10

Modificando a Rota Padrão do MVC

Transcrição

Continuando, implementaremos as demais views pertinentes ao nosso e-commerce. Para isso, teremos que criar outras actions, dentro do nosso pedido `Controller`. Vamos copiar a primeira action criada, e colar abaixo da existente, criando um método `Carrinho()`:

```
namespace CasaDoCodigo.Controllers
{
    public class PedidoController : Controller
    {
        public IActionResult Carrossel()
        {
            return View();
        }
        public IActionResult Carrinho()
        {
            return View();
        }
    }
}
```

Em seguida, criaremos uma nova action, à qual daremos o nome de `Cadastro()`, que é o espaço onde o usuário preencherá seus dados:

```
namespace CasaDoCodigo.Controllers
{
    public class PedidoController : Controller
    {
        public IActionResult Carrossel()
        {
            return View();
        }
        public IActionResult Carrinho()
        {
            return View();
        }
        public IActionResult Cadastro()
        {
        return View();
        }
    }
}
```

Feito isso, o pedido é fechado e o cliente poderá visualizar o resumo do pedido, sendo assim, criaremos um outro método, chamado `Resumo()`:

```
namespace CasaDoCodigo.Controllers
{
```

```
    public class PedidoController : Controller
    {
        public IActionResult Carrossel()
        {
            return View();
        }
        public IActionResult Carrinho(),
        {
            return View();
        }
        public IActionResult Cadastro()
        {
        return View();
        }
        public IActionResult Resumo()
        {
        return View();
        }
    }
}
```

Atualizaremos a aplicação e iremos acessá-la com essas novas páginas. Primeiro, abriremos a URL `localhost:58246/pedido/carrossel`. Teremos a view de carrossel funcionando perfeitamente. O botão "Adicionar" ainda não está funcionando corretamente, por enquanto, ele apenas nos remete à home page.

Para acessarmos as demais páginas, teremos que alterar diretamente o endereço da URL. Sendo assim, digitaremos `localhost:58246/pedido/carrinho` para abrir a página de carrinho. Nela, encontraremos os produtos selecionados e dois links: "adicionar produtos" e "preencher cadastro". Ambos não funcionam ainda.

A seguir, o cliente acessará a página de cadastro, sendo assim, acessaremos `localhost:58246/pedido/cadastro`, onde temos um formulário, para que o usuário complete com os seus dados. Por fim, ele clicará no botão "finalizar pedido", que o remeterá à página `localhost:58246/pedido/resumo`, com o resumo do pedido.

Todas as views são arquivos `cshtml`, ou seja, são arquivos estáticos, com quase nenhuma variação baseada nas escolhas do cliente porque nossa web designer criou estas páginas como um modelo somente. Nosso trabalho será transformar o HTML para que possa receber informações que serão enviadas pela aplicação.

Já temos as páginas, e conseguimos acessá-las a partir da URL, agora faremos uma alteração para que a página inicial não seja mais a do padrão ASP.NET Core MVC, e sim a do carrossel, que é o local onde o cliente poderá visualizar os produtos disponíveis. Para que isso aconteça, teremos que modificar a rota padrão.

Uma rota é uma regra que define como as páginas serão encontradas em uma aplicação MVC. No projeto, localizaremos o arquivo `Startup.cs`. Esta classe contém dois métodos, o primeiro é `ConfigureServices()`, e serve para a configuração de serviços, e outro chamado `Configure()`, onde encontramos um trecho de código referente à rota padrão:

```
namespace CasaDoCodigo
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }
```

```
        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc();
        }

        // This method gets called by the runtime. Use this method to configure the HTTP request pip
        public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseBrowserLink();
                app.UseDeveloperExceptionPage();
            }
            else
            {
                app.UseExceptionHandler("/Home/Error");
            }

            app.UseStaticFiles();

            app.UseMvc(routes =>
            {
                routes.MapRoute(
                    name: "default",
                    template: "{controller=Home}/{action=Index}/{id?}");
            });
        }
    }
}
```

O `routes.MapRoute()` define o mapeamento da rota padrão e o `template`, que contém uma string `{controller=Home}/{action=Index}`. Portanto, temos a rota padrão quando inserimos o endereço raiz do nosso site, sem mencionar nenhum outro caminho. Mas queremos direcionar o usuário para o carrossel. Sendo assim, substituiremos `Home` por `Pedido`, e `Index` por `Carrossel`:

```
namespace CasaDoCodigo
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc();
        }
```

```
        // This method gets called by the runtime. Use this method to configure the HTTP request pip
        public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseBrowserLink();
                app.UseDeveloperExceptionPage();
            }
            else
            {
                app.UseExceptionHandler("/Home/Error");
            }

            app.UseStaticFiles();

            app.UseMvc(routes =>
            {
                routes.MapRoute(
                    name: "default",
                    template: "{controller=Pedido}/{action=Carrossel}/{id?}");
            });
        }
    }
}
```

Salvaremos e executaremos a aplicação. Ao fazermos isso, caímos diretamente na página de carrossel. Na sequência, podemos eliminar as páginas de `Home`, já que não as utilizaremos mais. Especificamente o arquivo `Home` dentro da pasta "Views", e o arquivo `HomeController`, na pasta "Controllers". Executaremos novamente, para nos certificarmos de que tudo continua funcionando corretamente.