LEKCE 04

Testování kódu

```
git pull template main --allow-unrelated-histories
```

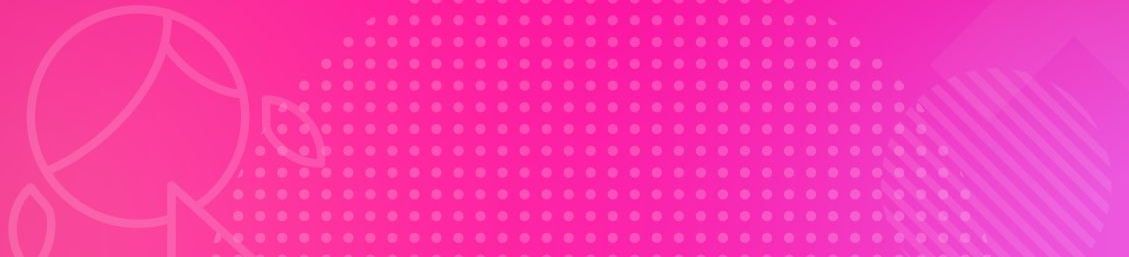
Lekce 04 - Testování kódu

0. Routing, DTO
1. Co je to **Unit Testing**
2. Testovací Frameworky
3. Jak psát testy?
4. Jak unit testovat REST API controller pomocí **xUnit**
5. Mockování závislostí pomocí **NSubstitute**

Naučíme se testovat kód a odhalíme benefity testování

czechitas

Routing



Co je to Routing?

Routing rozhoduje, jaká akce nebo část kódu se má spustit na základě cesty zadané v URL a podle použité HTTP metody (GET, POST, PUT, DELETE apod.).

