Multi Choice Type Question What is Swagger in the context of ASP.NET Core Web API? Marks : 1 Negative Marks : 0 image Answer here A testing framework for unit tests A database management system for storing test data A documentation tool for describing and visualizing APIs A build automation tool for compiling the application

The correct answer is:

**A documentation tool for describing and visualizing APIs** ✅

**Explanation:**  
Swagger (now part of the OpenAPI specification) is widely used in ASP.NET Core Web API projects to **document APIs**, provide an interactive **UI for testing endpoints**, and allow developers or consumers to understand available routes, request/response formats, and parameters. It is **not** a testing framework, database system, or build tool.

**You said:**

Multi Choice Type Question Which Swagger package is commonly used in ASP.NET Core Web API to enable API documentation generation? Marks : 1 Negative Marks : 0 image Answer here Swashbuckle.AspNetCore Microsoft.AspNetCore.Swagger ASP.NET Core Swagger API.Documentation.Swagger

The correct answer is:

**Swashbuckle.AspNetCore** ✅

**Explanation:**  
In ASP.NET Core Web API projects, **Swashbuckle.AspNetCore** is the most commonly used NuGet package to integrate Swagger. It automatically generates **OpenAPI/Swagger documentation** and provides a **Swagger UI** for testing and exploring API endpoints. The other options are not standard packages.

**You said:**

Multi Choice Type Question What is the first step to enable Swagger documentation in an ASP.NET Core Web API project? Marks : 1 Negative Marks : 0 image Answer here Adding the [Swagger] attribute to the controllers Installing the Swashbuckle.AspNetCore NuGet package Creating a separate Swagger configuration file Enabling the Swagger middleware in the Startup.cs file

The correct answer is:

**Installing the Swashbuckle.AspNetCore NuGet package** ✅

**Explanation:**  
The first step to enable Swagger in an ASP.NET Core Web API is to **install the Swashbuckle.AspNetCore package** via NuGet. After installing, you can configure Swagger in Program.cs (or Startup.cs for older versions) by adding services and middleware. Adding attributes or creating configuration files comes **after the package is installed**.

**You said:**

Multi Choice Type Question Which attribute is used to specify the API's summary or description in Swagger documentation? Marks : 1 Negative Marks : 0 image Answer here [Description] [APIInfo] [Summary] [SwaggerDoc]

The correct answer is:

**[Summary]** ✅

**Explanation:**  
In ASP.NET Core Web API with Swagger, the **[Summary] attribute** (or XML comments with <summary> tags) is used to provide a brief description of an API endpoint. Swagger reads these comments/attributes to display the **endpoint summary** in the generated documentation.

Other options like [Description], [APIInfo], or [SwaggerDoc] are not standard attributes for endpoint summaries in Swagger.

**You said:**

Multi Choice Type Question What is Swagger (OpenAPI) primarily used for? Marks : 1 Negative Marks : 0 image Answer here To create decoupled services To optimize API performance To describe and document REST APIs To generate random test data for APIs

The correct answer is:

**To describe and document REST APIs** ✅

**Explanation:**  
Swagger, now part of the **OpenAPI specification**, is primarily used to **describe, document, and visualize RESTful APIs**. It provides a standard way to define endpoints, request/response formats, parameters, and authentication details, and often comes with an interactive UI for testing APIs. It is **not** used for performance optimization, generating test data, or creating decoupled services.

**You said:**

Multi Choice Type Question Which security scheme is commonly used in Swagger documentation to represent API authentication? Marks : 1 Negative Marks : 0 image Answer here API Key OAuth 1.0 Basic Authentication Bearer Token

The correct answer is:

**Bearer Token** ✅

**Explanation:**  
In Swagger/OpenAPI documentation, **Bearer Token** is the most commonly used security scheme to represent **token-based API authentication**, especially for REST APIs using JWT (JSON Web Tokens). While API Key and Basic Authentication can also be used, **Bearer Token** is the standard for modern APIs requiring secure access. OAuth 1.0 is largely outdated in favor of OAuth 2.0.

**You said:**

Multi Choice Type Question How can API consumers interact with an API directly from the Swagger UI documentation? Marks : 1 Negative Marks : 0 image Answer here By using the provided API key for authentication By making HTTP requests to the API endpoints listed in the Swagger UI API consumers cannot interact with the API from Swagger UI directly By accessing the "/swagger/api" endpoint on the API server

The correct answer is:

**By making HTTP requests to the API endpoints listed in the Swagger UI** ✅

**Explanation:**  
Swagger UI provides an **interactive interface** where API consumers can directly **send HTTP requests** to the listed API endpoints, view responses, and test functionality. It often allows specifying authentication (like API keys or Bearer tokens) and parameters, making it a powerful **testing and exploration tool** for APIs.

**You said:**

Multi Choice Type Question What is the advantage of consuming APIs from Swagger UI during API development? Marks : 1 Negative Marks : 0 image Answer here It allows API consumers to directly modify the API documentation. It provides a convenient way to test and explore API endpoints without external tools. It reduces the load on the API server during testing. Consuming APIs from Swagger UI is not recommended during development.

The correct answer is:

**It provides a convenient way to test and explore API endpoints without external tools.** ✅

**Explanation:**  
Swagger UI allows developers and API consumers to **interactively test API endpoints**, inspect request and response formats, and explore functionality **directly from the browser**. This eliminates the need for separate tools like Postman during development and speeds up testing and debugging. It does **not** modify documentation, reduce server load, or pose a development risk when used properly.

