Multi Choice Type Question What is Postman primarily used for in the context of ASP.NET Core Web API development? Marks : 1 Negative Marks : 0 image Answer here To write and execute test scripts for API endpoints To create a database schema for the application To design the user interface of the web application To manage version control of the API endpoints

The correct answer is:

**To write and execute test scripts for API endpoints** ✅

**Postman** is primarily used to **test, debug, and interact with API endpoints** by sending HTTP requests and inspecting responses. It is not used for database design, UI design, or version control.

**You said:**

Multi Choice Type Question In the route template "{controller}/{action}/{id}", what does "{id}" represent? Marks : 1 Negative Marks : 0 image Answer here The action name The HTTP verb The controller name A parameter passed in the request URL

The correct answer is:

**A parameter passed in the request URL** ✅

In the route template "{controller}/{action}/{id}", **{id}** is a **placeholder for a value** passed in the URL, typically used to identify a specific resource.

**You said:**

Multi Choice Type Question In ASP.NET Core Web API, how are query string parameters typically extracted in a controller action? Marks : 1 Negative Marks : 0 image Answer here Using the [FromBody] attribute Using the [FromHeader] attribute Using the [FromRoute] attribute Using the [FromQuery] attribute

The correct answer is:

**Using the [FromQuery] attribute** ✅

In ASP.NET Core Web API, **query string parameters** are typically extracted in a controller action using the [FromQuery] attribute.

* [FromBody] is for request body data.
* [FromHeader] is for HTTP headers.
* [FromRoute] is for route parameters.

**You said:**

Multi Choice Type Question Which attribute is used to define a route template for an action method in an ASP.NET Core Web API controller? Marks : 1 Negative Marks : 0 image Answer here [Route] [HttpGet] [RoutePrefix] [RouteTemplate]

The correct answer is:

**[Route]** ✅

In ASP.NET Core Web API, the [Route] attribute is used to **define a route template** for a controller or an action method.

* [HttpGet] specifies the HTTP method.
* [RoutePrefix] and [RouteTemplate] are not used in ASP.NET Core (they were used in older Web API versions).

**You said:**

Multi Choice Type Question Which HTTP verb attribute is used to define an action method for handling HTTP POST requests? Marks : 1 Negative Marks : 0 image Answer here [HttpGet] [HttpPost] [HttpPut] [HttpDelete]

The correct answer is:

**[HttpPost]** ✅

In ASP.NET Core Web API, the [HttpPost] attribute is used to **define an action method that handles HTTP POST requests**, typically used to create new resources.

* [HttpGet] is for retrieving data.
* [HttpPut] is for updating data.
* [HttpDelete] is for deleting data.

**You said:**

Multi Choice Type Question What is the purpose of model binding in ASP.NET Core Web API? Marks : 1 Negative Marks : 0 image Answer here To map request data to action parameters automatically To apply data validation on incoming requests To enforce HTTPS for all API endpoints To authenticate and authorize incoming requests

The correct answer is:

**To map request data to action parameters automatically** ✅

In ASP.NET Core Web API, **model binding** automatically maps data from HTTP requests (query strings, route data, form data, or JSON body) to **action method parameters or model objects**, simplifying data handling in controllers.

**You said:**

Multi Choice Type Question What is the purpose of the Postman "Environment" feature? Marks : 1 Negative Marks : 0 image Answer here To manage the database connections for the API requests To define different environments (e.g., development, staging, production) and their corresponding variables To design the user interface for the web application To monitor the API response times

The correct answer is:

**To define different environments (e.g., development, staging, production) and their corresponding variables** ✅

In Postman, the **Environment** feature allows you to **store and switch between sets of variables** (like base URLs, tokens, or credentials) for different environments, making API testing more flexible and reusable.

**You said:**

Multi Choice Type Question Which logging level is typically used for critical error messages that may result in the application's failure? Marks : 1 Negative Marks : 0 image Answer here Trace Debug Error Info

The correct answer is:

**Error** ✅

The **Error** logging level is used for **critical issues** that indicate a failure in the application or a component, requiring attention.

