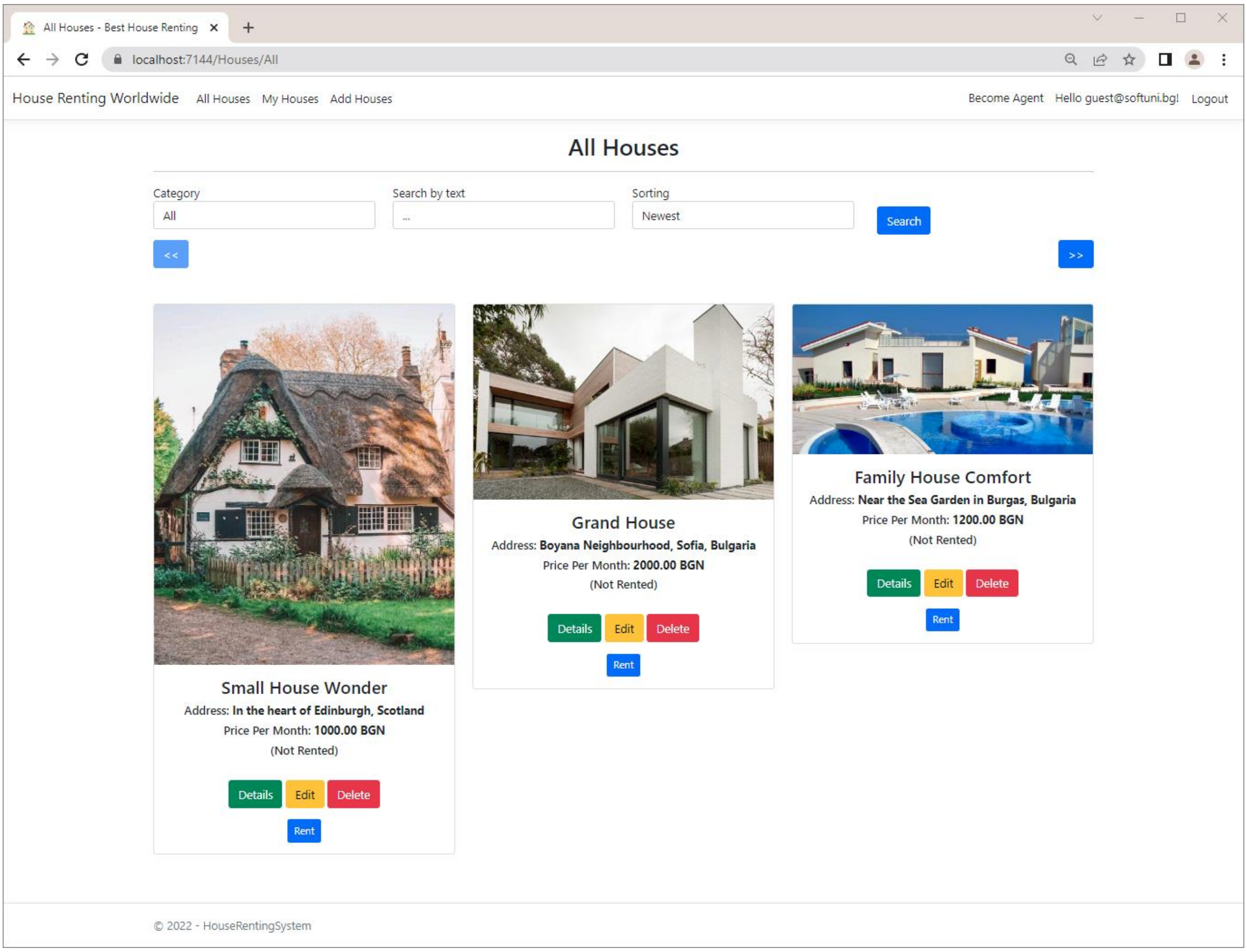
Workshop: Services and Error Pages

Workshop for the "ASP.NET Advanced" course @ SoftUni

The "**House Renting System**" **ASP.NET Core MVC App** is a Web application for **house renting**. **Users** can look at **all houses** with their **details**, **rent a house** and look at **their rented houses**. They can also **become Agents**. **Agents** can **add houses**, see their **details** and **edit** and **delete** only **houses they added**. The **Admin** has **all privileges** of **Users** and **Agents** and can see **all registrations** in the app and **all made rents**.

