ST10445601

CLDV6212

POE Part 2

Table of Contents

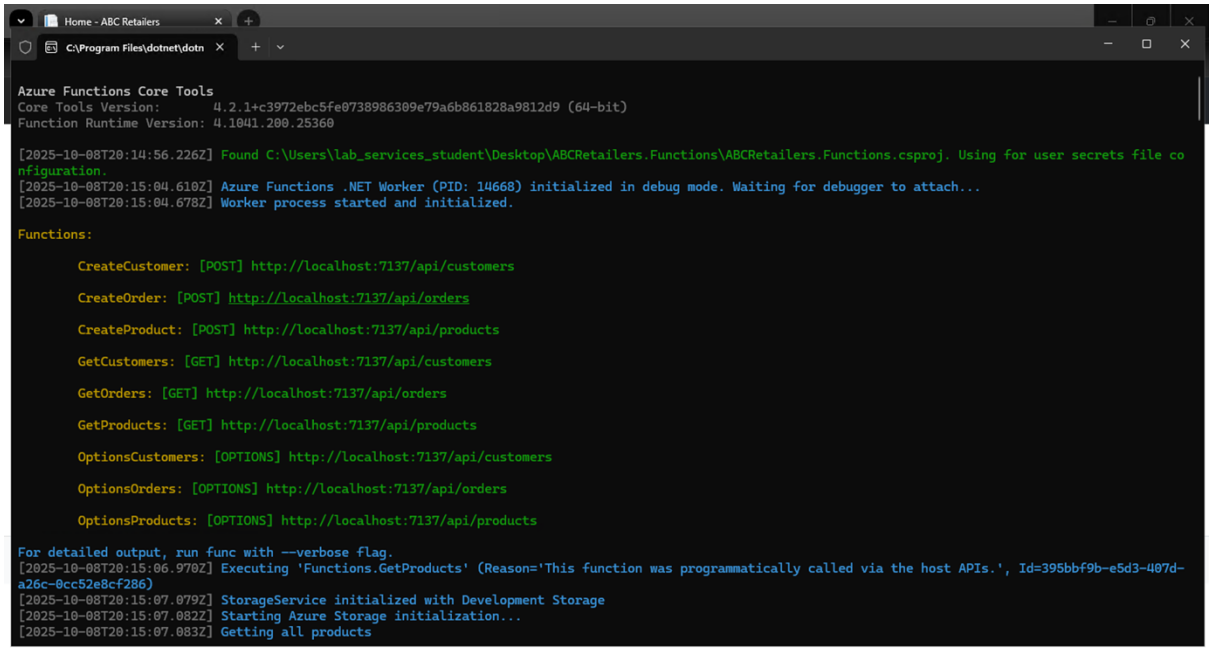
Youtube Link:

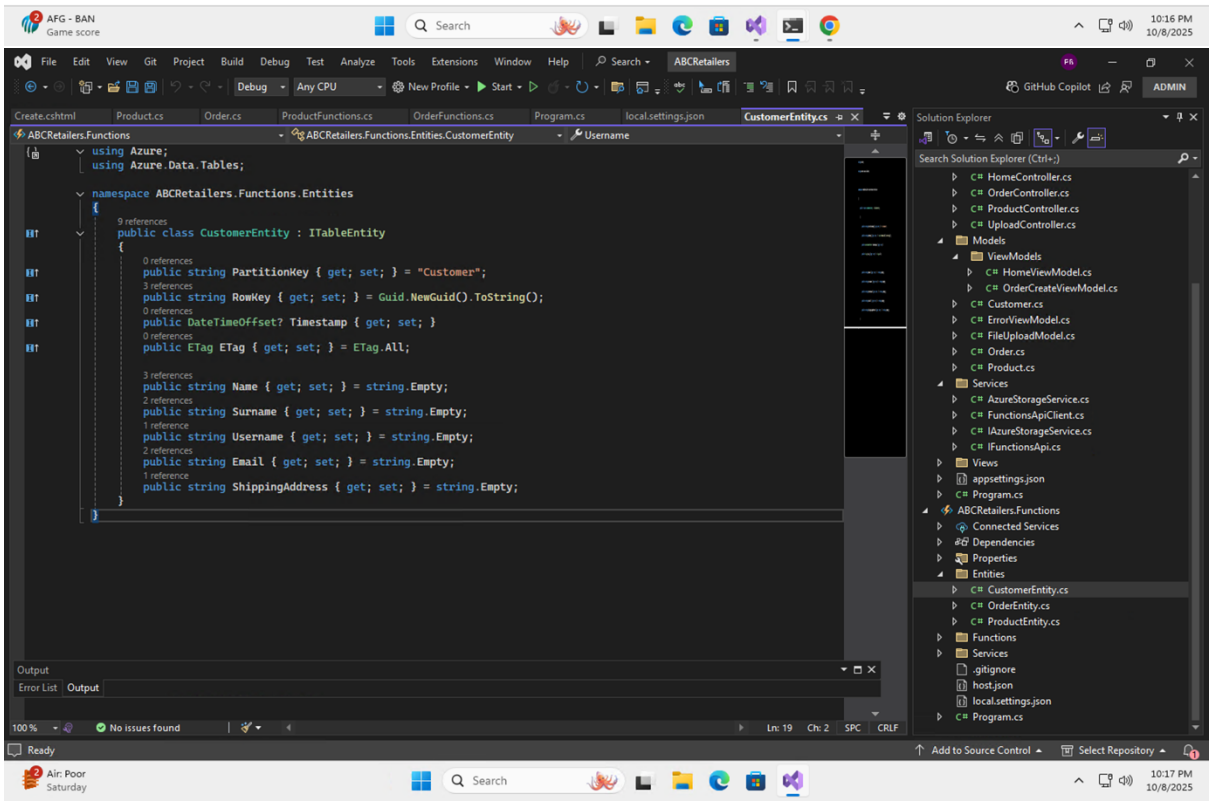
https://youtu.be/zdUm9_RAcwY?si=pIZJuG4M_jTgwxHf

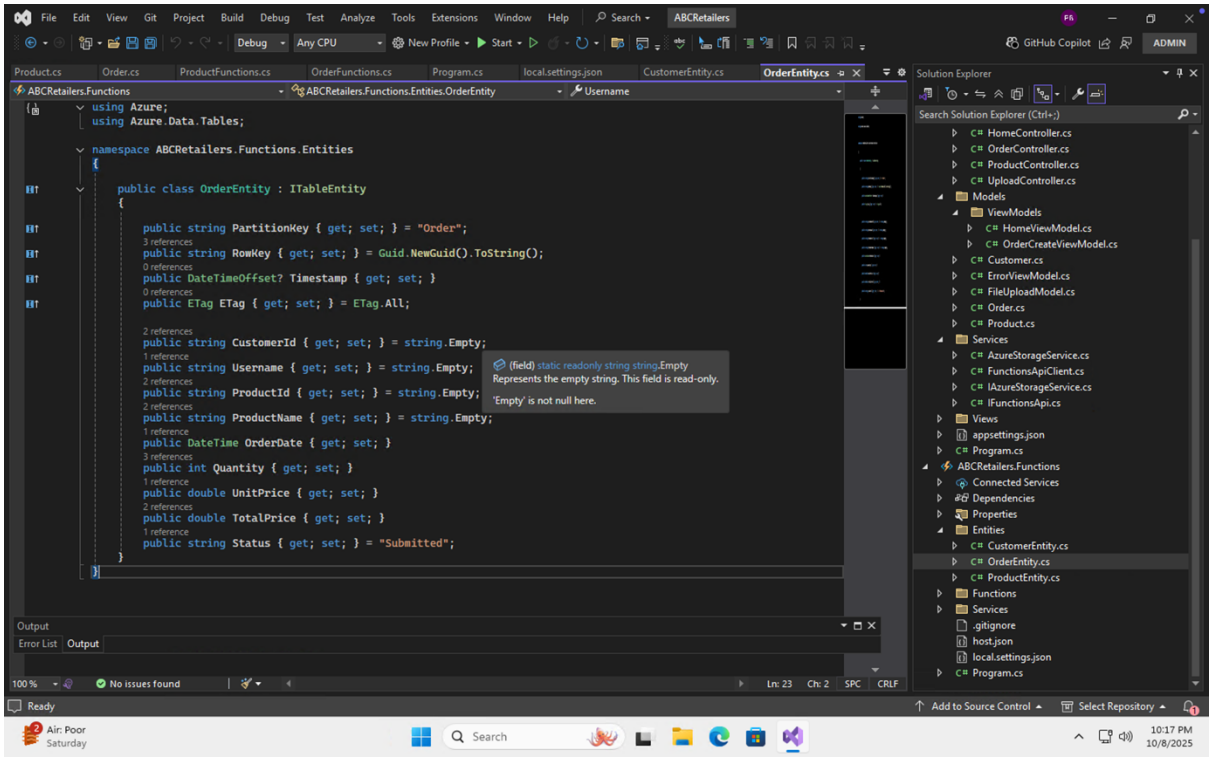
GITHub Link:

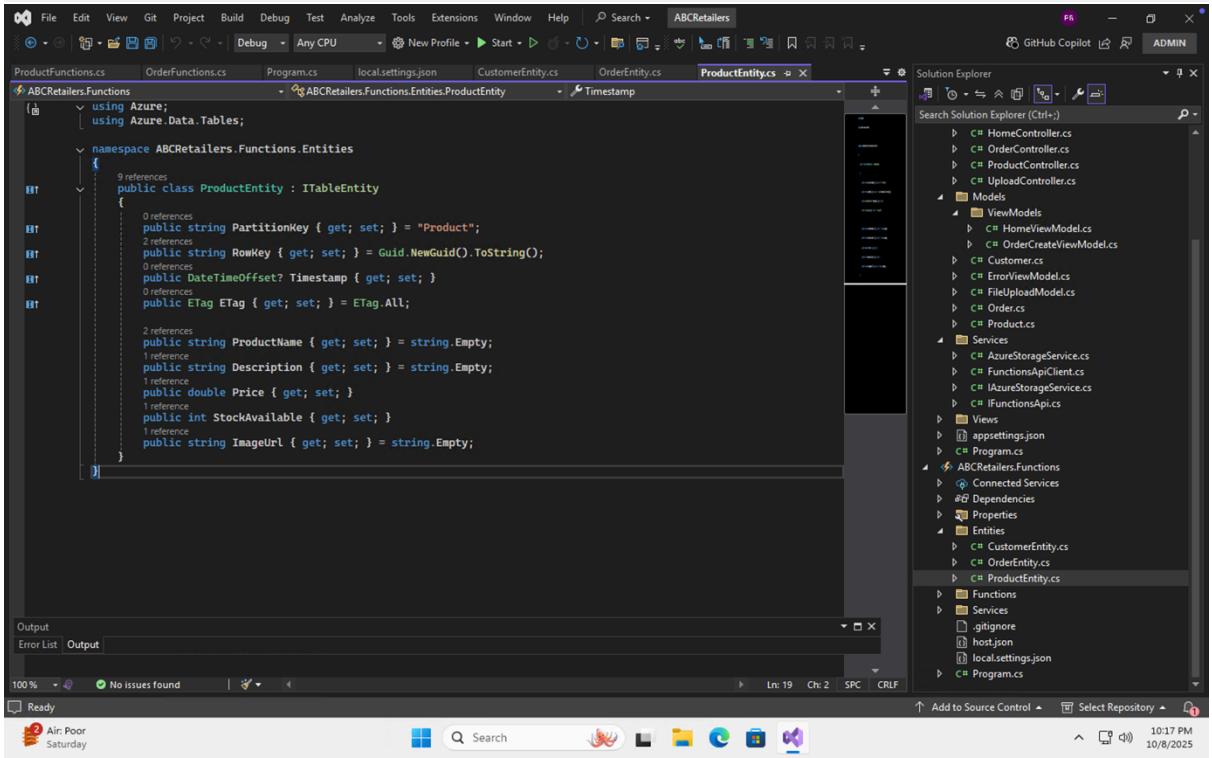
https://github.com/BritoPaulo/cldvpoepart2.git

