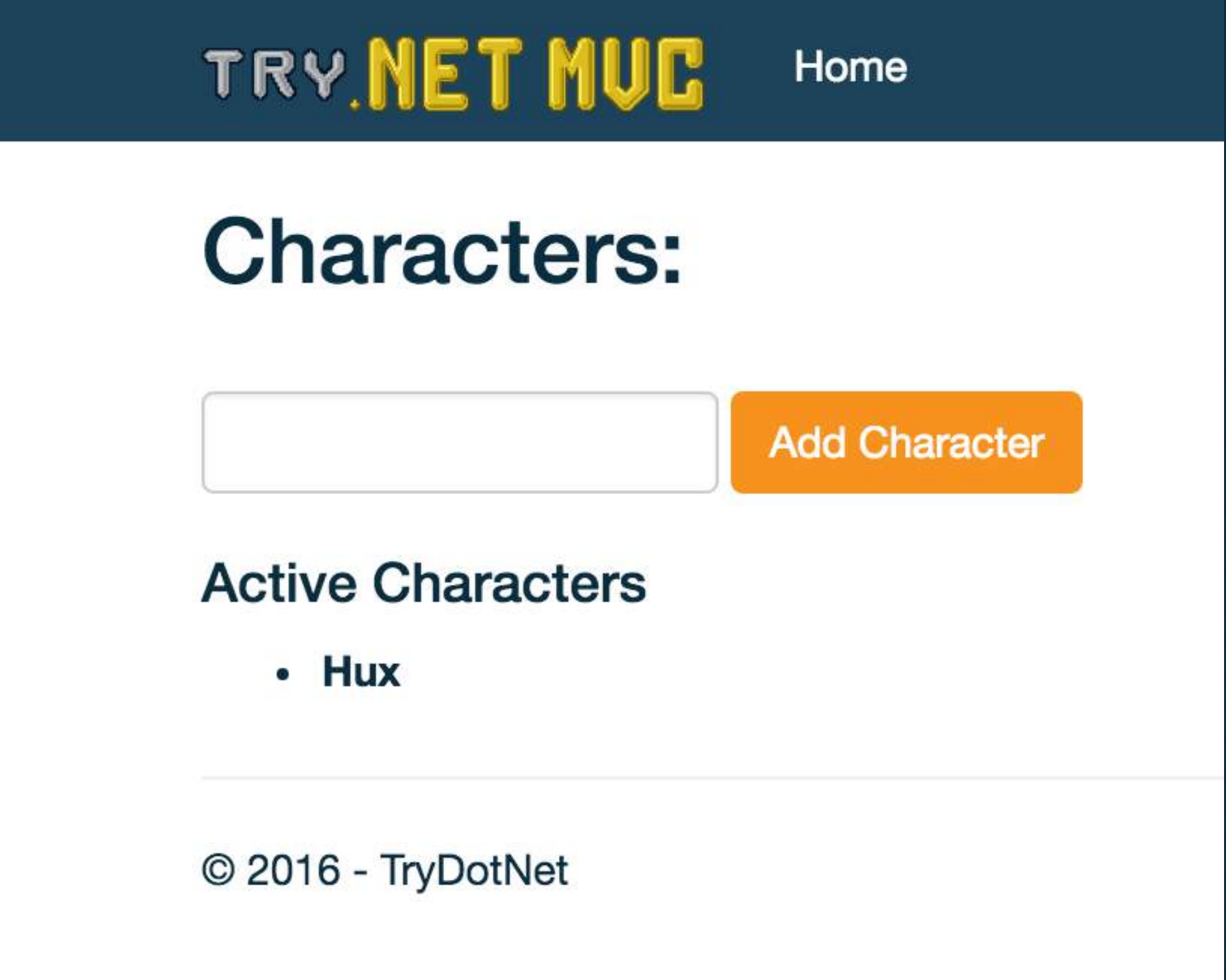
Application Needs Input

# The Problem...

In the last level, we set up our view to show our names dynamically, but then we hard-coded that name in the controller. Instead, we need to use user input.

