Appendix A — Answers to the Test Your Knowledge Questions

This appendix has the answers to the questions in the **Test Your Knowledge** section at the end of each chapter.

Chapter 1 — Introducing Apps and Services with .NET

1. Why is it good practice to add the following settings to your project files? And when should you not set it?

   ```
   <TreatWarningsAsErrors>true</TreatWarningsAsErrors>
   ```

   Answer: The compiler gives warnings about issues that the developer should fix. But compiler warnings can be ignored by default. Enabling the `TreatWarningsAsErrors` option forces the developer to fix the issues so that they can compile and run their project.

   In a .NET 7 project that uses the current gRPC tools, you cannot enable this option because the tools generate types with all-lowercase names. .NET 7 or later gives a warning if you use all lowercase for a type name. You can learn more about this warning wave at the following link: `https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/compiler-messages/warning-waves#cs8981---the-type-name-only-contains-lower-cased-ascii-characters`.

   If you find that you get too many errors after enabling this, you can disable specific warnings by using the `<WarningsNotAsErrors>` element with a comma-separated list of warning codes, as shown in the following markup:

   ```
   <WarningsNotAsErrors>0219,CS8981</WarningsNotAsErrors>
   ```

2. Which service technology requires a minimum HTTP version of 2?

   Answer: The gRPC service technology requires a minimum of HTTP/2.

3. In 2010, your organization created a service using .NET Framework and Windows Communication Foundation. What is the best technology to migrate to and why?

   Answer: Two options to migrate a WCF service and client are (1) use the Core WCF open-source project, or (2) re-implement the service and client using gRPC. If Core WCF has all the features you need and you want to use minimum effort, then choose Core WCF. If Core WCF does not support all the features you need and you have the time and resources to re-implement the service, then migrate to gRPC.

4. Which code editor or IDE should you install for .NET development?

   Answer: Most developers will benefit from installing both Visual Studio 2022 and Visual Studio Code and using the best for different project types. More experienced developers often tout the benefits of third-party choices like JetBrains Rider.

5. What should you beware of when creating Azure resources?

   Answer: Many Azure resources have a financial cost. You should delete resources when they are no longer needed as soon as possible to reduce your cost.

Chapter 1A (online-only section) — What's New in Modern C# and .NET

1. Which type of .NET release is higher quality, STS or LTS?

   Answer: Both STS and LTS releases have the same high quality. The only difference is the length of support, 18 months for STS and 3 years for LTS.

2. In new .NET projects, nullable checks are enabled. What are two ways to disable them?

   Answer: You can disable nullable checks in the project file by either setting the `<Nullable>` element to `disabled` or by deleting it completely. This is because nullable checks are off by default if the element is not explicitly set to `enabled`.

3. If you define any types in a top-level program, where must they go in the `Program.cs` file?

   Answer: At the bottom of the file below any executable statements, or you will see compiler error `CS8803`, as shown at the following link: `https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/compiler-messages/cs8803`.

4. How do you import a class like `Console` so that its static members like `WriteLine` are available in all code files throughout a project?

   Answer: In the project file, add a `<Using>` element to an `<ItemGroup>` element. The `<Using>` element must have its `Include` attribute set to the full namespace and name of the class, and its `Static` attribute set to `true`. For example, to import the `Console` class so that its static members like `WriteLine` or `ForegroundColor` are available in all code files throughout a project, add the following markup:

   ```
   <ItemGroup>
     <Using Include="System.Console" Static="true" />
   </ItemGroup>
   ```

5. What is the best new C# 12 language feature?

   Answer: Aliasing any type is the best new C# 12 language feature because it finally removes the long-standing limitation of only being able to alias a subset of types.

Chapter 1B (online-only section) — Benchmarking Performance and Testing

1. What information can you find out about a process?

    Answer: The `Process` class has many properties, including `ExitCode`, `ExitTime`, `Id`, `MachineName`, `PagedMemorySize64`, `ProcessorAffinity`, `StandardInput`, `StandardOutput`, `StartTime`, `Threads`, and `TotalProcessorTime`.

2. How accurate is the `Stopwatch` class?

    Answer: The `Stopwatch` class can be accurate to within a nanosecond (a billionth of a second), but you shouldn't rely on that.

Chapter 1C (online-only section) — Observing and Modifying Code Execution Dynamically

1. What are the four parts of a .NET assembly and which are optional?

    Answer: An assembly is made up of three mandatory parts and one optional part:

    - **Assembly metadata and manifest:** Name, assembly, file version, referenced assemblies, and so on.
    - **Type metadata:** Information about the types, their members, and so on.
    - **IL code:** Implementation of methods, properties, constructors, and so on.
    - **Embedded resources** (optional): Images, strings, JavaScript, and so on.

2. What can an attribute be applied to?

    Answer: There is an `enum` to control what an attribute can be applied to:

```
namespace System
{
  [Flags]
  public enum AttributeTargets
  {
    Assembly = 1,
    Module = 2,
    Class = 4,
    Struct = 8,
    Enum = 16,
    Constructor = 32,
    Method = 64,
    Property = 128,
    Field = 256,
    Event = 512,
```

```
      Interface = 1024,
      Parameter = 2048,
      Delegate = 4096,
      ReturnValue = 8192,
      GenericParameter = 16384,
      All = 32767
   }
 }
```

3. What are the names of the parts of a version number and what do they mean if they follow the rules of semantic versioning?

    Answer: Version numbers in .NET are a combination of three numbers, with two optional additions named **Prerelease** and **Build number**. If you follow the rules of semantic versioning, the three numbers denote the following:

    - **Major:** Breaking changes.
    - **Minor:** Non-breaking changes, including new features, and often bug fixes.
    - **Patch:** Non-breaking bug fixes.

4. How do you get a reference to the assembly for the currently executing console app?

    Answer: Call the `Assembly.GetEntryAssembly()` method.

5. How do you get all the attributes applied to an assembly?

    Answer: Call the `GetCustomAttributes()` method on a reference to an assembly.