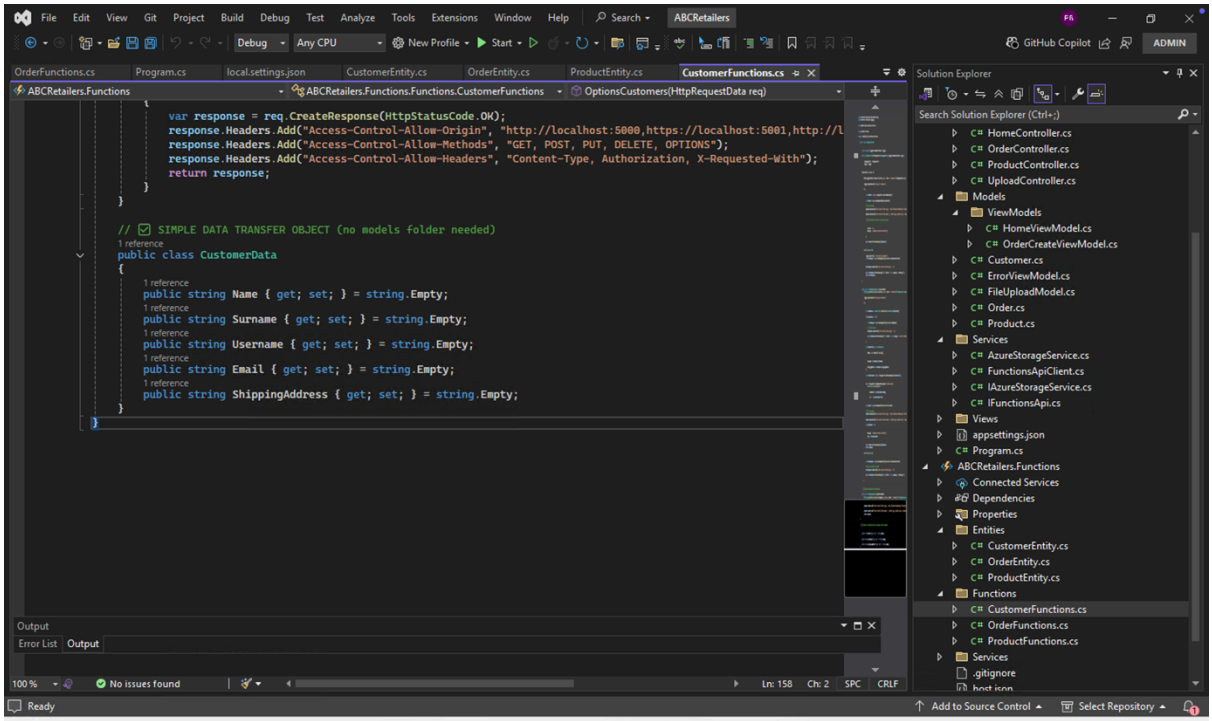
Screenshots:

