Abdallah Hany

* LinkedIn article about the Architecture pattern in general and MVC particularly

[Linked-in article](https://lnkd.in/p/dQZMs4PZ)

1. why we use IActionResult not ActionResult support ur answer with scenario or problems?

* ActionResult returns only built-in MVC results (View, Json, Redirect…).
* IActionResult is an interface, so it allows both built-in and custom results → more flexible and testable.

1. what the httpcontext request and response message consist of ?

* Request: Headers, URL, Query string, Cookies, Method (GET/POST…), Body.
* Response: Status code, Headers, Cookies, Content type, Body (HTML/JSON/etc).

1. what’s the diff btw https and http?

HTTP: Plain text, not secure.

HTTPS: Encrypted with SSL/TLS → ensures confidentiality & integrity.

1. what’s the segments and fragments in URL with real URL Example?

Segment:Path part after domain.  
Fragment:Client side bookmark.

1. what’s Builder and Dependency injection with a real life example clarify it?

* Builder: Helps construct complex objects step by step.
* Dependency Injection: Pass required services instead of creating them inside.

1. what’s the difference btw Web Pages (Razor) and MVC and state two business cases and compare btw them?

* Razor Pages: Page-focused, simpler for CRUD, less ceremony.
* MVC: Separation of Model, View, Controller, better for large apps.  
  Business cases:
* Small admin dashboard → Razor Pages (fast/simple).
* Large e-commerce with multiple modules → MVC (structured, scalable).

1. what’s Content type in response message and where we use it and why?

* Tells client how to interpret body (e.g., application/json, text/html).
* Why/Where: APIs use it so browser/app knows whether to parse JSON, show HTML, or download a file.

1. what’s minification, web bundle, webPack and lazy loading of client side and what’s its role in increasing performance through the network?

* Minification: Remove spaces/comments → smaller files.
* Bundling: Combine multiple CSS/JS → fewer requests.
* WebPack: Modern bundler for JS modules/assets.
* Lazy loading: Load resources only when needed (e.g., load images on scroll).  
  Role: All reduce network payload & requests → faster performance.