如何在實務上使用TDD來開發

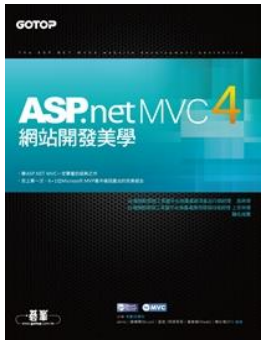
Joey Chen(91)

tw MVC

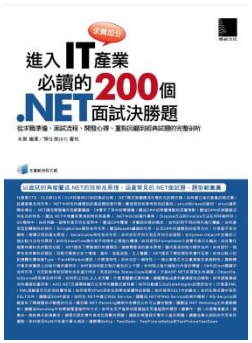
MVP Microsoft® Most Valuable Professional

About 91

- Microsoft MVP 2010~2013
- C#
- 共同作者，書籍譯者
- 軟體工程，測試，重構，CI，TDD，BDD…







TDD很難？

WHY?

講TDD之前

降龍十八掌



亢龍有悔

源自《易經》乾卦

卦象（䷀）

上九爻辭：「亢龍有悔。」

食指：巨龍高飛窮極，

終將導致悔恨。

團掌出擊，留有餘地。

發揮必勝，故力不可用盡，

招勢雖猛發則型，

三分吉番，退後勁終終。

左腿微屈

右臂內彎

右掌劃一圓圈

向外推去



一般人呢？



所以

降龍十八掌很爛？

跟錯老師



MVC

今天



Agenda

- Web UI Testing
- Refactoring
- Unit Testing
- BDD
- Interface
- 第一式打到最後一式

功能


商品名稱
商品名稱test
長
30
寬
20
高
10
重量
10
貨運商 黑貓
運費: 180
計算運費

程式碼



重構 start !!

Web UI Testing

Scenarios

黑貓：200

新竹貨運：254.16

郵局：180

工具

Firefox

Selenium IDE

Demo

- Record Scenarios
- Replay and Assert
- Export to C#
- Auto testing in C#

Demo – Selenium IDE & C#



Reference

- Selenium IDE
- [Selenium IDE]Export to C#/WebDriver/MSTest
- NuGet
  - Selenium WebDriver
  - Selenium WebDriver Support Classes