We will implement the app during the workshops in the course.



All Houses

Category: All

Search by text: ...

Sorting: Newest

Small House Wonder
Address: **In the heart of Edinburgh, Scotland**
Price Per Month: **1000.00 BGN**
(Not Rented)

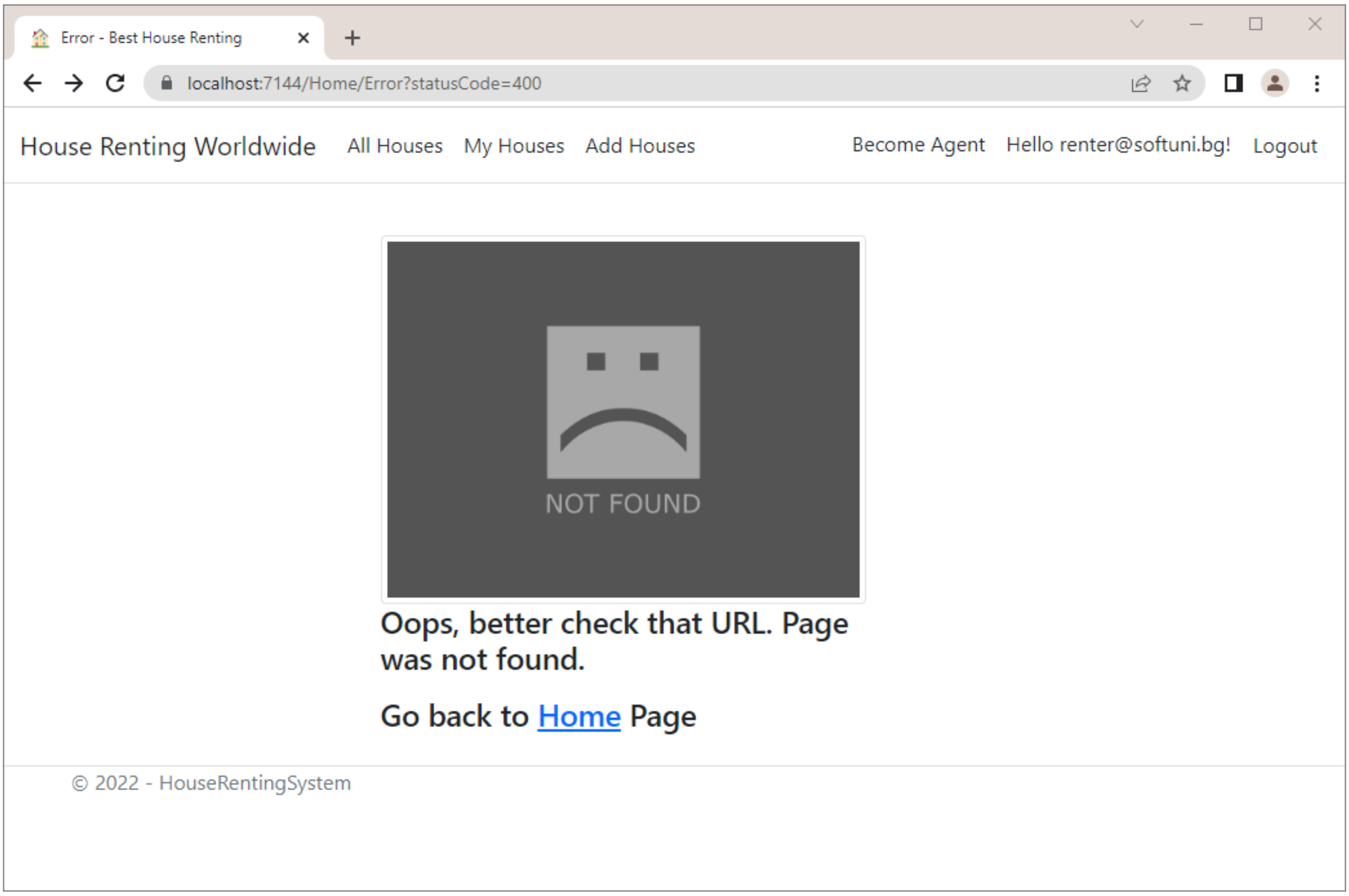
Grand House
Address: **Boyana Neighbourhood, Sofia, Bulgaria**
Price Per Month: **2000.00 BGN**
(Not Rented)

