

Do you like travelling on Economy class?

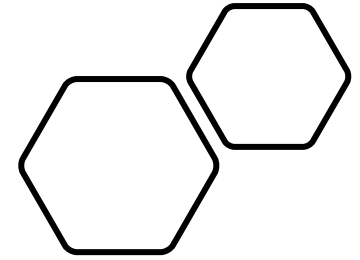


...inspired by real events



[This Photo](#) by Unknown Author is licensed under [CC BY-NC](#)

{code}





{code}

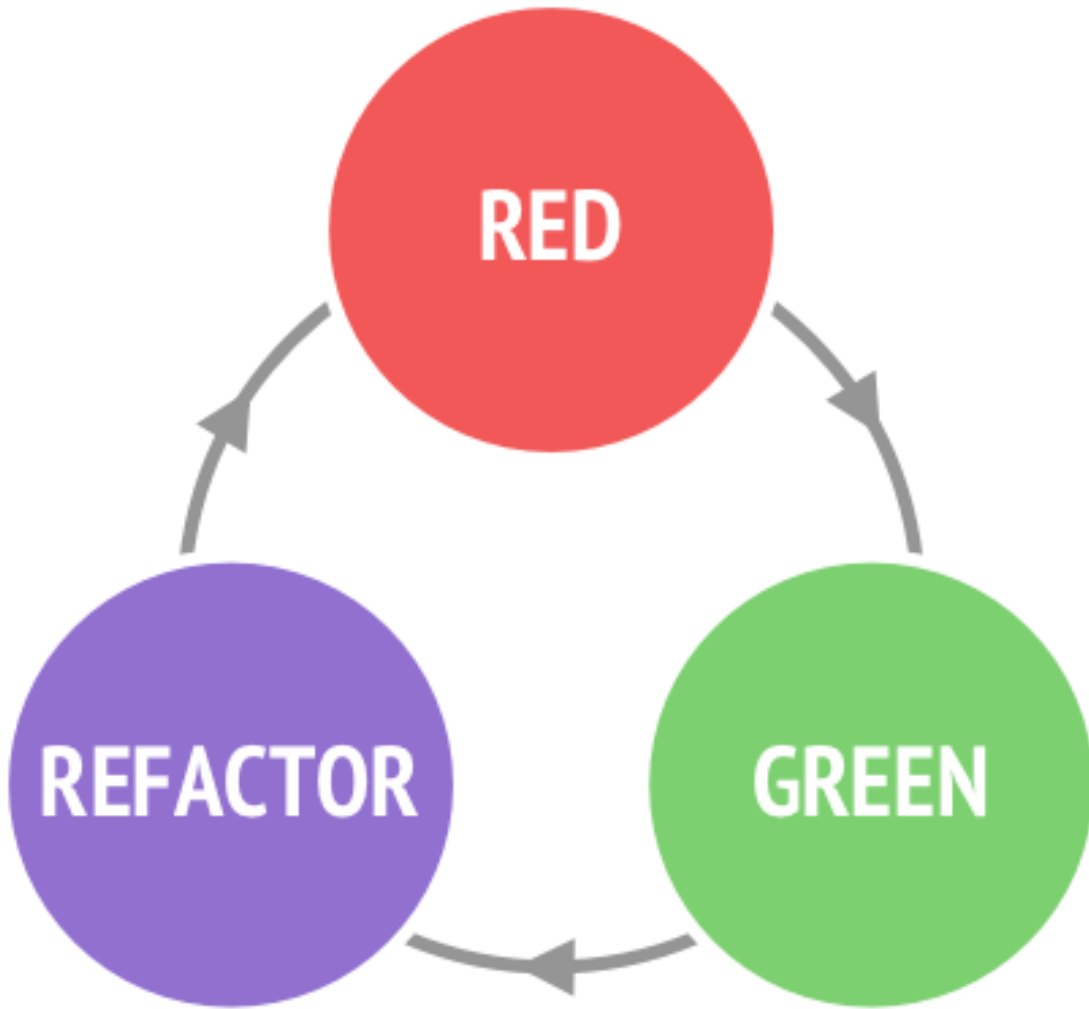




{code}

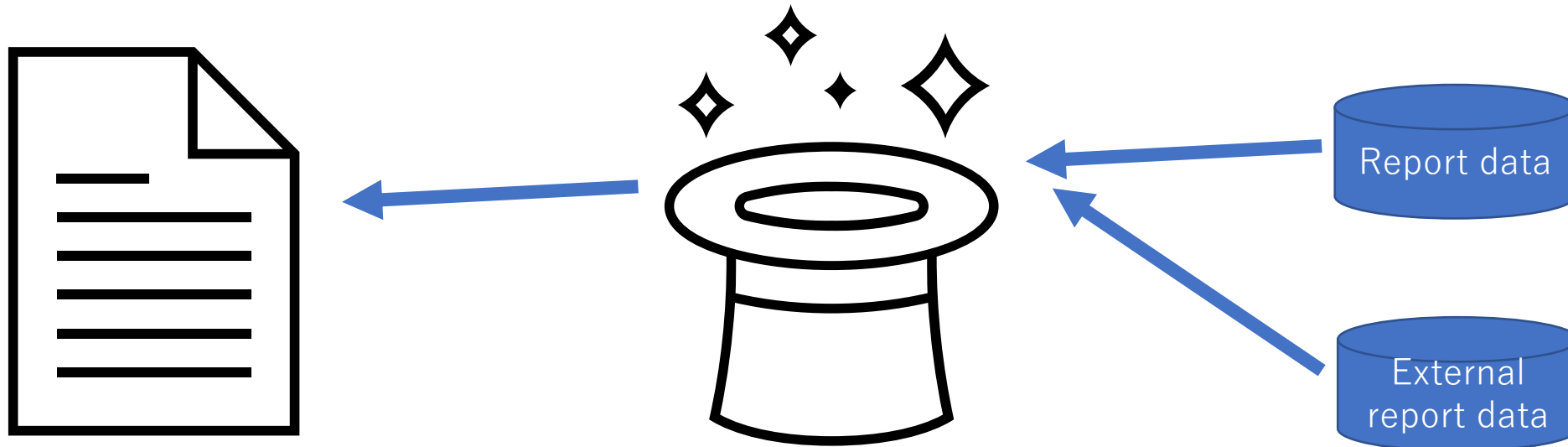


{code}



ALCOR
academy

{code}



ReportDataService.cs

{code}

Test doubles

- Stubs / Fakes
- Mocks / Spies
- (Dummies)



