ASP.Net

Part 1 ❤

Aliaa Ali

1- Add Singletone vs Add Scoped vs Add Transiet

الـ3بيحددو الـ LifeTime بتاع الـ Objectاللي انا هكريته

Add Transient
الـCLR بيكريتObject ويحطه في الـHeap طول فتره الـOperation
بيكريت ف كل مرهNew Object

Add Scoped
الـCLR بيكريتObjectويحطه في الـ Heap طول فتره الـRequest
اول ما Request يخلص بيتمسح من الـHeap

Add Singletone
الـCLR بيكريتObject ويحطه في الـHeap ما انا عامل Run
وكل ماتطلبه هيجيب نفس الـ Object

2- Try , Catch And Finally

ليه بستخدم try في الكود..

بستخدمها لان ممكن يطلع في الكود اللي بكتبه Exception
عشان كده بقول لل C# لو الكود اللي جاي مفيهوش مشاكل نفذيه تمام

طيب لو طلع فيه Exception
روح ع ال Block اللي اسمه Catch

ايه هي وظيفه ال Catch

لما بتلاقي ال Exception.. بتطلع لليوزر شاشه بتقوله ايه اللي حصل مثلا
او تطلب منه بيانات صحيحه

طيب لو انا دلوقتي في ال try بتاعي فتحت Connection مع ال Sql
ولسبب ما حصل Exception في الكود في السطر اللي قبل ماقفل فيه ال Connection

مش المفروض طالما باقى الكود متنفذش اقول لل C# يقفل ال Connectionده
عشان ميعملش مشاكل
لان احتمال افتح Connection مره تانيه فيحصلي مشاكل بدون سبب

عشان كده عملو Finally

دي وظيفتها تعمل Clean Up Resources تمسح كل حاجه ملهاش لازمه
في حاله فشل الكود في ال try…

3- Difference between Abstract Class and Interface

Abstract	Interface
Can have implementation for some of its method	Can`t have implementation for any of it`s method
Can have Fields	Can`t have Fields
Can Inherit from another interface or another abstract class	Can inherit from another interface Only
Abstract class member can have Access Modifiers	interface members can`t have Access Modifiers

4- What is the Sealed Class?

دا عبارة عن Class مينفعش اي Class تانيه ت Inherit منه
على عكس ال Abstract class اللي اصلا مينفعش يتعمل منها Object
ولازم عشان استخدمه اكون عامل Class بتInherit منه

5- Difference between (Public – protected –Private)

لما اعمل Method او Data Member في اي Class واخليها Public
معناها اقدر انادي عليها في اي مكان وكله شايفها عادي

لكن لو Protectedدي معناها ال Derived classesبس هي اللي تقدر تشوفها
زي مثلا الاب مينفعش يروح غير عند عياله بس

ال Private بقى مفيش حد يقدر يشوفه برا ال Class بتاعه

6- Difference between String - String Builder

String → Reference Type Work As Value type

String Builder → Reference Type Work As Reference type

كمان ال String Builder بيكون Mutable بقدر اعدل فيها
لكن ال String بيكون Immutable غير قابل للتعديل فيه

امتى استخدمهم
لو هغير في ال Variable هستخدم String Builder
لكن لو مش هغير استخدم ال String

6- List – LinkedList – Queue – Stack

ال **List** زيها زي ال Array
بس هي Dynamic Allocationبتكبر وتصغر زي مانت حابب

ال **Queue**
زي الطابور اللي بيجي الاول بيمشي الاول
FIFO (First in First Out)

ال **Stack**
زى علبه الاسطوانات كده(CD holder)اللي بيجي الاخر بيطلع الاول
FIFO (First in First Out)

ال **Linked List**
زي السبحه كده
بتتكون من جزئين .. جزء الداتا وجزء ال Pointer اللي بيشاور على Next Element

ال **Double Linked List**
بيتكون من 3 اجزاء
جزء الداتا وجزء ال Pointer اللي بيشاور على Next Element
والجزء التالت ال Pointer بردو بيشاور ع ال Previous Element