**You said:**

Multi Choice Type Question What is the purpose of configuring security schemes in the Swagger middleware? Marks : 1 Negative Marks : 0 image Answer here To specify the API version used for securing the API endpoints To define the allowed origins for cross-origin requests To control the visibility of the Swagger documentation To specify the authentication mechanisms required for accessing specific API endpoints

The correct answer is:

**To specify the authentication mechanisms required for accessing specific API endpoints** ✅

**Explanation:**  
Configuring **security schemes** in Swagger (OpenAPI) allows you to **define how API endpoints are secured**, such as using **Bearer tokens, API keys, or OAuth2**. This helps Swagger UI prompt users for the correct authentication credentials when testing endpoints. It does **not** control API versioning, CORS origins, or documentation visibility.

**You said:**

Multi Choice Type Question Which HTTP methods are supported for making requests to API endpoints from Swagger UI in ASP.NET Core Web API? Marks : 1 Negative Marks : 0 image Answer here Only GET, POST, and PUT Only GET, POST, PUT, and DELETE Only GET, POST, PUT, DELETE, and PATCH Swagger UI does not support making API requests.

The correct answer is:

**Only GET, POST, PUT, DELETE, and PATCH** ✅

**Explanation:**  
Swagger UI supports **all standard HTTP methods** used in REST APIs, including **GET, POST, PUT, DELETE, and PATCH**. This allows developers and API consumers to fully test and interact with API endpoints directly from the documentation interface. It is **not limited** to just a few methods.

**You said:**

Multi Choice Type Question How can you customize the appearance and behavior of the Swagger UI in an ASP.NET Core Web API project? Marks : 1 Negative Marks : 0 image Answer here By editing the Swagger UI HTML and CSS files directly By using the SwaggerUIOptions in the UseSwaggerUI() method By configuring the appsettings.json file By modifying the default API controllers

The correct answer is:

**By using the SwaggerUIOptions in the UseSwaggerUI() method** ✅

**Explanation:**  
In ASP.NET Core, you can **customize the Swagger UI**—such as the document title, endpoint URLs, default expansion of models, and OAuth settings—by configuring **SwaggerUIOptions** inside the app.UseSwaggerUI() method in Program.cs (or Startup.cs). Editing HTML/CSS directly or modifying controllers is unnecessary, and appsettings.json is not used for Swagger UI customization.

**You said:**

Multi Choice Type Question How do you specify a custom title and version for the Swagger documentation in an ASP.NET Core Web API project? Marks : 1 Negative Marks : 0 image Answer here Configure the SwaggerUIOptions in the UseSwaggerUI() method Add metadata to the API controllers Use the AddSwaggerGen() method in the ConfigureServices() method Modify the appsettings.json file

The correct answer is:

**Use the AddSwaggerGen() method in the ConfigureServices() method** ✅

**Explanation:**  
In ASP.NET Core Web API, you can specify **custom metadata** like the API **title, version, and description** by configuring **SwaggerGenOptions** in the services.AddSwaggerGen() method inside Program.cs (or Startup.cs). This is where you define SwaggerDoc() with the desired name and version. Customizing SwaggerUIOptions affects only the UI, not the underlying documentation metadata.

**You said:**

Multi Choice Type Question Which attribute can be used to hide an API endpoint from appearing in the Swagger documentation? Marks : 1 Negative Marks : 0 image Answer here [ApiExplorerSettings(IgnoreApi = true)] [Hidden] [SwaggerIgnore] [ExcludeFromDocs]

The correct answer is:

**[ApiExplorerSettings(IgnoreApi = true)]** ✅

**Explanation:**  
In ASP.NET Core Web API, applying the attribute **[ApiExplorerSettings(IgnoreApi = true)]** to a controller or action **hides it from Swagger documentation**. The other options ([Hidden], [SwaggerIgnore], [ExcludeFromDocs]) are not standard attributes in ASP.NET Core for controlling Swagger visibility.

**You said:**

Multi Choice Type Question Which method is used in the Program.cs file to add the Swagger middleware to the HTTP request pipeline? Marks : 1 Negative Marks : 0 image Answer here app.SwaggerUI() app.ConfigureSwagger() app.AddSwagger() app.UseSwagger()

The correct answer is:

**app.UseSwagger()** ✅

**Explanation:**  
In ASP.NET Core, the **app.UseSwagger()** method is used in Program.cs (or Startup.cs) to **add the Swagger middleware** to the HTTP request pipeline. This enables serving the generated Swagger JSON. To display the interactive UI, **app.UseSwaggerUI()** is also called, but the middleware that generates the Swagger JSON is specifically **UseSwagger()**.

**You said:**

Multi Choice Type Question In an ASP.NET Core Web API project, where should the Swagger configuration typically be added? Marks : 1 Negative Marks : 0 image Answer here appsettings.json Controller.cs Program.cs Web.config

The correct answer is:

**Program.cs** ✅

**Explanation:**  
In ASP.NET Core Web API, **Swagger configuration** is typically added in **Program.cs** (or Startup.cs in older versions). This includes:

* Adding services with builder.Services.AddSwaggerGen()
* Configuring the middleware with app.UseSwagger() and app.UseSwaggerUI()

It is **not** added in appsettings.json, controllers, or Web.config.