Keep Refactoring

釐清思緒

說明程式

擷取方法

Demo – Extract Method



分離職責

工作委託

主詞動詞分離

Demo – 建立Class



單元測試

3A Pattern

Arrange

Act

Assert

羊毛出在羊身上

Scenarios

黑貓：200

新竹貨運：254.16

郵局：180

Demo – 單元測試



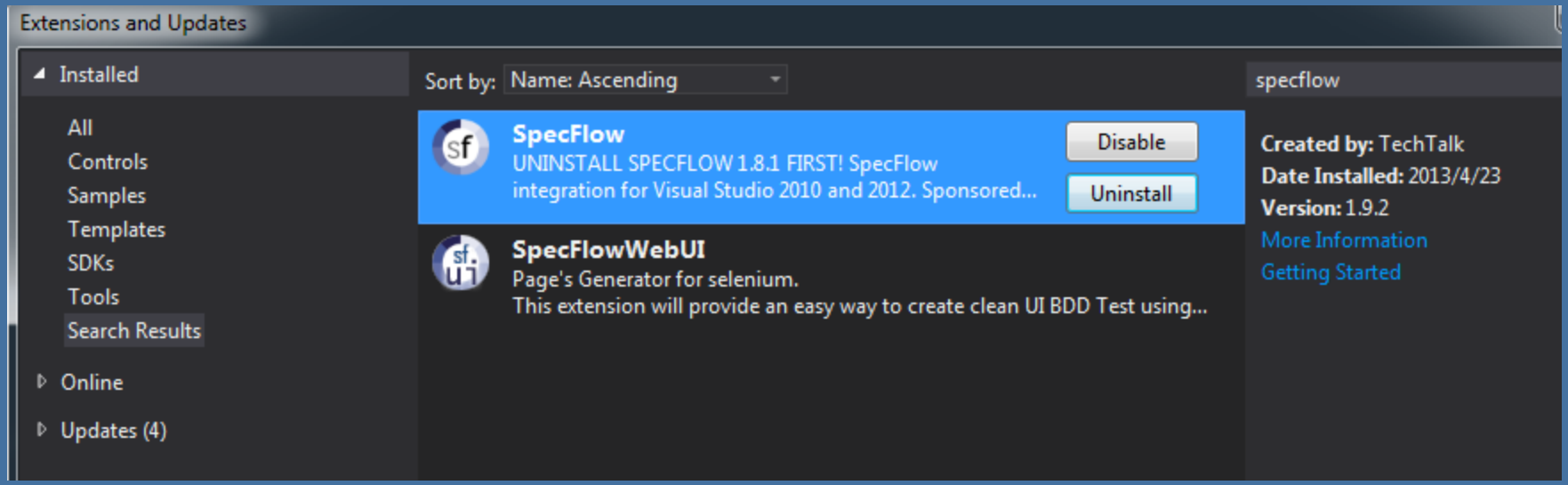
需求說明
by
測試案例

MVC

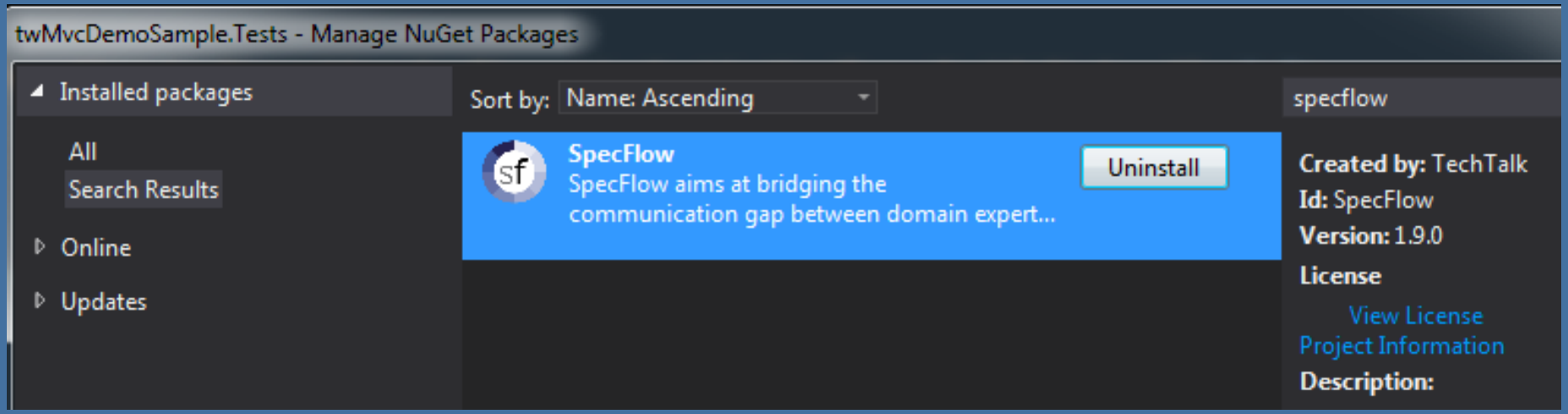
自然語言

Specflow

Visual Studio Extension



NuGet


twMvcDemoSample.Tests - Manage NuGet Packages
Installed packages
All
Search Results
Online
Updates
Sort by: Name: Ascending
SpecFlow
SpecFlow aims at bridging the communication gap between domain expert...
Uninstall
specflow
Created by: TechTalk
Id: SpecFlow
Version: 1.9.0
License
View License
Project Information
Description:

App.Config

```
<unitTestProvider name="MsTest" />
```

Scenarios

```
Scenario: 黑貓計算運費
    Given 商品資料
    | Length | Width | Height | Weight | Name     |
    | 30     | 20    | 10     | 10     | product1 |
    When 呼叫黑貓的計算運費
    Then 運費結果應為 200

Scenario: 新竹貨運計算運費
    Given 商品資料
    | Length | Width | Height | Weight | Name     |
    | 30     | 20    | 10     | 10     | product1 |
    When 呼叫新竹貨運的計算運費
    Then 運費結果應為 254.16

Scenario: 郵局計算運費
    Given 商品資料
    | Length | Width | Height | Weight | Name     |
    | 30     | 20    | 10     | 10     | product1 |
    When 呼叫郵局的計算運費
    Then 運費結果應為 180
```

Given = Arrange

When = Act

Then = Assert

Demo - Specflow



Reference

- Specflow
- Specflow Visual Studio Extension
- NuGet
  - Specflow

眼睛瞇起來

抽象觀察

思考需求

依頼介面

Demo – Extract Interface



Recap

- Web UI testing (acceptance testing)
- Unit testing
- Test cases by scenarios
- Depend on interface

從無到有
TDD

PO：

我要一個登入功能

沒了?!

Team:

登入要做什麼事？

PO：

成功要導到首頁

PO：

失敗要提醒

Team:

能給我個例子嗎？

PO & team - 登入成功

- 帳號輸入 joey
- 密碼輸入 1234
- 登入成功
- 導到首頁 ~/Home/Index

PO & team – 登入失敗

- 帳號輸入 joey
- 密碼輸入 abc
- 登入失敗
- 呈現錯誤訊息為 “帳號或密碼有誤”

Team:

是這個樣子嗎？

Demo – Prototype confirm



PO：

醜了點，但YES

Scenario寫下來

```
Scenario: 登入成功，導到首頁
    Given Login的頁面
    When 在帳號輸入"joey"
    And 在密碼輸入"1234"
    And 按下登入
    Then 導到首頁

Scenario: 登入失敗，顯示錯誤訊息
    Given Login的頁面
    When 在帳號輸入"joey"
    And 在密碼輸入"abc"
    And 按下登入
    Then 顯示"帳號或密碼有誤"
```

Demo – Describe Scenarios



如果有QE

Selenium Testing

沒有QE??

人人都可以是QE

Demo – Selenium testing



Controller's testing

登入成功

- Account 傳入 joey
- Password 傳入 1234
- 結果
  - Controller 應為 Home
  - Action 應為 Index

登入失敗

- Account 傳入 joey
- Password 傳入 abc
- 結果
  - ViewBag.Message 為*帳號或密碼有誤*

Demo – Controller’s testing



Next?

如何驗證合法?

bool IAuthentication.
Verify(id, password)

IoC

Mock framework - RhinoMocks



RhinoMocks

MockRepository.GenerateSub<T>()