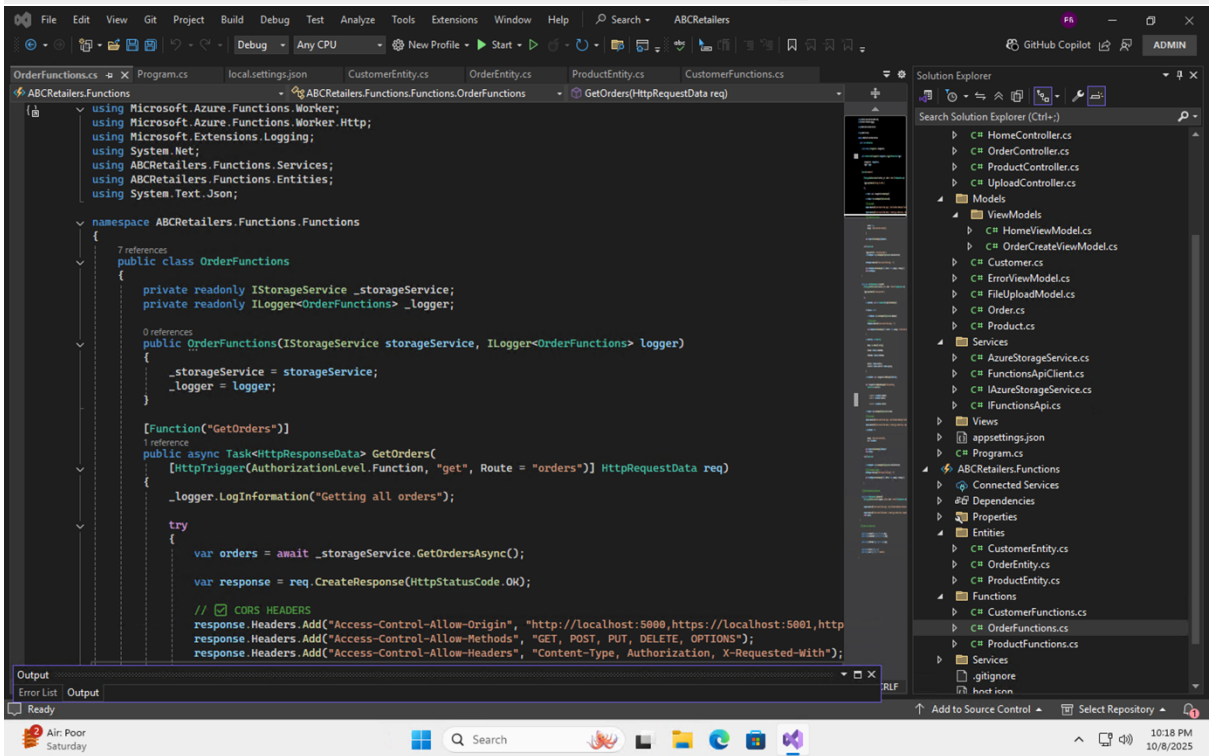


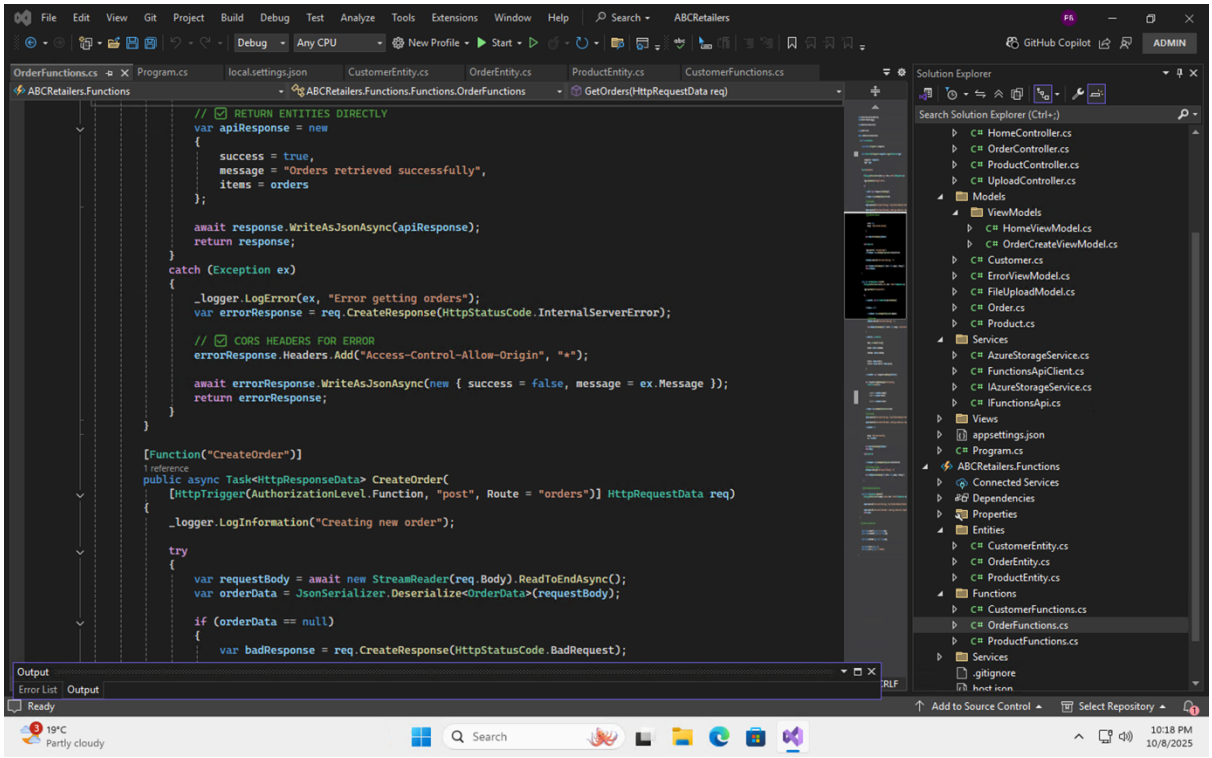


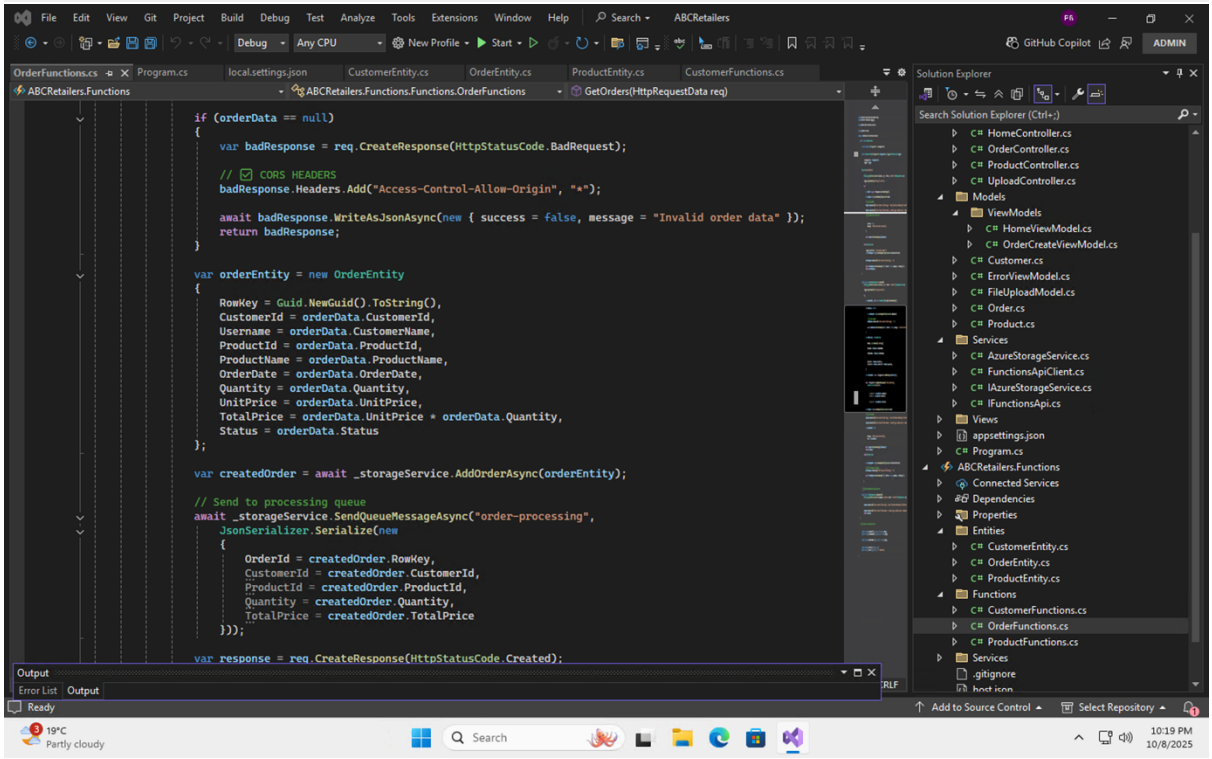

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
Product.cs
Order.cs
ProductFunctions.cs
OrderFunctions.cs
Program.cs
local.settings.json
CustomerEntity.cs
OrderEntity.cs
ABCRetailers.Functions
ABCRetailers.Functions.Entities.OrderEntity
Username
using Azure;
using Azure.Data.Tables;
namespace ABCRetailers.Functions.Entities
{
public class OrderEntity : ITableEntity
{
public string PartitionKey { get; set; } = "Order";
3 references
public string RowKey { get; set; } = Guid.NewGuid().ToString();
0 references
public DateTimeOffset? Timestamp { get; set; }
0 references
public ETag ETag { get; set; } = ETag.All;
2 references
public string CustomerId { get; set; } = string.Empty;
1 reference
public string Username { get; set; } = string.Empty;
2 references
public string ProductId { get; set; } = string.Empty;
2 references
public string ProductName { get; set; } = string.Empty;
1 reference
public DateTime OrderDate { get; set; }
3 references
public int Quantity { get; set; }
1 reference
public double UnitPrice { get; set; }
2 references
public double TotalPrice { get; set; }
1 reference
public string Status { get; set; } = "Submitted";
}
}
(field) static readonly string string.Empty
Represents the empty string. This field is read-only.
'Empty' is not null here.
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
Services
.gitignore
host.json
local.settings.json
Program.cs
Output
Error List
Output
100 %
No issues found
Ln: 23
Ch: 2
SPC
CRLF
Ready
Add to Source Control
Select Repository
Air: Poor
Saturday
Search
10:17 PM
10/8/2025

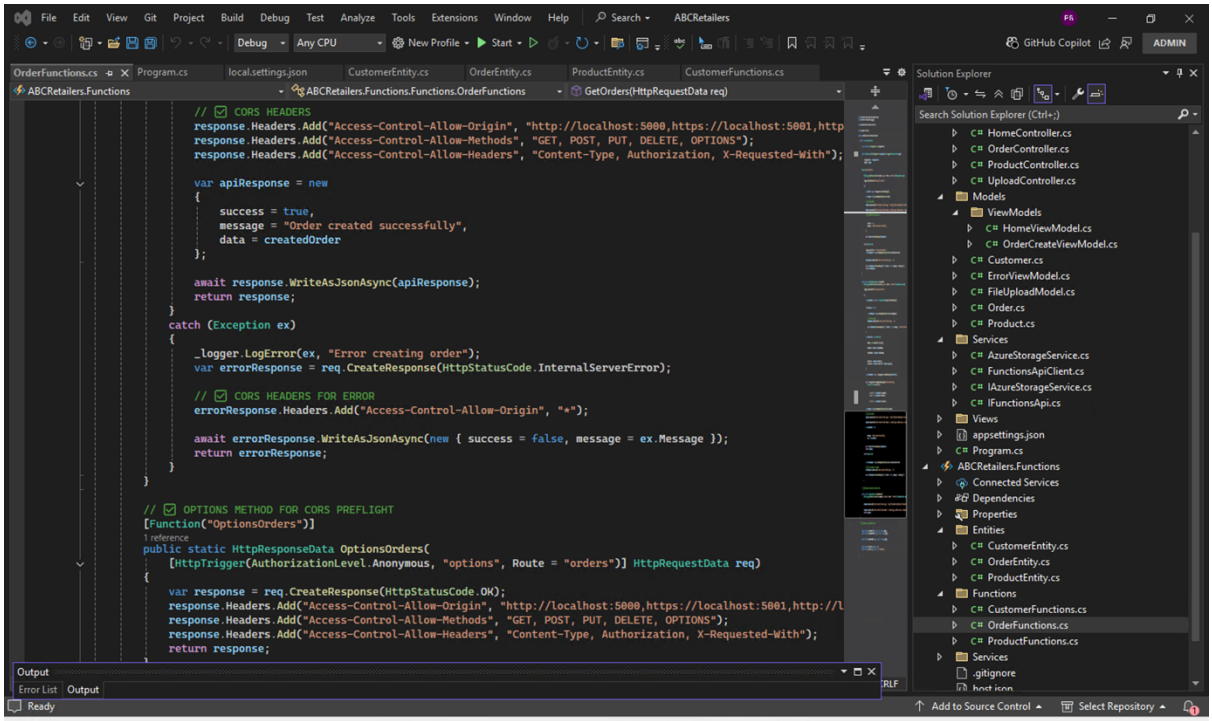

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
ProductFunctions.cs
OrderFunctions.cs
Program.cs
local.settings.json
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
ABCRetailers.Functions
ABCRetailers.Functions.Entities.ProductEntity
Timestamp
using Azure;
using Azure.Data.Tables;
namespace ABCRetailers.Functions.Entities
{
9 references
public class ProductEntity : ITableEntity
{
0 references
public string PartitionKey { get; set; } = "Product";
2 references
public string RowKey { get; set; } = Guid.NewGuid().ToString();
0 references
public DateTimeOffset? Timestamp { get; set; }
0 references
public ETag ETag { get; set; } = ETag.All;
2 references
public string ProductName { get; set; } = string.Empty;
1 reference
public string Description { get; set; } = string.Empty;
1 reference
public double Price { get; set; }
1 reference
public int StockAvailable { get; set; }
1 reference
public string ImageUrl { get; set; } = string.Empty;
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
Services
.gitignore
host.json
local.settings.json
Program.cs
Output
Error List
Output
100 %
No issues found
Ln: 19
Ch: 2
SPC
CRLF
Ready
Add to Source Control
Select Repository
Air: Poor
Saturday
Search
10:17 PM
10/8/2025

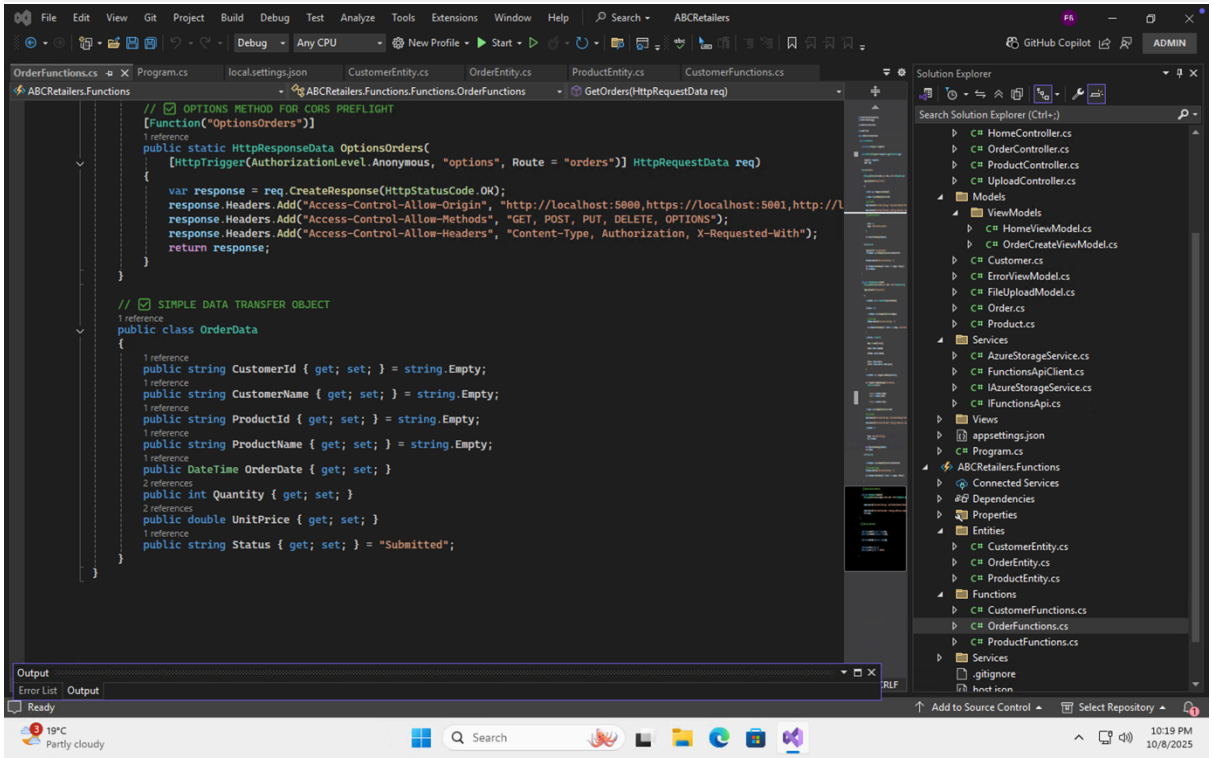

var response = req.CreateResponse(HttpStatusCode.OK);
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http://l
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
return response;
// ☑ SIMPLE DATA TRANSFER OBJECT (no models folder needed)
public class CustomerData
public string Name { get; set; } = string.Empty;
public string Surname { get; set; } = string.Empty;
public string Username { get; set; } = string.Empty;
public string Email { get; set; } = string.Empty;
public string ShippingAddress { get; set; } = string.Empty;