Family House Comfort
Address: **Near the Sea Garden in Burgas, Bulgaria**
Price Per Month: **1200.00 BGN**
(Not Rented)

© 2022 - HouseRentingSystem

1. Implement Custom Error Pages

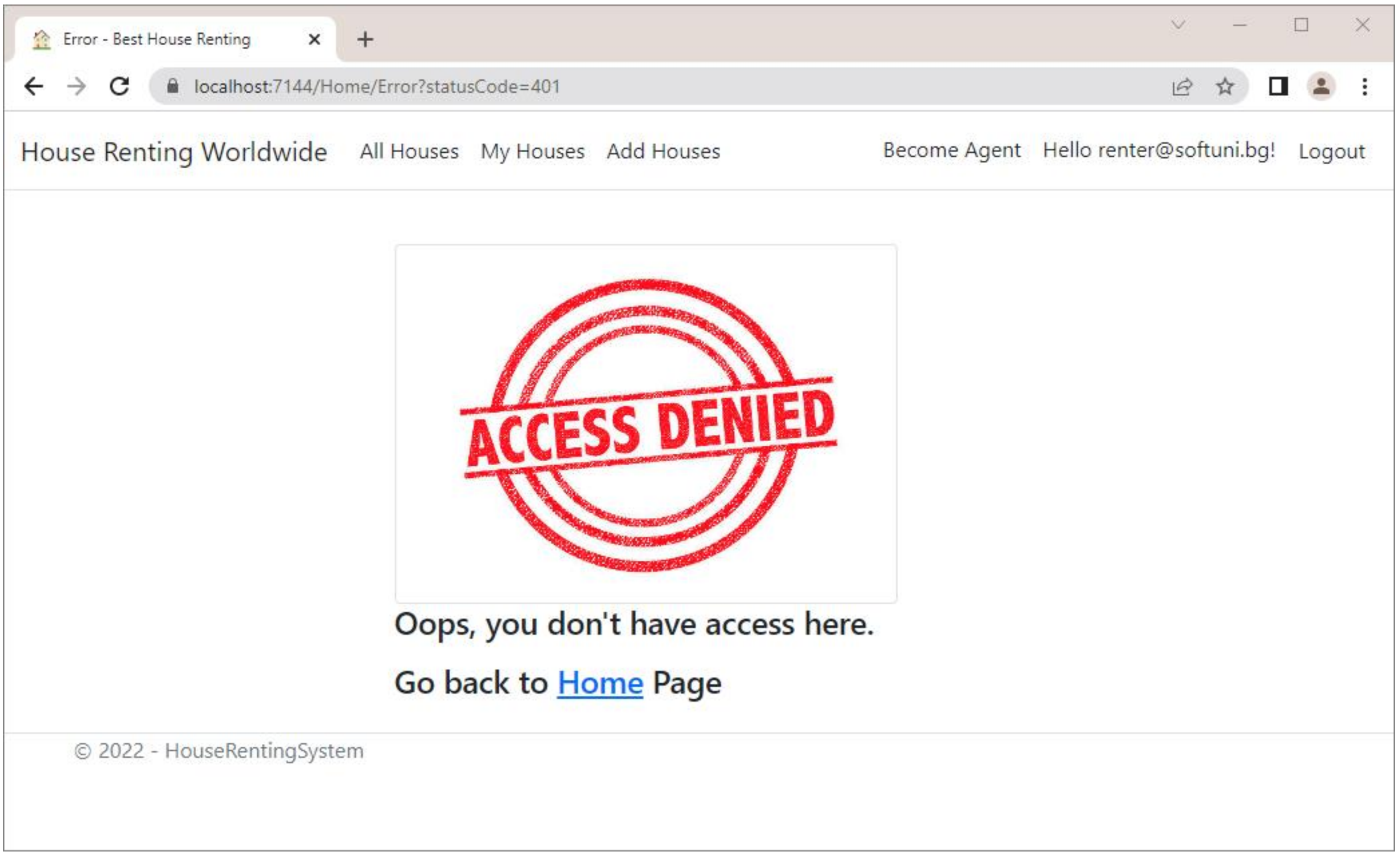
In this task, we will see how to **implement custom error pages** for "**400 Bad Request**" and "**401 Unauthorized**" **errors** in "**Production**" **environment**, so that errors look more user-friendly. They will look like this:



Oops, better check that URL. Page was not found.