RhinoMocks

```
stubObj
.Stub(x=>x.MyFun(arg1))
.Return(myResult);
```

Demo – IoC & Stub



Mock framework

解決出現的紅燈

Without DI

這個物件沒有定義無參數的建構函式。

描述: 在執行目前 Web 要求的過程中發生未處理的例外狀況。請檢閱堆疊追蹤以取得錯誤的詳細資訊，以及在程式碼中產生的位置。

例外狀況詳細資訊: System.MissingMethodException: 這個物件沒有定義無參數的建構函式。

並未將物件參考設定為物件的執行個體。

描述: 在執行目前 Web 要求的過程中發生未處理的例外狀況。請檢閱堆疊追蹤以取得錯誤的詳細資訊，以及在程式碼中產生的位置。

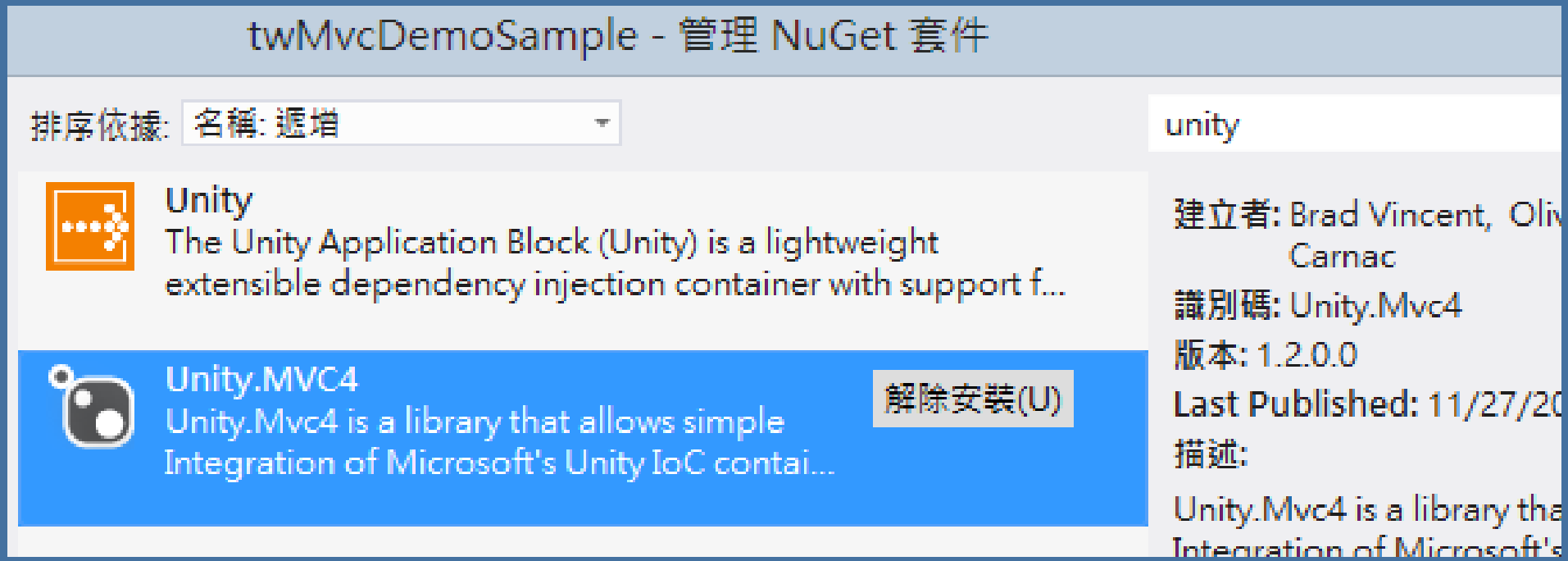
例外狀況詳細資訊: System.NullReferenceException: 並未將物件參考設定為物件的執行個體。

原始程式錯誤:

```
行 42:              //    return View();
行 43:              //}
行 44:              var isValid = this.authentication.Verify(account, password);
行 45:              if (isValid)
行 46:              {
```

原始程式檔: c:\Users\JoeyChen\SkyDrive\twMVC share\twMvcDemoSample\twMvcDemoSample\Controllers\LoginController.cs **行:** 44

DI framework – Unity.MVC4



Container.RegisterType<TFrom,TTo>

```
public static class Bootstrapper
{
  public static IUnityContainer Initialise()...

  private static IUnityContainer BuildUnityContainer()...

  public static void RegisterTypes(IUnityContainer container)
  {
      container.RegisterType<IAuthentication, MyAuthentication>();
  }
}
```

Demo – DI & MyAuthentication



MyAuthentication Testing

Get Password from Data Source

IAccountDao

Hash Password to Compare

IHash

Scenarios

IAccountDao回傳與IHash結果相同

- IAccountDao的GetPassword方法，傳入”joey”，回傳“Joey hashed password”
- IHash的GetHashResult方法，傳入”1234”，回傳“Joey hashed password”
- 結果
  - True

IAccountDao回傳與IHash結果不同

- IAccountDao的GetPassword方法，傳入”joey”，回傳“Joey hashed password”
- IHash的GetHashResult方法，傳入”abc”，回傳“abc hashed result”
- 結果
  - False