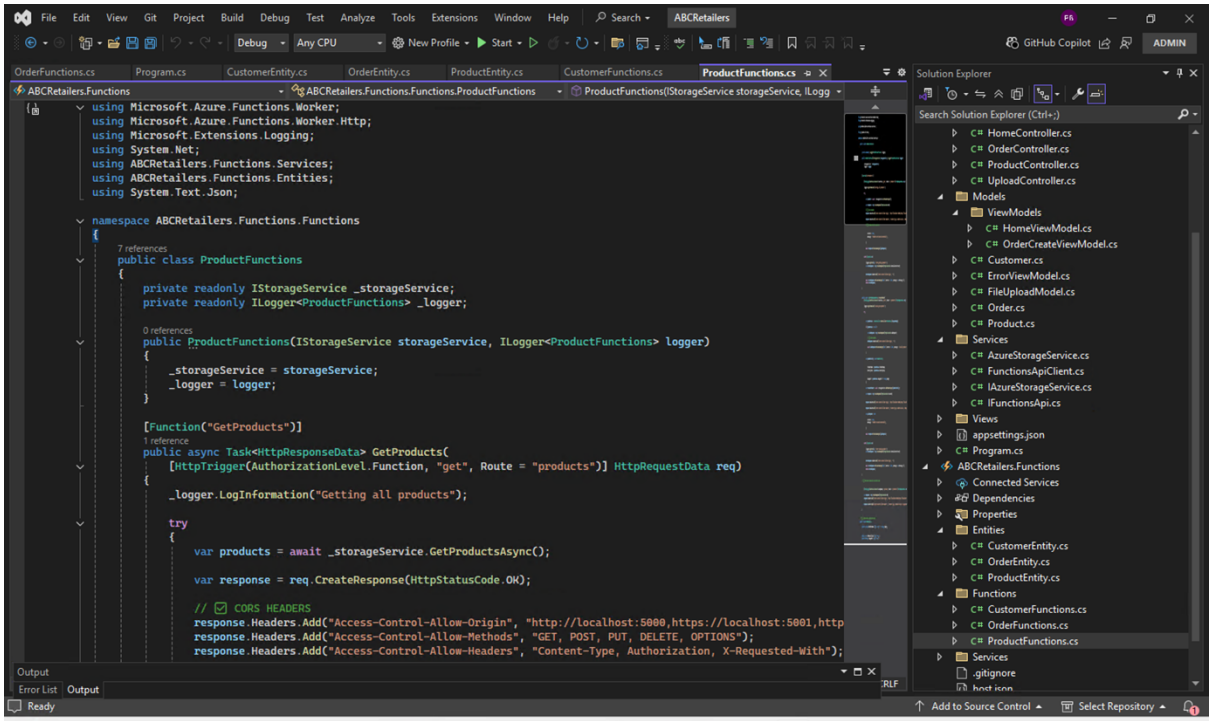

using Microsoft.Azure.Functions.Worker;
using Microsoft.Azure.Functions.Worker.Http;
using Microsoft.Extensions.Logging;
using System.Net;
using ABCRetailers.Functions.Services;
using ABCRetailers.Functions.Entities;
using System.Text.Json;
namespace ABCRetailers.Functions.Functions
public class OrderFunctions
private readonly IStorageService _storageService;
private readonly ILogger<OrderFunctions> _logger;
public OrderFunctions(IStorageService storageService, ILogger<OrderFunctions> logger)
_storageService = storageService;
_logger = logger;
[Function("GetOrders")]
public async Task<HttpResponseData> GetOrders(
[HttpTrigger(AuthorizationLevel.Function, "get", Route = "orders")] HttpRequestData req)
_logger.LogInformation("Getting all orders");
try
var orders = await _storageService.GetOrdersAsync();
var response = req.CreateResponse(HttpStatusCode.OK);
// ☑ CORS HEADERS
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");

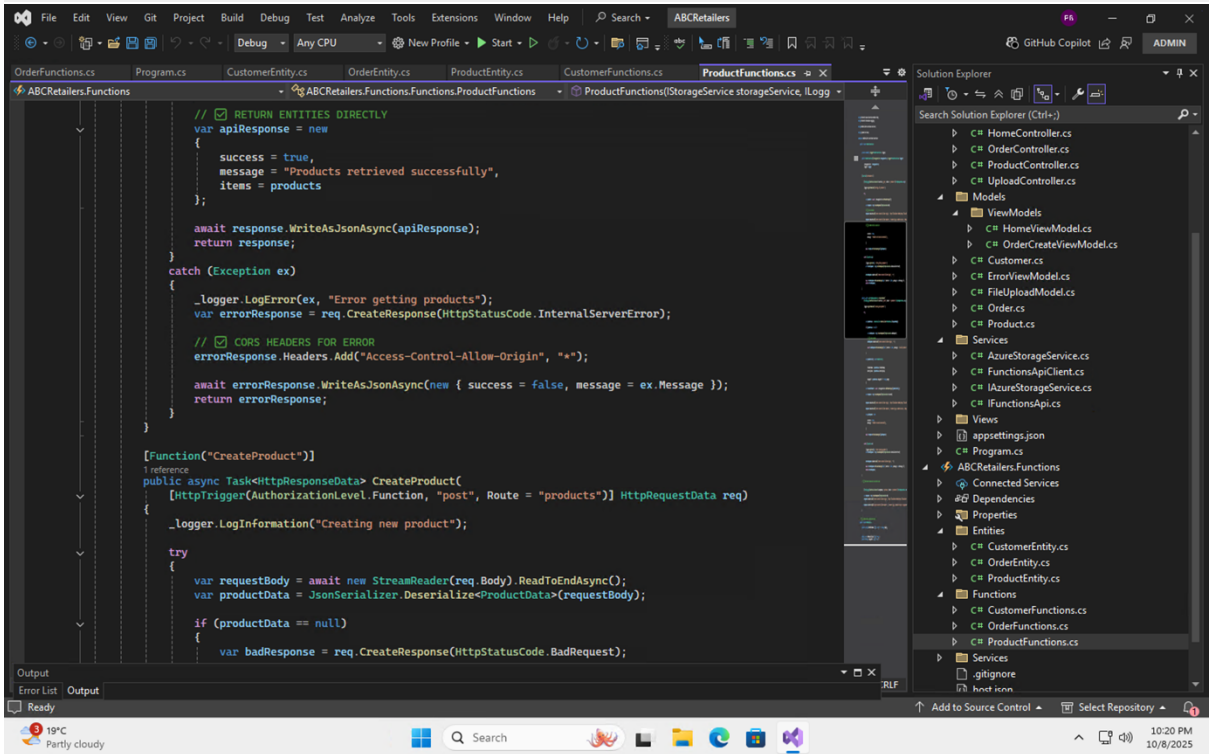

// ☑ RETURN ENTITIES DIRECTLY
var apiResponse = new
{
success = true,
message = "Orders retrieved successfully",
items = orders
};
await response.WriteAsJsonAsync(apiResponse);
return response;
}
catch (Exception ex)
{
_logger.LogError(ex, "Error getting orders");
var errorResponse = req.CreateResponse(HttpStatusCode.InternalServerError);
// ☑ CORS HEADERS FOR ERROR
errorResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await errorResponse.WriteAsJsonAsync(new { success = false, message = ex.Message });
return errorResponse;
}
}
[Function("CreateOrder")]
1 reference
public async Task<HttpResponseData> CreateOrder(
[HttpTrigger(AuthorizationLevel.Function, "post", Route = "orders")] HttpRequestData req)
{
_logger.LogInformation("Creating new order");
try
{
var requestBody = await new StreamReader(req.Body).ReadToEndAsync();
var orderData = JsonSerializer.Deserialize<OrderData>(requestBody);
if (orderData == null)
{
var badResponse = req.CreateResponse(HttpStatusCode.BadRequest);

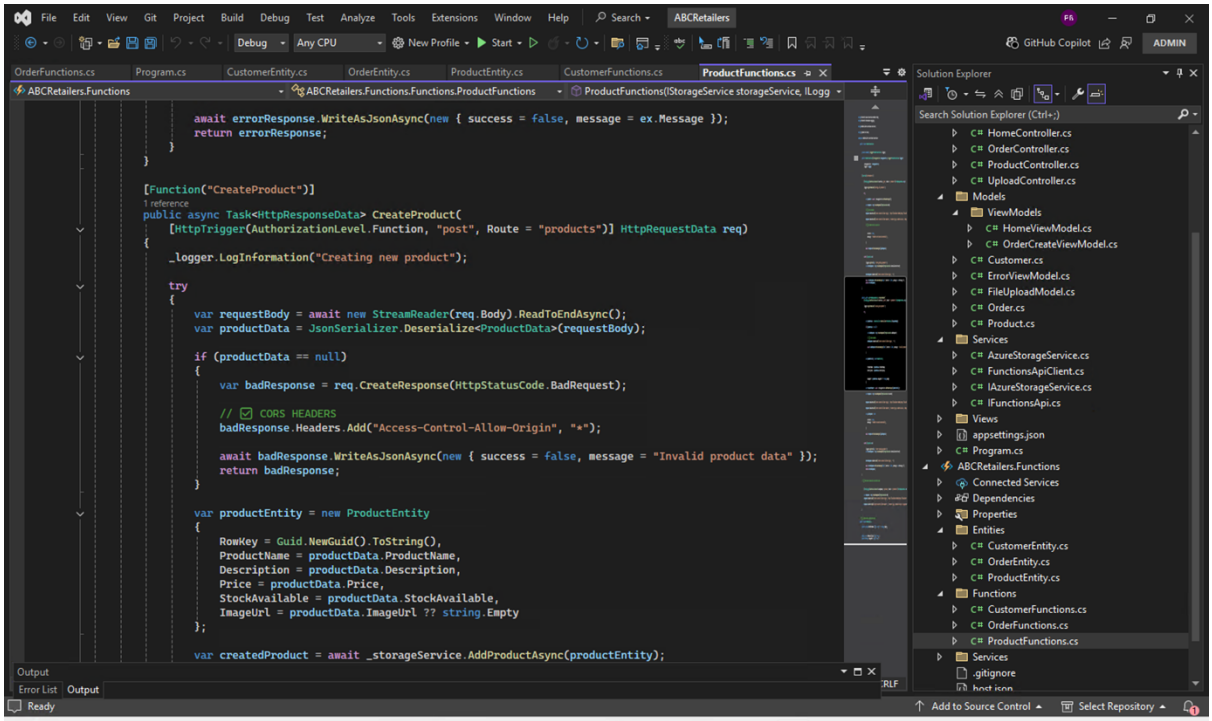

if (orderData == null)
{
var badResponse = req.CreateResponse(HttpStatusCode.BadRequest);
// ☑ CORS HEADERS
badResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await badResponse.WriteAsJsonAsync(new { success = false, message = "Invalid order data" });
return badResponse;
}
var orderEntity = new OrderEntity
{
RowKey = Guid.NewGuid().ToString(),
CustomerId = orderData.CustomerId,
Username = orderData.CustomerName,
ProductId = orderData.ProductId,
ProductName = orderData.ProductName,
OrderDate = orderData.OrderDate,
Quantity = orderData.Quantity,
UnitPrice = orderData.UnitPrice,
TotalPrice = orderData.UnitPrice * orderData.Quantity,
Status = orderData.Status
};
var createdOrder = await _storageService.AddOrderAsync(orderEntity);
// Send to processing queue
await _storageService.SendQueueMessageAsync("order-processing",
JsonSerializer.Serialize(new
{
OrderId = createdOrder.RowKey,
CustomerId = createdOrder.CustomerId,
ProductId = createdOrder.ProductId,
Quantity = createdOrder.Quantity,
TotalPrice = createdOrder.TotalPrice
}));
var response = req.CreateResponse(HttpStatusCode.Created);