Go back to Home Page

© 2022 - HouseRentingSystem

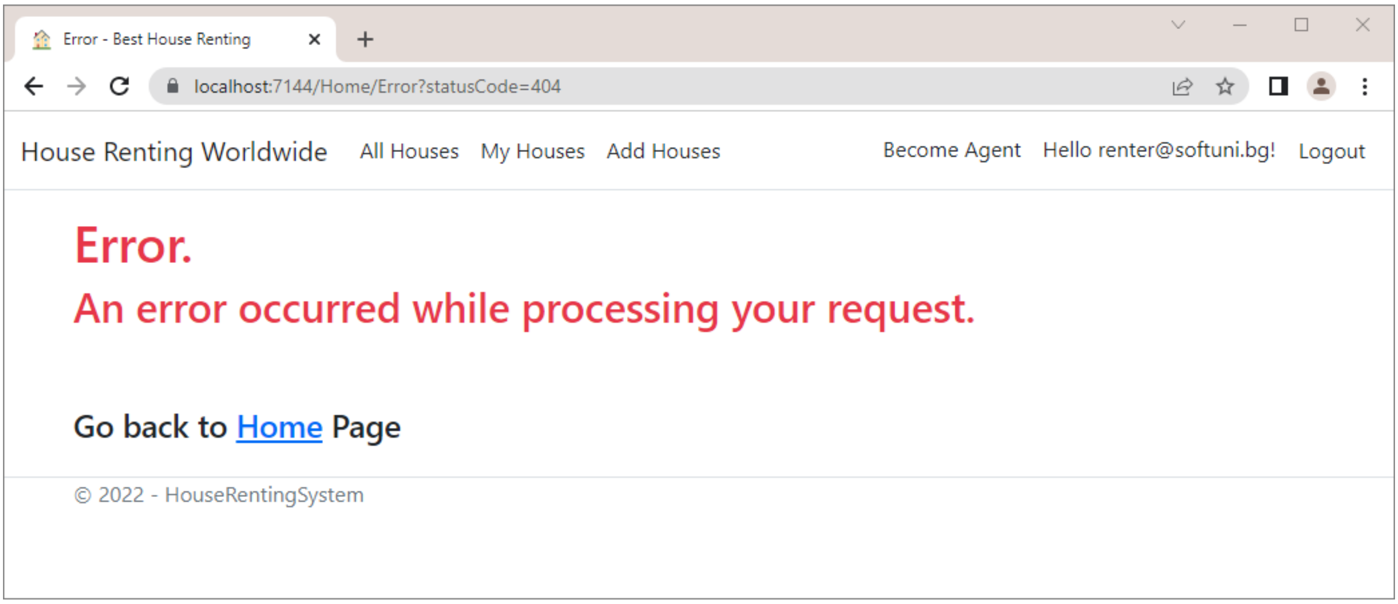


Oops, you don't have access here.

Go back to Home Page

© 2022 - HouseRentingSystem

The **error page** for **other status code** will be the following:



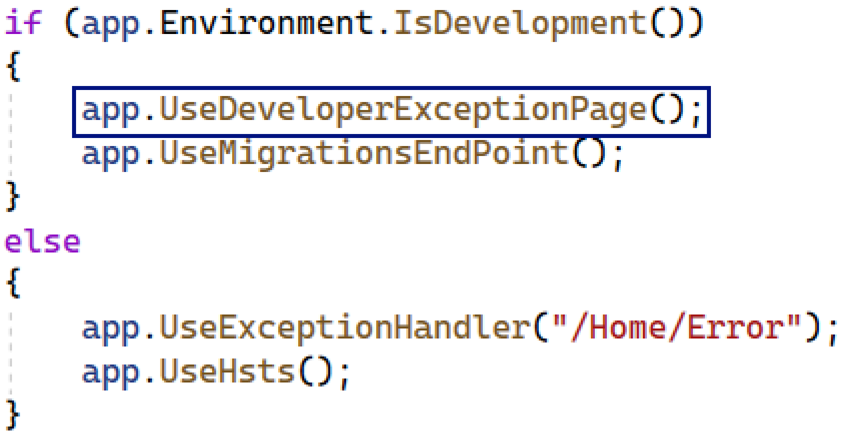
Error.

