ဒီတစ္ေခါက္မွာေတာ့ ကြ်န္ေတာ္ Java နဲ ့ေရးေနတဲ့ programming blog website အေၾကာင္းကုိဆက္ေျပာထားပါတယ္။

အေျပာင္းအလဲအခ်ဴိ ့ေတြနဲ ့က်ြန္ေတာ္ေရးေနတဲ့ project ရဲ့ code အခ်ဴိ ့ေတြအေၾကာင္းေတြကုိအဓိကေျပာထားပါတယ္။ ဖတ္ၾကည့္ေပးပါဦး။

**I’m going to create a blogging platform in Java (Part – 2)**

ဒီတစ္ေခါက္မွာေတာ့ ကြ်န္ေတာ္ Java နဲ ့ေရးေနတဲ့ programming blog website အေၾကာင္းကုိဆက္ေျပာသြားပါမယ္။ သင္ကဒီအေၾကာင္းေတြနဲ ့မရင္းနွီးေသးရင္ Part 1 ကုိေတာ့ဒီမွာဖတ္လုိ ့ရပါတယ္။

ဒီတစ္ေခါက္မွာေတာ့ အေျပာင္းအလဲအခ်ဴိ ့ေတြ က်ြန္ေတာ္ေရးေနတဲ့ project ရဲ့ code အခ်ဴိ ့ေတြအေၾကာင္းကုိေျပာသြားပါမယ္။

**Overview vision changes**

Research and development အခ်ိဴ ့လုပ္ျပီးတဲ့အခ်ိန္ သူငယ္ခ်င္းအခ်ိဴ ့နဲ ့တုိင္ပင္ျပီးတဲ့အခ်ိန္မွာေတာ့ အဓိကအခ်က္သုံးခ်က္ေျပာင္းလဲသြားပါတယ္။

1. Blog မွာလုိအပ္တဲ့အရာေတြမွန္သမွ်ကုိ ကုိယ့္ဘာကုိလုိက္ေရးတာမ်ဴိးမဟုတ္ေတာ့ပဲ အဆင္သင့္ရွိျပီး အဆင္ေျပနုိင္တဲ့ Library ေတြ third party tools ေတြကုိအဆင္ေျပရင္ေျပသလုိသုံးသြားပါမယ္။
2. Website အေနနဲ ့ iPhone, Android, PC ေတြအကုန္မွာအဆင္ေျပေျပၾကည့္နုိင္ရပါမယ္။ ဒါမဲ့ေလာေလာဆယ္ေတာ့ စမ္းဖုိ ့ devices ေတြအစုံမရွိေသးပါဘူး။ ☺
3. Project တခုလုံးကုိ Spring Boot ကုိသုံးျပီးေရးျဖစ္သြားပါတယ္။ အလုပ္ရူပ္သက္သာျပီးေတာ့ေတာ္ေတာ္ေလးကုိအဆင္ေျပပါတယ္။

**Tools changes**

**(Primefaces Extenstion CKEditor) To (CKEditor) –** CKEditor ကုိ primefaces ရဲ့ extension ကိုမသုံးေတာ့ပဲ သူ ့ခ်ည္းပဲသက္သက္ official website ကေန download လုပ္ျပီးေတာ့ project မွာေပါင္းထည့္ပါမယ္။ ဘာလုိ ့လဲဆုိေတာ့ primefaces extension မွာက editor ရဲ့ features ကေတာ္ေတာ္နည္းပါေသးတယ္။ ထပ္ထည့္လုိ ့ရေပမယ့္ extension lib ကုိျပင္ရမွာျဖစ္တဲ့အတြက္ေတာ္ေတာ္အလုပ္ရူပ္ပါတယ္။ ေရရွည္အတြက္လည္းအဆင္မေျပပါဘူး။

**(JSF 2.2) To (Spring MVC + Thymeleaf) –** Primefaces extension ckeditor မသုံးေတာ့တဲ့အတြက္ JSF2.2 ကုိမသုံးေတာ့ဖုိ ့ဆုံးျဖတ္လုိက္ပါတယ္။ Spring boot application လည္းျဖစ္ေနတဲ့အတြက္ Spring MVC + Thymeleaf ပဲသုံးသြားပါေတာ့မယ္။