8 - Difference between Linked list and Array list

امتى استخدمهم

1-Access

ال **Array List** بتستخدم Direct Access
لكن ال **Linked List** بتستخدم ال Sequential access

عشان مش فيها Index عشان كده هي مش Direct Access

لو عاوز Element معين لازم اعدي على كل اللى قبله
يعني لازم اعدي على كل ال Previous elements
يعني مثلا لو عاوز ال element رقم1000
لازم اعدي ع ال999 اللي قبله

2-Add and remove elements

طيب في حاله المسح والاضافه

ال Array List زيها زي ال String في الحته دي
يعني عشان ازود Element جديد ال C# بتعمل Array جديده بالحجم القديم **1+**
وبعدين ياخد ال Array القديمه ويعملها Paste في Arrayجديده
ويزود ال Elementالجديد ده

نفس الكلام لو جيت امسح
بيعمل Array جديده بالحجم القديم **1-**

فهيبقى الموضوع صعب لو عندي Arrayكبيره وعاوز اضيف element جديد
لكن ال Linked List كل اللي بتعمله بتخلي ال Pointer بتاع اخر Element
بدل ماكان بيشاور على Null بيشاور ع ال Element الجديد

كده نفهم ان لو عاوز اعمل Access استخدم Array List
لكن لو عاوز امسح او اضيف استخدمLinked List

9 - Polymorphism with an example

هوا عبار عن رسالة ببعتها لاكتر من Class وكل منهم بيترجمها
زي مهو عايز

زي متلا لو عندي Parent class اسمها Animal
وعملت Classes بت Inherit منها
اسمهم Snake – Frog – Elephant – Chicken
وعملت Function فى Animal Class سميتها Walk
هترجمها في ال Chicken class واقول بتمشي على رجلين
وفى Elephant class واقول بيمشي على 4
وفي Frog class واقول بينط
وفي Snake class واقول بيزحف
وهكذا....
يبقى هو عباره عن Function في ال Parent Class
ال derived classes بينفذوها بطرق مختلفه

Override, New and Virtual keywords

دي ال Keywords بتاعت ال polymorphism

ال **Virtual** دي فانكشن بتبقى في ال parent واللي هتتغير ف الderived classes
ال **Override** دي الفانكشن اللي كانت في ال Parent
ال **New** دي اسمها Method hiding يعني مش بتغير الفانكشن اللي كانت
في ال Parent انا بسيبها عادي بس بخفيها
وبكتب واحده جديده بنفس الاسم

Overloading and constructor overloading

هو اكتر من فانكشن كلهم ليهم نفس الاسم بس بنفرق بينهم بال Parameters
عددهم وترتيبهم وال Data type بتاعتهم

ال Overloading في ال Constructor
هو عباره عن ان انا بعمل ال Constructor بأكتر من شكل
كل منهم مناسب لشئ معين
في منهم default constructor و copy constructor...الخ

10 - Difference between Primary key and Foreign key

Primary key	Foreign Key
Uniquely identify each row	Refer to a primary key in another table
Doesn't allow null value	Allow null value
Cannot be duplicated	Can be duplicated
Table have single primary key	Table can have multiple foreign keys

11 - Relationship types

1 – One to One
2 – One to Many
3 – Many to Many

12 - How to do the many to many relationship between two tables

عن طريق إننا نعمل جدول جديد
يكون فيه ال Primary Key بتاع الجدول الاولاني كا Foreign key
وال Primary Key بتاع الجدول التاني كا Foreign Key
ويبقى ال Primary key الاثنين Foreign key دول

13 - Difference between where clause and having

الاتنين زي بعض بس الفرق ان Where مش بتشتغل مع Aggregate Function
لكن Having دي اصلا وظيفتها

14 - What is the index
• Difference between cluster and noncluster index

ال index لتسهيل عمليه ال Access لاي داتا موجوده في ال Table
زي مثلا الفهرس الموجود في الكتاب كده