An error occurred while processing your request.

Go back to Home Page

© 2022 - HouseRentingSystem

To make **custom error pages**, we should first **modify the app middleware**. Go to the **`Program.cs`:**

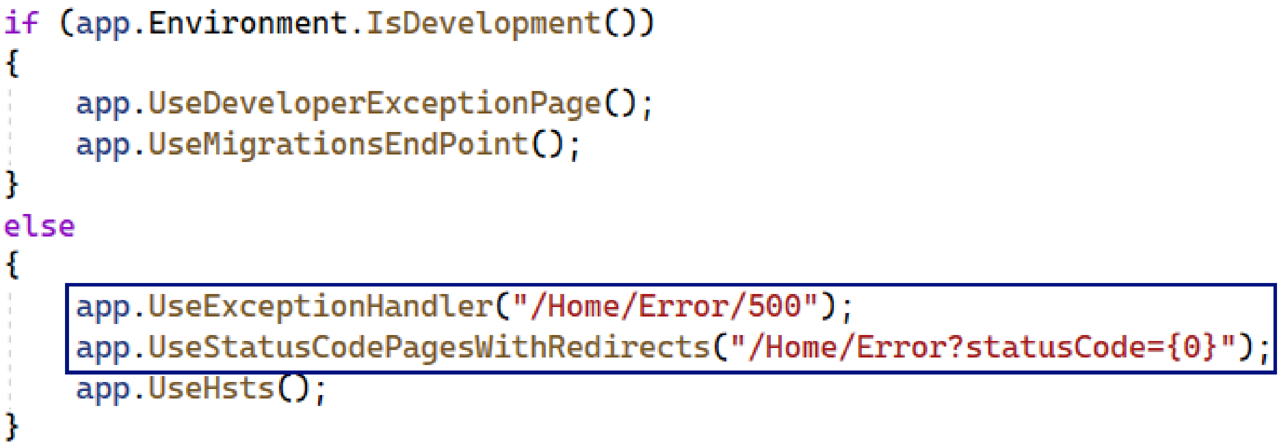


```
if (app.Environment.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
    app.UseMigrationsEndPoint();
}
else
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}
```

Note that when we are in "**`Development`**" **environment**, we have a **middleware** for using **developer exception pages**. When we are not, we **invoke** the **`Error()` method** of the **`HomeController` class**, which **returns an error view**.

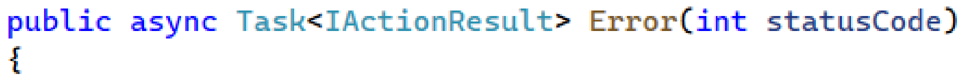
In our case, however, we want to **set custom error pages** when we are not in "**`Development`**", depending on the **error status code**.

For this reason, we should **change the second middleware**, responsible for **status code pages**. We want to be **redirected** again to "**`/Home/Error`**" but also to have the **status code as an URL query parameter**. Note that the "**`500 Internal Server Error`**" is treated differently than other status code pages, so we should **set a special exception handler** for it. Do it like this:



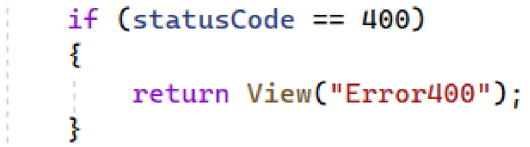
```
if (app.Environment.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
    app.UseMigrationsEndPoint();
}
else
{
    app.UseExceptionHandler("/Home/Error/500");
    app.UseStatusCodePagesWithRedirects("/Home/Error?statusCode={0}");
    app.UseHsts();
}
```

Now our **`Error()` method** in the **`HomeController`** should accept a **status code** like this:



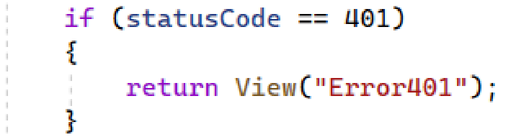
```
public async Task<IActionResult> Error(int statusCode)
{
```

Then, if we have a **status code** "**`400 Bad Request`**", we will return the **"`Error400.cshtml`" view** (which we will create later):



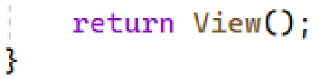
```
    if (statusCode == 400)
    {
        return View("Error400");
    }
```

