

# ABCRetailers Web Application - CustomerController Code Snapshots

## CustomerController.cs (Full Code Snapshot)

```
CustomerController.cs  + X
ABCRetailers

1  // Controllers/CustomerController.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 }
```

### CustomerController.cs (IAzureStorageService Action Code Snapshot)

```
private readonly IAzureStorageService _storageService;

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

### CustomerController.cs (Index Action Code Snapshot)

```
3 references
public async Task<IActionResult> Index()
{
    var customers = await _storageService.GetAllEntitiesAsync<Customer>();
    return View(customers);
}
```

### CustomerController.cs (Create Action Code Snapshot)

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

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create(Customer customer)
{
    if (ModelState.IsValid)
    {
        try
        {
            await _storageService.AddEntityAsync(customer);
            TempData["Success"] = "Customer created successfully!";
            return RedirectToAction(nameof(Index));
        }
        catch (Exception ex)
        {
            ModelState.AddModelError("", $"Error creating customer: {ex.Message}");
        }
    }
    return View(customer);
}
```

## CustomerController.cs (Edit Action Code Snapshot)

0 references

```
public async Task<IActionResult> Edit(string id)
{
    if (string.IsNullOrEmpty(id))
    {
        return NotFound();
    }

    var customer = await _storageService.GetEntityAsync<Customer>("Customer", id);
    if (customer == null)
    {
        return NotFound();
    }

    return View(customer);
}
```

[HttpPost]

[ValidateAntiForgeryToken]

0 references

```
public async Task<IActionResult> Edit(Customer customer)
{
    if (ModelState.IsValid)
    {
        try
        {
            await _storageService.UpdateEntityAsync(customer);
            TempData["Success"] = "Customer updated successfully!";
            return RedirectToAction(nameof(Index));
        }
        catch (Exception ex)
        {
            ModelState.AddModelError("", $"Error updating customer: {ex.Message}");
        }
    }

    return View(customer);
}
```

## CustomerController.cs (Delete Action Code Snapshot)

[HttpPost]

0 references

```
public async Task<IActionResult> Delete(string id)
{
    try
    {
        await _storageService.DeleteEntityAsync<Customer>("Customer", id);
        TempData["Success"] = "Customer deleted successfully!";
    }
    catch (Exception ex)
    {
        TempData["Error"] = $"Error deleting customer: {ex.Message}";
    }

    return RedirectToAction(nameof(Index));
}
```