Client side validation အတြက္အလုပ္ေတာ့နည္းနည္းပုိလာပါတယ္။

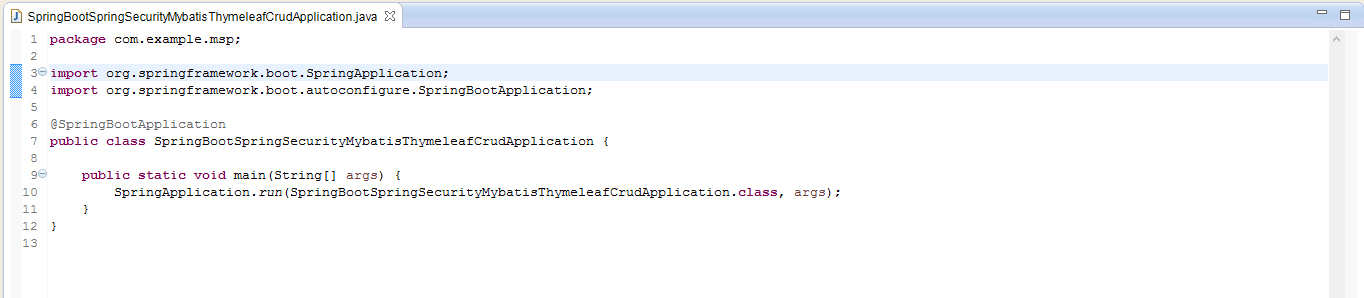
**Mybatis –** အစကေတာ့ ေခ်ြတာေရးအေနနဲ ့ storage အတြက္ကုိ database မသုံးပဲ Github မွာ File အေနနဲ ့သြားသိမ္းျပီးဆြဲထုတ္မလုိ ့လုပ္ပါတယ္၊ ဒါမဲ့လုိအပ္တဲ့အရာေတြက database နဲ ့သူ့ ့ functions ေတြသုံးမွရမွာပါ။ အဲ့ေတာ့ database နဲ ့ mybatis ကုိပဲသုံးဖုိ ့ေရြးလုိက္ပါတယ္။ Project မွာ select query ေတြကအဓိကသုံးမွာဆုိေတာ့ query ေကာင္းေကာင္းေရးပီးထိန္းလုိ ့လြယ္တဲ့ mybatis နဲ ့ဆုိပုိအဆင္ေျပပါတယ္။ Spring ရဲ့ transaction နဲ ့တြဲျပီးသုံးသြားပါမယ္။

**Junit 4 –** Testing အတြက္သုံးသြားပါမယ္။ Version 4 မွာထဲ့ထားတဲ့ *assertThat()* ဆုိတဲ့ method တစ္ခုနဲ ့တင္ေတာ္ေတာ္ေလးအသုံး၀င္ေနပါျပီ။