DTO

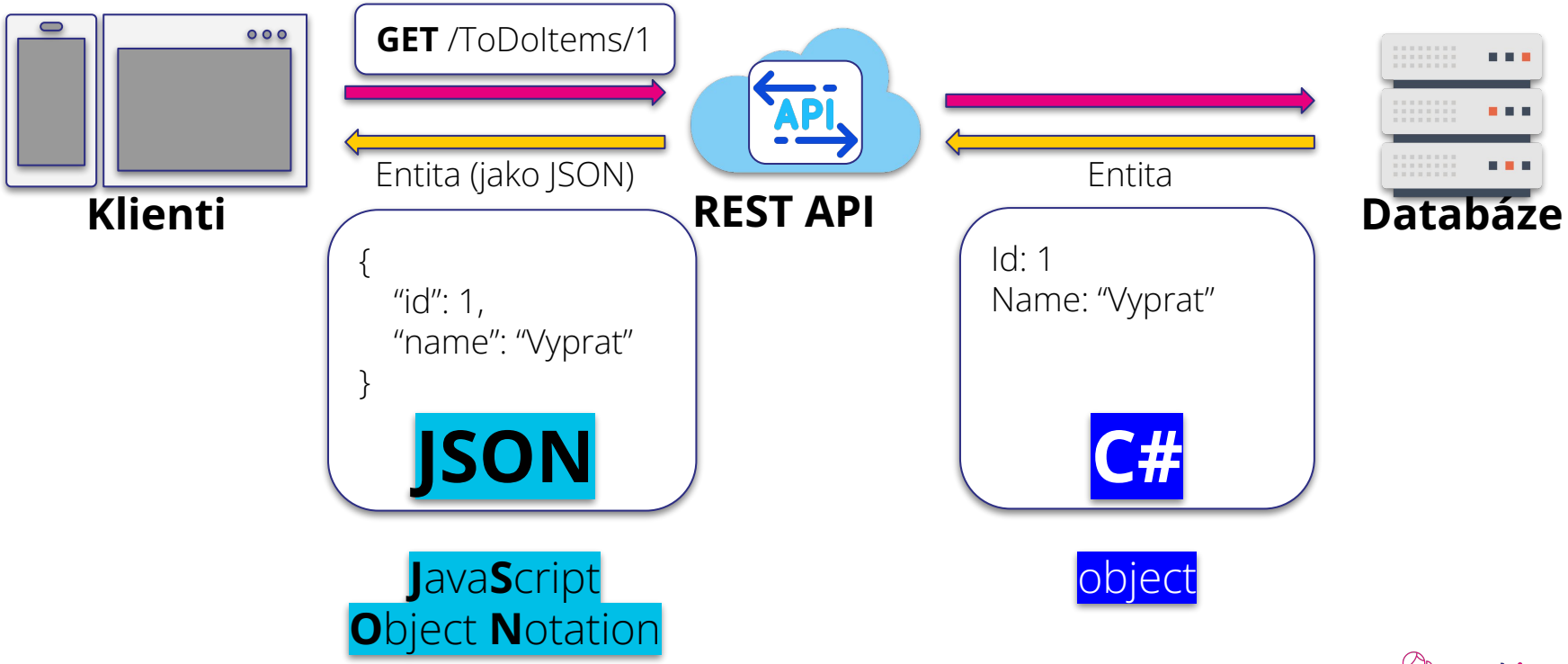
Data **T**ransfer **O**bject

Co je to DTO?

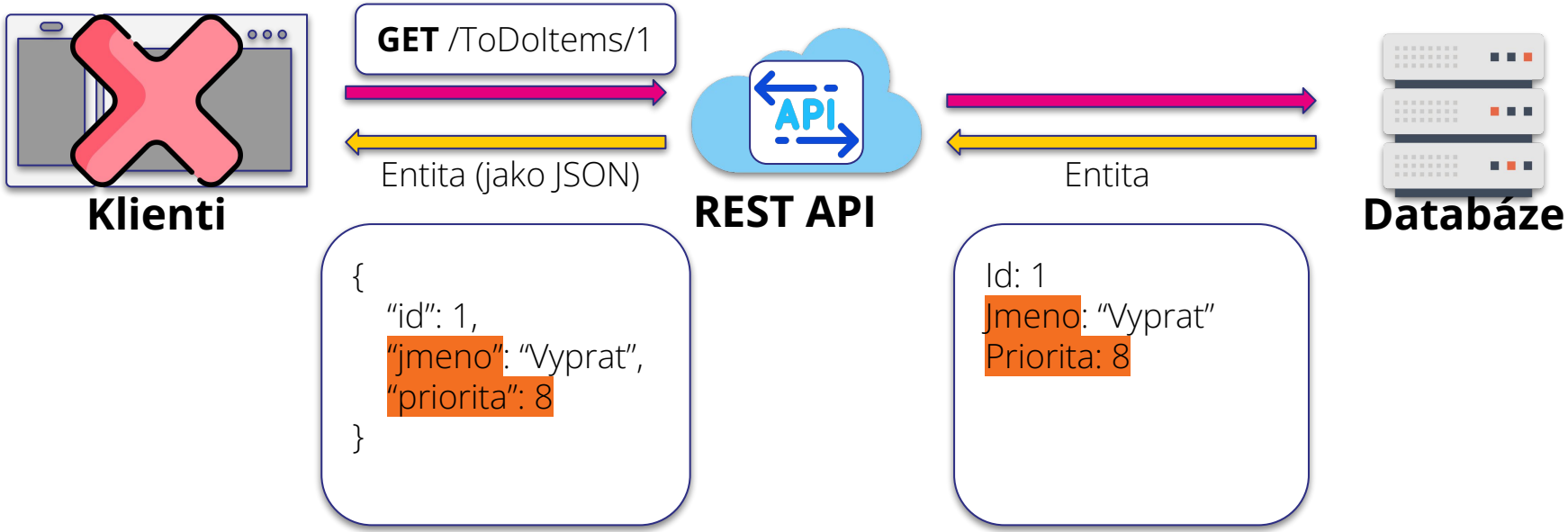
Data Transfer Object (DTO) je takový objekt,
který přenáší data mezi procesy nebo aplikacemi.

V kontextu REST API, DTO může být považován jako
kontrakt mezi klientem a serverem

Proč používat Data Transfer Objects?