6. How should you create a custom attribute?

    Answer: Create a class that inherits from the `Attribute` class. Decorate your class with `[AttributeUsage]` to indicate where it can be applied. Please see the answer to question 2 for the details of attribute targets.

7. What class do you inherit from to enable dynamic loading of assemblies?

    Answer: `AssemblyLoadContext`.

8. What is an expression tree?

    Answer: Expression trees represent code as a structure that you can examine or execute.

9. What is a source generator?

    Answer: Source generators allow a programmer to get a compilation object that represents all the code being compiled, dynamically generate additional code files, and then compile those too.

10. Which interface must a source generator class implement, and what methods are part of that interface?

    Answer: The `ISourceGenerator` interface has the `Initialize` and `Execute` methods.

Chapter 2 — Managing Relational Data Using SQL Server

1. Which NuGet package should you reference in a .NET project to get the best performance when working with data in SQL Server?

    Answer: `Microsoft.Data.SqlClient`.

2. What is the safest way to define a database connection string for SQL Server?

    Answer: Create an instance of `SqlConnectionStringBuilder`, set its properties like `DataSource`, `InitialCatalog`, `UserID`, and `Password`, and then read its `ConnectionString` property.

3. What must T-SQL parameters and variables be prefixed with?

    Answer: T-SQL parameters and variables must be prefixed with the @ character.

4. What must you do before reading an output parameter of a command executed using `ExecuteReader`?

    Answer: You must close the data reader before reading an output parameter.

5. What type does Dapper add its extension methods to?

    Answer: Dapper adds its extension methods to any type that implements `IDbConnection`.

6. What are the two most used extension methods provided by Dapper?

    Answer: The `Query<T>` extension method and the `Execute` extension method.

Chapter 3 — Building Entity Models for SQL Server Using EF Core

1. What can the `dotnet-ef` tool be used for?

    Answer: The `dotnet-ef` tool can perform design-time development tasks, like creating or applying migrations and generating a model from an existing database.

2. What type would you use for the property that represents a table, for example, the `Products` property of a data context?

    Answer: `DbSet<T>`, where `T` is the entity model type in the table, for example, `Product`.

3. What type would you use for the property that represents a one-to-many relationship, for example, the `Products` property of a `Category` entity?

    Answer: `ICollection<T>`, where `T` is the entity model type in the related table, for example, `Product`.

4.   What is the EF Core convention for primary keys?

Answer: The property named `ID` or `Id` or `ClassNameID` or `ClassNameId` is assumed to be the primary key. If the type of that property is any of the following, then the property is also marked as being an `IDENTITY` column: `tinyint`, `smallint`, `int`, `bigint`, or `guid`.

5.   Why might you choose the Fluent API in preference to annotation attributes?

Answer: You might choose Fluent API in preference to annotation attributes when you want to keep your entity classes free from extraneous code that is not needed in all scenarios. For example, when creating a .NET Standard 2.0 class library for entity classes, you might want to only use validation attributes so that the metadata can be read by Entity Framework Core, and by technologies like ASP.NET Core model binding validation and .NET MAUI desktop and mobile apps. However, you might want to use Fluent API to define Entity Framework Core-specific functionality, like mapping to a different table or column name.

6.   Why might you implement the `IMaterializationInterceptor` interface in an entity type?

Answer: EF Core interceptors enable interception, modification, and/or suppression of EF Core operations. The `IMaterializationInterceptor` interface allows interception before and after an entity instance is created, and before and after properties of that instance are initialized. This enables setting properties or calling methods needed for validation, computed values, or flags, using a factory to create instances, and creating a different entity instance than EF would normally create, such as an instance from a cache.

Chapter 4 — Managing NoSQL Data Using Azure Cosmos DB

1.   What are the APIs supported by Azure Cosmos DB?

Answer: The APIs supported by Azure Cosmos DB are the Core (SQL) API, Gremlin (Graph) API, MongoDB API, Azure Table Storage API, and Cassandra API.

2.   At what level do you select the API: account, database, container, or partition?

Answer: You select the API at the account level, and that applies to every database, container, or partition in that account.

3.   What does *embed* mean regarding data modeling with Cosmos DB?

Answer: Embedding means storing data, like the category and supplier information for a product, within the product data, even if that means duplicating that data across many products. It is like the concept of denormalization in a relational database.

4.   What is the unit of measurement for throughput for Cosmos DB, and what does 1 unit represent?

Answer: Throughput is measured as **request units per second (RU/s)**. A single **request unit (RU)** is about the cost of performing a `GET` request for a 1 KB document, using its unique identifier.

5. What package should you reference to programmatically work with Cosmos DB resources?

   Answer: `Microsoft.Azure.Cosmos`.

6. What language do you use to write Cosmos DB Core (SQL) API user-defined functions and stored procedures?

   Answer: JavaScript.

Chapter 4A (online-only section) — Managing Graph Data Using Gremlin API

1. What is the difference between a vertex and an edge in a graph database?

   Answer: A vertex represents a noun like a customer or product. An edge represents a relationship between two vertices, like bought or viewed.

2. What package should you reference to programmatically execute Gremlin scripts?

   Answer: `Gremlin.net`.

3. What Gremlin command returns all vertices that have a label of `product` and have a property named `unitsInStock`, with more than 10 units in stock?

   Answer: `g.V().has('product', 'unitsInStock', gt(10))`

4. Why do some edges have weights?

   Answer: A common property for an edge is a `weight` that could be used to give importance to the relationship, or the cost of traversing the edge in time, money, or distance.

Chapter 5 — Multitasking and Concurrency

1. By convention, what suffix should be applied to a method that returns `Task` or `Task<T>`?

   Answer: Add the suffix `Async` to the method name; for example, for a synchronous method named `Open`, use `OpenAsync` for the equivalent that returns a `Task` or `Task<T>`.

2. To use the `await` keyword inside a method, which keyword must be applied to the method declaration?

   Answer: The `async` keyword must be applied to the method declaration.

3. How do you create a child task?

   Answer: Call the `Task.Factory.StartNew` method with the `TaskCreationOptions.AttachToParent` option to create a child task.