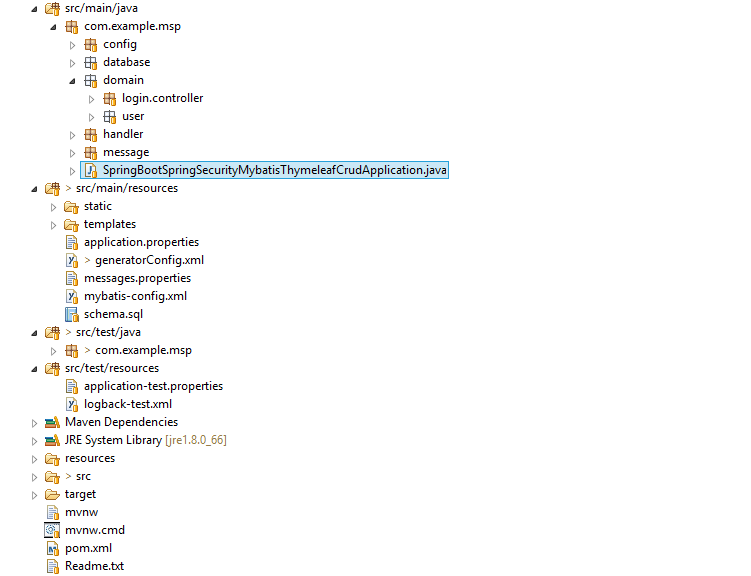
**Starting the project**

Spring Initializr ကုိသုံးျပီး project စေဆာက္ခဲ့ပါတယ္။ Spring boot application အတြက္လုိအပ္တဲ့ dependencies ေတြကုိစိတ္ၾကိဳက္အလြယ္တကူေရြးထည့္လုိ ့ရပီး သူက ျပင္ဆင္ပီးသား project တခုကုိယ့္အတြက္ထုတ္ေပးပါတယ္။ အဲ့ဒါကုိ ကုိယ့္ရဲ့ Eclipse ထဲထည့္လုိက္ယုံပါပဲ။ Eclipse ထဲ import နဲ ့ထဲ့တဲ့အခါ “Existing Maven Project” ကေနတဆင့္ပဲထည့္ဖုိ ့မေမ့ပါနဲ ့။

Spring Initializr ကေန generate ထုတ္ေပးျပီးတဲ့အခါထြက္လာတဲ့ application main class ကေအာက္ကအတုိင္းပါပဲ။

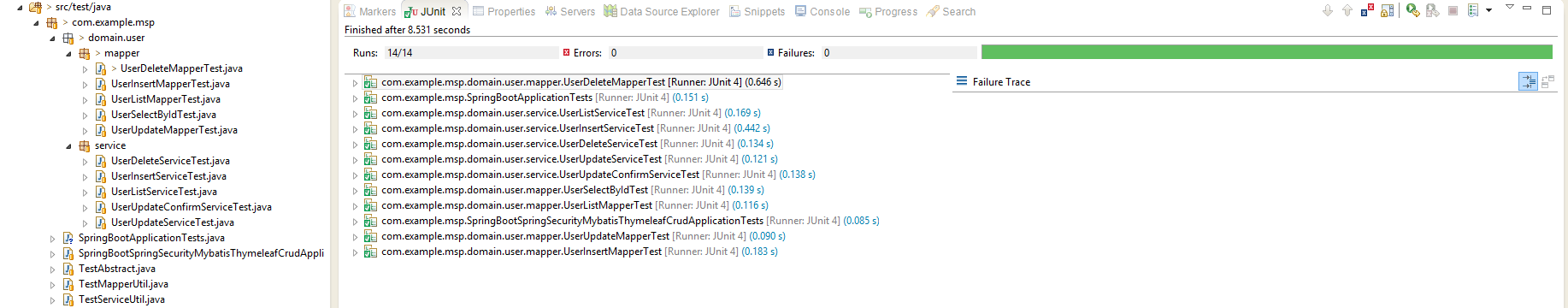


Project structure ကေတာ့ေအာက္ကအတုိင္းသုံးသြားပါမယ္။

****

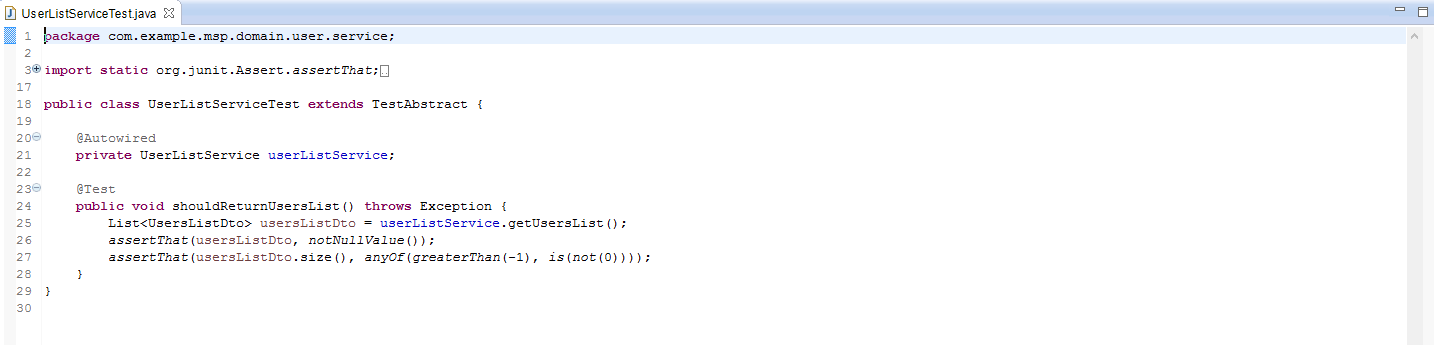
Testing လုပ္မဲ့ source folder ကတပုိင္း development တကယ္လုပ္တဲ့ source folder ကတပုိင္းပါ။ ျပီးေတာ့ project မွာ package structure ကုိ layer အလုိက္မဟုတ္ပဲ domain အလုိက္ခြဲထားပါတယ္။

