Toggle Service Sample

#Sara Silva

Agenda

- Context
- Feature Toggle Overview
- Toggle Service solution
- Aditional References

Context

This presentation has the goal to expose the solution related with the "Toggle Service Exercise" provided.

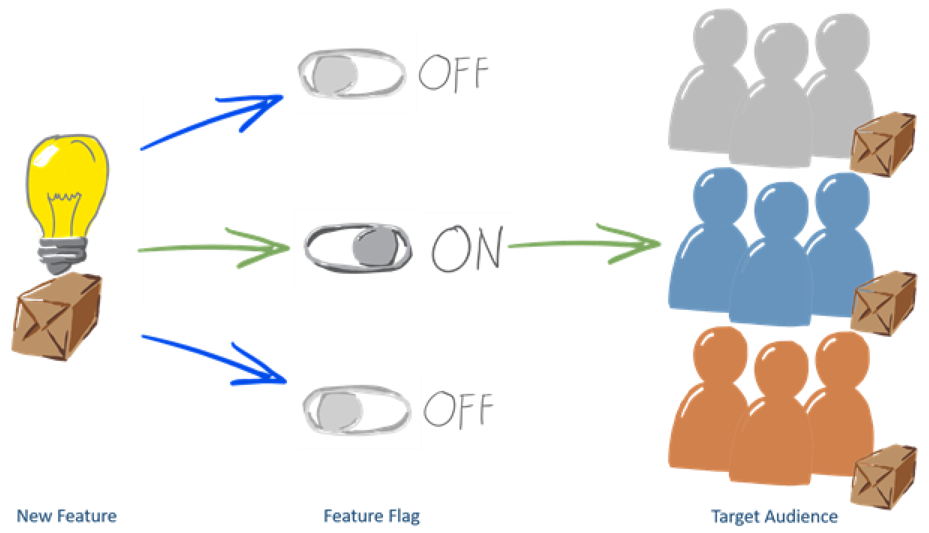
"Company X P T O
has a digital platform built under SOA and currently with 62 services/applications built withi n. All new features are implemented with system toggle so they can quickly deliver new val ue, and if something goes wrong, they just toggle it off. Currently, whenever X P T O

needs to change the toggle values in live, they need to change a file with the toggle proper ties values and restart the service/application so it can take effect."

…

Feature Toggle Overview

Feature Toggle Overview



A **feature toggle** (also feature switch, feature flag, feature flipper, conditional feature, etc.) **is a technique in software development** that attempts to provide an alternative to maintaining multiple source-code branches (known as feature branches), such that the **feature can be tested, even before it is completed and ready for release**.

Feature toggle is **used to hide, enable or disable the features**, …

https://en.wikipedia.org/wiki/Feature_toggle

Feature Toggle Overview

A framework for "feature flags" should:

- Allow the management of the flags outside of your application
- Allow you to change the configuration during runtime without any downtime
- Switch the configuration at once (on all servers and in all components)
- Have a minimal fingerprint / a very high performance
- Be failsafe (return a default value when the service is not available)
- Allow you to change the configuration per user, machine, percentage

Implementing a framework that meets these requirements is pretty complex.

by Michael Kaufmann
http://bit.ly/2izyY1s

Managing Features - Frameworks

“**There are a lot of open source frameworks** for the different languages. For Java there are Togglz, FF4J, Fitchy and Flip. For .Net there are FeatureSwitcher, NFeature, FlipIt, FeatureToggle or FeatureBee. Some use strings, some enums and some classes – **but none has a high scalable backend and a portal to manage your flags** (at least not that I know).

That’s why I played around with **LaunchDarkly** the last months. This is not just a framework – it’s a complete “**feature flag as a service**” solution. It has a SDK for .Net, Java, Python, Ruby, Go, Node, JavaScript, iOS, Android and PHP. It has a portal to manage your flags and to set up experiments. **It integrates with VSTS** and BitBucket Pipelines, with Slack and HipChat, with Optimizley and New Relic.”

by Michael Kaufmann

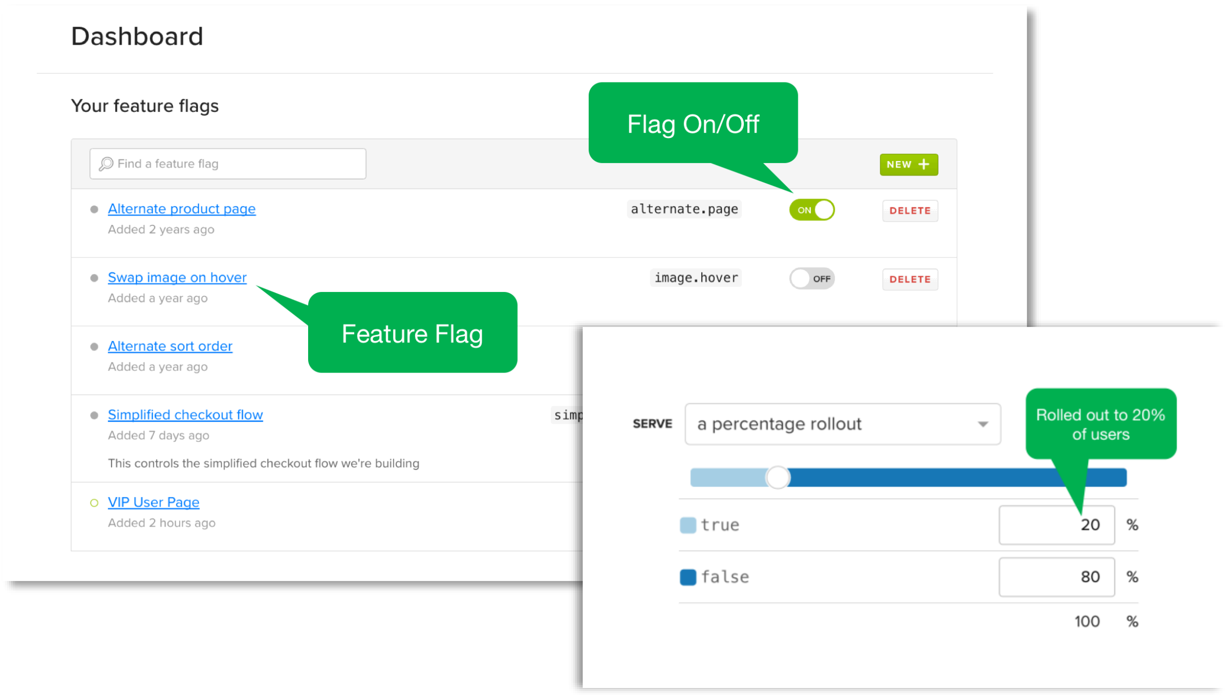
http://bit.ly/2izyY1s

Managing Features - Frameworks

- nToggle found in 06/12
- FeatureToggle found in 06/12
- NFeature found in 06/12
- Toggler found at 12/12/12
- Flipper found at 01/01/13
- Switcheroo found at 01/13/13
- FlipIt found at 01/13/13
- c24.FeatureSwitcher found at 12/07/13
- OnOff found at 01/12/14
- FeatureToggler found at 08/12/14
- Moon.Features found at 09/14/14
- FeatureSwitch found at 09/14/14
- FeatureFlipper found at 09/14/14
- ReallySimpleFeatureToggle found at 09/14/14
- FeatureBee found at 09/14/14
- Togglr found at 09/14/14
- toggler.net found at 09/14/14
- Ensign found at 09/14/14
- Fooidity found at 11/06/14
- FeatureSwitch found at 12/25/14
- DevCookie found at 7/1/16

LaunchDarkly - Feature flag management software designed for teams.

