

# Table of Contents

[Řešení Alza testovacího případu](#) [Řešení Alza testovacího případu](#)

[Zadání Alza testovacího případu](#) [Zadání Alza testovacího případu](#)

Tento projekt je řešením pro zadání Alza case study. Cílem je navrhnout plně automatizovanou test suitu pomocí .NET Core a C#. Běhové prostředí musí být kompatibilní s CI/CD.

## Instalace

Dle požadavků je test suite napsána pomocí .NET Core a C#, řešení používá .NET 6. Jako IDE bylo použito Visual Studio i VS Code.

Visual Studio - instalace probíhá automaticky po otevření AlzaTest.sln

VS Code - je potřeba otevřít složku projektu -> Open Folder -> AlzaTest případně použít příkaz:

- run: dotnet build ./AlzaTest.sln

Lokace na GitHubu <https://github.com/voigry/AlzaTest.git>

## CI/CD

V průběhu vývoje byly využity GitHub akce. Projekt byl po každém commit do master sestaven a otestován v prostředí ubuntu s verzí .NET 5.0.x (viz main.yaml).

## Jak spustit test

Test lze spustit pomocí Nunit test runneru. Ve Visual Studiu použijeme Test Explorer. Ve VS Code je k dispozici doplněk .NET Test Explorer. Nebo lze jednoduše použít cli příkaz dotnet test, např: dotnet test --filter "FullyQualifiedName~AlzaTest.Tests.TestZadaniPositive"

Na výběr je několik testů. Každý test rozšiřuje třídu AlzaBaseTest. Díky tomu je docíleno snadné rozšiřitelnosti o nové testy a znovu použitelnosti vlastností a metod.

### Test pozice softwarový tester - pozitivní testování

Spustí se pomocí příkazu: dotnet test --filter "FullyQualifiedName~AlzaTest.Tests.TestPositionSoftwareTesterPositive" Test obsahuje ověření hodnot pracovní pozice, jako popis a jednotlivé položky pozice, lokace a s kým se aplikant potká na pohovoru.

#### Možné chyby:

U popisu pozice nesedí velká-malá písmena.

### Test pozice softwarový tester - negativní testování

Spustí se pomocí příkazu: dotnet test --filter "FullyQualifiedName~AlzaTest.Tests.TestPositionSoftwareTesterNegative" Zde se testuje nevalidní country code a segment. V obou případech se očekává status odpovědi not found.

#### Možné chyby:

Při požití country kódu CzechiaCZ se očekává status odpovědi not found, aktuálně se ale vrací Ok.

## Logování

Primárně se loguje se pomocí knihovny Log4net, zde je zejména využita možnost kontinuálního logování logů z různých testů do jednoho souboru.

### Log4net

Podrobné nastavení včetně nastavení cesty k logům se nachází v log4net.config. Samotná inicializace log4net je implementována ve třídě SetupTrace, kde se nastavuje cesta k log4net.config souboru a properta LogFileName, která

nastavuje absolutní cestu k logovacímu souboru. Při nastavení těchto cest je brán ohled na nezávislost na operačním systému. Díky tomu, že je Log4net nastaven v kontextu OneTimeSetup je logger automaticky k dispozici ve všech testech nebo testFixtures a jednotlivé zápisy jsou zobrazeny při běhu v testovací konzoli (např. Test Explorer pro Visual Studio).

### Trace a další Loggers

Cesta k výsledkům a jiné vlastnosti jsou nastaveny v runsettings. Výsledky testů jsou zapisovány do souboru html a trx. Je možno logovat i do console, nicméně tuto možnost ponechávám zakomentovanou.

# Zadání:

Navrhnout plně automatizovanou test suitu pomocí .NET Core a C#. Běhové prostředí kompatibilní s CI/CD.

## Scope:

na stránkách [www.alza.cz/kariera](http://www.alza.cz/kariera) chceme najít inzerát na pozici software tester pro QA oddělení, kde zkontrolujeme, že inzerát obsahuje všechny potřebné informace pro kandidáta k pohovoru. (vyplněný popis pozice, fotka lidí, které potkáte na pohovoru a jejich krátký popis...).

Máme veřejný end point

<https://webapi.alza.cz/api/career/v2/positions/softwareovy-tester?country=cz>

## Zjistit výsledek:

- Vyplněný popis pozice
- Kde bude pracovat
- S kým se na pohovoru setká a co se o něm doví.
- Jestli je to práce pro studenty
- Požadavky: (Pokuste se dodržet zadání, doporučené je splnit všechny, není však povinné)
- Řešení je verzováno pomocí Gitu uloženo na GitHub a spustitelné
- Řešení používá .NET 5.0
- Řešení používá NUnit jako test runner
- Řešení používá REST API klient library RestSharp
- Doporučený nástroj pro vývoj je IDE - VS Code
- Řešení je zdokumentováno pomocí automaticky generované dokumentace (docfx)
- Řešení nabízí config, ve kterém je možné definovat cesty k logům
- Řešení podporuje logování pro snadnější debug a
- reportování. Logování průběhu testu
- Ověřit zachycení neplatného url segmentu. Např: jiná prac. pozice, jazyk