Proč používat Data Transfer Objects?



Proč používat Data Transfer Objects?

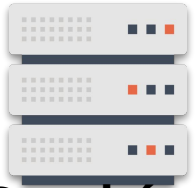
Klienti

GET /ToDoItems/1

DTO == Kontrakt

REST API

Entita



Databáze

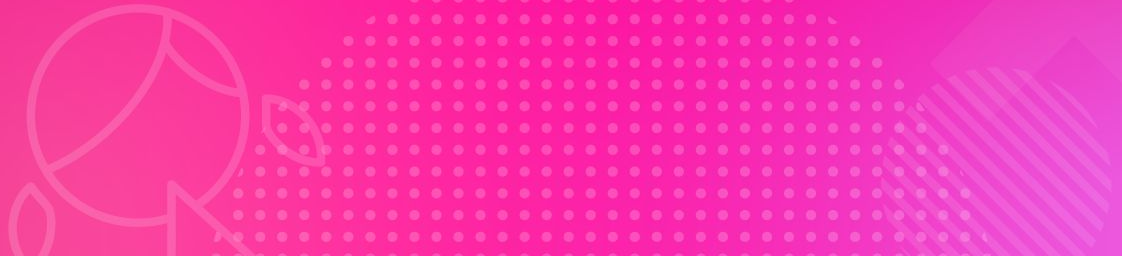
```
{
   “id”: 1,
   “name”: “Vyprat”
}
```

Id: 1
Jmeno: “Vyprat”
Priorita: 8

DTO se chová jako kontrakt, který definuje očekávání a požadavky na to, jak se budou posílat a přijímat data mezi klientem a serverem.