## Steps to the Solution

1. Create form in view
2. Add Create() method to controller
3. Move existing logic out of Index() into Create()



The screenshot shows the Try.NET MVC application interface. At the top, there is a dark blue header with the text "TRY.NET MVC" in yellow and "Home" in white. Below the header, the main content area is white. It features a section titled "Characters:" in dark blue. Underneath this title, there is a text input field and an orange button labeled "Add Character". Below the input field, there is a section titled "Active Characters" in dark blue, which contains a single bullet point: "• Hux". At the bottom of the page, there is a footer with the text "© 2016 - TryDotNet".





# Create Our New Form

./Views/Home/Index.cshtml

CSHTML

```
@model CharacterSheetApp.Models.Character
```

```
<h2>Characters:</h2>
```

```
<form>
```

```
  <div>
```

```
    <input name="CharacterName" />
```

```
    <input type="submit" value="Add Character" />
```

```
  </div>
```

```
</form>
```

*We've added some HTML to our view for a simple form.*



TRY.NET MVC

Home

## Characters:

Add Character

### Active Characters

- Hux

© 2016 - TryDotNet

# Inputs Are Matched to Parameters by Name

./Views/Home/Index.cshtml

CSHTML

```
@model CharacterSheetApp.Models.Character
```

```
<h2>Characters:</h2>
```

```
<form>
```

```
<div>
```

```
<input name="CharacterName" />
```

```
<input type="submit" value="Add Character" />
```

```
</div>
```

```
</form>
```

*This input's name, CharacterName, will match up with our action's parameter, so these will automatically map up.*

./Controllers/HomeController.cs

CS

```
public IActionResult Create(string characterName)
{
    return View();
}
```



# Creating an Action With an Input Parameter

./Controllers/HomeController.cs

CS

```
public IActionResult Index()
{
    var model = new CharSheetApp.Models.Character();
    model.Name = "Hux";

    return View(model);
}

public IActionResult Create(string characterName)
{
    return View();
}
```



# Moving Our Model to Create

CS

./Controllers/HomeController.cs

```
public IActionResult Index()  
{  
    return View();  
}
```

*We don't want our index creating our character anymore, so let's move that code to our Create method.*

```
public IActionResult Create(string characterName)  
{  
    var model = new CharSheetApp.Models.Character();  
    model.Name = "Hux";  
  
    return View(model);  
}
```





# Using Our Input Parameter

./Controllers/HomeController.cs

CS

```
public IActionResult Index()
{
    return View();
}

public IActionResult Create(string characterName)
{
    var model = new CharSheetApp.Models.Character();
    model.Name = characterName;

    return View(model);
}
```

*Change our hard-coded value to our parameter*



# Set Create Action to Use the Index View

./Controllers/HomeController.cs

CS

```
public IActionResult Index()  
{  
    return View();  
}
```

*This will redirect to /Home/Index after the model is created.*

```
public IActionResult Create(string characterName)  
{  
    var model = new CharSheetApp.Models.Character();  
    model.Name = characterName;  
    return View("Index", model);  
}
```



# Using Tag Helpers to Set the Called Action

./Views/Home/Index.cshtml

CSHTML

```
@model CharacterSheetApp.Models.Character
```

```
<h2>Characters:</h2>
```

```
<form asp-action="Create">
```

```
<div>
```

```
<input name="CharacterName" />
```

```
<input type="submit" value="Add Character" />
```

```
</div>
```

```
</form>
```

*asp-action is a tag helper that tells  
Razor what controller action we  
want to use on submit.*



# Optional: asp-controller Tag Helper

./Views/Home/Index.cshtml

CSHTML

```
@model CharacterSheetApp.Models.Character
<h2>Characters:</h2>

<form asp-action="Create" asp-controller="Home">
  <div>
    <input name="CharacterName" />
    <input type="submit" value="Add Character" />
  </div>
</form>
```

*We can also use the asp-controller tag helper to let our form know which controller to find the action in.*



*Without asp-controller here, MVC will use the controller that matches your view's directory.*

# Now We Can Accept User Input!

We're finally able to accept user input to create our list of characters, which means our developer doesn't need to manually update our character list anymore.

**TRY.NET MVC** Home

## Characters:

Add Character

### Active Characters

- Hux

© 2016 - TryDotNet





TRY

NETMUG