{code}

```
//SETUP
0 references | 0 changes | 0 authors, 0 changes
public ReportDataServiceShould()
{
```

```
    _logger = new();
```

```
    _accessControlMock = new Mock<AccessControl>();
```

```
    var reportDbContext = CreateDbContext();
```

```
    _reportConfigurationRepositoryMock = new Mock<IReportConfigurationRepository>();
```

```
    _reportDataRepositoryMock = new Mock<ReportDataRepository>(reportDbContext);
```

```
    _onPremiseDataServiceMock = new Mock<IOnPremiseDataService>(reportDbContext);
```

```
    _reportCleanerFactoryMock = new Mock<IReportCleanerFactory>();
```

```
    _reportServiceMock = new Mock<IReportService>();
```

```
    var reportData = new ReportData();
```

```
    _reportDataService = new ReportDataService();
```

```
    _onPremiseDataService = new OnPremiseDataService();
```

```
    _reportDataRepository = new ReportDataRepository();
```

```
    _reportCleanerFactory = new ReportCleanerFactory();
```

```
    _reportService = new ReportService();
```

```
    _reportCleanerFactoryMock.Setup(cleanerFactory => cleanerFactory.GetReportCleaner(It.IsAny<string>()))
```

```
    _reportServiceMock.Setup(service => service.GetReportData(CompanyName, ReportTypeCode, PeriodFrom, PeriodTo))
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportCleanerFactoryMock.Returns(new WeeklyReportCleaner(_reportServiceMock.Object));
```

```
    _reportConfigurationRepositoryMock.Setup(repository => repository.Setup(m => m.GetReportData(CompanyName, ReportTypeCode, PeriodFrom, PeriodTo))
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

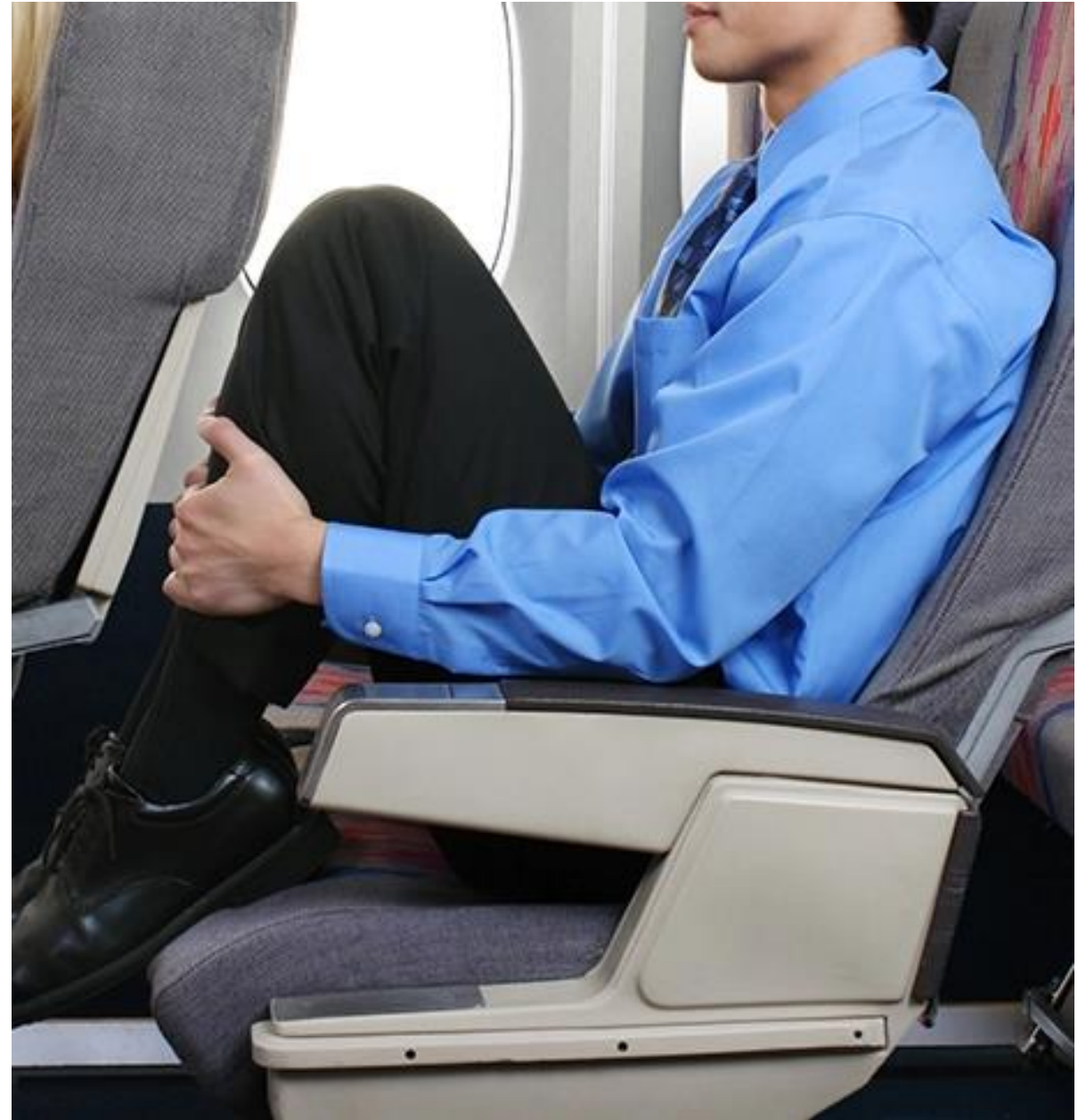
```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

```
    _reportConfigurationRepositoryMock.ReturnsAsync(reportData);
```