czechitas

Testování



Pojďme otestovat tuto raketu

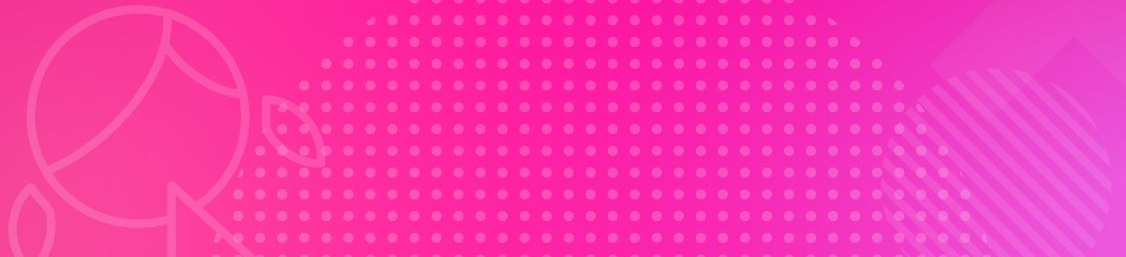
czechitas

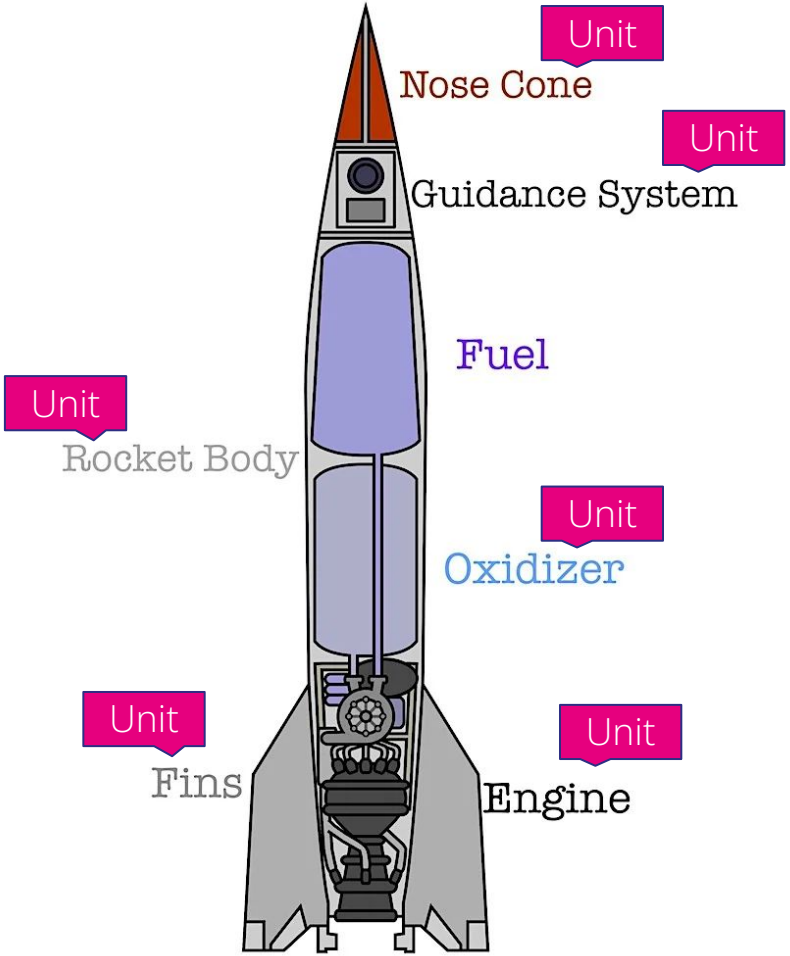
Co by se mohlo pokazit?

czechitas

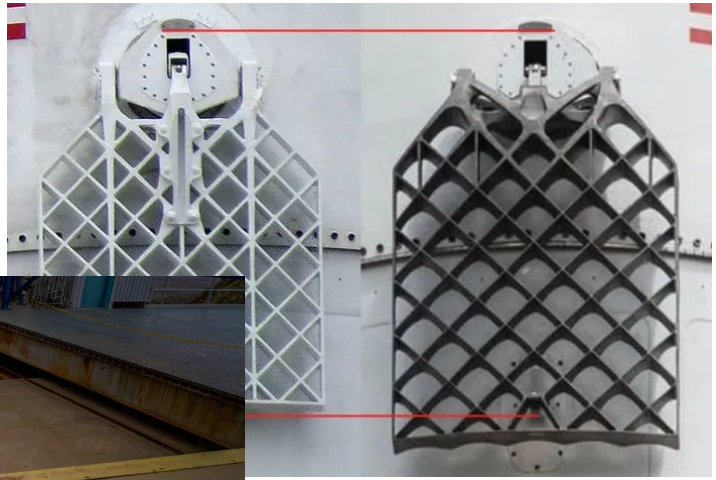

SPEED
ALTITUDE
98 KM
T+00:03:20
SPEED
ALTITUDE
95 KM

Unit Testing




Unit
Nose Cone
Unit
Guidance System
Fuel
Unit
Rocket Body
Unit
Oxidizer
Unit
Fins
Unit
Engine