4. Why should you avoid the `lock` keyword?

   Answer: The `lock` keyword does not allow you to specify a timeout; this can cause deadlocks. Use the `Monitor.Enter` method, pass a `TimeSpan` argument as a timeout, and then call the `Monitor.Exit` method explicitly to release the lock at the end of your work instead.

5. When should you use the `Interlocked` class?

   Answer: You should use the `Interlocked` class to modify integers and floating-point numbers that are shared between multiple threads.

6. When should you use the `Mutex` class instead of the `Monitor` class?

   Answer: Use `Mutex` when you need to share a resource across process boundaries. `Monitor` only works on resources inside the current process.

7. What is the benefit of using `async` and `await` in a website or web service?

   Answer: In a website or web service, using `async` and `await` improves scalability, but not the performance of a specific request because of the extra work required to hand over control between threads.

8. Can you cancel a task? If so, how?

   Answer: Yes, you can cancel a task, as described at the following link: `https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/cancel-an-async-task-or-a-list-of-tasks`.

Chapter 6 — Implementing Popular Third-Party Libraries

1. What is the most downloaded third-party `NuGet` package of all time?

   Answer: `Newtonsoft.Json`.

2. What method do you call on the `ImageSharp Image` class to make a change like resizing the image or replacing colors with grayscale?

   Answer: `Mutate`.

3. What is a key benefit of using `Serilog` for logging?

   Answer: `Serilog` can be told to write serialized structured data to the log. The @ symbol prefixing a parameter tells `Serilog` to serialize the object passed in, instead of just the result of calling the `ToString` method. Later, that complex object can be queried for improved search and sort capabilities in the logs.

4. What is a `Serilog` sink?

   Answer: A `Serilog` sink is where you record your logs. For example, there are sinks for writing to the console, to a file, to SQL Server, and to Azure's Application Insights.

5. Should you always use a package like `AutoMapper` to map between objects?

   Answer: No. There is a debate about when `AutoMapper` should be used that you can read about at the following link: `https://www.anthonysteele.co.uk/AgainstAutoMapper.html`.

6. Which `FluentAssertions` method should you call to start a fluent assertion on a value?

   Answer: `Should`.

7. Which `FluentAssertions` method should you call to assert that all items in a sequence conform to a condition, like a `string` item must have fewer than six characters?

   Answer: `OnlyContain`.

8. Which `FluentValidation` class should you inherit from to define a custom validator?

   Answer: `AbstractValidator<T>`.

9. With `FluentValidation`, how can you set a rule to only apply in certain conditions?

   Answer: Call the `When` method with a lambda expression that returns `true`. You can optionally also call the `Otherwise` method to run if the lambda expression returns `false`.

10. With `QuestPDF`, which interface must you implement to define a document for a PDF, and what methods of that interface must you implement?

    Answer: You must implement the `IDocument` interface and, therefore, implement the `Compose` and `GetMetadata` methods.

Chapter 7 — Handling Dates, Times, and Internationalization

1. What is the difference between localization, globalization, and internationalization?

   Answer:

   - Localization affects the user interface of your application. Localization is controlled by a neutral (language only) or specific (language and region) culture. You provide multiple language versions of text and other values. For example, the label of a text box might be "First name" in English, and "Prénom" in French.
   - Globalization affects the data of your application. Globalization is controlled by a specific (language and region) culture, for example, `en-GB` for British English, or `fr-CA` for Canadian French. The culture must be specific because a decimal value formatted as a currency must know to use Canadian dollars instead of French euros.
   - Internationalization is the combination of localization and globalization.

2. What is the smallest measurement of time available in .NET?

   Answer: Nanoseconds.

3. How long is a "tick" in .NET?

   Answer: One tick is 100 nanoseconds.

4. In what scenario might you use a `DateOnly` value instead of a `DateTime` value?

   Answer: Use a `DateOnly` value when you are mapping to a date column in SQL Server, or when you do not need to store the time or know the time zone.

5. For a time zone, what does its `BaseUtcOffset` property tell you?

   Answer: The `BaseUtcOffset` property is a `TimeSpan` that represents the difference between this time zone and the UTC time zone, ignoring any potential Daylight Saving adjustments.

6. How can you get information about the local time zone in which your code executes?

   Answer: Read the properties of the `TimeZoneInfo.Local` object.

7. For a `DateTime` value, what does its `Kind` property tell you?

   Answer: The `Kind` property is a `DateTimeKind` value that indicates if the `DateTime` value is `Unspecified`, `Utc`, or `Local`.

8. How can you control the current culture for your executing code?

   Answer: Set the `CultureInfo.CurrentCulture`, `CultureInfo.CurrentUICulture`, `Thread.CurrentThread.CurrentCulture`, or `Thread.CurrentThread.CurrentUICulture` property to an appropriate `CultureInfo` instance.

9. What is the ISO culture code for Welsh?

   Answer: `cy-GB`.

10. How do localization resource file fallbacks work?

    Answer: Resource files are XML files with the `.resx` extension. The filename includes a culture code, for example, `PacktResources.en-GB.resx` or `PacktResources.da-DK.resx`. The automatic culture fallback search path for resources goes from a specific culture (language and region) to a neutral culture (language only) to an invariant culture. If the current thread culture is `en-AU` (Australian English), then it will search for the resource file in the following order:

    - Australian English: `PacktResources.en-AU.resx`
    - Neutral English: `PacktResources.en.resx`
    - Invariant: `PacktResources.resx`

Chapter 8 — Building and Securing Web Services Using Minimal APIs

1. List six method names that can be specified in an HTTP request.

   Answer: `GET`, `HEAD`, `POST`, `PUT`, `PATCH`, and `DELETE`. Others include `TRACE`, `OPTIONS`, and `CONNECT`.

2. List six status codes and their descriptions that can be returned in an HTTP response.

   Answer: 200 OK, 201 Created, 301 Moved Permanently, 400 Bad Request, 404 Not Found (missing resource), and 500 Internal Server Error. Others include 101 Switching Protocols (e.g., from HTTP to WebSocket), 202 Accepted, 204 No Content, 304 Not Modified, 401 Unauthorized, 403 Forbidden, 406 Not Acceptable (for example, requesting a response format that is not supported by a website), and 503 Service Unavailable.