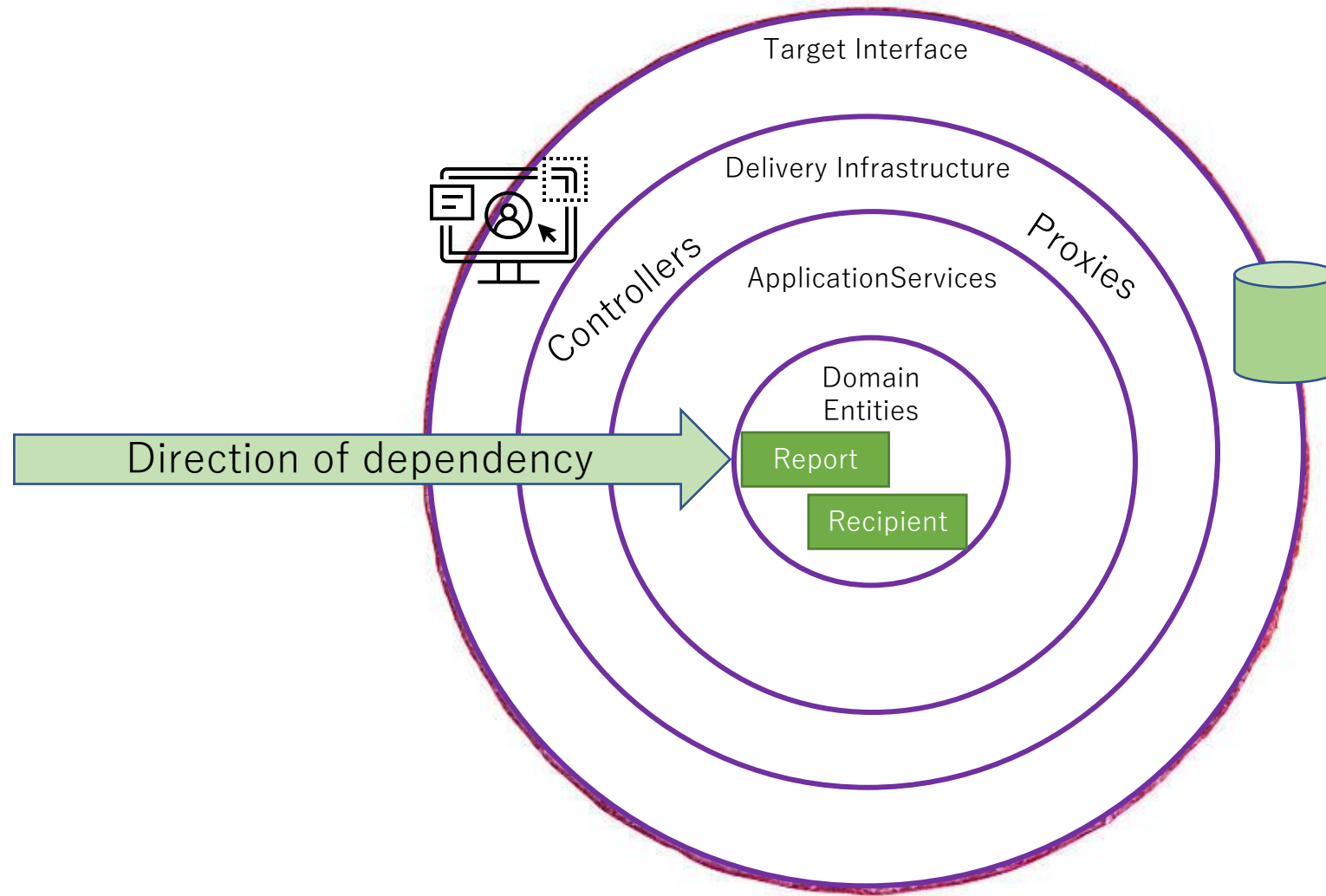


moq.github.io/

```
    var reportData = _reportDataService.GetReportData(CompanyName, ReportTypeCode, PeriodFrom, PeriodTo);
```