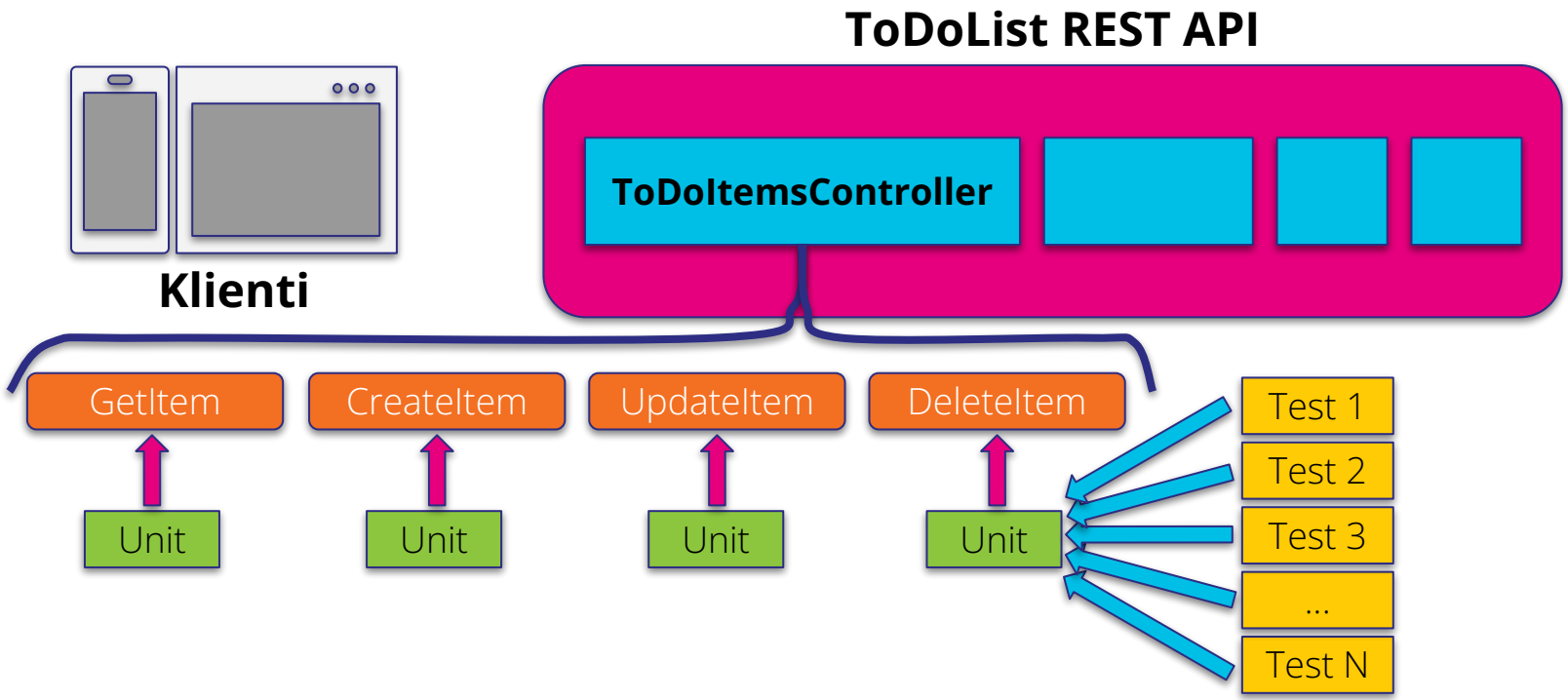
Units



Co je to Unit Testing?

Unit Testing je testování kódu,
kdy každá část kódu je testována v izolaci bez externích závislostí

