Memo

CS445 – Spring15

VIN

Wine Club with REST interface

Lalit Patil

A20345732

Status of project: Complete

Use cases attempted:

A. Subscriber:

1. Add Subscriber
2. Modify Subscriber
3. View subscriber
4. Search subscriber
5. View Delivery info
6. Modify Delivery info

B. Shipment:

1. View Shipments
2. View Shipment details
3. Add Shipment Note
4. View Shipment Notes
5. View Shipment Note Details
6. Delete Shipment Note

C. Wine:

1. View Wines
2. View Wine details
3. Add Wine Note
4. View Wine Notes
5. View Wine Note Details
6. Modify Wine Note
7. Add Wine Rating
8. View Wine Rating

D. Admin:

1. Add admin
2. View Admins
3. View Admin details
4. Modify Admin
5. Add_monthly_selection
6. View_monthly_selection
7. View Revenue

E. Partner:

1. View subscriber list
2. Add receipt
3. View receipts
4. View Receipt details

VIN REST Request Examples:

"http://localhost:8080/VIN/receipt",

"http://localhost:8080/VIN/admin/monthly_selection

"http://localhost:8080/VIN/sub/2/wines/8/rating",

"http://localhost:8080/VIN/admin/1",

"http://localhost:8080/VIN/sub/2/wines/8/rating",

"http://localhost:8080/VIN/admin",

"http://localhost:8080/VIN/sub/2/shipments/11/notes/17",

"http://localhost:8080/VIN/admin/",

"http://localhost:8080/VIN/sub/2/wines/8/notes",

"http://localhost:8080/VIN/sub/2/search?q=\"Round\"",

"http://localhost:8080/VIN/sub/2/wines/8/notes",

"http://localhost:8080/VIN/sub/2/shipments/10",

"http://localhost:8080/VIN/sub/2/shipments/11/notes/15",

"http://localhost:8080/VIN/sub/2/shipments",

"http://localhost:8080/VIN/admin/revenue",

"http://localhost:8080/VIN/sub/2/shipments/11/notes",

"http://localhost:8080/VIN/receipt",

"http://localhost:8080/VIN/sub",

"http://localhost:8080/VIN/wines/5",

"http://localhost:8080/VIN/sub",

"http://localhost:8080/VIN/receipt",

"http://localhost:8080/VIN/sub/3",

"http://localhost:8080/VIN/sub/2/delivery",

"http://localhost:8080/VIN/sub/2/wines/8/notes", "http://localhost:8080/VIN/admin/1",

"http://localhost:8080/VIN/sub/2/wines/8/notes/15",

"http://localhost:8080/VIN/sub/2/search",

"http://localhost:8080/VIN/sub/2",

"http://localhost:8080/VIN/receipt/14",

"http://localhost:8080/VIN/admin/monthly_selection/12",

"http://localhost:8080/VIN/sub/2/shipments/11/notes",

"http://localhost:8080/VIN/sub/2/shipments/11/notes/14",

"http://localhost:8080/VIN/sub/2/wines/8"

Difficulties:

Before starting this project I was not aware about how the web services are created. So for me this was a good learning experience with lot of difficulties.

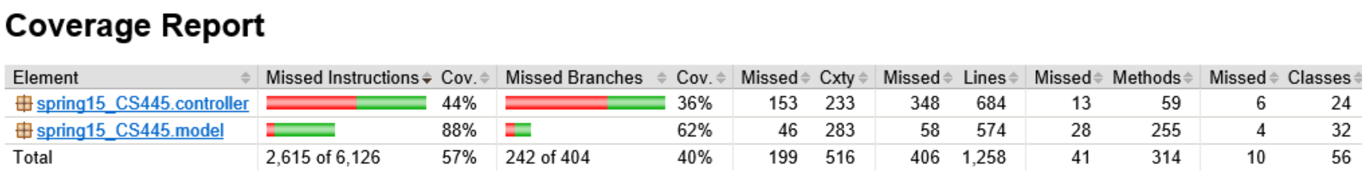
1. First problem I faced was, GET methods are ok but how to provide input to POST methods? How to test POST method? How to access the JSON input?

2. Another problem that I faced was related to linux environment. I was very much unfamiliar with build scripts and dependencies. Running junit test cases took a bit of time. Learning about ANT script build.xml, ivy.xml.

3. For code coverage I tried different plugins like EMMA, Clover, JaCoCo. Integrating them into script was a bit challenging. EMMA did not give good results. I finally settled with JaCoCo.

Statistics:

1. Unit test coverage:

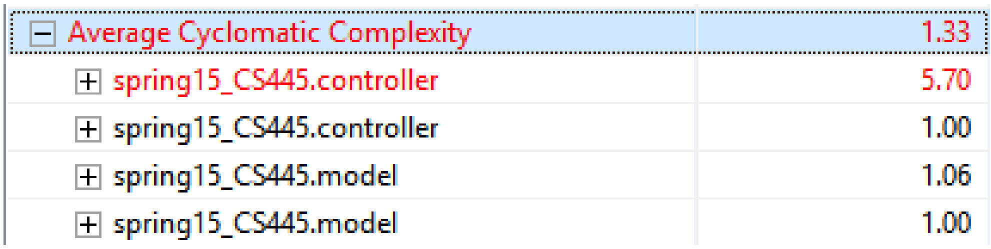


Coverage Report

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
spring15_CS445.controller	🟥🟩	44%	🟥🟩	36%	153	233	348	684	13	59	6	24
spring15_CS445.model	🟥🟩	88%	🟥🟩	62%	46	283	58	574	28	255	4	32
Total	2,615 of 6,126	57%	242 of 404	40%	199	516	406	1,258	41	314	10	56

Please refer to README.txt for instructions to regenerate test coverage report.

2. Cyclomatic Complexity:



⊟ Average Cyclomatic Complexity	1.33
⊞ spring15_CS445.controller	5.70
⊞ spring15_CS445.controller	1.00
⊞ spring15_CS445.model	1.06
⊞ spring15_CS445.model	1.00

3. Lines of Code (approx. excluding comments):

Controller: 1300 lines

Model : 1100 lines

4. Lines of Unit tests (approx., excluding comments):

Controller: 550 lines

Model: 600 lines

Design Patterns:

1. I have used MVC design pattern or you can say structure pattern. Although I haven't specified VIEW separately as in this project we are generally returning status code or json response. Controller handles processing and visualization of model data in this project.

2. I have used template pattern. MothlySelection model class is abstract class. Classes RW, AR and AW extend MonthlySelection class.

3. Iterator design pattern is used while accessing VinClub subscriber list. It simplifies traversing through subscriber without knowing about the index.