Launch → Control → Measure your features



Dashboard

Your feature flags

Find a feature flag | NEW +

Alternate product page Added 2 years ago	alternate.page	ON	DELETE
Swap image on hover Added a year ago	image.hover	OFF	DELETE
Alternate sort order Added a year ago			
Simplified checkout flow Added 7 days ago This controls the simplified checkout flow we're building	simp		
VIP User Page Added 2 hours ago			

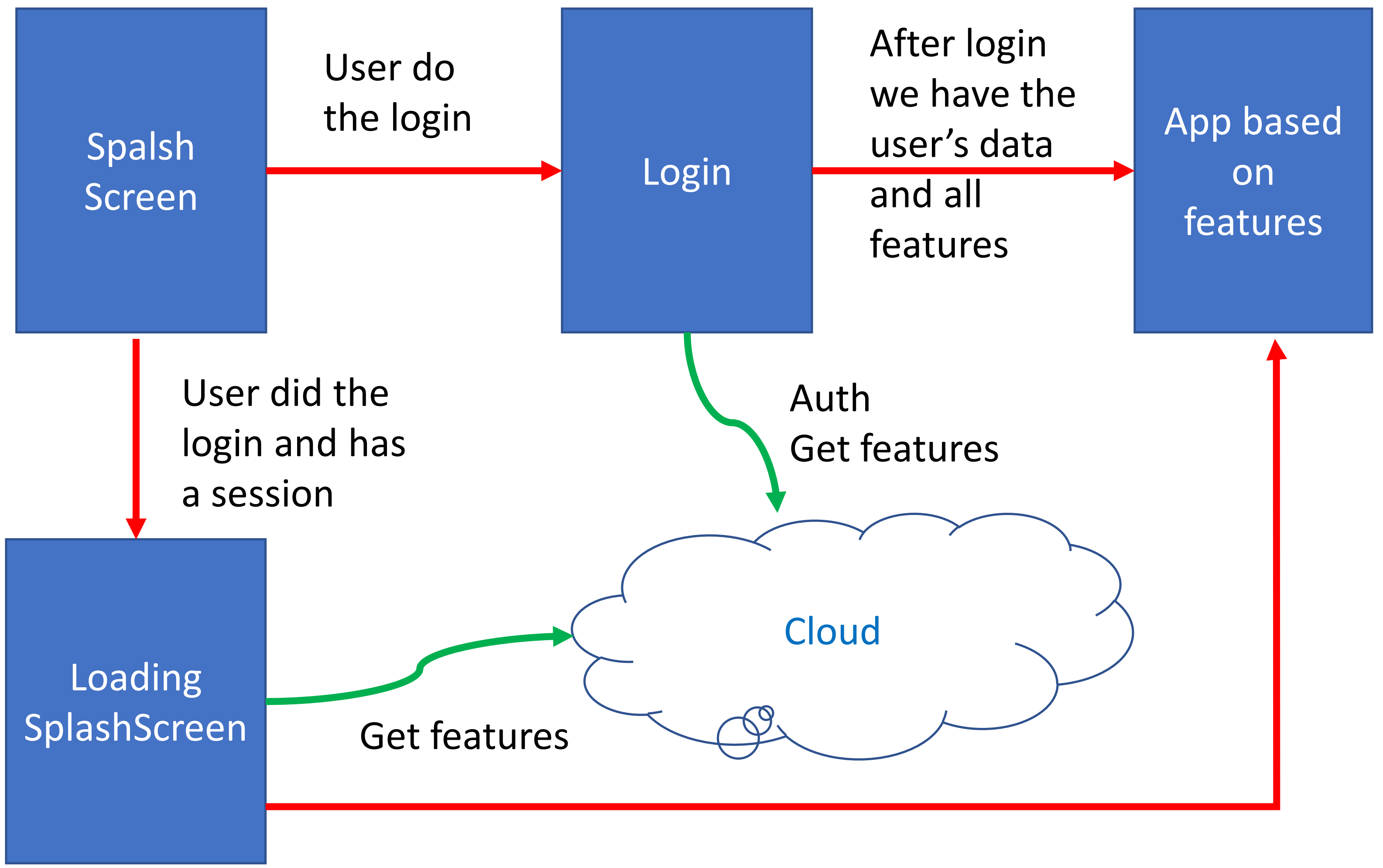
SERVE a percentage rollout

true	20	%
false	80	%
	100	%

Considerations

- **Feature toggles** require a robust engineering process, solid technical design and a mature toggle life-cycle management.
- Without these 3 key considerations, use of feature toggles **can be counter-productive**.
- Remember the **main purpose of toggles is to perform release with minimum risk**, once release is complete toggles need to be removed.