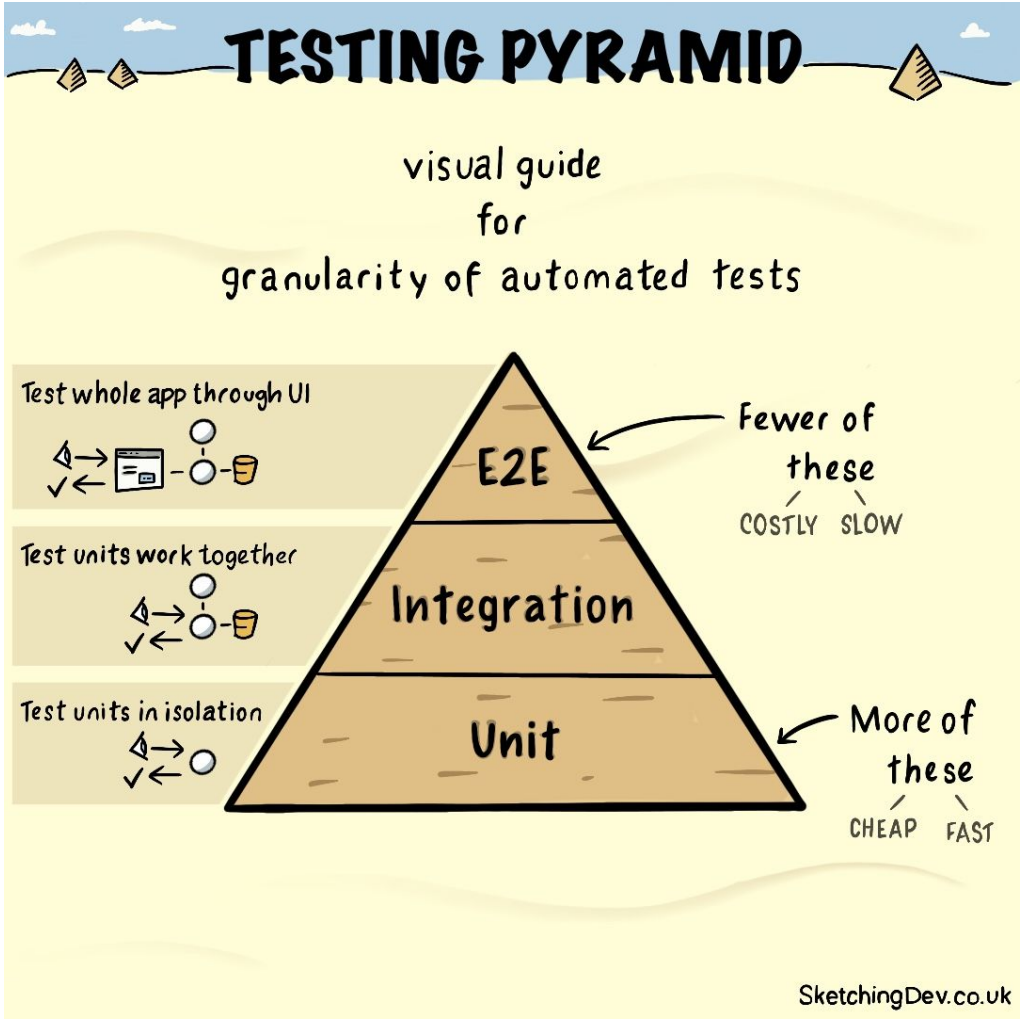
Unit Testing REST API



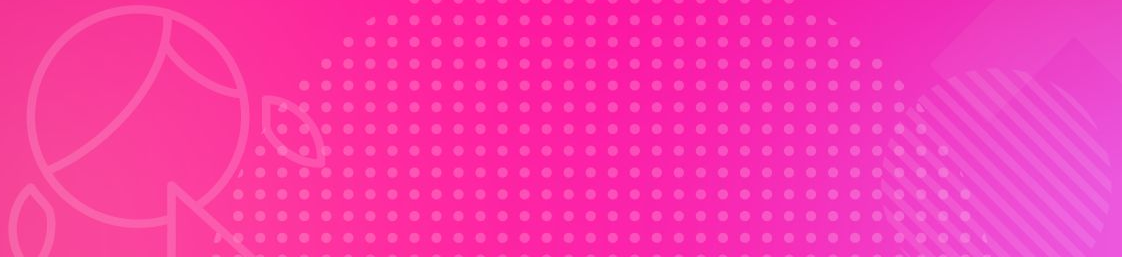
czechitas

Proč unit testovat?

- Rychlá kontrola správnosti kódu
- Možnost dělat změny v kódu bez obav
- Snížení dopadu chyb
- Živá dokumentace a specifikace


TESTING PYRAMID
visual guide
for
granularity of automated tests
Test whole app through UI
Test units work together
Test units in isolation
E2E
Integration
Unit
Fewer of these
COSTLY
SLOW
More of these
CHEAP
FAST
SketchingDev.co.uk