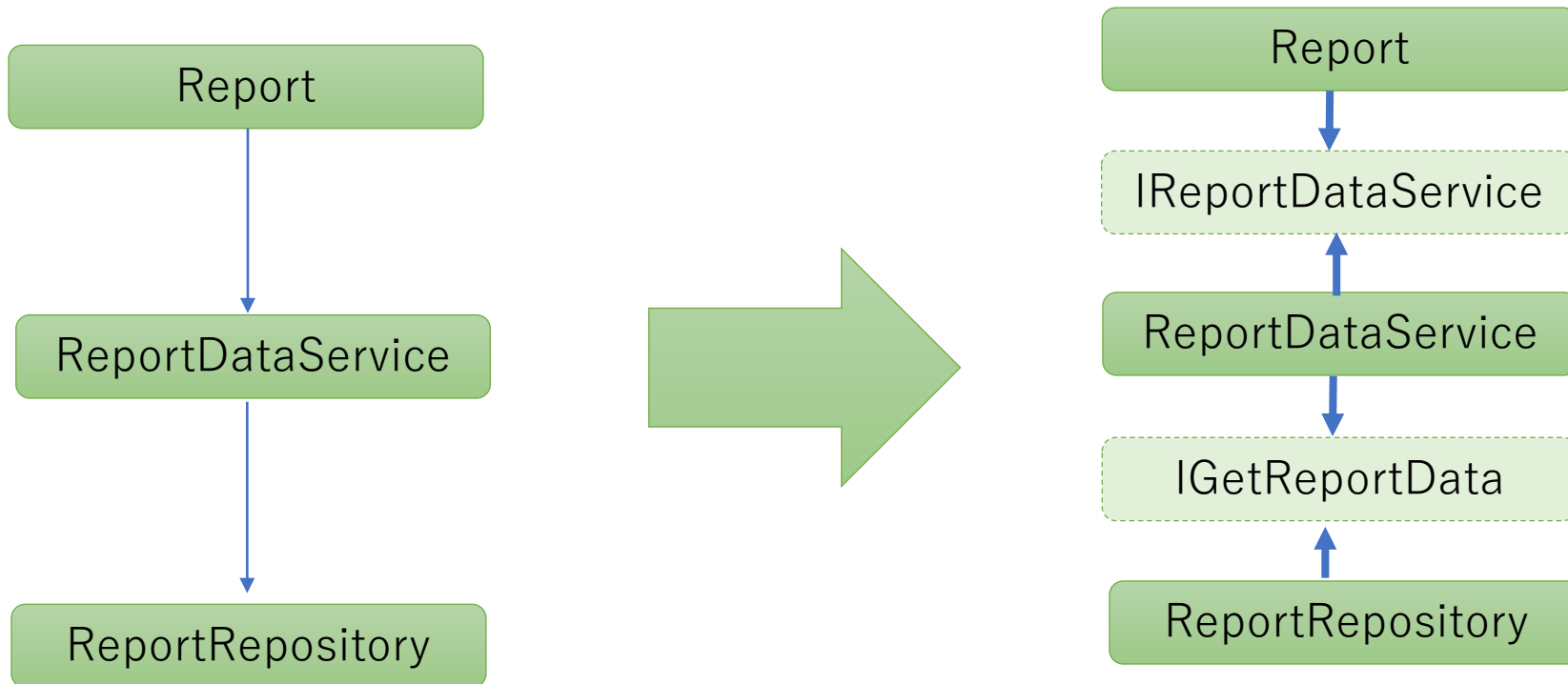



{code}

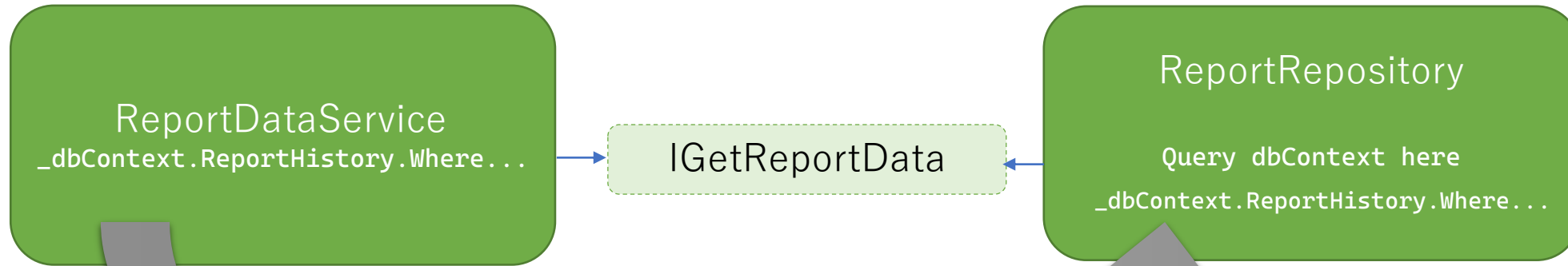


{code}

Dependency Inversion / Liskow Substitution Principle



{code}



Move direct usage of EF dbContext in here



{code}



```
0 references | 0 changes | 0 authors, 0 changes
public class ReportDataSourceFactory : IReportDataSourceFactory
{
    private readonly Dictionary<ReportDataSourceType, IDataSource> _dataSources;

    0 references | 0 changes | 0 authors, 0 changes
    public ReportDataSourceFactory() => _dataSources = new()
    {
        { ReportDataSourceType.Local, new LocalDataSourceRepository() },
        { ReportDataSourceType.OnPremise, new OnPremiseDataSourceRepository() },
        { ReportDataSourceType.WaterfallCoLtd, new WaterfallDataSourceRepository() }
    };

    0 references | 0 changes | 0 authors, 0 changes
    public IDataSource GetDataSource(ReportDataSourceType dataSourceType)
    {
        return !_dataSources.ContainsKey(dataSourceType)
            ? throw new Exception($"The system does not support data source type {dataSourceType}")
            : _dataSources[dataSourceType];
    }
}
```

