3. WebApi\_Handson Objectives Answers

# 1. Demonstrate creation of an Action method to return list of custom class entity

- Create a model class (e.g., `Employee`) with properties like Id, Name, Salary, etc.  
- Create an API controller (e.g., `EmployeeController`) with a `[HttpGet]` method that returns `List<Employee>`.  
- Mark the method with `[AllowAnonymous]` to allow public access without authorization.  
- Use `[ProducesResponseType(typeof(List<Employee>), 200)]` for Swagger support.  
- Example:  
 [HttpGet]  
 [AllowAnonymous]  
 [ProducesResponseType(typeof(List<Employee>), 200)]  
 public ActionResult<List<Employee>> GetEmployees()  
 {  
 return Ok(GetStandardEmployeeList());  
 }

# 2. Explain the usage of FromBody attribute

- `[FromBody]` is used in `[HttpPost]` or `[HttpPut]` methods to bind the request body to a model object.  
- It is necessary when you want to send complex data (JSON/XML) in the request body rather than query parameters.  
- Example:  
 [HttpPost]  
 public IActionResult PostEmployee([FromBody] Employee emp)  
 {  
 \_employees.Add(emp);  
 return Ok(emp);  
 }  
- Without `[FromBody]`, the model binding might fail or look for data in query string.

# 3. Demonstrate Custom filter

- Create a custom filter class by inheriting `ActionFilterAttribute`.  
- Override `OnActionExecuting(ActionExecutingContext context)` to intercept requests before hitting action methods.  
- Example: Validate presence of `Authorization` header or handle exceptions globally.  
- For custom exception filter, implement `IExceptionFilter` and override `OnException(ExceptionContext context)`.  
- Log exception details and return generic 500 error.  
- Example - Authorization Filter:  
 public class CustomAuthFilter : ActionFilterAttribute  
 {  
 public override void OnActionExecuting(ActionExecutingContext context)  
 {  
 var headers = context.HttpContext.Request.Headers;  
 if (!headers.ContainsKey("Authorization"))  
 {  
 context.Result = new BadRequestObjectResult("Invalid request - No Auth token");  
 return;  
 }  
  
 var token = headers["Authorization"].ToString();  
 if (!token.Contains("Bearer"))  
 {  
 context.Result = new BadRequestObjectResult("Invalid request - Token present but Bearer unavailable");  
 }  
 }  
 }  
- Example - Exception Filter:  
 public class CustomExceptionFilter : IExceptionFilter  
 {  
 public void OnException(ExceptionContext context)  
 {  
 var exception = context.Exception;  
 File.AppendAllText("log.txt", $"[{DateTime.Now}] {exception.Message}\n");  
 context.Result = new ObjectResult("Internal Server Error")  
 {  
 StatusCode = 500  
 };  
 }  
 }  
- Install `Microsoft.AspNetCore.Mvc.WebApiCompatShim` if needed for backward compatibility.