أهم نوعي فيه هما ال Cluster و Noncluster

ال **Cluster** هو الترتيب الفعلي في ال Physical data
يعني مثلا لو عندي ال Id اللي هو Primary key لو انا عملت عليه index وخليته cluster
هيبقى الترتيب الفعلي بتاعهم 1وبعدين 2 وبعدين 3 وبعدين مثلا 9... فالاكبر
لان مثلا ال IDs اللي بين ال3 و 9 مش موجود

طيب انا عاوز اعمل insert ل rowجديد ال id بتاعه مثلا 7
اللي هو اكبر من 3 واصغر من 9
فلما اعمل insert المتوقع ان ال rowالجديد دا هيكون ف اخر الجدول
بس ال Cluster هيتدخل وهيعيد ترتيبهم م الاول
عشان ال row الجديده دي تكون بين ال row اللي ال id بتاعها 3 و 9

ال **Noncluster** بقى
بيرتبهم Logically
زي مثلا الكتاب الفهرس بتاعه مش مترتب ابجدي لكن مترتب على حسب كل موضوع
وموجود في الصفحه الكام

يعني مثلا لو عندي جدول فيه ال column بتاع ال salary دا
وعندي فيه القيم 200 بعدين 220 بعدين 190 بعدين 500 بعدين 120
وعاوز اعمل insert ل rowجديد ال salary بتاعه 50
هنلاقيه عمل nsert في الاخر خالص

طيب لو كنا شغالين Cluster كان زمانه حطها ف اول الجدول خالص لانها اصغر قيمه
او حسب ترتيبها ف الجدول يعني

15 – What is Normalization

هو عباره عن تحويل ال bad schema ل good schema
او تحويل ال bad relationship ل good relationship
وظيفته ان لو عندي حاجه متكرره ف الجدول يشيلها

التكرار هنا مش قصده منقصدش بيه ان ال rows بتكرر نفسها
يعني مش wrong data entry لانها دي ممكن تتعالج ب constraint
لكن التكرار هنا نقصد بيه ذكر العنصر الواحد اكتر من مره

ID	Name	Department	Department Head
1	Tarek	CS	Abdul-Hamid
2	Ahmed	IS	Maher Ali
3	Ali	IS	Maher Ali
4	Basel	CS	Abdul-Hamid
5	Ziad	CS	Abdul-Hamid
6	Hany	IS	Maher Ali

في الجدول دا عندي حاجتين القسم ورئيس القسم
لو غيرنا مثلا ال CS ده هنروح نغيره في كل row بادينا
طيب لو عندي بقي 10000 Rowصعب نغير بأدينا كل ده وممكن ننسى واحد او اتنين
فأصبح الاسم القديم موجود عندي في كذا Rowمن اللي ناسيه وهو لا يدل على شئ

هنا بقى تبقى اسمها data inconsistency
ودي وظيفه Normalization .. ينضف بقى الليله دي كلها… عن طريق انه يقسمهم لجدولين ويعمل بينهمRelationship

في عندي بقى 3 انواع من Normalization
الاول NF1
دا بننفذه في حاله واحده بس…. Atomic data
يعني ايه Atomic data
يعني data منفصله زي كدا

ID	Movie	Actor 1	Actor 2	Actor 3
1	Awesome	Adam	Marie	John
2	Cool	Jim	Adam	Helbert
3	Wonderful	Marie	Leonardo	

هنا اسمها Atomic لان الداتا متجزأه واتكرر فيها اسماء وفي اسماء متحطتش ب null يعني

طيب لو مش Atomic هيبقى شكلها عامل ازاي

ID	Movie	Actors
1	Awesome	Adam, Marie, John
2	Cool	Jim, Adam, Helbert
3	Wonderful	Marie, Leonardo

طيب بنطبق NF1 فبنقسم الجدول دا لجدولين
الجدول الاول فيه اسما الممثلين و ال IDs بتاعتهم
والجدول التاني بيمثل ال relationship