**Testing (Junit4)**

****

Junit 4 ကုိသုံးျပီးအဓိကေရးသြားပါမယ္။ Test Driven Development (TDD) ပုံစံမ်ိဴးေရးေပမယ့္ တကယ့္ TDD အစစ္ေတာ့မဟုတ္ေသးပါဘူး။ က်ြန္ေတာ္ TDD နဲ ့ပတ္သက္ျပီး အေတြ ့အၾကံဳလုိေနေသးတာလည္းပါပါတယ္။ ပုံမွာျပထားတဲ့အတုိင္း service and dao layers ကုိအဓိကစစ္ထားပါတယ္။ အဲ့ဒီ class တစ္ခုစီအတြက္ test class တစ္ခုရွိပါမယ္။ UI ပုိင္းကုိစစ္ဖုိ ့အတြက္ေတာ့ website release လုပ္ျပီးမွပဲဆက္ျပီးေတာ့ေရးသြားပါမယ္။

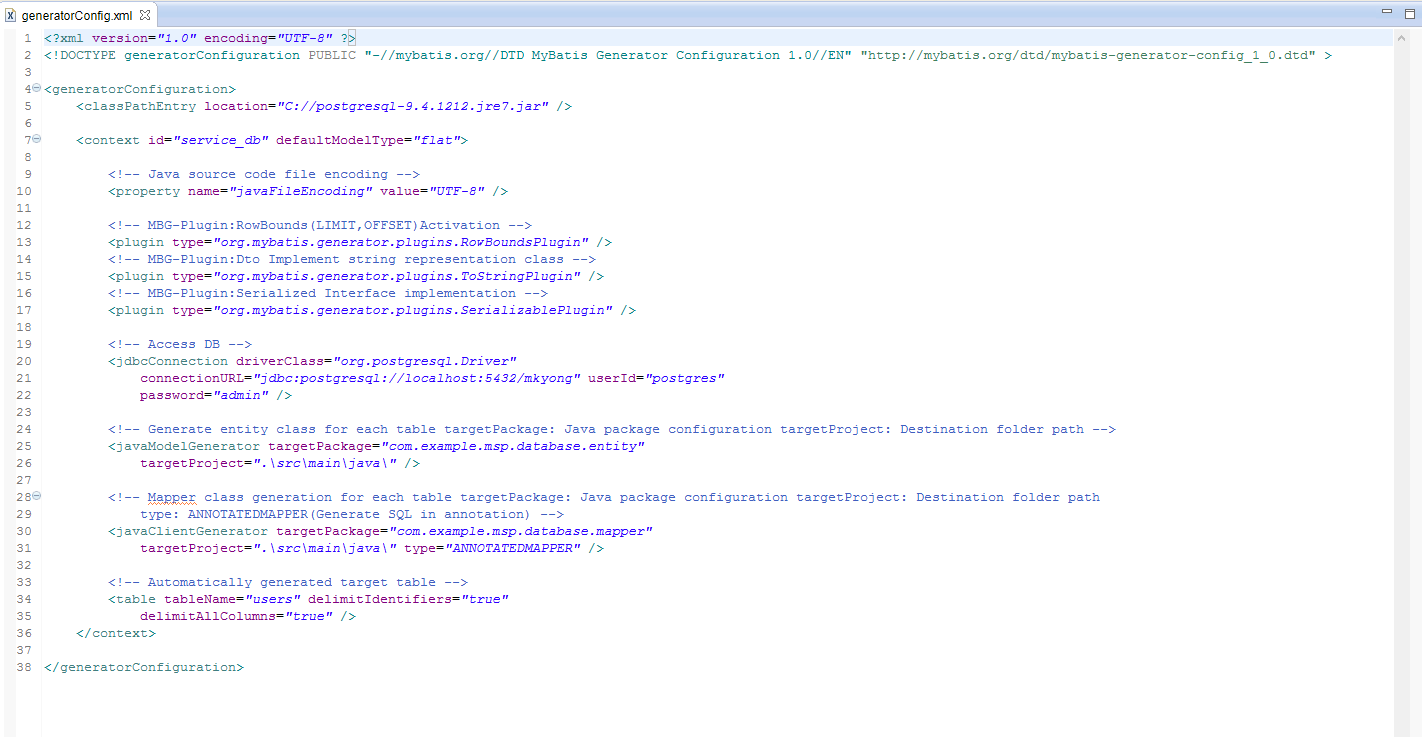
Test code sample ေရးထားတဲ့ပုံစံကုိေတာ့ ဒီမွာၾကည့္လုိ ့ရပါတယ္။