Feature Toggle - Mobile App's Flow



Toogle Service Solution

References

The references related with the exercise developed are

- The source code

https://github.com/saramgsilva/ToogleServiceSample

- Platform running on Azure (Web Apps) & Azure Database

http://toggleserviceplatform.azurewebsites.net/

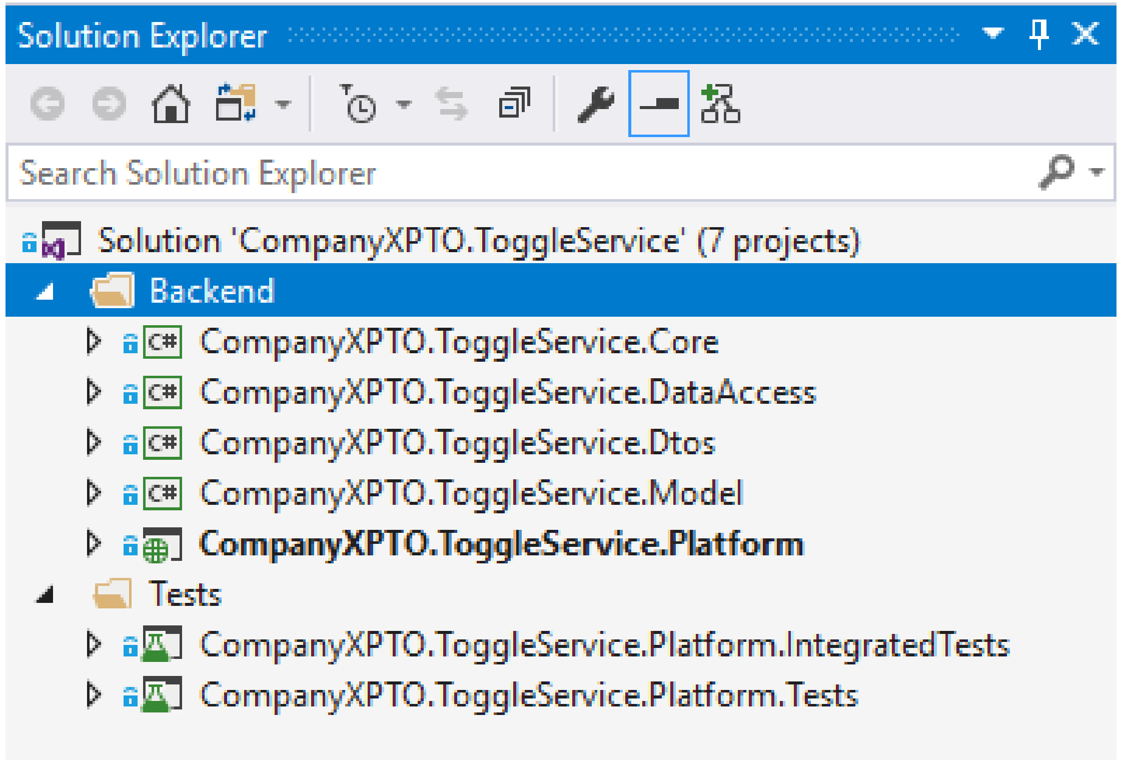
How to run

- Go to ***ToggleServiceSample\src***
- Open ***CompanyXPTO.ToggleService.sln***
- Define ***CompanyXPTO.ToggleService.Platform*** as “Startup Project”
- Open the ***WebConfig*** and define the connection string
- Apply migration(*) using ***update-database*** (database and tables will be created, and the seed da ta is applied with initial data.

(*) The solution is using EF Code First

The solution

- The solution has two solution folders – Backend & Tests



Solution 'CompanyXPTO.ToggleService' (7 projects)
Backend
CompanyXPTO.ToggleService.Core
CompanyXPTO.ToggleService.DataAccess
CompanyXPTO.ToggleService.Dtos
CompanyXPTO.ToggleService.Model
CompanyXPTO.ToggleService.Platform
Tests
CompanyXPTO.ToggleService.Platform.IntegratedTests
CompanyXPTO.ToggleService.Platform.Tests

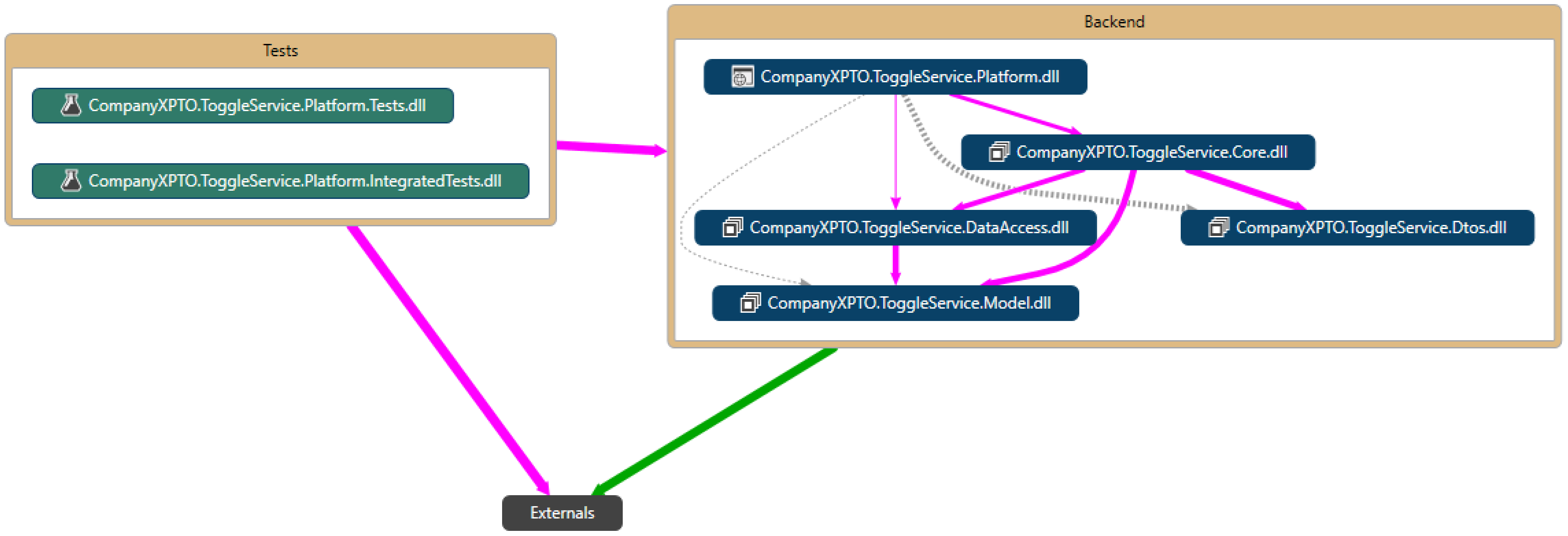
- **Backend** folder contains all projects that define the platform.

- **Tests** folder contains all projects that defines the de tests created.