ID	Actors
1	Adam
2	Marie
3	John
4	Jim
5	Helbert
6	Leonardo

دا الجدول الاول
طيب المفروض ال Relationship دي تكون ايه
ايوة بالظبط Many to Many

ID	Movie	Actor ID
1	Awesome	1
1	Awesome	2

1	Awesome	3
2	Cool	4
2	Cool	1
2	Cool	5
3	Wonderful	2
3	Wonderful	6

طيب .. ال NF2 .. ايه هي ؟!
لما الجدول بتاعي دا يكون فيه column مش بتعتمد إعتماد كلي على ال Primary key
زي كدا

ID	Name	Department	Department Head
1	Tarek	CS	Abdul-Hamid
2	Ahmed	IS	Maher Ali
3	Ali	IS	Maher Ali
4	Basel	CS	Abdul-Hamid
5	Ziad	CS	Abdul-Hamid
6	Hany	IS	Maher Ali

هنا رئيس القسم .. بيعتمد على القسم نفسه .. مش على ال Primary key
لذلك لو انا غيرت في القسم ... لازم اغير رئيس القسم بتاعه
تمام كدا ؟!
دي عشان نتعمل .. بنقسم الجدول دا لاتنين .. الاول بيشيل معلومات الأقسام .. والتاني بيشيل معلومات الطلبة
relation بال

ID	Department Name	Department Head
1	CS	Abdul-Hamid
2	IS	Maher Ali

الجدول التاني

ID	Name	Department ID
1	Tarek	1
2	Ahmed	2
3	Ali	2

4	Basel	1
5	Ziad	1
6	Hany	2

تمام ؟!

طيب ال NF3
دى فى حالة الcolumn اللى مش بيعتمد نهائياً على ال primary key .. وفى نفس الوقت بيعتمد على column تانى ..
خلي بالك .. في ال NF2 .. قولنا مش بيعتمد إعتماد كلي .. هنا قولنا نهائيا

Order ID	Customer Number	Unit Price	Quantity	Total
1	202	10$	2	20$
2	203	9$	20	180$
3	204	19$	1	19$
4	205	12$	10	120$

هنا الColumn بتاع ال Total . مش بيعتمد على Primary key ..
دا كمان ممكن بتحسب .. مش لازم بتخزن .. انا ممكن وانا بعرض الريبورت احسبها في الريبورت
إذن انا ضيعت مساحه عالفاضي .. مش كدا ؟!
D: ال NF3 .. بقا بيقول .. امسح الحاجات اللي زى دى .. مش تاقصاهم هي
فا بعد ما نطبق ال NF3 عالجدول دا
هتبقا كدا

Order ID	Customer Number	Unit Price	Quantity
1	202	10$	2
2	203	9$	20
3	204	19$	1
4	205	12$	10

حد حس بحاجه ؟!
كدا خلصت الNF3

16 – What is Cursors

ال Cursor دا حاجه زي ال pointer كده بيشير ل row معين
ومن خلاله اقدر اعمل Loop تمشي row y row

يعني مثلا لو عندي جدول فيه id , product name , unit price
وشويه Columns تانيه خاصه بكل وحده
وفرضنا الجدول فيه 10000 منتج... هما 3 منتجات بس... بس عندنا 10000 وحده م المنتجات دي.. زي مثلا سوبر ماركت معندوش غير 3 انواع من البضاعه بس عنده 10000 قطعه منهم

وعاوز اعمل update للاسعار دي
وعلى سبيل المثال عندنا 3 منتجات .. اسمهم 65و 66و67
وعاوز احدد سعر لكل منهم .. واقول لو اسم المنتج65 خلي سعر الوحده بيساوي 30 .. و لو اسمه 66 خلي سعر الوحده يساوي 40 .. ولو اسمه 67 خلي سعر الوحده 60
سعر الوحده

هنا بقا هحتاج حاجه زي اوبجيكت تكون بتشيل كل حاجه ف ال row
او بتشير ليه بحيث اعمل retrieve للداتا في كل row على حدى