3. How is the ASP.NET Core Minimal APIs service technology different from the ASP.NET Core Web API service technology?

    Answer: An ASP.NET Core Web API service requires a `controller` class to define the endpoints. An ASP.NET Core Minimal APIs service does not need a controller. A Minimal APIs service can therefore be simpler, and implemented in fewer statements, hence the name.

4. With the ASP.NET Core Minimal APIs service technology, how do you map an HTTP `PUT` request to `api/customers` to a lambda statement block?

    Answer: Call the `MapPut` method, and pass a route pattern and a lambda statement block, as shown in the following code:

    ```
    app.MapPut("api/customers", () =>
      {
        // Do something.
      });
    ```

5. With the ASP.NET Core Minimal APIs service technology, how do you map a method or lambda parameter to a value in a route, query string, or the body of the request?

    Answer: Decorate the parameter with `[FromRoute]`, `[FromQuery]`, or `[FromBody]`.

6. Does enabling CORS increase security for a web service?

    Answer: No. Enabling CORS deliberately weakens security for a web service to allow functionality to work by relaxing the same origin policy of web browsers in specific scenarios. Enabling CORS has no effect on non-web browser clients.

7. You have added statements to `Program.cs` to enable HTTP logging, but HTTP requests and responses are not being logged. What is the most likely reason, and how can you fix it?

    Answer: If no HTTP activity is being logged, then the most likely reason is that the log level is set to `Warning` (2) or `Error` (1). The log level for HTTP logging must be `Information` (3) or higher. In the `appsettings.json` file for your environment, make sure that HTTP logging is at a suitable level, as shown highlighted in the following configuration:

    ```
    {
      "Logging": {
        "LogLevel": {
          "Default": "Information",
          "Microsoft.AspNetCore": "Warning",
          "Microsoft.AspNetCore.HttpLogging": "Information"
        }
      }
    }
    ```

8. How do you limit the rate of requests for a specific client using the `AspNetCoreRateLimit` package?

   Answer: First, configure the name of the HTTP request header that the client must send, as shown in the following configuration:

   ```
   "ClientRateLimiting": {
     "ClientIdHeader": "X-Client-Id",
   ```

   Second, provide the client with a unique ID that they must set that header to in their requests, for example, a GUID value for their account. Third, configure rules for that client ID, as shown in the following configuration:

   ```
   "ClientRateLimitPolicies": {
     "ClientRules": [
       {
         "ClientId": "abc123-...",
         "Rules": [
   ```

9. How do you limit the rate of requests for a specific endpoint using the `Microsoft.AspNetCore.RateLimiting` package?

   Answer: First, create a `RateLimiterOptions` object, configured with the limits that you want, and a name for the policy, and then pass it to a call to the `UseRateLimiter` method. Second, call the `RequireRateLimiting` method on the endpoint and pass in the name of the policy.

10. What does JWT mean?

    Answer: JSON Web Token (JWT) is an open standard that defines a compact and self-contained way to transmit information as a JSON object. It is commonly used for authorization and other small data exchanges.

Chapter 8A (online-only section) — Exposing Data via the Web Using OData

1. What transport protocol does an OData service use?

   Answer: OData uses the **Hyper Text Transport Protocol** (HTTP).

2. Why is an OData service more flexible than a traditional ASP.NET Core Web API service?

   Answer: OData uses query strings for its queries that enable the client to control what is returned, minimizing round trips. A traditional web API defines all the methods and what gets returned.

3. What must you do to an action method in an OData controller to enable query strings to customize what it returns?

   Answer: You must decorate an action method in an OData controller with the `[EnableQuery]` attribute to enable query strings to customize what it returns.

4. What URL path would return customers in Germany who have made more than one order?

    Answer:

    ```
    /ordersystem/customers
      ?$select=CustomerId,CompanyName,City,Country
      &$filter=(Country eq 'Germany') and (Orders/$count gt 1)
    ```

    > You can learn more about using count in filters at the following link: https://devblogs.microsoft.com/odata/adding-support-for-count-segment-in-filter-collections-in-odata-webapi/.

5. How do you get related entities?

    Answer: Use $expand to specify the name of the navigation property, as shown in the following example:

    ```
    /ordersystem/customers
      ?$select=CustomerId,CompanyName,Orders
      &$expand=Orders
    ```

Chapter 9 – Caching, Queuing, and Resilient Background Services

1. How much longer does it take to read 1 MB of data from SSD compared to memory?

    Answer: It takes about four times longer to read 1 MB of data from SSD compared to memory.

2. What is the difference between absolute and sliding expirations?

    Answer: Absolute is a fixed date/time. Sliding is a time span that resets if the item is read so it can potentially last forever.

3. What unit of measurement is used by Size for the in-memory cache?

    Answer: The unit of measurement is arbitrary.

4. You have written the following statement to get information about in-memory caching, but stats is null. What must you do to fix this issue?

    ```
    MemoryCacheStatistics? stats = _memoryCache.GetCurrentStatistics();
    ```

    Answer: You must set TrackStatistics to true in the MemoryCacheOptions object when registering the MemoryCache as a dependency service, as shown in the following code:

    ```
    builder.Services.AddSingleton<IMemoryCache>(new MemoryCache(
      new MemoryCacheOptions
      {
    ```

```
    TrackStatistics = true,
    // ...set other options.
  }));
```

5. What data types can be stored in (a) an in-memory cache, and (b) a distributed cache?

    Answer: Any data type can be stored in an in-memory cache. Only byte arrays can be stored in a distributed cache.

6. What are the differences between the Retry and Circuit Breaker patterns?

    Answer: The **Retry** pattern enables clients to automatically retry a failed action with the expectation that the fault will succeed if retried after a short delay. The **Circuit Breaker** pattern prevents calls when a threshold of faults is reached. In effect, it is a way for a service to detect if a fault is *not* transient, or not transient enough to keep retrying.