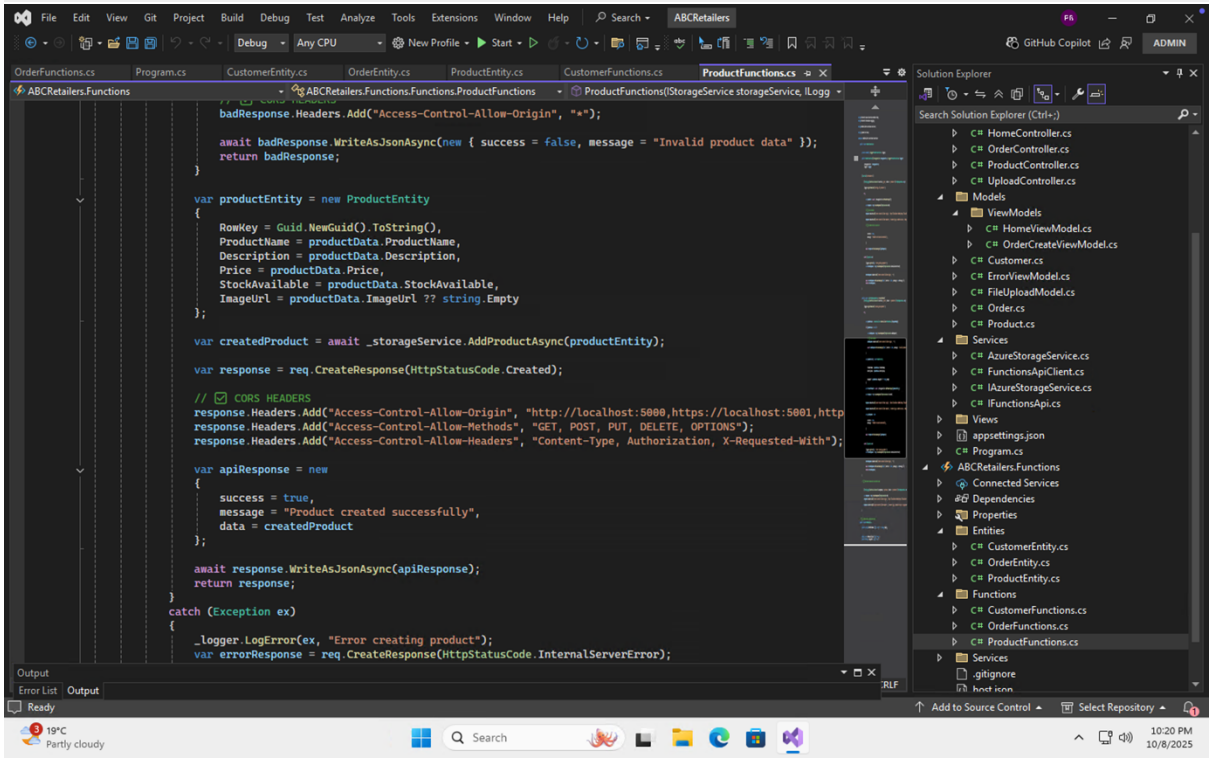

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
ADMIN
OrderFunctions.cs
Program.cs
local.settings.json
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.OrderFunctions
GetOrders(HttpRequestData req)
// ☑ CORS HEADERS
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
var apiResponse = new
{
success = true,
message = "Order created successfully",
data = createdOrder
};
await response.WriteAsJsonAsync(apiResponse);
return response;
}
catch (Exception ex)
{
_logger.LogError(ex, "Error creating order");
var errorResponse = req.CreateResponse(HttpStatusCode.InternalServerError);
// ☑ CORS HEADERS FOR ERROR
errorResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await errorResponse.WriteAsJsonAsync(new { success = false, message = ex.Message });
return errorResponse;
}
}
// ☑ OPTIONS METHOD FOR CORS PREFLIGHT
[Function("OptionsOrders")]
1 reference
public static HttpResponseData OptionsOrders(
[HttpTrigger(AuthorizationLevel.Anonymous, "options", Route = "orders")] HttpRequestData req)
{
var response = req.CreateResponse(HttpStatusCode.OK);
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http://l
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
return response;
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
Output
Error List
Output
Ready
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:19 PM
10/8/2025

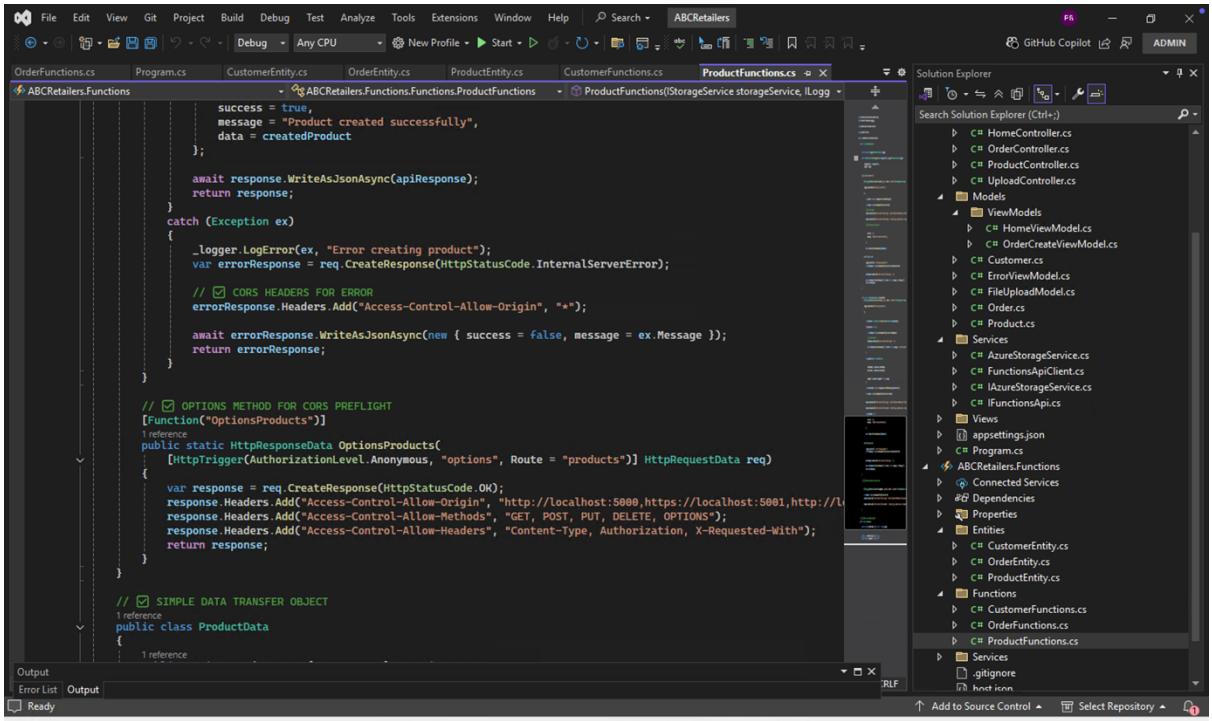

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
ADMIN
OrderFunctions.cs
Program.cs
local.settings.json
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.OrderFunctions
GetOrders(HttpRequestData req)
// ☑ OPTIONS METHOD FOR CORS PREFLIGHT
[Function("OptionsOrders")]
1 reference
public static HttpResponseData OptionsOrders(
[HttpTrigger(AuthorizationLevel.Anonymous, "options", Route = "orders")] HttpRequestData req)
{
var response = req.CreateResponse(HttpStatusCode.OK);
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http://l
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
return response;
}
}
// ☑ SIMPLE DATA TRANSFER OBJECT
1 reference
public class OrderData
{
1 reference
public string CustomerId { get; set; } = string.Empty;
1 reference
public string CustomerName { get; set; } = string.Empty;
1 reference
public string ProductId { get; set; } = string.Empty;
1 reference
public string ProductName { get; set; } = string.Empty;
1 reference
public DateTime OrderDate { get; set; }
2 references
public int Quantity { get; set; }
2 references
public double UnitPrice { get; set; }
1 reference
public string Status { get; set; } = "Submitted";
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
Output
Error List
Output
Ready
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:19 PM
10/8/2025