The solution

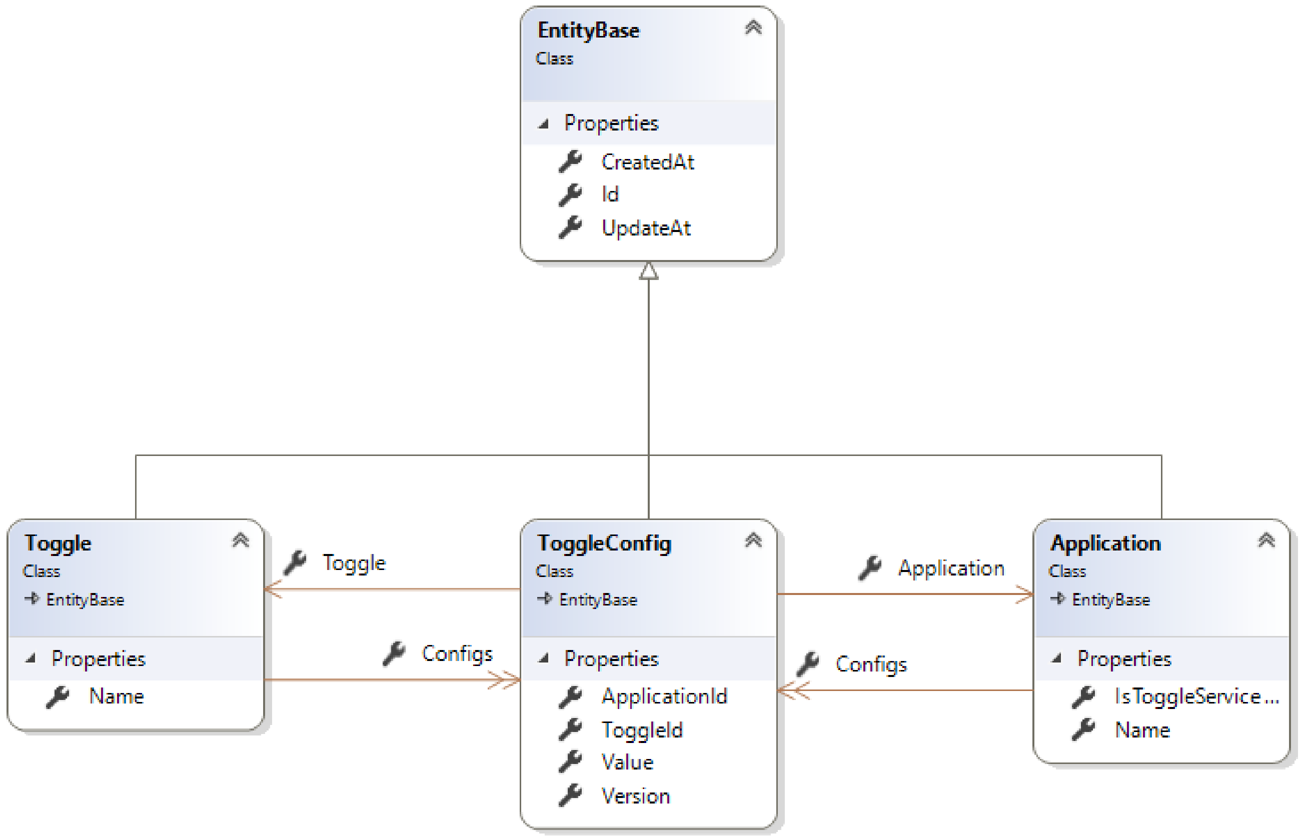
Project's name	Description
CompanyXPTO.ToggleService.Core	Contains the classes that defines the business logic from the platform – these classes can be used by Service or by MVC Controllers.
CompanyXPTO.ToggleService.DataAccess	Contains all classes to access data from data base - DBContext, Repository Pattern, UnitOfWork, ….
CompanyXPTO.ToggleService.Dtos	Contains all classes that defines by data object transfer from the platform.
CompanyXPTO.ToggleService.Model	Contains all classes that defines the model which is used by DataAccess Layer.
CompanyXPTO.ToggleService.Platform	ASP.Net MVC & WEBAPI project
CompanyXPTO.ToggleService.Platform.IntegratedTests	Contains the integrated tests to test if the http requests are working correctly.
CompanyXPTO.ToggleService.Platform.Tests	Contains the unit tests from the BackEnd projects.

Architecture – Code Map for Solution


Tests
CompanyXPTO.ToggleService.Platform.Tests.dll
CompanyXPTO.ToggleService.Platform.IntegratedTests.dll
Backend
CompanyXPTO.ToggleService.Platform.dll
CompanyXPTO.ToggleService.Core.dll
CompanyXPTO.ToggleService.DataAccess.dll
CompanyXPTO.ToggleService.Dtos.dll
CompanyXPTO.ToggleService.Model.dll
Externals