7. When using the RabbitMQ default direct exchange, what must the routing key be for a queue named `product`?

    Answer: The routing key must match the queue name, so it must be `product`.

8. What is the difference between a Fanout and a Topic exchange?

    Answer: A Fanout exchange delivers messages to all queues that are bound to it and the routing key is ignored. A Topic exchange delivers messages based on a routing key and criteria defined in the binding between the exchange and a queue.

9. What port does RabbitMQ listen on by default?

    Answer: `5672`.

10. When inheriting from the `BackgroundService` class, what method must you override that is called automatically by the host to run your service?

    Answer: `ExecuteAsync`.

Chapter 10 — Building Serverless Nanoservices Using Azure Functions

1. What is the difference between the in-process and isolated hosting models for Azure Functions?

    Answer: The in-process hosting model requires your Azure Function to be loaded alongside other code and to target a predefined version of an LTS release, like .NET Core 8 or .NET 6. The isolated hosting model allows your Azure Function to load in its own process, and it can use any version of .NET that you choose.

2. What attribute do you use to cause a function to trigger when a message arrives in a queue?

    Answer: Decorate the `Run` method parameter that represents the received message with the `[QueueTrigger]` attribute, and specify the queue name, as shown highlighted in the following code:

```
public static async Task Run(
  [QueueTrigger("checksQueue")] QueueMessage message,
  ILogger log)
{
```

3. What attribute do you use to make a queue available to send messages to?

Answer: Decorate the Run method parameter that represents the queue with the [Queue] attribute, and specify the queue name, as shown highlighted in the following code:

```
public static async Task<IActionResult> Run(
  [HttpTrigger(AuthorizationLevel.Anonymous,
    "get", "post", Route = null)] HttpRequest req,
  [Queue("checksQueue")] ICollector<string> collector,
  ILogger log)
{
  ...
  collector.Add("This is the message."); // Send message to queue.
  ...
```

4. What schedule does the following NCRONTAB expression define?

```
0 0 */6 * 6 6
```

Answer: Assuming the occurrences are during the year 2023, the NCRONTAB expression defines that they occur on Saturdays (the last 6 digit) during June (the preceding 6 digit) in 2023, on any day of the month, at zero minutes and seconds (the first and second 0 digits) of every 6 hours (*/6). For example:

```
Sat, 03 Jun 2023 00:00:00
Sat, 03 Jun 2023 06:00:00
Sat, 03 Jun 2023 12:00:00
Sat, 03 Jun 2023 18:00:00
Sat, 10 Jun 2023 00:00:00
Sat, 10 Jun 2023 06:00:00
Sat, 10 Jun 2023 12:00:00
Sat, 10 Jun 2023 18:00:00
Sat, 17 Jun 2023 00:00:00
Sat, 17 Jun 2023 06:00:00
Sat, 17 Jun 2023 12:00:00
Sat, 17 Jun 2023 18:00:00
Sat, 24 Jun 2023 00:00:00
Sat, 24 Jun 2023 06:00:00
Sat, 24 Jun 2023 12:00:00
Sat, 24 Jun 2023 18:00:00
```

5. How can you configure a dependency service for use in a function?

    Answer: To register a dependency service for use in a function, create a class that inherits from the `FunctionsStartup` class and override its `Configure` method. Add the `[FunctionsStartup]` assembly attribute to specify the class name registered for startup. Add services to the `IFunctionsHostBuilder` instance passed to the method.

Chapter 11 — Broadcasting Real-Time Communication Using SignalR

1. What transport does SignalR use, and which is the default?

    Answer: SignalR prefers to use WebSockets as its transport, then it will fall back to Server-side events, and finally, it will use Long polling if neither of the others is supported by the client and server.

2. What is good practice for RPC method signature design?

    Answer: Good practice for RPC method signature design is to define a single parameter using a complex type. This allows additional properties to be added to the type in the future without breaking the contract between the client and server.

3. What tool can you use to download the SignalR JavaScript library?

    Answer: At the command line or terminal, you can use the Library Manager CLI tool. Visual Studio 2022 includes a graphical tool; right-click a web project, and then choose **Add** | **Client Side Libraries**.

4. What happens if you send a SignalR message to a client with a connection ID that does not exist?

    Answer: Nothing.

5. What are the benefits of separating a SignalR service from other ASP.NET Core components?

    Answer: Once you separate the SignalR hosting, you can take advantage of Azure SignalR Service. This offers global reach and a world-class data center and network, and scales up to millions of connections, while meeting SLAs like providing compliance and high security.

Chapter 12 — Combining Data Sources Using GraphQL

1. What transport protocol does a GraphQL service use?

    Answer: GraphQL can use HTTP or others, like WebSocket.

2. What media type does GraphQL use for its queries?

    Answer: GraphQL can use its own media type, `application/graphql`.

3. How can you parameterize GraphQL queries?

    Answer: In the query definition, you define parameter names prefixed with `$` with a data type, and then reference them in the query, as shown in the following query:

```
query getOrdersByDateAndCountry($country: String, $orderDate: String) {
  order(orderDate: $orderDate {
    orderId
    orderDate
    customer(country: $country) {
      companyName
      country
    }
  }
}
```

4. What are the benefits of using Strawberry Shake over a regular HTTP client for GraphQL queries?

   Answer: Strawberry Shake includes a tool to generate strongly typed proxies that call a GraphQL service and can be more easily used by a .NET client, instead of manually making requests to the service.

5. How might you insert a new product into the Northwind database?

   Answer: You would define a mutation named addProduct, consisting of a type to represent the input, a type to represent the payload, and a class with a method to perform the insertion. Then, you could submit a mutation, as shown in the following GraphQL document:

```
mutation AddProduct {
  addProduct(
    input: {
      productName: "Tasty Burgers"
      supplierId: 1
      categoryId: 2
      quantityPerUnit: "6 per box"
      unitPrice: 40
      unitsInStock: 0
      unitsOnOrder: 0
      reorderLevel: 0
      discontinued: false
    }
  )
  {
    product {
      productId
      productName
    }
  }
}
```

Chapter 13 — Building Efficient Microservices Using gRPC

1. What are three benefits of gRPC that make it a good choice for implementing services?

   Answer: Three benefits of gRPC that make it a good choice for implementing services are (1) its Protobuf binary serialization that minimizes network usage, (2) its requirement of HTTP/2 that provides significant performance benefits, and (3) its support by almost all languages and platforms.