If we have a **status code** "**`401 Unauthorized`**", we will return the **"`Error401.cshtml`" view**:



```
    if (statusCode == 401)
    {
        return View("Error401");
    }
```

If none of the above cases happen, we should just return the **"`Error.cshtml`" view**:



```
    return View();
}
```

Now let's **create the view files**. In the "**`/Views/Home`**" **folder** create the **"`Error.cshtml`"**, **"`Error400.cshtml`"** and **"`Error401.cshtml`" views**:

Remove the **"`Error.cshtml`"** file from the "**`/Views/Shared`**" **folder**, as we won't need it. Also, **remove** the **`ErrorViewModel` class** from the "**`/Models`**" **folder**:

Write the **"`Error.cshtml`" view** like this

```
@{
    ViewData["Title"] = "Error";
}

<h1 class="text-danger">Error.</h1>
<h2 class="text-danger">An error occurred while processing your request.</h2>
<br /><br />
<div class="text-left h4">  Go back to <a asp-controller="Home" asp-
action="Index">Home</a> Page</div>
```

The **"`Error400.cshtml`" view** should be the following:

```
@{
    ViewData["Title"] = "Error";
}

<div class="row">
    <div class="col-3"></div>
    <div class="col-6">
        <br>
        <div class="row">
            <div class="col-10">
                <img
src="https://www.publicdomainpictures.net/pictures/280000/velka/not-found-image-
15383864787lu.jpg" alt="Alternative image" class="img-thumbnail">
                <div class="text-left h4">
                    <p>Oops, better check that URL. Page was not found.</p>
                </div>
                <div class="text-left h4">  Go back to <a asp-controller="Home" asp-action="Index">Home</a> Page</div>
            </div>
        </div>
    </div>
    <div class="col-3"></div>
</div>
```

Get the **image URL** from here:

```
https://www.publicdomainpictures.net/pictures/280000/velka/not-found-image-15383864787lu.jpg
```

The last view is **"`Error401.cshtml`".** Write it like this:

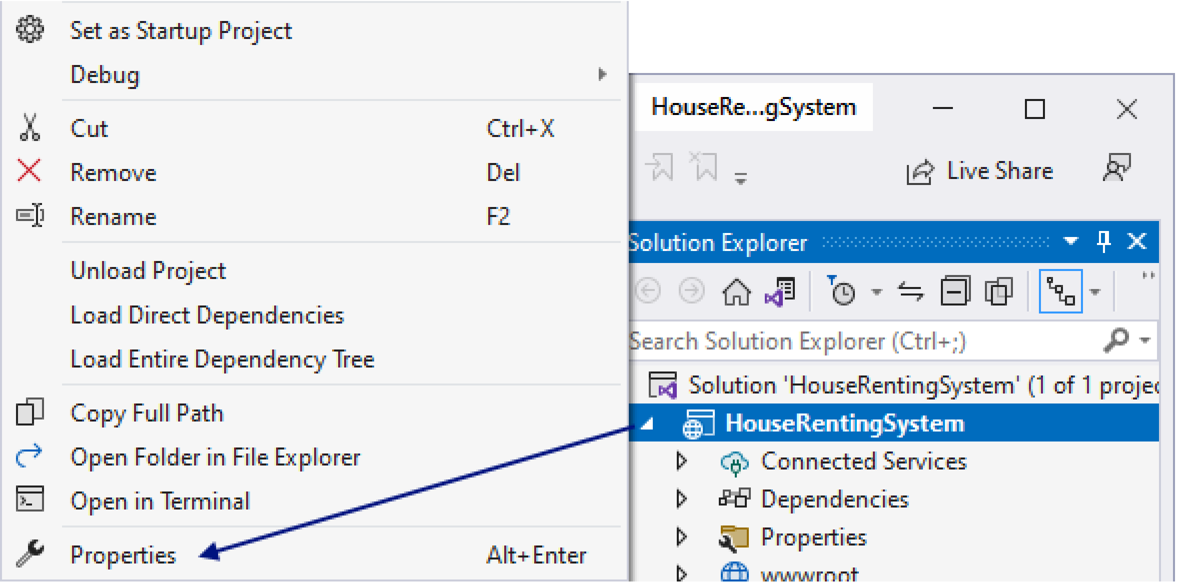
```
@{
    ViewData["Title"] = "Error";
}

<div class="row">
    <div class="col-3"></div>
    <div class="col-6">
        <br>
        <div class="row">
            <div class="col-10">
                <img src="https://media.istockphoto.com/photos/rubber-stamp-picture-id545992942?k=6&m=545992942&s=612x612&w=0&h=XD1kddQeDrbG9DEQQxGNf6_3snw4gdFZv8GqSFta6Is=" alt="Alternative image" class="img-thumbnail">
                <div class="text-left h4">
                    <p>Oops, you don't have access here.</p>
                </div>
                <div class="text-left h4">  Go back to <a asp-controller="Home" asp-action="Index">Home</a> Page</div>
            </div>
        </div>
    </div>
    <div class="col-3"></div>
</div>
```