The class diagram from Model

- The class diagram with the model from the platform is



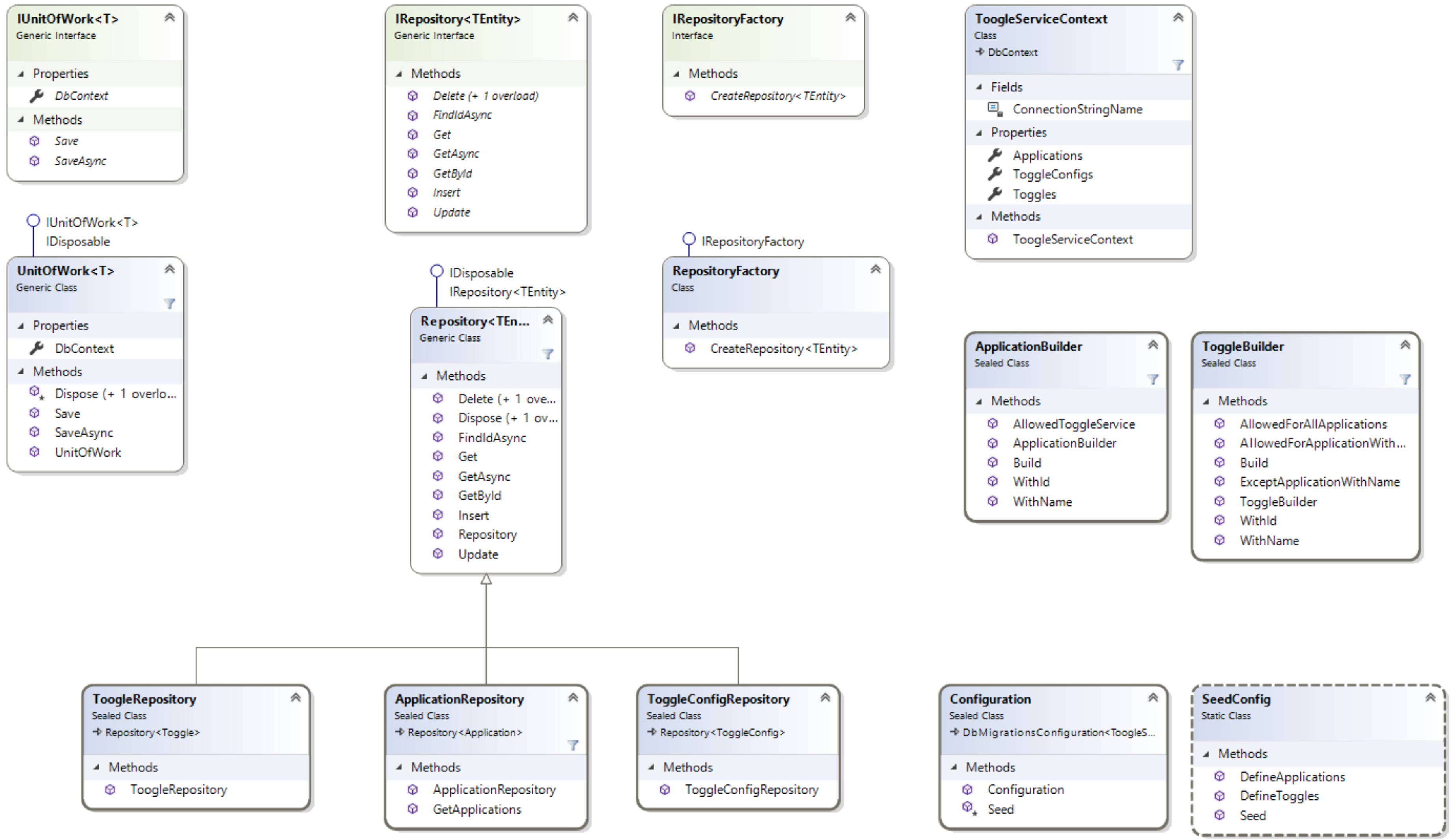
The model

Description for each class define in model

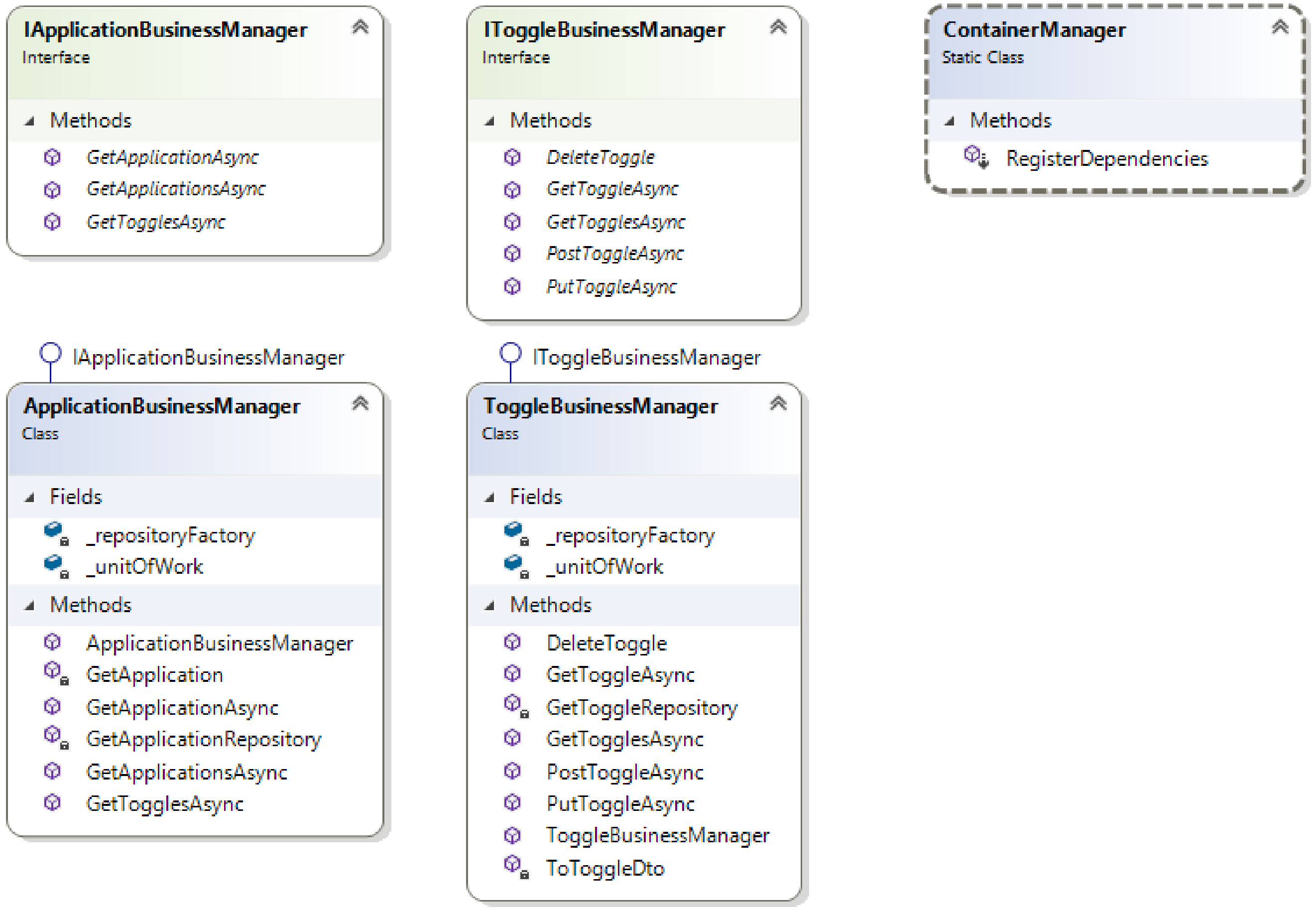
Class	Description
Toggle	Define the entity to define the toggle feature from platform
Application	Define the entity to define the applications existing in system
ToggleConfig	Defines the entity that define the relationship between Toggle and Application, and its configurations (Version, Value – if toggle is on/off for the service,…)
EntityBase	Defines the entity base with common properties cross all model's class.

Note: In real project it can be more complex because can have more configurations, requirements, …

The class diagram from DataAccess project



The class diagram from Core project



Desing Pattern

In the platform it was used differents design pattern:

- Repository Pattern
- UnitOfWork Pattern
- Builder Pattern
- Bridge Pattern
- Factory Pattern
- …

All dependencies were injected using ***SimpleInjector*** which are configured in ***IocConfig.cs*** and ***ContainerManager.cs.***

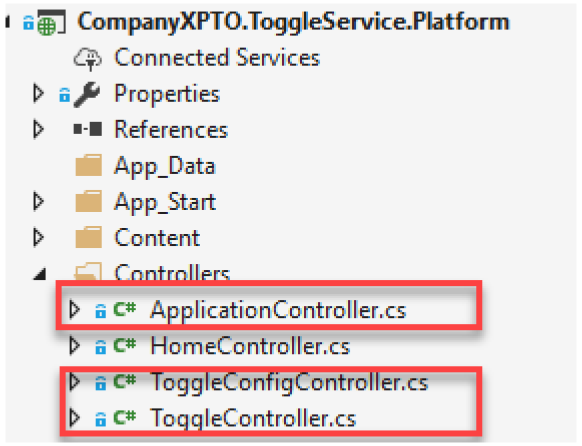
Services

The service are defined in *Controllers* folder:

ApplicationController – defines the application's service (for CRUD operations)

ToggleController – defines the toggle's service (for CRUD operations)

ToggleConfigController – define the service that allow to get toggle by application



```
CompanyXPTO.ToggleService.Platform
    Connected Services
  ▷ Properties
  ▷ References
    App_Data
  ▷ App_Start
  ▷ Content
  ◢ Controllers
    ▷ C# ApplicationController.cs
    ▷ C# HomeController.cs
    ▷ C# ToggleConfigController.cs
    ▷ C# ToggleController.cs
```

Services - GetToggles by applicationId

`http://toggleserviceplatform.azurewebsites.net/api/v1/ToggleConfig/{id}`

```
[HttpGet]
[Route("{applicationId}")]
[AllowAnonymous]
[ResponseType(typeof(Response<IEnumerable<ToggleServiceConfigDto>>))]
public async Task<IHttpActionResult> GetTogglesAsync(string applicationId)
{
    try
    {
        // the application id is always required
        if (string.IsNullOrEmpty(applicationId))
        {
            // if application id is null the request is wrong
            return BadRequest();
        }

        // get toggles configured in database from the applicationid provided
        var result = await _applicationBusinessManager.GetTogglesAsync(applicationId);

        return Ok(result);
    }
    catch (ArgumentOutOfRangeException e)
    {
        return NotFound();
    }
    catch (Exception ex)
    {
        // in some cases the exception could not be sent and it is only written in log system
        return InternalServerError(ex);
    }
}
```

Core - GetToggles by applicationId

ApplicationBusinessManager define the implementation used by service

```
public async Task<Response<IEnumerable<ToggleServiceConfigDto>>> GetTogglesAsync(string applicationId)
    {
        if (string.IsNullOrEmpty(applicationId))
        {
            throw new ArgumentNullException(nameof(applicationId));
        }

        var applicationRepository = GetApplicationRepository();
        var application = await GetApplication(applicationId, applicationRepository);
        if (!application.IsToggleServiceAllowed)
        {
            return new Response<IEnumerable<ToggleServiceConfigDto>> { IsValid = false , ErrorCode = "102", Message = $"{application.Name} does not have permission to use toggle service."};
        }
        var items = application.Configs?.Select(config => new ToggleServiceConfigDto
        {
            //●When the application/service request their toggles, they must only provide their id and version.
            Id = config.ToggleId,
            Version = config.Version

        }).ToList() ?? new List<ToggleServiceConfigDto>();

        return new Response<IEnumerable<ToggleServiceConfigDto>> { Result = items, IsValid = true };
    }
```

Tests

In the project ***CompanyXPTO.ToggleService.Platform.Tests*** it was used the Moq framework to create mock data from each entity.