2. How are contracts defined in gRPC?

   Answer: In gRPC, contracts are defined using `.proto` files.

3. Which of the following .NET types require extensions to be imported: `int`, `double`, or `DateTime`?

   Answer: `DateTime` is not supported by default and so requires an extension to be imported.

4. Why should you set a deadline when calling a gRPC method?

   Answer: Setting a deadline for a gRPC call is recommended practice because it controls the upper limit on how long a gRPC call can run for. It prevents gRPC services from potentially consuming too many server resources.

5. Why should you be cautious if implementing your own custom type to handle `decimal` values?

   Answer: There is no standard for handling `decimal` values, so any clients will need to know how you have handled them so that they can do so in the same way. This is difficult to guarantee.

6. What do you have to do to use date and time values in a gRPC message?

   Answer: You must use well-known type extensions: `google.protobuf.Timestamp` and `google.protobuf.Duration`.

7. What are some scenarios when you might implement an interceptor?

   Answer: Interceptors are a way to perform additional processing during requests and responses in the client or service. For example, they can be used for logging, monitoring, and validation.

8. What are the benefits of enabling gRPC JSON transcoding to a gRPC service hosted in ASP.NET Core?

   Answer: gRPC has some strict requirements that not all clients can support. Making HTTP requests and passing JSON objects is supported by all clients. Enabling gRPC JSON transcoding for a gRPC service therefore enables the best of both worlds.

Chapter 14 — Building Web User Interfaces Using ASP.NET Core

1. What is the advantage of declaring a strongly typed Razor view, and how do you do it?

   Answer: The advantage of declaring a strongly typed Razor view is IntelliSense, which will help you write the markup and code, and compiler warnings and errors at design time. If you do not specify a model type, then the Razor view assumes `dynamic` for the type, the use of `Model` cannot show IntelliSense, and errors will not occur until runtime. You specify a model type by adding a `@model` directive with the type you want to use at the top of the Razor view. In the markup, use `Model` to represent the object. For example:

   ```
   @model Customer
   ...
   <h1>@Model.CompanyName</h1>
   ...
   ```

2. How do you enable Tag Helpers in a view?

   Answer: At the top of the Razor view, or in `Views\_ViewImports.cshtml`, add the following statement:

   ```
   @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
   ```

3. What are the pros and cons of HTML Helper methods compared to Tag Helpers?

   Answer: Tag Helpers are generally easier to work with so should be used in most cases, but there are notable situations where Tag Helpers cannot be used, like in Razor components. In those cases, HTML Helper methods should be used instead. HTML Helper methods can do everything you need but require messier code.

4. How can a browser request a preferred language for localization?

   Answer: A browser can request what culture it prefers by adding a query string parameter (for example, `?culture=en-US&ui-culture=en-US`), sending a cookie (for example, `c=en-US|uic=en-US`), or setting an HTTP header (for example, `Accept-Language: en-US,en;q=0.9,fr-FR;q=0.8,fr;q=0.7,en-GB;q=0.6`).

5. How do you localize text in a view?

   Answer: Create resource files (`*.resx`) for each language with the text values that need to be localized. Inject the registered dependency service for `IViewLocalizer` into the Razor view. Retrieve each text value using the unique key that you assigned to it. In `Program.cs`, call the `AddLocalization` method, and specify the folder that contains the `.resx` files. After the call to `AddControllersWithViews`, call the `AddViewLocalization` method. Call the `UseRequestLocalization` method while building the HTTP pipeline with an array of ISO culture codes that you want to support.

6. What is the prefix for attributes recognized by Tag Helpers?

    Answer: `asp-`. For example, `asp-action` and `asp-controller` in an anchor tag element:

    ```
    <a asp-action="Privacy" asp-controller="Home">View our privacy policy.</
    a>
    ```

7. How can you pass a complex object as a query string parameter?

    Answer: Create a method that returns a dictionary with string values for both the key and value. Call the method to set the `asp-all-route-data` attribute of an anchor tag element, as shown in the following markup:

    ```
    <a asp-controller="Home" asp-action="Shipper"
        asp-all-route-data="await GetShipperData()"
        class="btn btn-outline-primary">Shipper</a>
    ```

8. How can you control how long the contents of the `<cache>` element are cached for?

    Answer: The `<cache>` element has the following attributes that control how long the contents are cached for:

    -   expires-after: A TimeSpan value to expire after. The default is `00:20:00`, meaning 20 minutes.
    -   expires-on: A DateTimeOffset value to expire at. No default.
    -   expires-sliding: A TimeSpan value to expire after if the value has not been accessed during that time. No default.

9. What is the `<environment>` element used for?

    Answer: The Environment Tag Helper renders its content only if the current environment matches one of the values in a comma-separated list of names. This allows you to only show content that is needed during development, or only show content on the production website.

10. How does cache busting with Tag Helpers work?

    Answer: When `asp-append-version` is specified with a `true` value in a `<link>`, `<img>`, or `<script>` element, the Tag Helper for that tag type is invoked. They bust the cached version of whatever is referenced by the link, image, or script by automatically appending a query string value, named `v`, that is generated from a SHA256 hash of the referenced source file.

Chapter 15 — Building Web Components Using Blazor

1. What is the benefit of the new Blazor Full Stack hosting model in .NET 8 compared to legacy hosting models like Blazor Server?

Answer: Legacy hosting models like Blazor Server limit where components can run. The new Blazor Full Stack hosting model provides complete flexibility and even adds more options. Components can execute on the server and generate static markup, and each individual component can be switched to any of the following: streaming rendering, interactive server-side with live updates of the COM using SignalR, or interactive client-side with WebAssembly. The Blazor WebAssembly hosting model is still useful for standalone apps hosted on a static website.