Demo – MyAuthentication BDD



紅燈？

實作 IAccountDao

MyAccountDao: IAccountDao

MyAccountDao with hard-code

```
public class MyAccountDao : IAccountDao
{
    public string GetPassword(string account)
    {
        //todo hard-code
        if (account == "joey")
        {
            return "Joey hashed password";
        }
        else
        {
            return string.Empty;
        }
    }
}
```

實作 IHash

MyHash: IHash

MyHash with hard-code

```
public class MyHash : IHash
{
    public string GetHashResult(string password)
    {
        //todo hard-code
        if (password == "1234")
        {
            return "Joey hashed password";
        }
        else
        {
            return "abc hashed result";
        }
    }
}
```

Container.RegisterType

```
public static class Bootstrapper
{
  public static IUnityContainer Initialise()...

  private static IUnityContainer BuildUnityContainer()...

  public static void RegisterTypes(IUnityContainer container)
  {
      container.RegisterType<IAuthentication, MyAuthentication>();

      container.RegisterType<IAccountDao, MyAccountDao>();
      container.RegisterType<IHash, MyHash>();
  }
}
```

Demo – MyAccountDao & MyHash

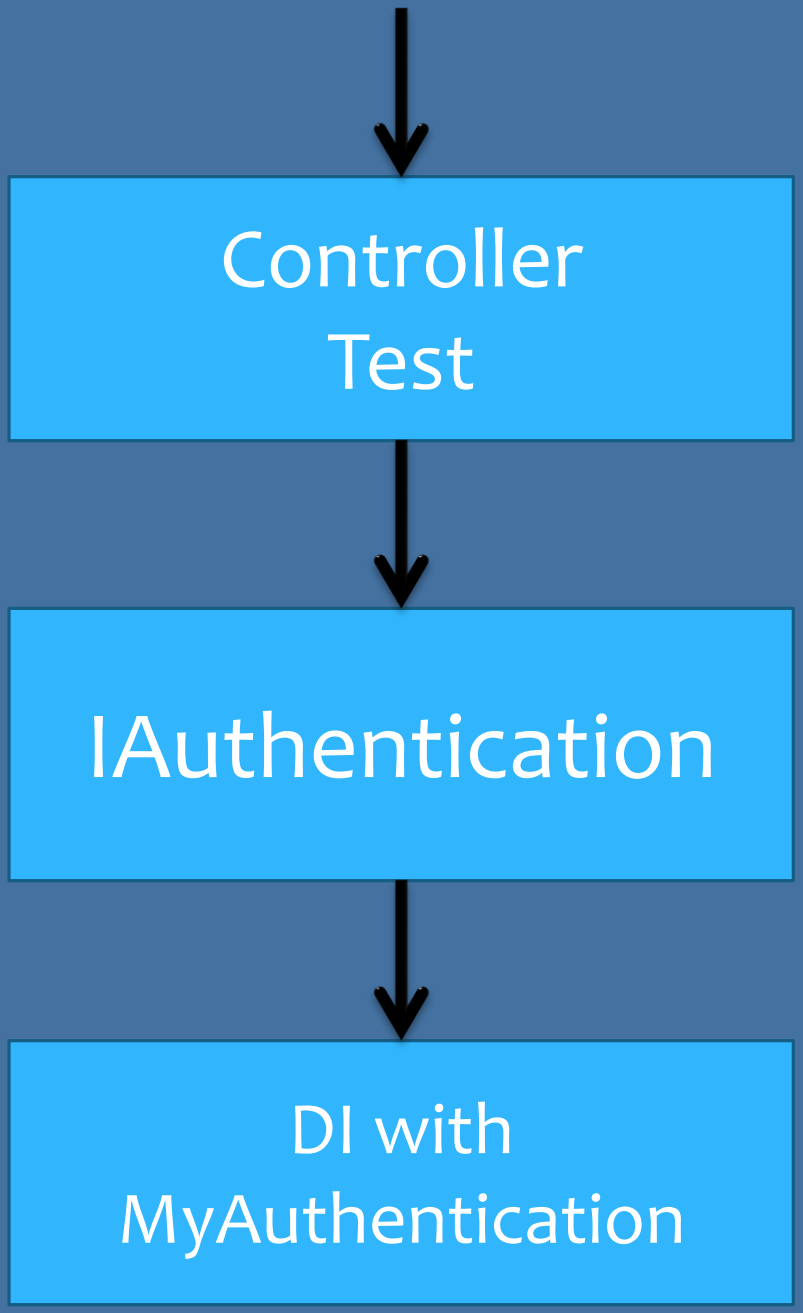


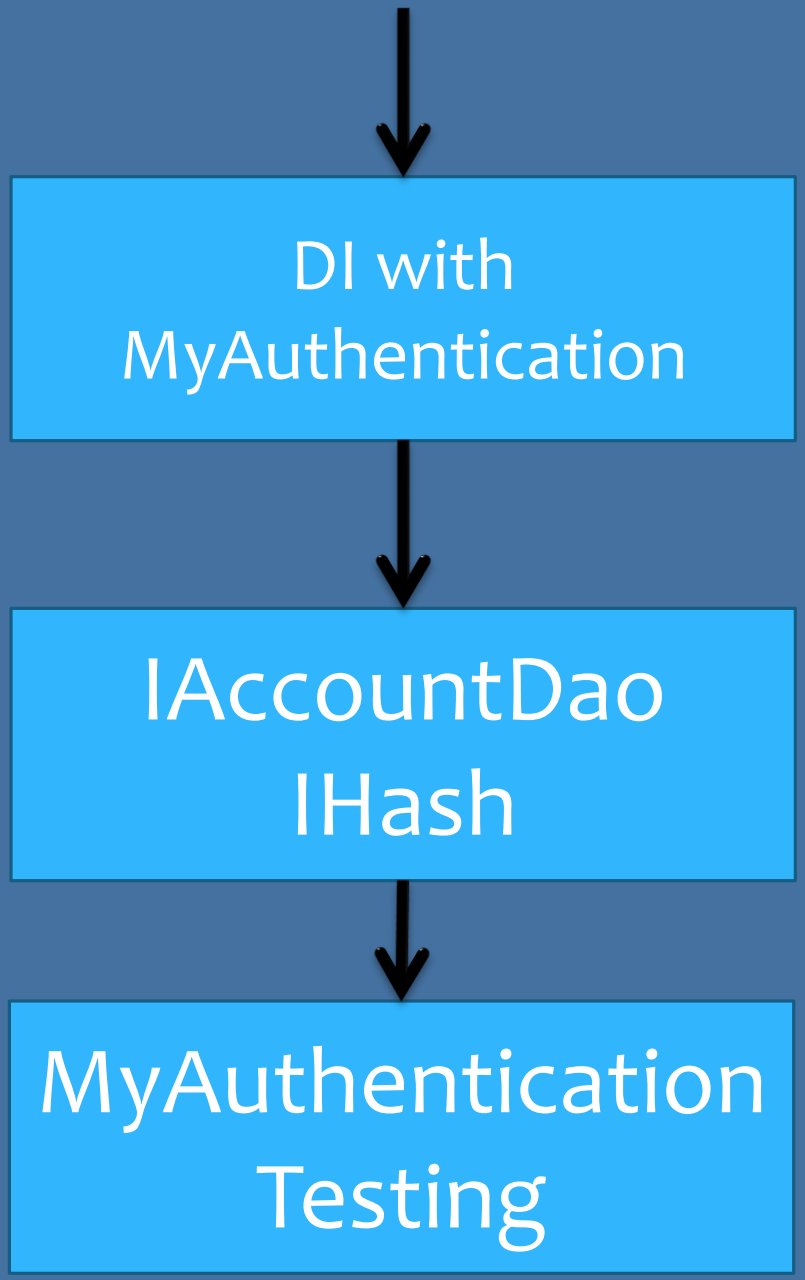
Next?