The **image URL** for this page is the following:

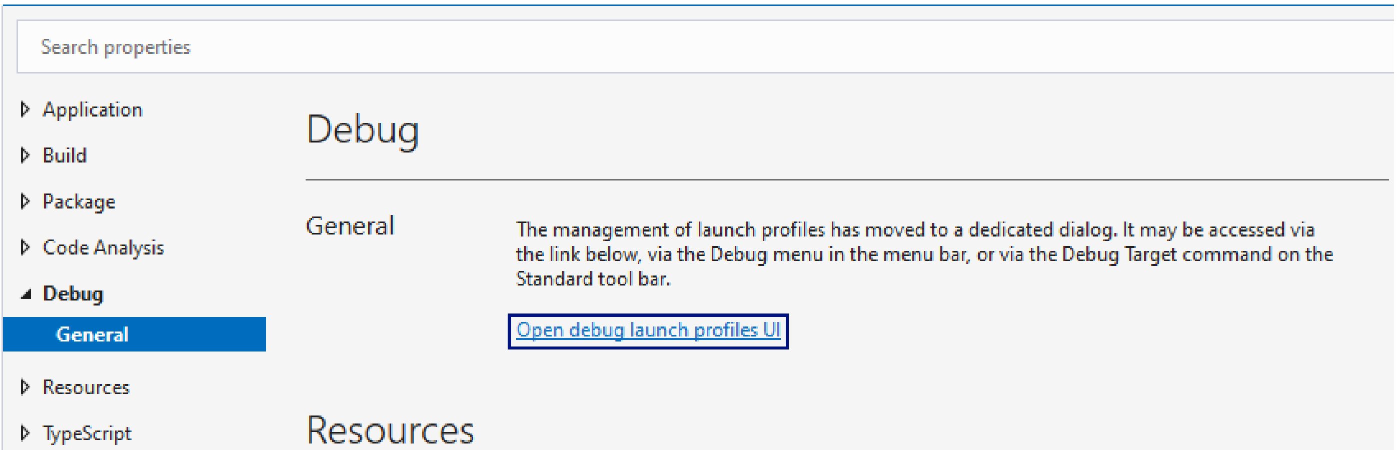
```
https://media.istockphoto.com/photos/rubber-stamp-picture-id545992942?k=6&m=545992942&s=612x612&w=0&h=XD1kddQeDrbG9DEQQxGNf6_3snw4gdFZv8GqSFta6Is=
```

Now let's **change the environment** to "**`Production`**", so that we can **test the error pages** we created. To do this, we need to **change** the **ASPNETCORE_ENVIRONMENT variable value**.