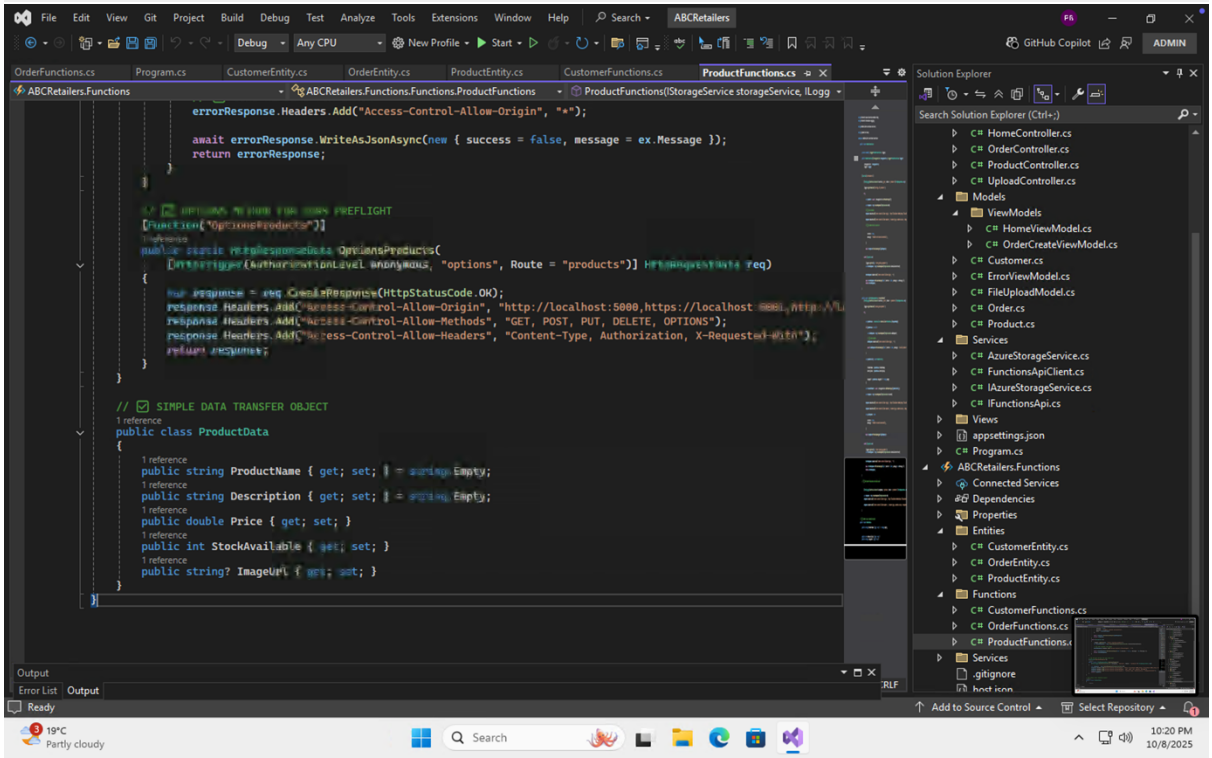

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
Program.cs
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.ProductFunctions
ProductFunctions(IStorageService storageService, ILogg
using Microsoft.Azure.Functions.Worker;
using Microsoft.Azure.Functions.Worker.Http;
using Microsoft.Extensions.Logging;
using System.Net;
using ABCRetailers.Functions.Services;
using ABCRetailers.Functions.Entities;
using System.Text.Json;
namespace ABCRetailers.Functions.Functions
{
7 references
public class ProductFunctions
{
private readonly IStorageService _storageService;
private readonly ILogger<ProductFunctions> _logger;
0 references
public ProductFunctions(IStorageService storageService, ILogger<ProductFunctions> logger)
{
_storageService = storageService;
_logger = logger;
}
[Function("GetProducts")]
1 reference
public async Task<HttpResponseData> GetProducts(
[HttpTrigger(AuthorizationLevel.Function, "get", Route = "products")] HttpRequestData req)
{
_logger.LogInformation("Getting all products");
try
{
var products = await _storageService.GetProductsAsync();
var response = req.CreateResponse(HttpStatusCode.OK);
// ☑ CORS HEADERS
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
Output
Error List
Output
Ready
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
host.json
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:20 PM
10/8/2025

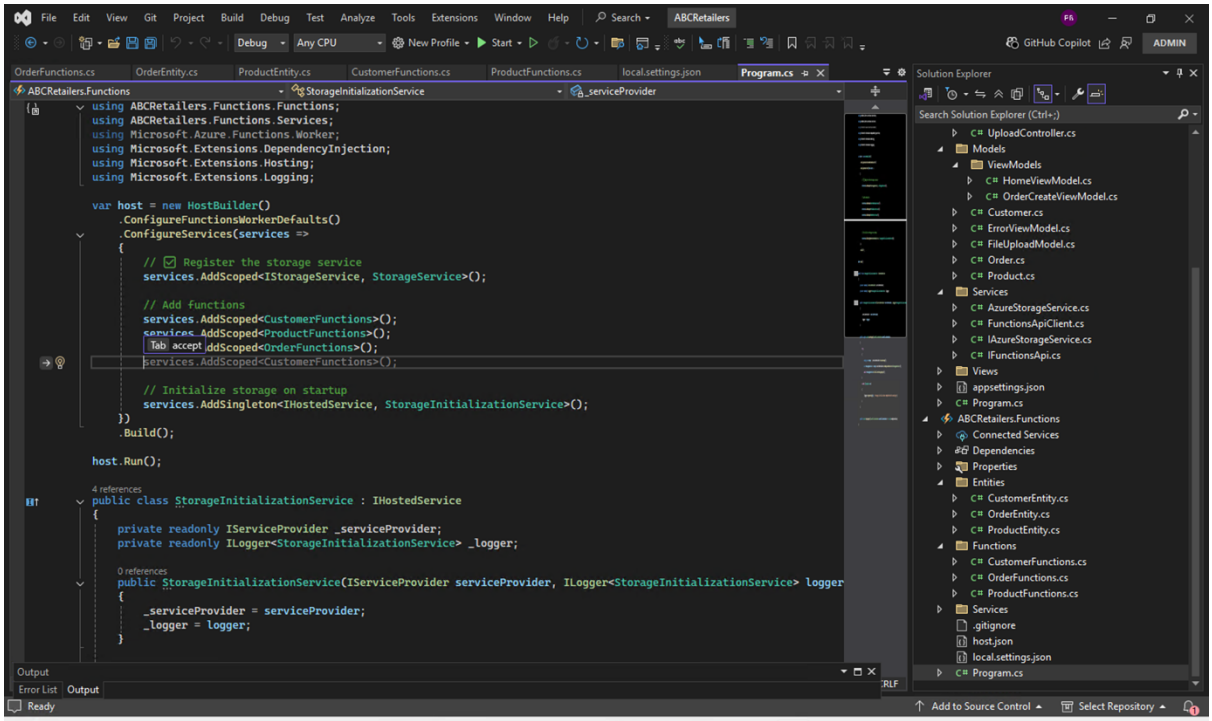

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
Program.cs
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.ProductFunctions
ProductFunctions(IStorageService storageService, ILogg
// ☑ RETURN ENTITIES DIRECTLY
var apiResponse = new
{
success = true,
message = "Products retrieved successfully",
items = products
};
await response.WriteAsJsonAsync(apiResponse);
return response;
}
catch (Exception ex)
{
_logger.LogError(ex, "Error getting products");
var errorResponse = req.CreateResponse(HttpStatusCode.InternalServerError);
// ☑ CORS HEADERS FOR ERROR
errorResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await errorResponse.WriteAsJsonAsync(new { success = false, message = ex.Message });
return errorResponse;
}
}
[Function("CreateProduct")]
1 reference
public async Task<HttpResponseData> CreateProduct(
[HttpTrigger(AuthorizationLevel.Function, "post", Route = "products")] HttpRequestData req)
{
_logger.LogInformation("Creating new product");
try
{
var requestBody = await new StreamReader(req.Body).ReadToEndAsync();
var productData = JsonSerializer.Deserialize<ProductData>(requestBody);
if (productData == null)
{
var badResponse = req.CreateResponse(HttpStatusCode.BadRequest);
Output
Error List
Output
Ready
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
host.json
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:20 PM
10/8/2025

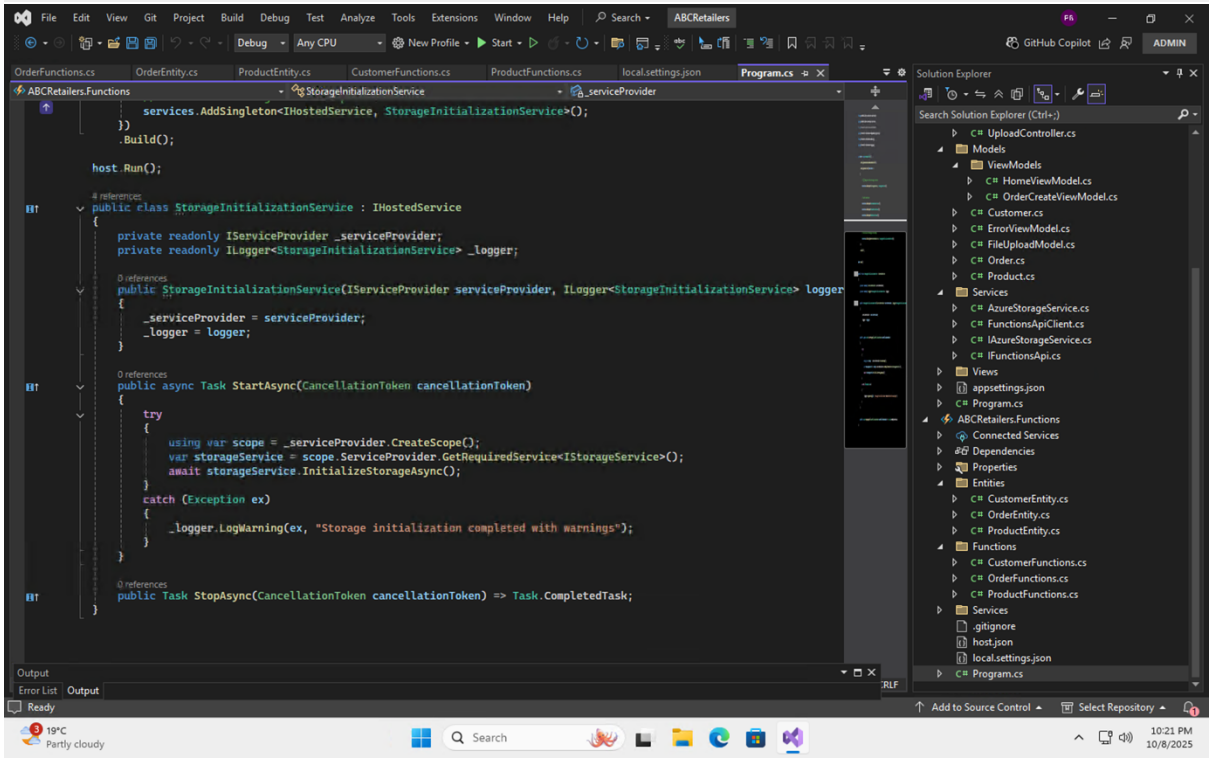

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
Program.cs
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.ProductFunctions
ProductFunctions(IStorageService storageService, ILogg
await errorResponse.WriteAsJsonAsync(new { success = false, message = ex.Message });
return errorResponse;
}
}
[Function("CreateProduct")]
1 reference
public async Task<HttpResponseData> CreateProduct(
[HttpTrigger(AuthorizationLevel.Function, "post", Route = "products")] HttpRequestData req)
{
_logger.LogInformation("Creating new product");
try
{
var requestBody = await new StreamReader(req.Body).ReadToEndAsync();
var productData = JsonSerializer.Deserialize<ProductData>(requestBody);
if (productData == null)
{
var badResponse = req.CreateResponse(HttpStatusCode.BadRequest);
// ☑ CORS HEADERS
badResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await badResponse.WriteAsJsonAsync(new { success = false, message = "Invalid product data" });
return badResponse;
}
var productEntity = new ProductEntity
{
RowKey = Guid.NewGuid().ToString(),
ProductName = productData.ProductName,
Description = productData.Description,
Price = productData.Price,
StockAvailable = productData.StockAvailable,
ImageUrl = productData.ImageUrl ?? string.Empty
};
var createdProduct = await _storageService.AddProductAsync(productEntity);
Output
Error List
Output
Ready
Add to Source Control
Select Repository
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
host.json
19°C
Partly cloudy
Search
10:20 PM
10/8/2025