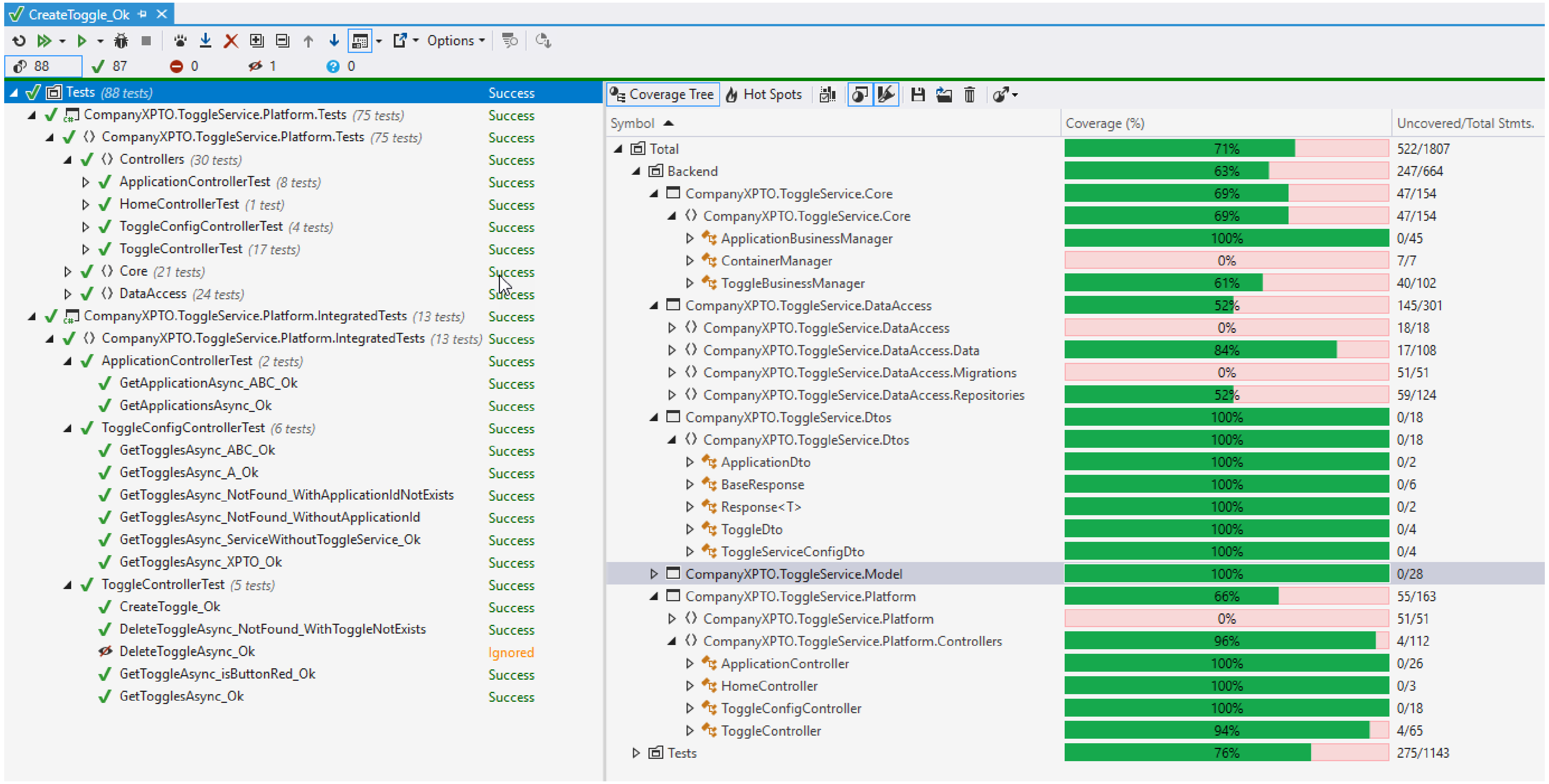
```
[TestMethod]
    public async Task GetTogglesAsync_GetResponseWith2ToggleDtos_Ok()
    {
      var toggles = ToggleFakeData.GetToggles();
      var unitoOfWorkMock = new Mock<IUnitOfWork<DbContext>>();
      var repositoryMock = new Mock<IRepository<Toggle>>();
      repositoryMock.Setup(r => r.GetAsync()).Returns(Task.FromResult(toggles));

      var repositoryFactoryMock = new Mock<IRepositoryFactory>();
      repositoryFactoryMock.Setup(r => r.CreateRepository<Toggle>(unitoOfWorkMock.Object)).Returns(repositoryMock.Object);

      var manager = new ToggleBusinessManager(unitoOfWorkMock.Object, repositoryFactoryMock.Object);

      var response = await manager.GetTogglesAsync();
      Assert.IsNotNull(response);
      Assert.IsNotNull(response.IsValid);
      Assert.AreEqual(response.Result.Count(),2);
      foreach (var item in response.Result)
      {
        Assert.IsNotNull(item.Name);
        Assert.IsNotNull(item.Id);
        Assert.IsNotNull(item.Applications);
      }
    }
```