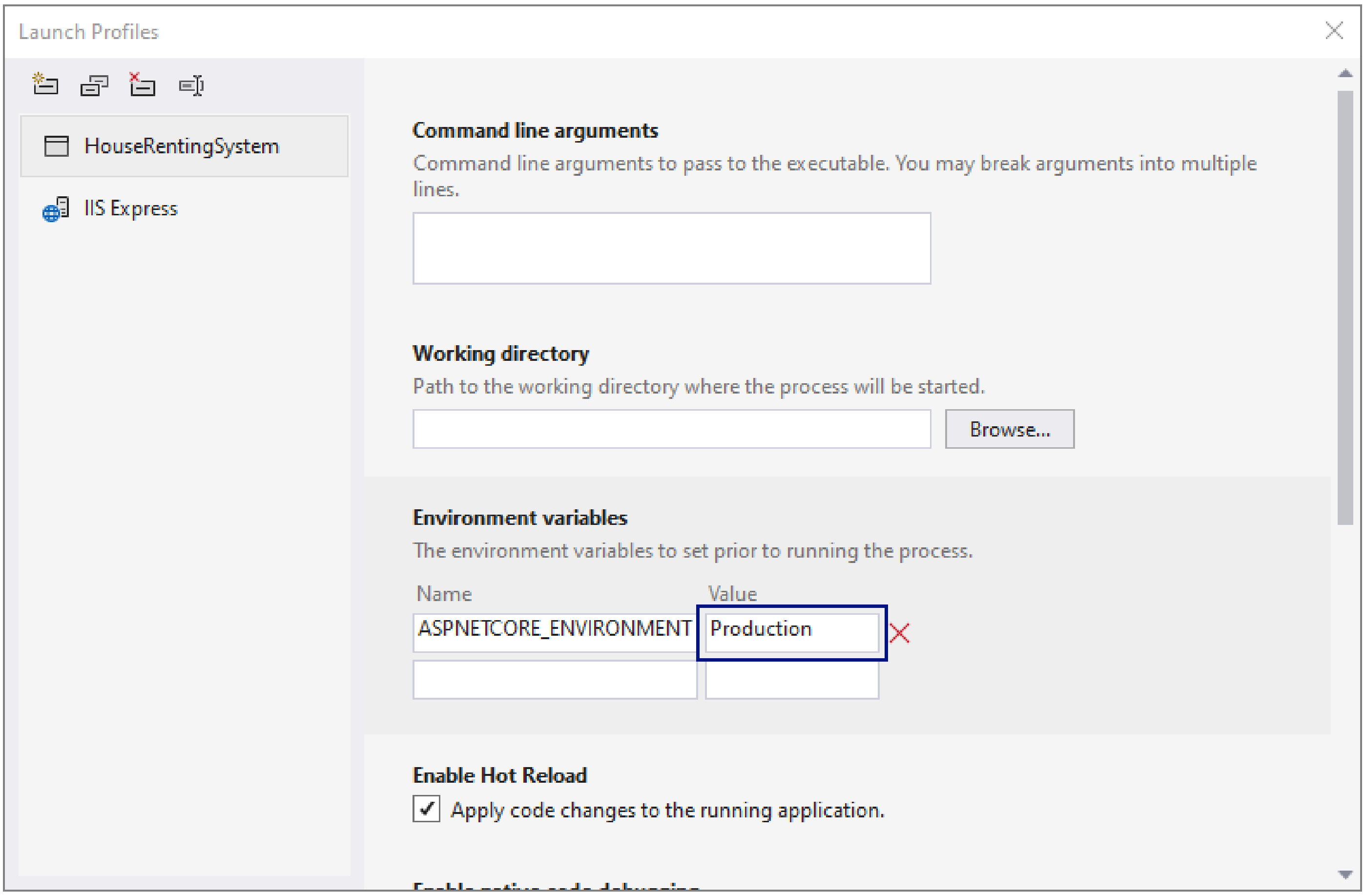
Right-click on the **project** in **Visual Studio** and go to **`[Properties]`**:



Then, go to the **`[Debug]` menu** and change the **`ASPNETCORE_ENVIRONMENT` variable value** from "**`Development`**" to "**`Production`**":



The management of launch profiles has moved to a dedicated dialog. It may be accessed via the link below, via the Debug menu in the menu bar, or via the Debug Target command on the Standard tool bar.


Launch Profiles

HouseRentingSystem

IIS Express

Command line arguments

Command line arguments to pass to the executable. You may break arguments into multiple lines.

Working directory

Path to the working directory where the process will be started.

Environment variables

The environment variables to set prior to running the process.

Name	Value
ASPNETCORE_ENVIRONMENT	Production

Enable Hot Reload

☑ Apply code changes to the running application.

Now **run the app**. Go to a **non-existing page** like https://localhost:44342/Houses/Details/1000 to see the "**`400 Bad Request`**" **error page**. You should be **redirected** to https://localhost:44342/Home/Error?statusCode=400.

Test the "**`401 Unauthorized`**" **error page** by going to https://localhost:44342/Houses/Edit/{id} with a **house id from the database**, but you **should not be the creator of the house**. You should be **redirected** to https://localhost:44342/Home/Error?statusCode=401.