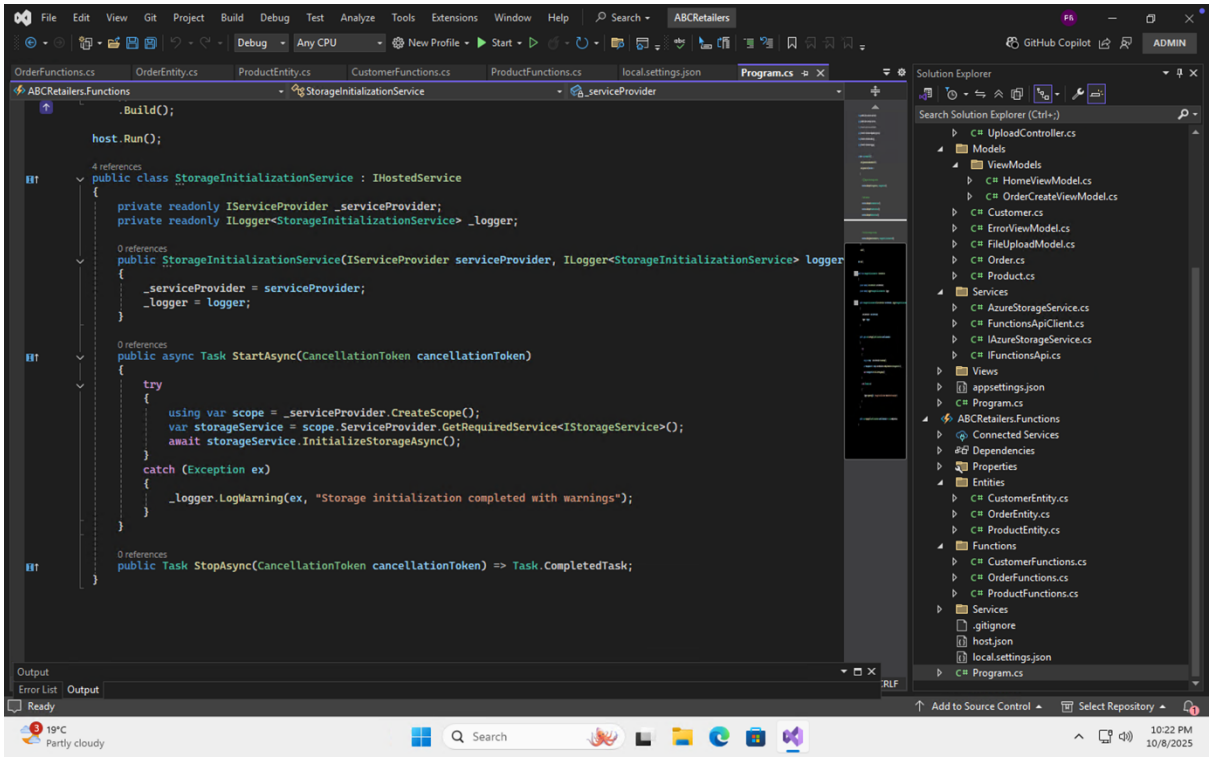

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
Program.cs
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.ProductFunctions
ProductFunctions(IStorageService storageService, ILogg
badResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await badResponse.WriteAsJsonAsync(new { success = false, message = "Invalid product data" });
return badResponse;
}
var productEntity = new ProductEntity
{
RowKey = Guid.NewGuid().ToString(),
ProductName = productData.ProductName,
Description = productData.Description,
Price = productData.Price,
StockAvailable = productData.StockAvailable,
ImageUrl = productData.ImageUrl ?? string.Empty
};
var createdProduct = await _storageService.AddProductAsync(productEntity);
var response = req.CreateResponse(HttpStatusCode.Created);
// ☑ CORS HEADERS
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
var apiResponse = new
{
success = true,
message = "Product created successfully",
data = createdProduct
};
await response.WriteAsJsonAsync(apiResponse);
return response;
}
catch (Exception ex)
{
_logger.LogError(ex, "Error creating product");
var errorResponse = req.CreateResponse(HttpStatusCode.InternalServerError);
Output
Error List
Output
Ready
Add to Source Control
Select Repository
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
host.json
19°C
Partly cloudy
Search
10:20 PM
10/8/2025

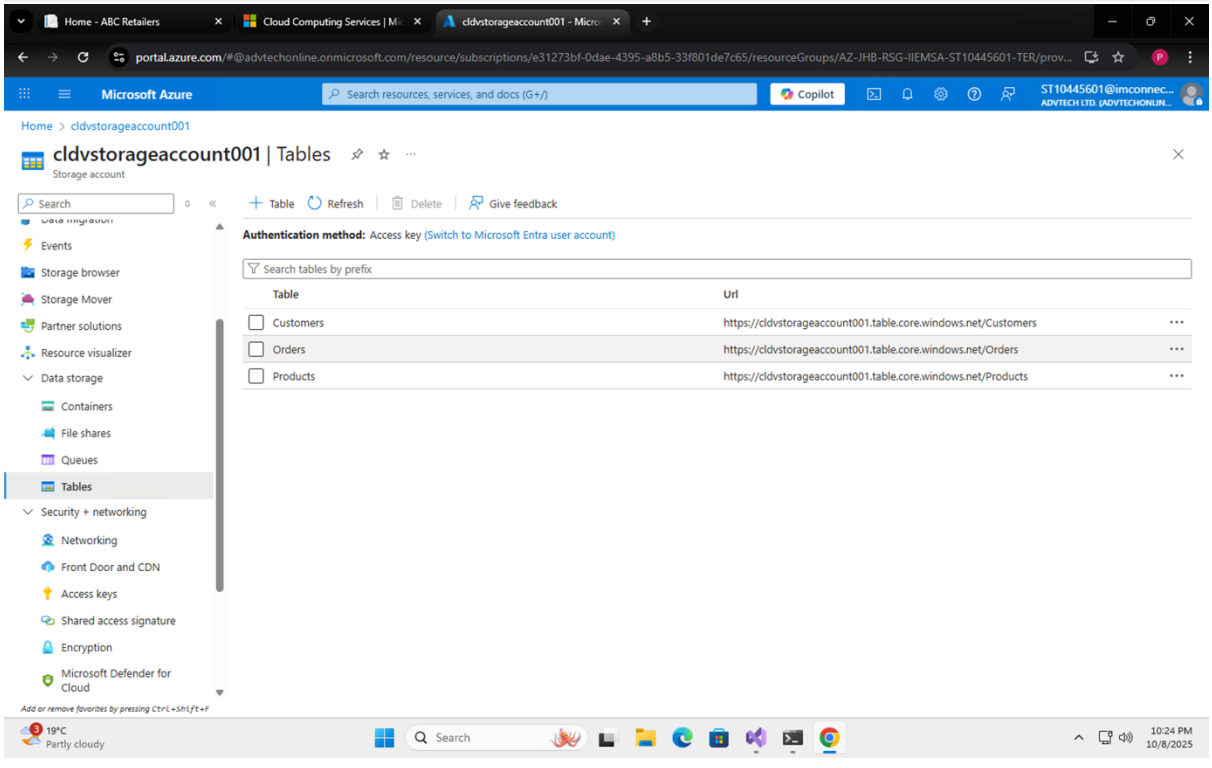

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
Program.cs
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.ProductFunctions
ProductFunctions(IStorageService storageService, ILogg
success = true,
message = "Product created successfully",
data = createdProduct
};
await response.WriteAsJsonAsync(apiResponse);
return response;
}
catch (Exception ex)
{
_logger.LogError(ex, "Error creating product");
var errorResponse = req.CreateResponse(HttpStatusCode.InternalServerError);
// ☑ CORS HEADERS FOR ERROR
errorResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await errorResponse.WriteAsJsonAsync(new { success = false, message = ex.Message });
return errorResponse;
}
}
// ☑ OPTIONS METHOD FOR CORS PREFLIGHT
[Function("OptionsProducts")]
1 reference
public static HttpResponseData OptionsProducts(
[HttpTrigger(AuthorizationLevel.Anonymous, "options", Route = "products")] HttpRequestData req)
{
var response = req.CreateResponse(HttpStatusCode.OK);
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost:5001,http://l
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
return response;
}
}
// ☑ SIMPLE DATA TRANSFER OBJECT
1 reference
public class ProductData
{
1 reference
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
Output
Error List
Output
Ready
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:20 PM
10/8/2025