Code coverage results



CreateToggle_Ok

Options

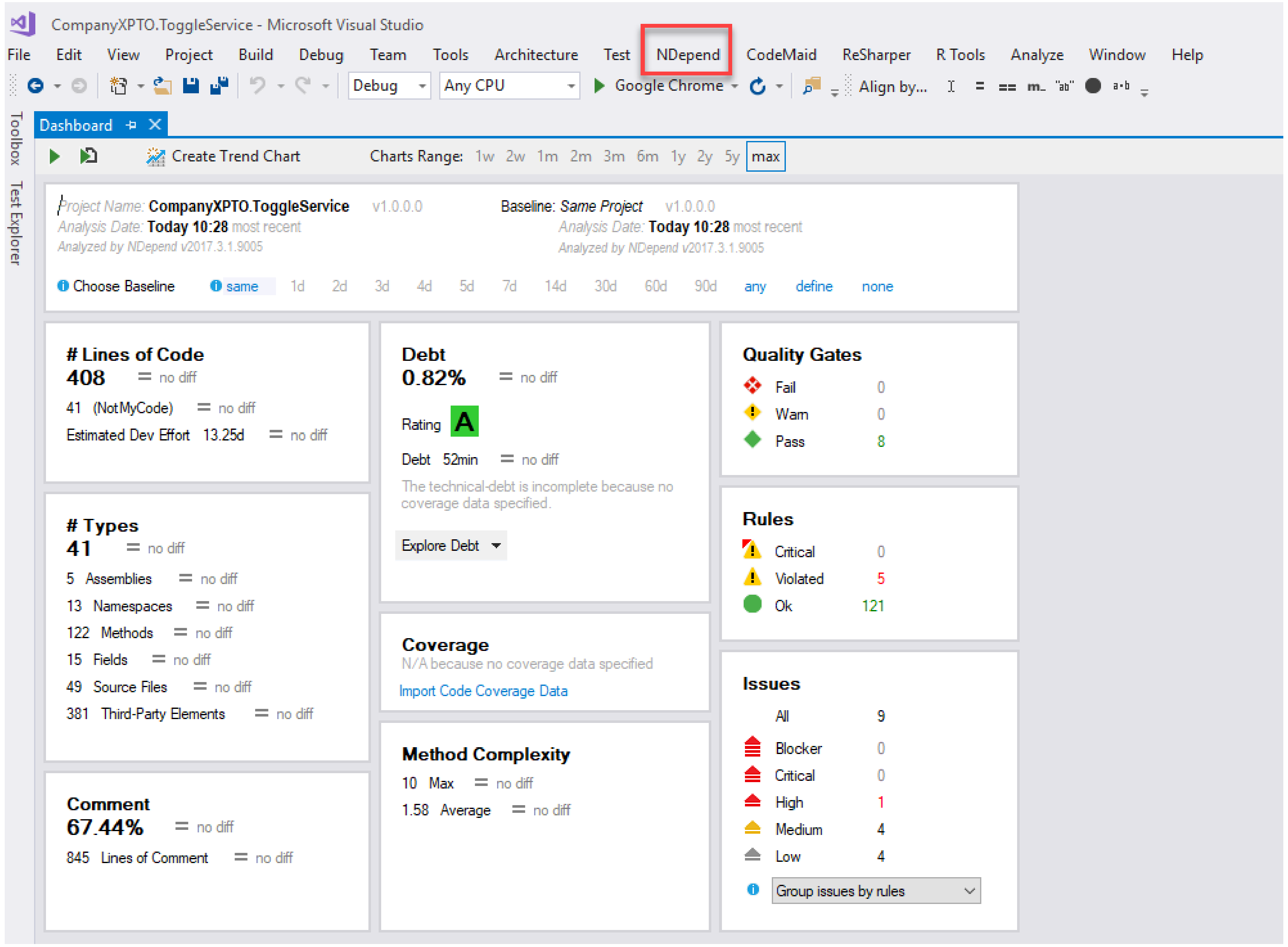
88 | ✓ 87 | 0 | 1 | 0

Tests (88 tests)	Success
CompanyXPTO.ToggleService.Platform.Tests (75 tests)	Success
CompanyXPTO.ToggleService.Platform.Tests (75 tests)	Success
Controllers (30 tests)	Success
ApplicationControllerTest (8 tests)	Success
HomeControllerTest (1 test)	Success
ToggleConfigControllerTest (4 tests)	Success
ToggleControllerTest (17 tests)	Success
Core (21 tests)	Success
DataAccess (24 tests)	Success
CompanyXPTO.ToggleService.Platform.IntegratedTests (13 tests)	Success
CompanyXPTO.ToggleService.Platform.IntegratedTests (13 tests)	Success
ApplicationControllerTest (2 tests)	Success
GetApplicationAsync_ABC_Ok	Success
GetApplicationsAsync_Ok	Success
ToggleConfigControllerTest (6 tests)	Success
GetTogglesAsync_ABC_Ok	Success
GetTogglesAsync_A_Ok	Success
GetTogglesAsync_NotFound_WithApplicationIdNotExists	Success
GetTogglesAsync_NotFound_WithoutApplicationId	Success
GetTogglesAsync_ServiceWithoutToggleService_Ok	Success
GetTogglesAsync_XPTO_Ok	Success
ToggleControllerTest (5 tests)	Success
CreateToggle_Ok	Success
DeleteToggleAsync_NotFound_WithToggleNotExists	Success
DeleteToggleAsync_Ok	Ignored
GetToggleAsync_isButtonRed_Ok	Success
GetTogglesAsync_Ok	Success

Coverage Tree | Hot Spots

Symbol	Coverage (%)	Uncovered/Total Stmts.
Total	71%	522/1807
Backend	63%	247/664
CompanyXPTO.ToggleService.Core	69%	47/154
CompanyXPTO.ToggleService.Core	69%	47/154
ApplicationBusinessManager	100%	0/45
ContainerManager	0%	7/7
ToggleBusinessManager	61%	40/102
CompanyXPTO.ToggleService.DataAccess	52%	145/301
CompanyXPTO.ToggleService.DataAccess	0%	18/18
CompanyXPTO.ToggleService.DataAccess.Data	84%	17/108
CompanyXPTO.ToggleService.DataAccess.Migrations	0%	51/51
CompanyXPTO.ToggleService.DataAccess.Repositories	52%	59/124
CompanyXPTO.ToggleService.Dtos	100%	0/18
CompanyXPTO.ToggleService.Dtos	100%	0/18
ApplicationDto	100%	0/2
BaseResponse	100%	0/6
Response<T>	100%	0/2
ToggleDto	100%	0/4
ToggleServiceConfigDto	100%	0/4
CompanyXPTO.ToggleService.Model	100%	0/28
CompanyXPTO.ToggleService.Platform	66%	55/163
CompanyXPTO.ToggleService.Platform	0%	51/51
CompanyXPTO.ToggleService.Platform.Controllers	96%	4/112
ApplicationController	100%	0/26
HomeController	100%	0/3
ToggleConfigController	100%	0/18
ToggleController	94%	4/65
Tests	76%	275/1143

Code quality results


CompanyXPTO.ToggleService - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Architecture Test NDepend CodeMaid ReSharper R Tools Analyze Window Help
Debug
Any CPU
Google Chrome
Align by...
Toolbox
Test Explorer
Dashboard
Create Trend Chart
Charts Range: 1w 2w 1m 2m 3m 6m 1y 2y 5y max

Project Name: **CompanyXPTO.ToggleService** v1.0.0.0
Analysis Date: **Today 10:28** most recent
Analyzed by NDepend v2017.3.1.9005

Baseline: *Same Project* v1.0.0.0
Analysis Date: **Today 10:28** most recent
Analyzed by NDepend v2017.3.1.9005

Choose Baseline same 1d 2d 3d 4d 5d 7d 14d 30d 60d 90d any define none

# Lines of Code
408 = no diff
41 (NotMyCode) = no diff
Estimated Dev Effort 13.25d = no diff

# Types
41 = no diff
5 Assemblies = no diff
13 Namespaces = no diff
122 Methods = no diff
15 Fields = no diff
49 Source Files = no diff
381 Third-Party Elements = no diff