Testovací frameworky



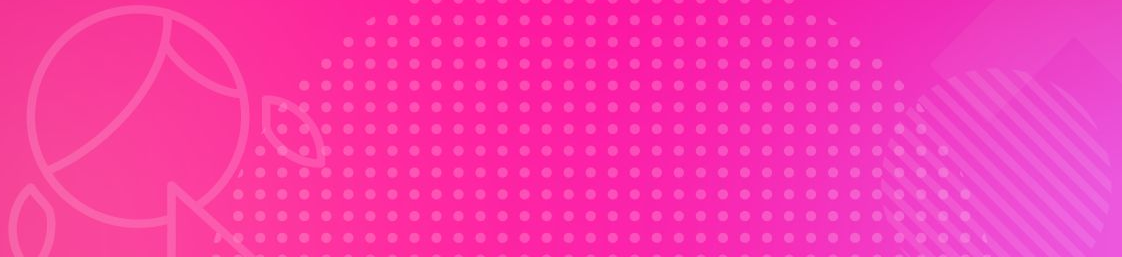
Testovací frameworky v .NET

MSTest

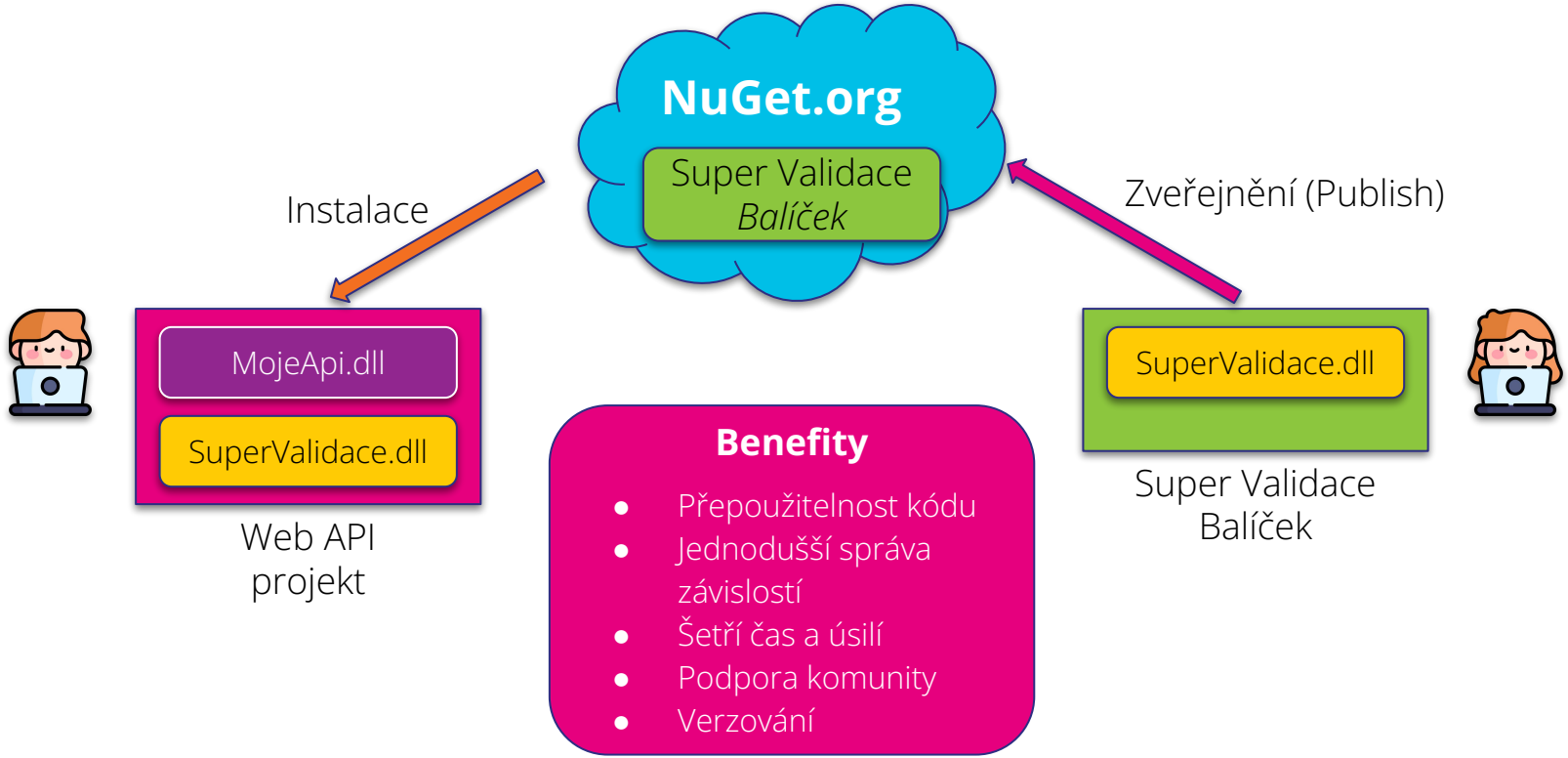
xUnit.net

Má preference

NuGet



Jak sdílet kód v .NET?