{code}

```
//SETUP
0 references | 0 changes | 0 authors, 0 changes
public ReportDataServiceShould()
{
    ILogger<ReportDataService> logger = Substitute.For<ILogger<ReportDataService>>();
    IAccessControl accessControl = Substitute.For<IAccessControl>();
    _reportConfigurationRepository = Substitute.For<IReportConfigurationRepository>();
    OnPremReportDataService onPremiseDataService = Substitute.For<OnPremReportDataService>();
    _reportCleanerFactory = Substitute.For<IReportCleanerFactory>();
    _reportService = Substitute.For<IReportService>();
    reportDataRepository = Substitute.For<IReportDataRepository>();

    // NSubstitute
    IReportDataRepository repository = Substitute.For<IReportDataRepository>();
    repository.GetData(Arg.Any<string>()).Returns(reportDataObject);
}

// Moq
Mock<ReportDataRepository> reportDataRepositoryMock = new Mock<ReportDataRepository>();
reportDataRepositoryMock
    .Setup(repository => repository.GetData(It.IsAny<string>()))
    .Returns(reportDataObject);

// to pass object as parameter:
reportDataRepositoryMock.Object

    _reportDataRepository.GetPreviousReport(Arg.Any<Guid>(), Arg.Any<DateTime>(), Arg.Any<DateTime>())
        .Returns(previousReport);
    _reportCleanerFactory.GetReportCleaner(Arg.Any<string>()).Returns(new WeeklyReportCleaner(_reportService));

    var reportData = _reportDataService.GetReportData(CompanyName, ReportTypeCode, PeriodFrom, PeriodTo);

    Assert.NotNull(reportData);
    OnlyFirstLineOfDataShouldBeCopied(reportData); //helper method containing asserts
}
```

te

locking libraries

e.github.io/







Questions?

Elisabeth Forland
Senior developer at Bouvet
elisabeth.forland@bouvet.no

//Thanks for your attention;P

Resources

- <https://nsubstitute.github.io/help/replacing-return-values/>
- <https://github.com/moq/moq4>
- Slides from Alcor Academy TDD course
- <https://alcor.academy/>

- Pictures:

- <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRL6KCV-779Vr40hV42Ptjf0KBZp2fiEizVnw&usqp=CAU>
- https://m.media-amazon.com/images/M/MV5BYzNkY2E5MTgtYTE4NC00MjVhLWE5NzEtMjRjZjdiM2ZlOGU4XkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_.jpg
- <https://www.skyclub.com/wp-content/uploads/2017/09/How-to-get-cheaper-business-class-flight.jpg>
- <https://c.ekstatic.net/ecl/photo-gallery/b777/economy-class/dining/emirates-b777-economy-class-dining-couple-720x480.jpg>
- <https://opinion8ed2.files.wordpress.com/2015/03/coach-class2.jpg>
- <https://media.istockphoto.com/id/157403494/photo/no-leg-room.jpg?s=612x612&w=0&k=20&c=IBZdC6ctPGkaY3kwBfjXfQQL0QTL-vknRTfr-2ye408=>
- <https://i.chzbgr.com/original/6488802048/h794DDB1E/bicycle-ferris-wheel-illusion-perspective-photography-6488802048>
- <https://s3.envato.com/files/352736585/DSC07338.jpg>
- <https://media.istockphoto.com/id/184275334/photo/red-onion-portion-on-white.jpg?s=612x612&w=0&k=20&c=YCnWnCxzBERzvz4GDqcyJrCEjXEB0luOoODpbcRYUZ0=>
- <https://thumbs.dreamstime.com/z/blue-white-small-jet-airplane-flying-air-isolated-background-domestic-flight-166936233.jpg>
- https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.csitech.com%2F&psig=AOvVaw3xJ6z_i0wz-hfqtjoeU-9f&ust=1685457502670000&source=images&cd=vfe&ved=0CBEQjRxxqFwoTCOIY1c_gmv8CFQAAAAAdAAAAABAE
- https://d347cldnsmtg5x.cloudfront.net/pic_3268_2_5fa3da53be6d9.jpg?width=587