**🏗 Step 1: Create the Azure Function Project**

In Visual Studio 2022:

1. **Create a new project → Azure Functions**
2. Project Name: PolicyQueueFunction
3. **.NET 8 (Isolated Worker)**
4. Function type: **Queue Trigger**
5. Storage Emulator: select Use Azure Storage Emulator or connect your real Azure Storage Account.

**📁 Step 2: Project File (.csproj)**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<TargetFramework>net8.0</TargetFramework>

<AzureFunctionsVersion>v4</AzureFunctionsVersion>

<OutputType>Exe</OutputType>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.Azure.Functions.Worker.Sdk" Version="1.17.4" OutputItemType="Analyzer" />

<PackageReference Include="Microsoft.Azure.Functions.Worker" Version="1.22.0" />

<PackageReference Include="Microsoft.Azure.Functions.Worker.Extensions.Storage.Queues" Version="5.3.0" />

<PackageReference Include="Microsoft.Azure.Cosmos" Version="3.39.1" />

</ItemGroup>

</Project>

**⚙️ Step 1: Folder Structure**

**PolicyQueueFunction/**

**├── Functions/**

**│ ├── QueueProducer.cs 👈 HTTP-triggered producer**

**│ ├── QueueListener.cs 👈 Queue-triggered consumer**

**├── Models/**

**│ └── PolicyHolder.cs**

**├── Program.cs**

**├── local.settings.json**

**└── PolicyQueueFunction.csproj**

**🧩 Step 2: Model**

**Models/PolicyHolder.cs**

**namespace PolicyQueueFunction.Models**

**{**

**public class PolicyHolder**

**{**

**public string Id { get; set; } = Guid.NewGuid().ToString();**

**public string Name { get; set; }**

**public string PolicyNumber { get; set; }**

**public DateTime StartDate { get; set; }**

**public decimal PremiumAmount { get; set; }**

**}**

**}**

**🚀 Step 3: Queue Producer Function (adds message)**

**Functions/QueueProducer.cs**

**using Microsoft.Azure.Functions.Worker;**

**using Microsoft.Azure.Functions.Worker.Http;**

**using Microsoft.Extensions.Logging;**

**using PolicyQueueFunction.Models;**

**using System.Net;**

**using System.Text.Json;**

**namespace PolicyQueueFunction.Functions**

**{**

**public class QueueProducer**

**{**

**private readonly ILogger<QueueProducer> \_logger;**

**public QueueProducer(ILogger<QueueProducer> logger)**

**{**

**\_logger = logger;**

**}**

**[Function("AddToQueue")]**

**public async Task<HttpResponseData> Run(**

**[HttpTrigger(AuthorizationLevel.Function, "post")] HttpRequestData req,**

**[Queue("policy-queue", Connection = "AzureWebJobsStorage")] IAsyncCollector<string> queueCollector)**

**{**

**\_logger.LogInformation("HTTP trigger received a request to add a message.");**

**var body = await new StreamReader(req.Body).ReadToEndAsync();**

**var policy = JsonSerializer.Deserialize<PolicyHolder>(body);**

**if (policy == null)**

**{**

**var badResponse = req.CreateResponse(HttpStatusCode.BadRequest);**

**await badResponse.WriteStringAsync("Invalid PolicyHolder payload.");**

**return badResponse;**

**}**

**// Serialize and add message to queue**

**string message = JsonSerializer.Serialize(policy);**

**await queueCollector.AddAsync(message);**

**\_logger.LogInformation("Message added to queue: {PolicyNumber}", policy.PolicyNumber);**

**var response = req.CreateResponse(HttpStatusCode.OK);**

**await response.WriteStringAsync($"Policy {policy.PolicyNumber} added to queue successfully.");**

**return response;**

**}**

**}**

**}**

**✅ This function:**

* **Listens on HTTP POST**
* **Accepts a JSON body**
* **Adds serialized content to policy-queue**

**🧠 Step 4: Queue Listener (consumer)**

**Functions/QueueListener.cs**

**using Microsoft.Azure.Functions.Worker;**

**using Microsoft.Extensions.Logging;**

**using PolicyQueueFunction.Models;**

**using System.Text.Json;**

**namespace PolicyQueueFunction.Functions**

**{**

**public class QueueListener**

**{**

**private readonly ILogger<QueueListener> \_logger;**

**public QueueListener(ILogger<QueueListener> logger)**

**{**

**\_logger = logger;**

**}**

**[Function("ProcessPolicyQueue")]**

**public void Run(**

**[QueueTrigger("policy-queue", Connection = "AzureWebJobsStorage")] string queueMessage)**

**{**

**\_logger.LogInformation("Queue message received at: {time}", DateTime.UtcNow);**

**\_logger.LogInformation("Raw message: {msg}", queueMessage);**

**try**

**{**

**var policy = JsonSerializer.Deserialize<PolicyHolder>(queueMessage);**

**\_logger.LogInformation("Processed Policy: {PolicyNumber}, Name: {Name}",**

**policy?.PolicyNumber, policy?.Name);**

**}**

**catch (Exception ex)**

**{**

**\_logger.LogError(ex, "Error while processing message.");**

**}**

**}**

**}**

**}**

**⚙️ Step 5: local.settings.json**

**{**

**"IsEncrypted": false,**

**"Values": {**

**"AzureWebJobsStorage": "UseDevelopmentStorage=true",**

**"FUNCTIONS\_WORKER\_RUNTIME": "dotnet-isolated"**

**}**

**}**

**If you’re using a real Azure Storage account, replace "UseDevelopmentStorage=true" with your connection string.**

**🧰 Step 6: Program.cs**

**using Microsoft.Extensions.Hosting;**

**var host = new HostBuilder()**

**.ConfigureFunctionsWorkerDefaults()**

**.Build();**

**await host.RunAsync();**

**🧪 Step 7: Run Locally**

**Start your function host:**

**func start**

**You’ll see:**

**Found functions:**

**- AddToQueue: [POST] http://localhost:7071/api/AddToQueue**

**- ProcessPolicyQueue: queueTrigger -> policy-queue**

**🧫 Step 8: Send a Message (to test producer)**

**Using cURL or Postman:**

**curl -X POST http://localhost:7071/api/AddToQueue \**

**-H "Content-Type: application/json" \**

**-d '{**

**"Name": "John Doe",**

**"PolicyNumber": "POL-3001",**

**"StartDate": "2025-10-09T00:00:00Z",**

**"PremiumAmount": 2500.75**

**}'**

**Output**

**Policy POL-3001 added to queue successfully.**

**Immediately in your console:**

**[2025-10-09T06:20:00Z] Queue message received at: 2025-10-09T06:20:00Z**

**[2025-10-09T06:20:00Z] Processed Policy: POL-3001, Name: John Doe**