ال Cursor بقى بيعمل كده
بيشير لل row ويخليه يعملretrieve للداتا اللي فيه
فهنا نقدر نعمل loop تمشي بيها ع ال rows اللي في الجدول row by row

17 - Difference between stored procedure and user defined function

Stored Procedure	User Defined Function
Can return Zero or n values	Return only one value, which is mandatory
Can have input/output parameters for it	Have only input parameters for it
Allows DML in it (Data manipulation language) select, insert, update, delete-	Allows only select statements
Cannot be called from a function	Can be called from a procedure
Exceptions can be handled by try catch block &	catch block & Cannot use try
Cannot be embedded in select statement	Can be embedded in select statement

18 – What is XML

عباره عن لغه بتستخدم ال **tags** لتخزين البيانات في شكل Hierarchy

```
<Persons>
    <Person>
        <ID> 1 <ID>
        <Name> Aliaa <Name>
    <Person>
<Persons>
```

مميزاتها انها Unified يعني مش بتحتاج برنامج معين عشان يقراها
وكمان سهله التعامل والتخزين والنقل عبر البرامج

19 – What is UML

ال UML مش لغه
هي طريقه لل Analysis and design
بنستخدمها ل Design برنامج

ليها 9 انواع

- Business Use Case diagram
- Use Case diagram
- Activity diagram
- Sequence diagram
- Collaboration diagram
- Class diagram
- Statechart diagram
- Component diagram
- Deployment diagram

20 - Difference between Authentication and Authorization

سؤال مهم

Authentication

بتجاوب ع السؤال ده who are you
يعني بتسأله عن اسمه رقم الباسورد
يبقى الAuthentication بتتأكد هو دا اليوزر ولا لا
طب لو هو يجي بقى دور ال

Authorization

بتجاوب ع السؤال ده what to do
يعني ايه ال Limits اللي عليه
يعني زائر ادي ولا موظف ولا مدير

21 - Validation controls

عندنا 6 انواع

1- Required field validator
2- Range validator
3- Compare validator
4- Regular expression validator
5- Custom validator
6- Validation summary

دي كلها بتحصل في ال Client side مش في ال Server Side

22 - Difference between ASP and ASP.Net

ASP	ASP.Net
only two languages available for scripting .VBScript and Jscript/Javascript	Any fully compliant .NET language can now be used with ASP.NET, including C# and .VB.NET
ASP is interpreted	ASP.NET is compiled
Classic ASP uses a technology called ADO to .connect and work with databases	ASP.NET uses the ADO.NET technology
ASP has Mixed HTML and coding logic	asp.net html and coding part are separated .by code behind files
ASP is partially object oriented	ASP.NET purely object oriented
ASP No in-built support for XML	ASP.NET full XML Support for easy data .exchange

23- What are the View State, Session State and Application State

وهيسألك كل واحده بتتخزن فين

ال View State

- دا Variable بيستخدم لحفظ البيانات في نفس ال Page او ال Webform في حاله الpostback يعني لو اتنقلنا ل Page تانيه خلاص كده القيمه اللي كانت مخزناها ال View state اتمسحت
- ودي بتتخزن في ال Page في hidden field اسمه ViewState
- وفعليا كل ال ASP.Net Controls بتستخدم ال view state دا by default

ال Session State

دي بتستخدم لحفظ البيانات وسهوله نقلها بين كل ال Pages
بس لنفس اليوزر
بمجرد مايختلف اليوزر او البراوزر بتتمسح ال Session state دي
ودي بتتخزن في ال Web Server مش في ال Pages
وبتتمسح زي ماقولنا بمجرد ماليوزر يقفل او Time Out ودا ال Defaultبتاعه 20 دقيقة

ال Application State

دي بتستخدم لحفظ البيانات ونقلها بين كل ال Pages وبين كل اليوزرس
بتتخزن بردو في ال Web Server
وبتتمسح لما ال Application يتعملها Restart

24- Difference between view - function - stored procedure

ال View

ال View بتمثل جدول افتراضي مش موجود فعلا في data base.
بتظهرلك البيانات من خلال استعلام مخزن في ال View.