2. Does Blazor WebAssembly support all features of the latest .NET APIs?

   Answer: No. Due to limitations imposed by running inside a web browser, not all features are supported. Use of an unsupported feature will cause a `PlatformNotSupportedException` to be thrown. A browser compatibility analyzer will warn of issues during development time.

3. What is the file extension for Blazor components?

   Answer: `.razor`.

4. How do you set the default layout for all Blazor page components?

   Answer: In `App.razor`, set the `DefaultLayout` property of the `<RouteView>` element to a Razor expression that sets the type of a layout class, as shown highlighted in the following markup:

```
<Router AppAssembly="@typeof(App).Assembly">
  <Found Context="routeData">
    <RouteView RouteData="@routeData" DefaultLayout="@typeof(MainLayout)"
/>
    ...
  </Found>
  ...
</Router>
```

5. How do you register a route for a Blazor page component?

   Answer: At the top of the `.razor` file for a page component, add a `@page` directive with the relative route, as shown in the following code: `@page "my-route"`

6. When would you set the `Match` property of a `<NavLink>` component to `NavLinkMatch.All`?

   Answer: When multiple `<NavLink>` component instances could have a partial match on their relative paths; for example, by default, `/customers` would match both `/customers` and `/customers/USA`, so you would want the `<NavLink>` for `/customers` to be set to `NavLinkMatch.All`.

7. You have imported a custom namespace in the `_Imports.razor` file, but when you try to use a class in that namespace in a code-behind file for the Blazor component, the class is not found. Why? How can you fix the issue?

   Answer: The `_Imports.razor` file only applies to `.razor` files. If you use code-behind `.cs` files to implement component code, then they must have namespaces imported separately. To fix the issue, use global usings to implicitly import the namespace.

8. What must you do to a property in a component class to have it set to a query string parameter automatically?

    Answer: You must decorate the property with the `[Parameter]` and `[SupplyParameterFromQuery]` attributes, as shown in the following code:

    ```
    [Parameter]
    [SupplyParameterFromQuery(Name = "country")]
    public string? Country { get; set; }
    ```

9. What is QuickGrid?

    Answer: QuickGrid is an open-source basic grid Blazor component from Microsoft.

10. How can a Blazor component access browser features like local storage?

    Answer: A Blazor component accesses browser features like local storage by using JavaScript interop, using the interface named `IJSRuntime`. It can dynamically load a JavaScript module file using its `InvokeAsync<IJSObjectReference>` method. The JavaScript file can then interact with all browser features.

Chapter 15A (online-only section) — Leveraging Open-Source Blazor Component Libraries

1. Why is the Radzen Blazor component library a good choice compared to alternatives like DevExpress or SyncFusion?

    Answer: The Radzen Blazor component library is a good choice because it is free and open-source. Other component libraries are implemented in the same way, so learning Radzen prepares you to use the other component libraries too.

2. Does using the Radzen Blazer component library require your project to also use Bootstrap?

    Answer: Some Radzen Blazor themes require Bootstrap. If you want to avoid Bootstrap, then you can reference the `_content/Radzen.Blazor/css/default-base.css` file instead of `_content/Radzen.Blazor/css/default.css`, but then you can only use the base default theme without advanced layouts.

3. Which four Radzen Blazer components require you to register dependency services?

    Answer: `Dialog`, `Notification`, `Tooltip`, and `ContextMenu`.

4. In a `NotificationMessage`, what does setting the `Severity` property to 1, 2, or 3 do?

    Answer: Setting `Severity` to 1 makes the color of the notification box light blue for information. Setting `Severity` to 2 makes the color of the notification box green for success. Setting `Severity` to 3 makes the color of the notification box yellow for warning.

5. What is the name of the icon library that can be used by default to set icons for tabs and other components?

   Answer: Google Material Icons.

6. How can you customize the formatting of data values in a RadzenChart?

   Answer: Create a method that has an object input parameter and returns a `string` value using the format you want, as shown in the following code:

   ```
   private string FormatAsUSD(object value)
   {
     return ((double)value).ToString("C0",
       CultureInfo.GetCultureInfo("en-US"));
   }
   ```

   Set that method as the `Formatter` for the `RadzenValueAxis` element, as shown in the following markup:

   ```
   <RadzenValueAxis Formatter="@FormatAsUSD">
   ```

7. How do you two-way data-bind a `RadzenTextBox` component to a property?

   Answer: Set the `Value` property of the `RadzenTextBox` using the special `@bind-` prefix to an expression for a private field object, like `customer`, and one of its properties, like `CompanyName`, as shown in the following code:

   ```
   <RadzenTextBox Name="CompanyName" @bind-Value="customer.CompanyName" />
   ```

8. What three properties should be set on a `RadzenListBox` or `RadzenDropDown` component to display the list of items to select from?

   Answer: `Data` should be set to the data items. `TextProperty` should be set to the name of a property on each data item that is visible to the user. `ValueProperty` should be set to the name of a property on each data item that is stored.

9. How can you trigger a `Submit` event for a form?

   Answer: Add a `RadzenButton` with a `ButtonType` of `ButtonType.Submit`.

10. Which component provides visual grouping in a form?

    Answer: `RadzenFieldset`.

Chapter 16 — Building Mobile and Desktop Apps Using .NET MAUI

1. What are the four categories of .NET MAUI user interface components, and what do they represent?

    Answer: The four categories of .NET MAUI user interface components are:

    - Pages: This represents mobile application screens.
    - Layouts: This represents the structure of a combination of the user interface components.
    - Views: This represents a single user interface component.
    - Cells: This represents a single item in a list or table view.

2. What is the benefit of the `Shell` component, and what kinds of UI does it implement?

    Answer: The benefit of the `Shell` component is simplified app development by providing standardized navigation and search capabilities. It defines either a tab bar or flyout for user interface navigation.

