Hi Mosh,

I’m a big fan of your tutorial videos and thank you for offering this opportunity allowing me to ask questions about unit testing. I have some understanding about unit testing, it’s just when I’m writing my test code, I can’t be certain that I’m doing the right thing and how deep should I test. I have collected some code examples and put down some comments in code showing which part I am struggling with. Here goes, the following are in Windows Forms:

Question 1: How to test event handler, constructor and private methods?

using System;

using System.Messaging;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class Form1 : Form

{

private readonly MessageQueue \_messageQueue = new MessageQueue();

public Form1(string messageQueuePath)

{

InitializeComponent();

//How to test the following are being executed

\_messageQueue.ReceiveCompleted += new System.Messaging.ReceiveCompletedEventHandler(\_messageQueue\_ReceiveCompleted);

\_messageQueue.Path = messageQueuePath;

\_messageQueue.BeginReceive(new TimeSpan(0, 0, 3));

}

private void \_messageQueue\_ReceiveCompleted(object sender, ReceiveCompletedEventArgs e)

{

//How to test this

MessageQueue mq = (MessageQueue)sender;

mq.Formatter = new XmlMessageFormatter(new Type[]

{

typeof(Employee)

});

System.Messaging.Message m = mq.EndReceive(e.AsyncResult);

switch (m.Body.GetType().ToString())

{

case "Employee":

Process((Employee)m.Body);

break;

}

}

private void Process(Employee employee)

{

//How to test this

}

private void button1\_Click(object sender, EventArgs e)

{

//How to test the logic here

}

}

public class Employee

{

public string FirstName { get; set; }

public string LastName { get; set; }

public string Department { get; set; }

public string JobTitle { get; set; }

}

}

1. Should I test the UI? If so, how to test the UI? Should I use setter or getter or public methods?

public string Output

{

get { return lblOutput.Text; }

set { lblOutput.Text = value; }

}

public string GetOuput()

{

return lblOutput.Text;

}

public void SetOuput(string s)

{

lblOutput.Text = s;

}

1. Test database. How would you test external dependencies like the database. I understand in unit test you should not test the database. What about memory table (DataTable), like the following example. Should you test if data has been added to the data table? If so, what’s the right way to do it

private DataTable \_dataTable= new DataTable("Employee");

public Form1(string messageQueuePath)

{

InitializeComponent();

\_dataTable.Columns.Add("FirstName", typeof(string));

\_dataTable.Columns.Add("LastName", typeof(string));

\_dataTable.Columns.Add("Department", typeof(string));

\_dataTable.Columns.Add("JobTitle", typeof(string));

}

private void button1\_Click(object sender, EventArgs e)

{

//How or should I test the following

\_dataTable.Rows.Add(txtInput.Text);

}

1. Would you recommend any mock framework? I use Moq, it does the job but I only know the basic features, would you consider doing a course using Moq and/or any good unit testing tools?

Thank you for your time. I hope above example helped. If you have any questions please do not hesitate to contact me.

Best regards,

John Suen

Email: Johnnie.suen@gmail.com