طيب ايه هي استخدامات ال view:
قالك بتستخدم ف تبسيط عملية إدارة ال queries المعقدة وإعادة استخدامها.
وكمان ف الأمان والرقابة على الوصول للبيانات. باننا بندي صلاجيه للمستخدمين ع ال view بدل الجداول الاساسيه الأساسية.
وانه ينفع نغيير ال view من غير م يأثر على البيانات

ال Function

ال Function زي الإجراء بتقوم بعملية معينة وبترجع قيمة.
ممكن تاخد مدخلات وترجع مخرجات.

طيب ايه هي استخدامات ال Function:
تستخدم ف إعادة استخدام الكود: يعني اننا ممكن نستخدم ال function دي اكتر من مره ب مدخلات مختلفه
وبتبسط ال queries المعقده بحيث اننا نحطها ف function ونستدعيها بسهوله وخلاص

ال Stored Procedure

ال Stored Procedure عبارة عن سلسلة أوامر SQL مخزنة في data base.
بتنفذ الأوامر عند استدعائها وممكن تاخد مدخلات وترجع مخرجات.

طيب ايه هي استخدامات ال Stored Procedure:
قالك برضو زياده ف الامان :بحيث اننا بندي صلاحيات محدده لل users لتنفيذ ال stored procedures بدل الوصول المباشر ل table.
وقالك اهم نقطه
تقليل حركة الشبكة: في حالة وجود عمليات معقدة بتتم على server، بيؤدي استخدام ال stored procedures إلى تنفيذ الكود مرة واحدة بدلا من كل مرة يتم فيها استدعاء تلك العمليات من التطبيق.
سهوله العديل: التعديل السهل ع ال Stored Procedure من غير م يأثر ع اي تطبيق.
وبيستخدم خوارزميات معينه ف تحسين اداء ال Stored Procedure ف ال data base

25 – Difference between IEnumerable Vs IQueryable

تخيل معايا ان انا عندي Table employee فيه row 1000 ونا عايز ارجع ال employees الي ال salary بتاعهم اكبر من 2000

```
IEnumerable<Employee> employees = context.Employees;
var EmpWithsalaryGT2000 = employees.Where(e => e.Salary > 2000);
```

بالشكل ده كده هيروح يجيب كل الموظفين من الداتا بيز ويحطهم عندي ف ال memory وبعد كده يروح يعمل الفلتريشن ال انا طالبها منه ف انت مخيل اللفه ال بيعملها
ولو معاك داتا كبيره هيحصل اي طبعا الكلام ده وانا بتعامل مع ال EF

اومال لو استخدمنا ال IQueryable ف المثال ده هيحصل اي ؟

قالك لا هو كده هيروح يجيب الداتا ويعملها فلتريشن هناك ف الداتا بيز ويرجعلي ف الداتا ال انا محتاجها لانه بيشتغل بطريقه ال (deferred execution) وكده زي الفل شوفت الفرق
طيب هل ده معناه ان ال IEnumerable مش بيشتغل ب (deferred execution)

هما الاتنين بيعملو defered عادي بس ال IQueryable مش بيتعامل مع delegates بطريقه مباشره زي ال IEnumerable هو بيتعامل مع ال Expression وهو
بيتعامل مع ال delegates بيجمع ال query كلها في ال Expression وبينفذها في السيرفر على عكس ال IEnumerable بيتعامل مباشر مع ال delegates
فيجيب الداتا لل client وينفذ ال extended query إنما الاتنين عندهم support لل defered execution عادي

طيب م خلاص ي هندسه انا كده مستخدمش IEnumerable قالك لا هنستخدمه بس مع الداتا ال عندي ف ال memory عندك مثلا dictionary , list كده هستخدم IEnumerable وبرضو مع ال streaming operation

ال IEnumerable بتفيد جدا لما تتعامل مع مجموعة بيانات كبيرة وعايز توفر استهلاك ال memory. يعني مثلا لو عندك قاعدة بيانات كبيرة وعايز تعمل عمليات معينة على البيانات، ممكن تستخدم الـIEnumerable مع الـStreaming Operations.