* **Trace** and **Debug** are for detailed diagnostic information.
* **Info** is for general informational messages.

**You said:**

Multi Choice Type Question How can you pass complex data (e.g., JSON) from the client to the server in an HTTP POST request to an ASP.NET Core Web API endpoint? Marks : 1 Negative Marks : 0 image Answer here Using the request body Using query string parameters Using HTTP headers Complex data cannot be sent in a POST request

The correct answer is:

**Using the request body** ✅

In an HTTP POST request to an ASP.NET Core Web API, **complex data like JSON** is typically sent **in the request body**, which can then be deserialized into model objects using model binding.

* Query string parameters and headers are used for simple data or metadata, not complex JSON objects.

**You said:**

How can you specify a default value for a route parameter in ASP.NET Core attribute routing? Marks : 1 Negative Marks : 0 image Answer here By using the DefaultValue property of the [Route] attribute By appending a default value directly to the route parameter in the route template By using the [DefaultValue] attribute on the action method Default values cannot be specified for route parameters in attribute routing

The correct answer is:

**By appending a default value directly to the route parameter in the route template** ✅

In ASP.NET Core attribute routing, you can specify a default value like this:

[Route("api/products/{id=1}")]

public IActionResult GetProduct(int id) { ... }

Here, if id is not provided in the URL, it defaults to 1.

**You said:**

Multi Choice Type Question When using attribute routing in ASP.NET Core, where can you specify route constraints? Marks : 1 Negative Marks : 0 image Answer here In the controller's constructor Within the action method's parameter list In the ConfigureServices method of the Startup class In the [Route] attribute applied to the action method

The correct answer is:

**In the [Route] attribute applied to the action method** ✅

In ASP.NET Core **attribute routing**, route constraints are specified **directly in the route template** within the [Route] attribute. For example:

[Route("api/products/{id:int}")]

public IActionResult GetProduct(int id) { ... }

Here, {id:int} ensures that only integer values are matched.

**You said:**

When using attribute routing in ASP.NET Core, how can you create a route with multiple parameters? Marks : 1 Negative Marks : 0 image Answer here By concatenating the parameters directly in the [Route] attribute By separating parameters with commas in the [Route] attribute By defining separate [Route] attributes for each parameter By using curly braces {} around each parameter in the [Route] attribute

The correct answer is:

**By using curly braces {} around each parameter in the [Route] attribute** ✅

In ASP.NET Core **attribute routing**, multiple parameters are defined in the route template using **curly braces**. Example:

[Route("api/products/{categoryId}/{productId}")]

public IActionResult GetProduct(int categoryId, int productId) { ... }

This allows the route to capture values for both categoryId and productId.

**You said:**

Multi Choice Type Question In ASP.NET Core Web API, what is the purpose of the [ApiController] attribute? Marks : 1 Negative Marks : 0 image Answer here To specify the route template for the entire controller To indicate that the controller is an API controller and enable automatic HTTP 400 responses To define the HTTP methods supported by the controller To handle authentication and authorization for the controller

The correct answer is:

**To indicate that the controller is an API controller and enable automatic HTTP 400 responses** ✅

The [ApiController] attribute in ASP.NET Core:

* Marks the class as a Web API controller.
* Enables **automatic model validation**, returning HTTP 400 responses when the model state is invalid.
* Simplifies parameter binding and routing conventions for API controllers.

**You said:**

Multi Choice Type Question What does the [HttpGet] attribute signify when applied to an action method in an ASP.NET Core Web API controller? Marks : 1 Negative Marks : 0 image Answer here It specifies that the action method should only respond to HTTP GET requests. It defines the route template for the action method. It indicates that the action method should handle HTTP POST requests. It sets the logging level for the action method.

The correct answer is:

**It specifies that the action method should only respond to HTTP GET requests.** ✅

The [HttpGet] attribute in ASP.NET Core Web API **maps the action method to handle HTTP GET requests**, which are typically used to retrieve data from the server.

Top of Form