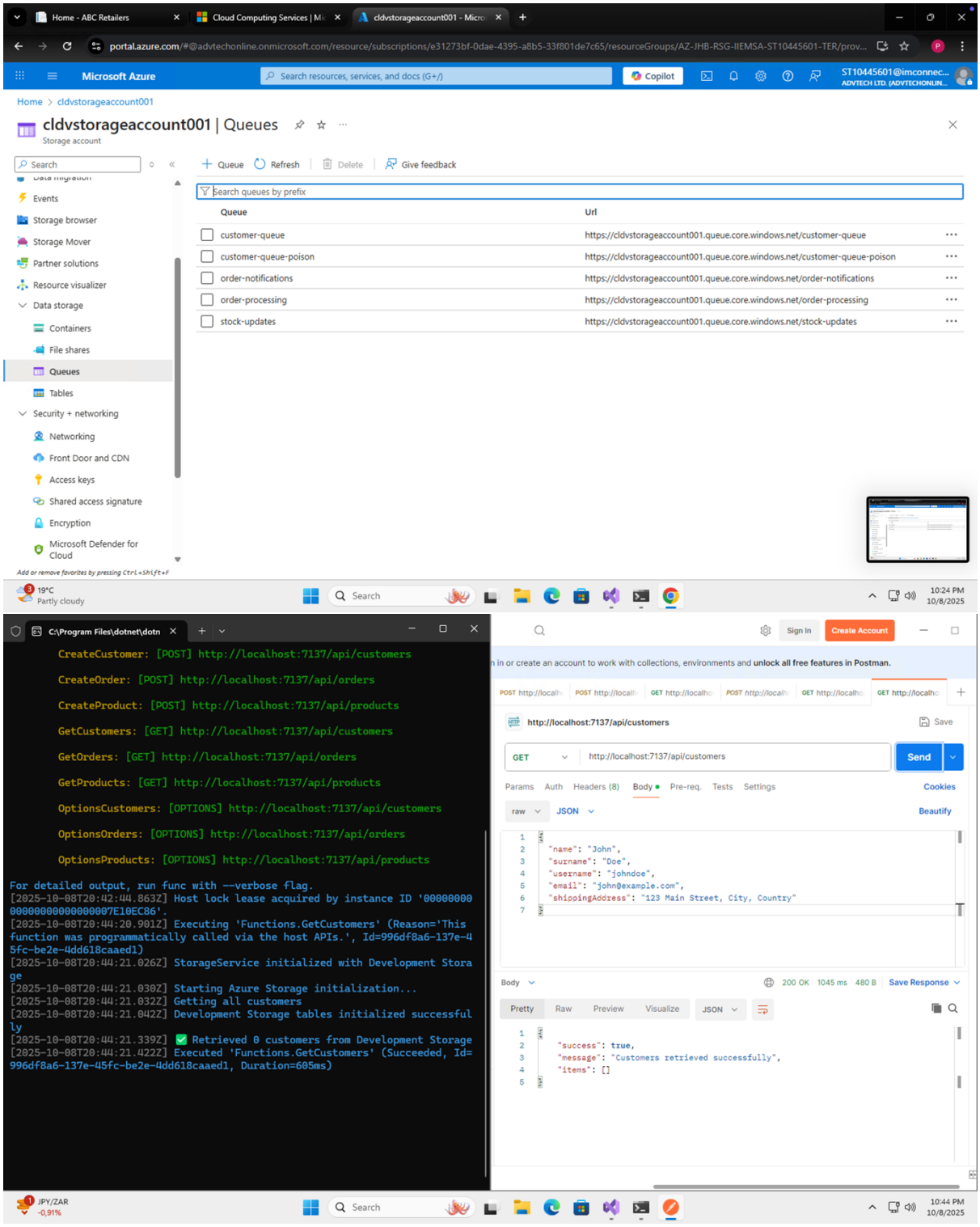

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
Program.cs
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
ABCRetailers.Functions
ABCRetailers.Functions.Functions.ProductFunctions
ProductFunctions(IStorageService storageService, ILogg
errorResponse.Headers.Add("Access-Control-Allow-Origin", "*");
await errorResponse.WriteAsJsonAsync(new { success = false, message = ex.Message });
return errorResponse;
}
}
// ☑ OPTIONS METHOD FOR CORS PREFLIGHT
[Function("OptionsProducts")]
public static HttpResponseData OptionsProducts(
[HttpTrigger(AuthorizationLevel.Anonymous, "options", Route = "products")] HttpRequestData req)
{
var response = req.CreateResponse(HttpStatusCode.OK);
response.Headers.Add("Access-Control-Allow-Origin", "http://localhost:5000,https://localhost
response.Headers.Add("Access-Control-Allow-Methods", "GET, POST, PUT, DELETE, OPTIONS");
response.Headers.Add("Access-Control-Allow-Headers", "Content-Type, Authorization, X-Requested-With");
return response;
}
}
// ☑ SIMPLE DATA TRANSFER OBJECT
1 reference
public class ProductData
{
1 reference
public string ProductName { get; set; } = string.Empty;
1 reference
public string Description { get; set; } = string.Empty;
1 reference
public double Price { get; set; }
1 reference
public int StockAvailable { get; set; }
1 reference
public string? ImageUrl { get; set; }
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
HomeController.cs
OrderController.cs
ProductController.cs
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
Output
Error List
Output
Ready
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:20 PM
10/8/2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
local.settings.json
Program.cs
ABCRetailers.Functions
StorageInitializationService
_serviceProvider
using ABCRetailers.Functions.Functions;
using ABCRetailers.Functions.Services;
using Microsoft.Azure.Functions.Worker;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
var host = new HostBuilder()
.ConfigureFunctionsWorkerDefaults()
.ConfigureServices(services =>
{
// Register the storage service
services.AddScoped<IStorageService, StorageService>();
// Add functions
services.AddScoped<CustomerFunctions>();
services.AddScoped<ProductFunctions>();
Tab accept ddScoped<OrderFunctions>();
services.AddScoped<CustomerFunctions>();
// Initialize storage on startup
services.AddSingleton<IHostedService, StorageInitializationService>();
})
.Build();
host.Run();
4 references
public class StorageInitializationService : IHostedService
{
private readonly IServiceProvider _serviceProvider;
private readonly ILogger<StorageInitializationService> _logger;
0 references
public StorageInitializationService(IServiceProvider serviceProvider, ILogger<StorageInitializationService> logger
{
_serviceProvider = serviceProvider;
_logger = logger;
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
host.json
local.settings.json
Program.cs
Output
Error List
Output
Ready
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:21 PM
10/8/2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
Debug
Any CPU
New Profile
Start
GitHub Copilot
ADMIN
OrderFunctions.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
local.settings.json
Program.cs
ABCRetailers.Functions
StorageInitializationService
_serviceProvider
services.AddSingleton<IHostedService, StorageInitializationService>();
})
.Build();
host.Run();
4 references
public class StorageInitializationService : IHostedService
{
private readonly IServiceProvider _serviceProvider;
private readonly ILogger<StorageInitializationService> _logger;
0 references
public StorageInitializationService(IServiceProvider serviceProvider, ILogger<StorageInitializationService> logger
{
_serviceProvider = serviceProvider;
_logger = logger;
}
0 references
public async Task StartAsync(CancellationToken cancellationToken)
{
try
{
using var scope = _serviceProvider.CreateScope();
var storageService = scope.ServiceProvider.GetRequiredService<IStorageService>();
await storageService.InitializeStorageAsync();
}
catch (Exception ex)
{
_logger.LogWarning(ex, "Storage initialization completed with warnings");
}
}
0 references
public Task StopAsync(CancellationToken cancellationToken) => Task.CompletedTask;
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
host.json
local.settings.json
Program.cs
Output
Error List
Output
Ready
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:21 PM
10/8/2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
ABCRetailers
ADMIN
OrderFunctions.cs
OrderEntity.cs
ProductEntity.cs
CustomerFunctions.cs
ProductFunctions.cs
local.settings.json
Program.cs
ABCRetailers.Functions
StorageInitializationService
_serviceProvider
.Build();
host.Run();
4 references
public class StorageInitializationService : IHostedService
{
private readonly IServiceProvider _serviceProvider;
private readonly ILogger<StorageInitializationService> _logger;
0 references
public StorageInitializationService(IServiceProvider serviceProvider, ILogger<StorageInitializationService> logger
{
_serviceProvider = serviceProvider;
_logger = logger;
}
0 references
public async Task StartAsync(CancellationToken cancellationToken)
{
try
{
using var scope = _serviceProvider.CreateScope();
var storageService = scope.ServiceProvider.GetRequiredService<IStorageService>();
await storageService.InitializeStorageAsync();
}
catch (Exception ex)
{
_logger.LogWarning(ex, "Storage initialization completed with warnings");
}
}
0 references
public Task StopAsync(CancellationToken cancellationToken) => Task.CompletedTask;
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
UploadController.cs
Models
ViewModels
HomeViewModel.cs
OrderCreateViewModel.cs
Customer.cs
ErrorViewModel.cs
FileUploadModel.cs
Order.cs
Product.cs
Services
AzureStorageService.cs
FunctionsApiClient.cs
IAzureStorageService.cs
IFunctionsApi.cs
Views
appsettings.json
Program.cs
ABCRetailers.Functions
Connected Services
Dependencies
Properties
Entities
CustomerEntity.cs
OrderEntity.cs
ProductEntity.cs
Functions
CustomerFunctions.cs
OrderFunctions.cs
ProductFunctions.cs
Services
.gitignore
host.json
local.settings.json
Program.cs
Output
Error List
Output
Ready
Add to Source Control
Select Repository
19°C
Partly cloudy
Search
10:22 PM
10/8/2025


Home - ABC Retailers
Cloud Computing Services | M
cldvstorageaccount001 - Micro
portal.azure.com/#@advtechonline.onmicrosoft.com/resource/subscriptions/e31273bf-0dae-4395-a8b5-33f801de7c65/resourceGroups/AZ-JHB-RSG-IIEMSA-ST10445601-TER/prov...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
ST10445601@imconnec...
ADVTECH LTD. (ADVTECHONLIN...
Home > cldvstorageaccount001
cldvstorageaccount001 | Tables
Storage account
Search
Table
Refresh
Delete
Give feedback
Authentication method: Access key (Switch to Microsoft Entra user account)
Search tables by prefix
Table
Url
Customers
https://cldvstorageaccount001.table.core.windows.net/Customers
Orders
https://cldvstorageaccount001.table.core.windows.net/Orders
Products
https://cldvstorageaccount001.table.core.windows.net/Products
Events
Storage browser
Storage Mover
Partner solutions
Resource visualizer
Data storage
Containers
File shares
Queues
Tables
Security + networking
Networking
Front Door and CDN
Access keys
Shared access signature
Encryption
Microsoft Defender for Cloud
Add or remove favorites by pressing Ctrl+Shift+F
19°C
Partly cloudy
Search
10:24 PM
10/8/2025


Home - ABC Retailers
Cloud Computing Services | Mi
cldvstorageaccount001 - Micro
portal.azure.com/#@advtechonline.onmicrosoft.com/resource/subscriptions/e31273bf-0dae-4395-a8b5-33f801de7c65/resourceGroups/AZ-JHB-RSG-IIEMSA-ST10445601-TER/prov...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
ST10445601@imconnec...
ADVTECH LTD. (ADVTECHONLIN...
Home > cldvstorageaccount001
cldvstorageaccount001 | Queues
Storage account
Search
Events
Storage browser
Storage Mover
Partner solutions
Resource visualizer
Data storage
Containers
File shares
Queues
Tables
Security + networking
Networking
Front Door and CDN
Access keys
Shared access signature
Encryption
Microsoft Defender for Cloud
Add or remove favorites by pressing Ctrl+Shift+F
Queue
Refresh
Delete
Give feedback
Search queues by prefix
Queue
Url
customer-queue
https://cldvstorageaccount001.queue.core.windows.net/customer-queue
customer-queue-poison
https://cldvstorageaccount001.queue.core.windows.net/customer-queue-poison
order-notifications
https://cldvstorageaccount001.queue.core.windows.net/order-notifications
order-processing
https://cldvstorageaccount001.queue.core.windows.net/order-processing
stock-updates
https://cldvstorageaccount001.queue.core.windows.net/stock-updates
19°C
Partly cloudy
Search
10:24 PM
10/8/2025
C:\Program Files\dotnet\dotn
CreateCustomer: [POST] http://localhost:7137/api/customers
CreateOrder: [POST] http://localhost:7137/api/orders
CreateProduct: [POST] http://localhost:7137/api/products
GetCustomers: [GET] http://localhost:7137/api/customers
GetOrders: [GET] http://localhost:7137/api/orders
GetProducts: [GET] http://localhost:7137/api/products
OptionsCustomers: [OPTIONS] http://localhost:7137/api/customers
OptionsOrders: [OPTIONS] http://localhost:7137/api/orders
OptionsProducts: [OPTIONS] http://localhost:7137/api/products
For detailed output, run func with --verbose flag.
[2025-10-08T20:42:44.863Z] Host lock lease acquired by instance ID '00000000000000000000007E10EC86'.
[2025-10-08T20:44:20.901Z] Executing 'Functions.GetCustomers' (Reason='This function was programmatically called via the host APIs.', Id=996df8a6-137e-45fc-be2e-4dd618caaed1)
[2025-10-08T20:44:21.026Z] StorageService initialized with Development Storage
[2025-10-08T20:44:21.030Z] Starting Azure Storage initialization...
[2025-10-08T20:44:21.032Z] Getting all customers
[2025-10-08T20:44:21.042Z] Development Storage tables initialized successfully
[2025-10-08T20:44:21.339Z] Retrieved 0 customers from Development Storage
[2025-10-08T20:44:21.422Z] Executed 'Functions.GetCustomers' (Succeeded, Id=996df8a6-137e-45fc-be2e-4dd618caaed1, Duration=605ms)
Sign In
Create Account
unlock all free features in Postman.
http://localhost:7137/api/customers
Save
GET
http://localhost:7137/api/customers
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
{
"name": "John",
"surname": "Doe",
"username": "johndoe",
"email": "john@example.com",
"shippingAddress": "123 Main Street, City, Country"
}
Body
200 OK
1045 ms
480 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"success": true,
"message": "Customers retrieved successfully",
"items": []
}
JPY/ZAR
-0,91%
Search
10:44 PM
10/8/2025