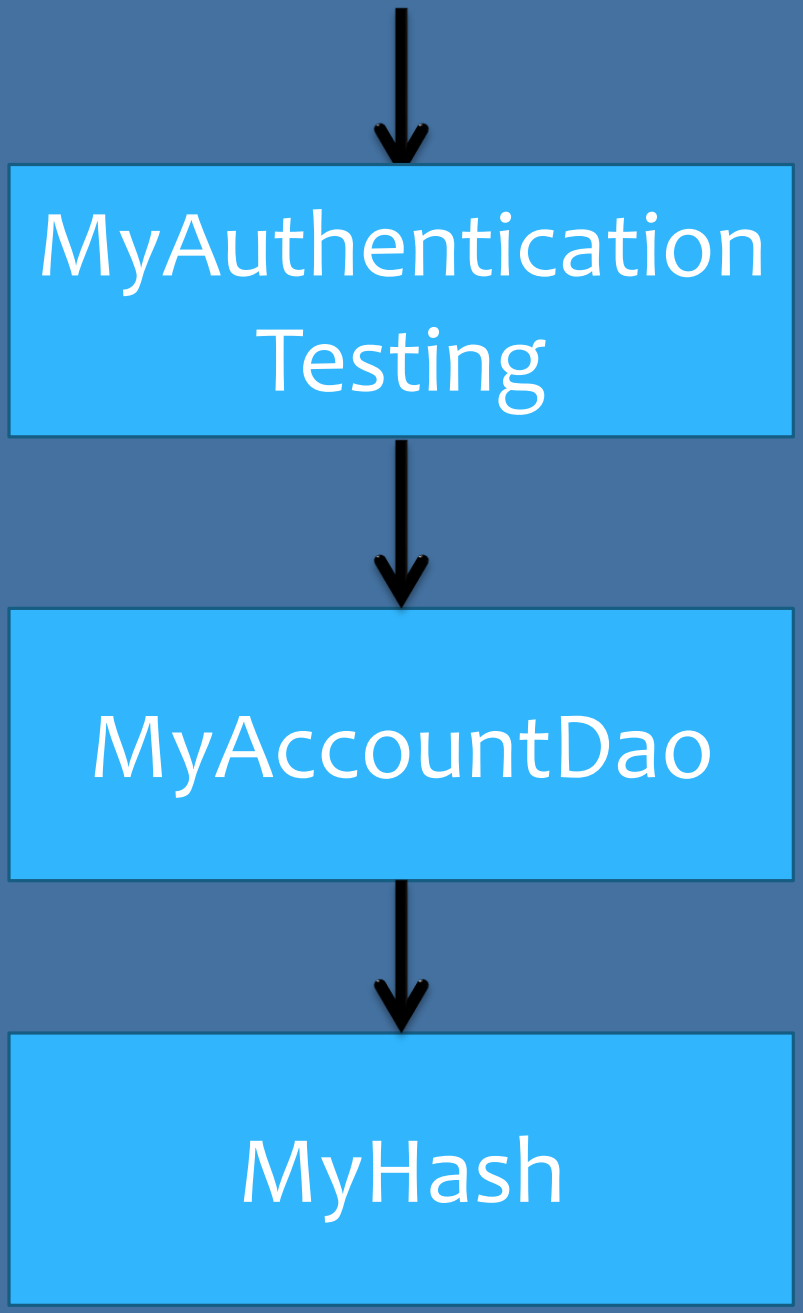
自己搞定
MyAccountDao
MyHash

Recap


需求
雛形
Selenium Test
Controller Test


Controller
Test
IAuthentication
DI with
MyAuthentication


DI with
MyAuthentication
IAccountDao
IHash
MyAuthentication
Testing


MyAuthentication Testing
MyAccountDao
MyHash

打完收工

結論


H.GravurePrint PAT.
Producted by HYUN-DAI CO., LTD. Taipei Taiwan

練習+練習+練習

你也可以！



Q & A

Thanks for your listening

謝謝各位

http://mvc.tw

- 本投影片所包含的商標與文字皆屬原著作者所有。
- 本投影片使用的圖片皆從網路搜尋。
- 本著作係採用 Creative Commons 姓名標示-非商業性-相同方式分享 3.0 台灣 (中華民國) 授權條款授權。