Comment
67.44% = no diff
845 Lines of Comment = no diff

Debt
0.82% = no diff
Rating **A**
Debt 52min = no diff
The technical-debt is incomplete because no coverage data specified.
Explore Debt

Coverage
N/A because no coverage data specified
Import Code Coverage Data

Method Complexity
10 Max = no diff
1.58 Average = no diff

Quality Gates

Fail	0
Warn	0
Pass	8

Rules

Critical	0
Violated	5
Ok	121

Issues

All	9
Blocker	0
Critical	0
High	1
Medium	4
Low	4

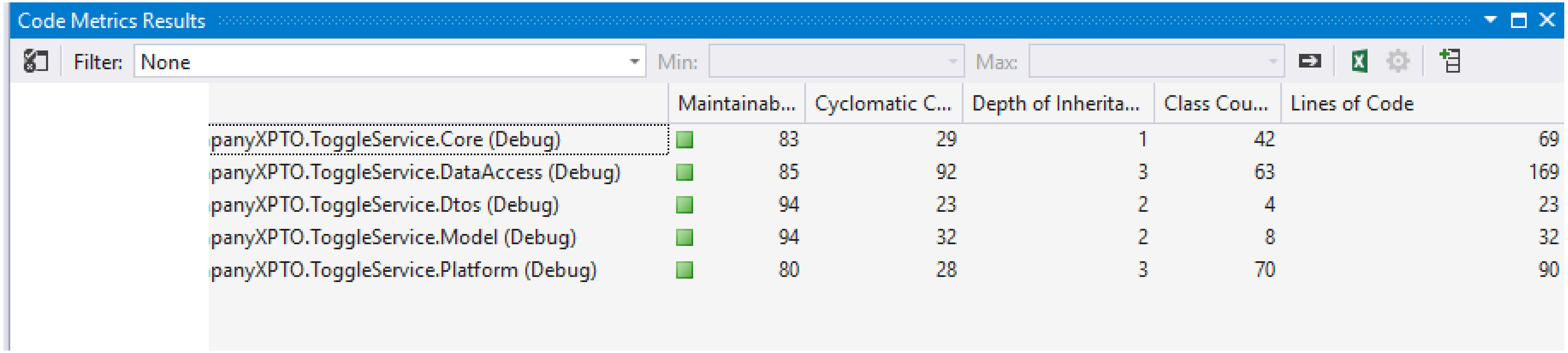
Group issues by rules

NDepend allow to analyze the quality code developed, in this case the Rating A is great, and in Rules there is 5 violed rules that is not possible to solve – we should change the query from NDepend to avoid these cases.

Note: See the NDepend output on src folder – a web based output

Code metrics result

The Maintainability index has high value which are good

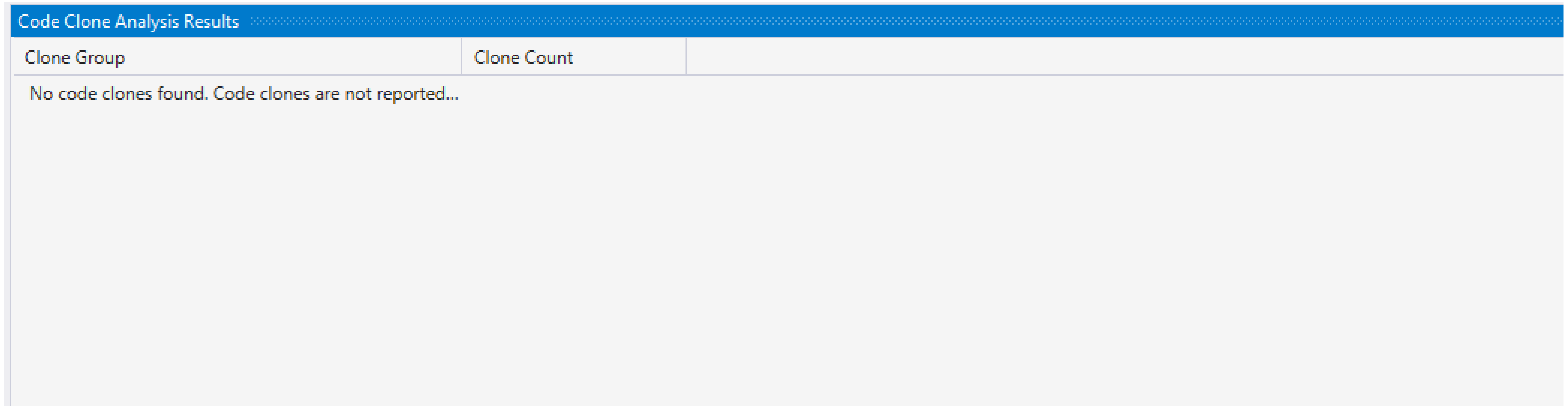


Code Metrics Results

Filter: None Min: Max:

		Maintainab...	Cyclomatic C...	Depth of Inherita...	Class Cou...	Lines of Code
panyXPTO.ToggleService.Core (Debug)	🟩	83	29	1	42	69
panyXPTO.ToggleService.DataAccess (Debug)	🟩	85	92	3	63	169
panyXPTO.ToggleService.Dtos (Debug)	🟩	94	23	2	4	23
panyXPTO.ToggleService.Model (Debug)	🟩	94	32	2	8	32
panyXPTO.ToggleService.Platform (Debug)	🟩	80	28	3	70	90

Code clone result



Code Clone Analysis Results

Clone Group	Clone Count
No code clones found. Code clones are not reported...	

Considerations

“When setting a new toggle, only users with admin permission may create a toggle.”

It is all about authentication and authorization that is implemented using an method attribute in each controller’s method with the role permission.

- There any solution to authentication – custom, external, social network, Company AD, Azure AD,…

- To implement the authorization in a simple scenario the user can have a property that says the role type – admin, regular user, …

In my opinion I think it should be implement according the strategy defined and with a robust solution

Considerations

"Try to find a way to alert each client application/service that the toggle was changed."

ASP.NET SignalR could be used to update the application/services when a toogle is changed, but I would like to recommend/test the Realtime framework - which is developed by portuguese company and it is focused in real-time communications

Considerations

API Helper - Swagger is a machine-readable representation of a RESTful API that enables support for interactive documentation, client SDK generation, and discoverability.

API Documentation - GhostDoc is a Visual Studio extension that automatically generates XML documentation comments for methods and properties based on their type, parameters, name, and other contextual information.

Logs – depending the server (on-promise or cloud) it should have a log strategy

Resources – depending the requirements the resources should be persisted in database to allow manage them using admin panel.

DevOps – Today there is any solutions to apply DevOps for a project, the solution provided has all necessary to apply it – only need a strategy to manage the database and data.

Two solutions I could use: Jenkins or Visual Studio Team Services

Aditional References

Related articles written by me and published on Microsoft TechNet Wiki

- Class diagram: a easy way to understand code
- Creating Framework Documentation with Ghostdoc
- Analyzing C# code using NDepend

Some NetPonto presentation related with share and quality code

- Como deixar de fazer "copy and paste" entre Windows Store e Windows Phone Apps
- Como analizar o código C# com o NDepend

Thank you!

Sara Silva