3. How can you enable a user to perform an action on a cell in a list view?

    Answer: To enable a user to perform an action on a cell in a list view, you can set some context actions, which are menu items that raise an event, as shown in the following markup:

    ```
    <TextCell Text="{Binding CompanyName}"
              Detail="{Binding Location}"
              TextColor="{DynamicResource PrimaryTextColor}"
              DetailColor="{DynamicResource PrimaryTextColor}" >
      <TextCell.ContextActions>
        <MenuItem Clicked="Customer_Phoned" Text="Phone" />
        <MenuItem Clicked="Customer_Deleted" Text="Delete"
                  IsDestructive="True" />
      </TextCell.ContextActions>
    </TextCell>
    ```

4. When would you use an `Entry` instead of an `Editor`?

    Answer: Use an `Entry` for a single line of text and an `Editor` for multiple lines of text.

5. What is the effect of setting `IsDestructive` to `true` for a menu item in a cell's context actions?

    Answer: The menu item is colored red as a warning to the user.

6. You have defined a `Shell` with a content page, but no navigation is shown. Why might this be?

    Answer: A shell with only one content page does not show any navigation because there is nothing to navigate to. You must have at least two shell content items.

7. What is the difference between `Margin` and `Padding` for an element like a `Button`?

    Answer: The difference between `Margin` and `Padding` for an element like a `Button` is that `Margin` is outside the `Border`, and `Padding` is inside the `Border`.

8. How are event handlers attached to an object using XAML?

    Answer: Event handlers are attached to an object using XAML by setting an attribute for the event name to the name of a method in the code-behind class, as shown in the following markup:

    ```
    <Button Clicked="SaveButton_Clicked">
    ```

9. What do XAML styles do?

    Answer: XAML styles enable the setting of one or more properties.

10. Where can you define resources?

    Answer: You can define resources in any element depending on where you want to share those resources:

    -   To share resources throughout an app, define resources in the `<Application.Resources>` element.
    -   To share resources only within a page, define resources in its `<Page.Resources>` element.
    -   To share resources only in a single element like a button, define resources in its `<Button.Resources>` element.

Chapter 16A (online-only section) — Implementing Model-View-ViewModel for .NET MAUI

1. What interface must a type implement to support two-way binding?

    Answer: The `INotifyPropertyChanged` interface.

2. What class should a view model inherit from the support automatic notifications when an item is added and removed from a collection?

    Answer: The `ObservableCollection<T>` class.

3. What class should a view model inherit from, and what keyword must be applied to it to allow automatic definitions of bindable properties?

    Answer: The `ObservableObject` class.

4. What attribute must you decorate private fields with to allow automatic definitions of bindable properties?

    Answer: The `[ObservableProperty]` attribute.

5. What naming conventions must you use for private fields to allow automatic definitions of bindable properties?

    Answer: The private fields must use the `camelCase`, `_camelCase`, or `m_camelCase` naming conventions.

Chapter 16B (online-only section) — Integrating .NET MAUI Apps with Blazor and Native Platforms

1. Modern .NET is cross-platform. Windows Forms and WPF apps can run on modern .NET. Can those apps, therefore, run on macOS and Linux?

    Answer: No. Although Windows Forms and WPF apps can run on modern .NET, they also need to make calls to Win32 APIs and so are limited to running on Windows. When you download the .NET SDK for Windows, it includes an extra workload to support WPF and Windows Forms apps.

2. Is .NET MAUI a .NET developer's only choice for creating cross-platform graphical user interfaces?

    Answer: No. .NET MAUI is not the only choice for creating cross-platform GUIs. A .NET developer could create a Blazor app using its Progressive Web App support, or use a third-party platform like Uno.

3. How do you enable a .NET MAUI app that targets iOS and Android to allow unsecured HTTP connections, i.e., use `http` as well as `https`?

    Answer: For iOS, modify the `Info.plist` file to add a new key named `NSAppTransportSecurity`, which is a dictionary, and in it, add a key named `NSAllowsArbitraryLoads` that has a value of `true`. For Android, create an XML file containing a `<network-security-config>` element, with a `<domain-config cleartextTrafficPermitted="true">` element that lists the domains allowed to use cleartext. In `AndroidManifest.xml`, add an attribute to the `<application>` element to reference the new XML file.

4. What does the .NET MAUI Community Toolkit do to make sure you have configured it?

    Answer: The .NET MAUI Community Toolkit runs a code analyzer that checks to see if you have called the extension method to use the toolkit.

5. What domain must a .NET MAUI app use to connect to a locally hosted web service when testing in (a) the Android emulator, and (b) the iOS simulator?

    Answer: When testing in the Android emulator, the domain must use the IP address `10.0.2.2`. When testing in the iOS simulator, the domain can use `localhost`.

6. In a .NET MAUI app, how would you store the text, "Hello, World!", in the system clipboard?

    Answer: `await Clipboard.Default.SetTextAsync("Hello, World!");`

7. As well as writing code that uses file and media pickers, what must you do?

   Answer: iOS and Android require you to configure intents and request permissions in their manifest and `plist` files.

8. When picking a file in a .NET MAUI app, how do you set a filter to limit which types of files can be selected?

   Answer: Create a `FilePickerFileType` object, with a dictionary that uses a `DevicePlatform` as a key and a sequence of `string` values to specify the file extensions or media types to filter, as shown in the following code:

```
FilePickerFileType textFileTypes = new(
  new Dictionary<DevicePlatform, IEnumerable<string>>
  {
    { DevicePlatform.iOS, new[] { "public.plain-text" } },
    { DevicePlatform.Android, new[] { "text/plain" } },
    { DevicePlatform.WinUI, new[] { ".txt" } },
    { DevicePlatform.Tizen, new[] { "*/*" } },
    { DevicePlatform.macOS, new[] { "txt" } }
  });
```

9. What information about a device can you access in a .NET MAUI app?

   Answer: Device model, device type, operating system and version, battery charge state, screen resolution, pixel density, orientation, and so on.

10. What are some benefits of using the .NET MAUI Community Toolkit?

    Answer: The .NET MAUI Community Toolkit includes animations, behaviors, converters, effects, and helpers. For example, it makes it easy to display Toast notifications.