

ABCRetailers Web Application - HomeController Code Snapshots

HomeController.cs (Full Code Snapshot)

```
HomeController.cs  X
ABCRetailers  ABCRetailers.Controllers

1  // Controllers/HomeController.cs
2  using ABCRetailers.Models;
3  using ABCRetailers.Models.ViewModels;
4  using ABCRetailers.Services;
5  using Microsoft.AspNetCore.Mvc;
6  using System.Diagnostics;
7
8  namespace ABCRetailers.Controllers
9  {
10     1 reference
11     public class HomeController : Controller
12     {
13
14         0 references
15         private readonly IAzureStorageService _storageService;
16
17         public HomeController(IAzureStorageService storageService)
18         {
19             _storageService = storageService;
20         }
21
22         1 reference
23         public async Task<IActionResult> Index()
24         {
25             var products = await _storageService.GetAllEntitiesAsync<Product>();
26             var customers = await _storageService.GetAllEntitiesAsync<Customer>();
27             var orders = await _storageService.GetAllEntitiesAsync<Order>();
28
29             var viewModel = new HomeViewModel
30             {
31                 FeaturedProducts = products.Take(5).ToList(),
32                 ProductCount = products.Count,
33                 CustomerCount = customers.Count,
34                 OrderCount = orders.Count
35             };
36
37             return View(viewModel);
38         }
39
40         0 references
41         public IActionResult Privacy()
42         {
43             return View();
44         }
45
46         [HttpPost]
47         0 references
48         public async Task<IActionResult> InitializeStorage()
49         {
50             try
51             {
52                 // Force re-initialization of storage
53                 await _storageService.GetAllEntitiesAsync<Customer>(); // This will trigger initialization
54                 TempData["Success"] = "Azure Storage initialized successfully!";
55             }
56             catch (Exception ex)
57             {
58                 TempData["Error"] = $"Failed to initialize storage: {ex.Message}";
59             }
60             return RedirectToAction(nameof(Index));
61         }
62
63         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
64         0 references
65         public IActionResult Error()
66         {
67             return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
68         }
69     }
70 }
```

HomeController.cs (IAzureStorageService Action Code Snapshot)

```
private readonly IAzureStorageService _storageService;

0 references
public HomeController(IAzureStorageService storageService)
{
    _storageService = storageService;
}
```

HomeController.cs (Index Action Code Snapshot)

```
1 reference
public async Task<IActionResult> Index()
{
    var products = await _storageService.GetAllEntitiesAsync<Product>();
    var customers = await _storageService.GetAllEntitiesAsync<Customer>();
    var orders = await _storageService.GetAllEntitiesAsync<Order>();

    var viewModel = new HomeViewModel
    {
        FeaturedProducts = products.Take(5).ToList(),
        ProductCount = products.Count,
        CustomerCount = customers.Count,
        OrderCount = orders.Count
    };

    return View(viewModel);
}
```

HomeController.cs (Privacy Action Code Snapshot)

```
0 references
public IActionResult Privacy()
{
    return View();
}
```

HomeController.cs (InitializeStorage Action Code Snapshot)

[HttpPost]

0 references

public async Task<IActionResult> InitializeStorage()

```
{
    try
    {
        // Force re-initialization of storage
        await _storageService.GetAllEntitiesAsync<Customer>(); // This will trigger initialization
        TempData["Success"] = "Azure Storage initialized successfully!";
    }
    catch (Exception ex)
    {
        TempData["Error"] = $"Failed to initialize storage: {ex.Message}";
    }
    return RedirectToAction(nameof(Index));
}
```

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

0 references

public IActionResult Error()

```
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
}
```

CustomerController.cs (Full Code Snapshot)

```
CustomerController.cs
ABCRetailers

1 // Controllers/CustomController.cs
2 using Microsoft.AspNetCore.Mvc;
3 using ABCRetailers.Models;
4 using ABCRetailers.Services;
5
6 namespace ABCRetailers.Controllers
7 {
8     [reference]
9     public class CustomerController : Controller
10     {
11         private readonly IAzureStorageService _storageService;
12
13         [reference]
14         public CustomerController(IAzureStorageService storageService)
15         {
16             _storageService = storageService;
17         }
18
19         [reference]
20         public async Task<ActionResult> Index()
21         {
22             var customers = await _storageService.GetAllEntitiesAsync<Customer>();
23             return View(customers);
24         }
25
26         [reference]
27         public IActionResult Create()
28         {
29             return View();
30         }
31
32         [HttpPost]
33         [ValidateAntiForgeryToken]
34         [reference]
35         public async Task<ActionResult> Create(Customer customer)
36         {
37             if (ModelState.IsValid)
38             {
39                 try
40                 {
41                     await _storageService.AddEntityAsync(customer);
42                     TempData["Success"] = "Customer created successfully!";
43                     return RedirectToAction(nameof(Index));
44                 }
45                 catch (Exception ex)
46                 {
47                     ModelState.AddModelError("", $"Error creating customer: {ex.Message}");
48                 }
49             }
50             return View(customer);
51         }
52
53         [reference]
54         public async Task<ActionResult> Edit(string id)
55         {
56             if (string.IsNullOrEmpty(id))
57             {
58                 return NotFound();
59             }
60
61             var customer = await _storageService.GetEntityAsync<Customer>("Customer", id);
62             if (customer == null)
63             {
64                 return NotFound();
65             }
66             return View(customer);
67         }
68
69         [HttpPost]
70         [ValidateAntiForgeryToken]
71         [reference]
72         public async Task<ActionResult> Edit(Customer customer)
73         {
74             if (ModelState.IsValid)
75             {
76                 try
77                 {
78                     await _storageService.UpdateEntityAsync(customer);
79                     TempData["Success"] = "Customer updated successfully!";
80                     return RedirectToAction(nameof(Index));
81                 }
82                 catch (Exception ex)
83                 {
84                     ModelState.AddModelError("", $"Error updating customer: {ex.Message}");
85                 }
86             }
87             return View(customer);
88         }
89
90         [HttpPost]
91         [reference]
92         public async Task<ActionResult> Delete(string id)
93         {
94             try
95             {
96                 await _storageService.DeleteEntityAsync<Customer>("Customer", id);
97                 TempData["Success"] = "Customer deleted successfully!";
98             }
99             catch (Exception ex)
100             {
101                 TempData["Error"] = $"Error deleting customer: {ex.Message}";
102             }
103             return RedirectToAction(nameof(Index));
104         }
105     }
106 }
```

ProductController.cs

OrderController.cs

UploadController.cs