ال Streaming Operations هي عمليات بتشتغل على البيانات واحدة ورا واحدة من غير ما تحمل البيانات كلها في الذاكرة. يعني بتقدر تعالج البيانات بالتدريج
ومش بتحملها كلها في ال memory وبتوفر في استهلاك ال memory.

في مثال بسيط، لو عندك قاعدة بيانات فيها مليون سجل، وعايز تطبق على كل سجل عملية معينة، بتستخدم الـIEnumerable مع ال Streaming Operations عشان متحملش المليون سجل في الذاكرة مرة واحدة، هتجيب السجلات واحدة ورا واحدة وتعالجها.

ايه هي ال Solid Principles وايه الفرق بينها وبين ال Design Pattern

ال solid principles هي مجموعة من المبادئ بتحسن جوده ال software بتاعي عن طريق كتابة كود clean وسهل ف الصيانه والتطوير.

الـ SOLID principles تتكون من:

ال S - Single responsibility principle

يعني كل كلاس أو function لازم يكون له مسؤولية واحدة بس، ما يشتتش الfocus.

ال O - Open closed principle

يعني الكلاسات والfunctions لازم تكون open for extension بس closed for modification. يعني أقدر أضيف عليها features جديدة من غير ما أعدل في الكود الأساسي.

ال L - Liskov substitution principle

الsubclass لازم يقدر يحل محل الparent class من غير ما يبوظ أي حاجة.

ال I - Interface segregation principle

لازم أجزئ الinterfaces عشان كل كلاس يستخدم اللي هو محتاجه بس مش كله ف interface واحد.

ال D - Dependency inversion principle

يعني الكلاسات ماتعتمدش على implementation details بل على abstractions.
والاعتمادات بتكون على interfaces بدل classes.

طيب ايه الفرق بينها وبين ال design pattern

قالك ال design patterns هي حلول لمشاكل برمجية شائعة و متكرره .
هي بتوفرلك templates جاهزة لحل المشكلات دي زي singleton وfactory وobserver وغيرها.

يعني ممكن تكتب كود يطبق design patterns بدون ما يطبق SOLID principles وده مش كويس. لكن SOLID principles بتعتبر أساس سليم لأي كود هتكتبه.

- **مفكرتش قبل كده ليه ف ال api مش بستخدم Cookie زي ال MVC**
بدل الحوارات الكتير بتاعت ال Token دي

ال token بيكون أكتر أمانا من ال cookie ليه ؟
علشان ف ال web api انت مش بتتعامل direct مع ال provider لا انت بتتعامل مع ال consumer عن طريق ان ال provider بي check authentication ويبعتها لل consumer وال consumer هو ال بيحدد ال token تتخزن ازاي ف ال client side

ال token مش هتحتاج تخزنه في ال browser زي ال cookie علشان بيتبعت في كل request في ال header وده بيخليه أسهل في الاستخدام مع ال APIs.

ال token بيدي مرونة أكتر في التحكم في ال access وال permissions علشان ممكن تضيف تفاصيل اكتر في ال payload بتاعته بدل من مجرد user id زي ال cookie.

ال token بيشتغل على معظم ال platforms وال languages بدون مشاكل ع عكس ال cookie.

ممكن تستخدم ال token في ال authentication وال authorization في نفس الوقت بدل م تستخدام cookie
لل authentication و token لل authorization.

ال token بيسهل تطبيق مفاهيم مهمة زي ال statelessness في ال REST APIs.

مع ال token ممكن تستخدم حاجات زي ال refresh token عشان تجدد ال access بسهولة.

ف من الاخر ال token ملاءم اكتر ل APIs من حيث ال ال security وال scalability