czechitas

Jak psát testy?

AAA pattern

Arrange

Příprava kódu pro test

Act

Vykonání akce, kterou chceme testovat

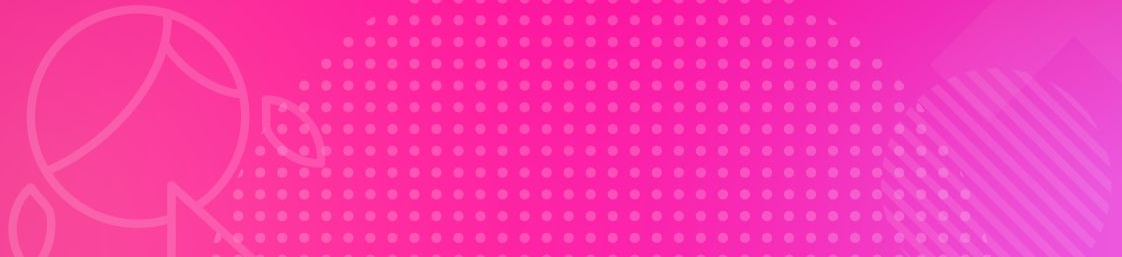
Assert

Kontrola výstupu vykonané akce

AAA pattern - ukázka v xUnit

```
[Fact]
public CalculatorSumReturnsCorrectSumOfThreeNumbers()
{
    // Arrange
    var calculator = new Calculator();
    // Act
    var result = calculator.Sum(1,2,3);
    // Assert
    Assert.Equals(result, 6);
    //result.Should().Be(6); //FluentAssertions
}
```

Mocking



Co je to Mockování?

Mockování je proces vytváření objektů (Mocků), které simulují chování reálných objektů.

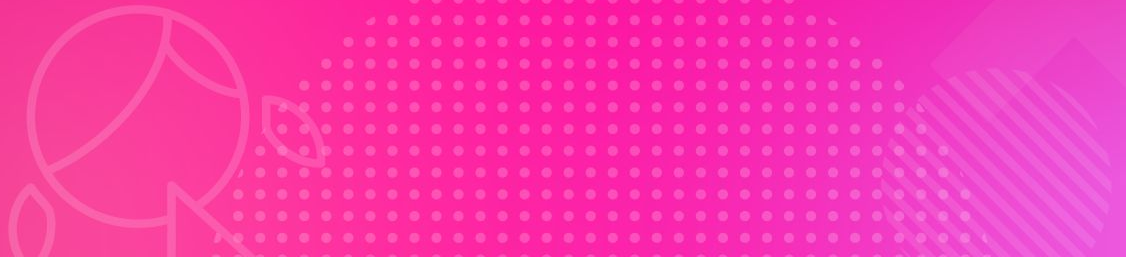
Tyto Mocky lze v testovacím prostředí upravovat a nastavovat jejich chování.

EXTRA: Typy zástupných objektů

- Dummy
- Stub
- Spy
- Mock
- Fake

Pro naše použití používáme Mocky!

Mockovací frameworky



Mocking frameworks v .NET

Má preference

Tipy na závěr

Tipy

- Testujeme pouze **jeden scénář**
- Jméno testu mi přesně říká, co testuje (e.g. Metoda_Scénář_Výsledek)
- Nejjednodušší procházející test
- Konzistentní struktura
- Co nejméně logiky
- Debugger je náš kamarád
- [Theory]
- FluentAssertions