****

**Mybatis Generator**

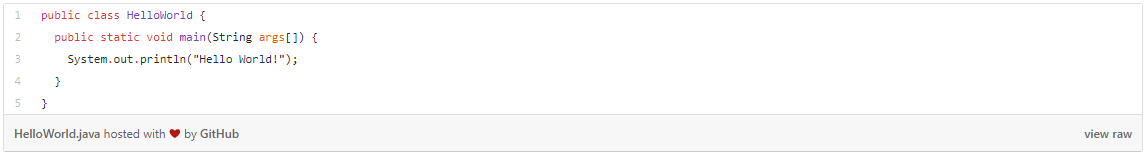
Mybatis generator ကေန ကုိယ္သုံးမဲ့ table ေတြရဲ့ entity class ေတြ query ေရးပီးသား class ေတြကိုထုတ္ေပးပါတယ္။ Generator plugin ထဲ့ပီး config file ေရး run လုိက္တာနဲ ့ရပါတယ္။

Productivity ကေတာ္ေတာ္တက္သြားပီး လုပ္ရကုိင္ရတာတကယ္အဆင္ေျပပါတယ္။ Config sample file ေလးေလာက္ပဲဒီမွာတြဲတင္ျပလုိက္ပါမယ္၊ အေသးစိတ္ကေတာ့ထုံးစံအတုိင္းေပးထားတဲ့လင့္ကေနတဆင့္သြားၾကည့္လုိ ့ရပါတယ္။



**Code Snippet (Gists)**

Github ရဲ့ feature တစ္ခုျဖစ္တဲ့ gists ဟာကုိယ့္ရဲ့ source code file ေတြကုိတစ္ခုခ်င္းစီအလုိက္သိမ္းထားလုိ ့ရပါတယ္။ ျပီးေတာ့ကုိယ့္ရဲ့ blog post ေတြမွာ code ေတြျပတဲ့အခါ code snippet တစ္ခုအေနနဲ ့ေခၚျပီးျပလုိ ့ရပါတယ္။ ျပရုံအျပင္ Github နဲ ့ပါခ်ိတ္ထားတဲ့အတြက္တခါတည္း download သြားလုပ္လုိ ့ရလုိ ့ပုိအဆင္ေျပပါတယ္။ နမူနာကုိေတာ့ေအာက္မွာၾကည့္ပါ။

****

**Conclusion**

ဒါဆုိေတာ္ေတာ္မ်ားမ်ားအဆင္ေျပသြားပါပီ။ Part-3 (final part) ကုိေတာ့ website release လုပ္ျပီးတဲ့အခါထပ္တင္သြားပါမယ္။ အဲ့ဒီအပုိင္းမွာေတာ့ website ၾကည့္တဲ့လူေတြအေနနဲ ့ comments ေပးတဲ့အပုိင္းေတြ share လုပ္လုိ ့ရမယ့္အပုိင္းေတြမွာသုံးထားတဲ့ technology အေၾကာင္းေတြနည္းနည္းထပ္ေျပာသြားပါမယ္။ ျပီးေတာ့ က်ြန္ေတာ္၀ယ္သုံးမဲ့ hosting နဲ ့ domain အေၾကာင္းေတြ၊ production site ကုိ release လုပ္ဖုိ ့အတြက္ continuous integration server ဘာေတြသုံးလည္းဆုိတာနဲ ့ blog website မွာဘာေတြအဓိကထားေရးသြားမလဲဆုိတာေတြထပ္ေျပာသြားပါမယ္။

**Final Words**

ဒီထဲကအၾကာင္းေတြနဲ ့ပတ္သက္ျပီးသင့္အတြက္လုိအပ္တဲ့ဟာရွိရင္မက္ေဆ့ျဖစ္ျဖစ္

email ျဖစ္ျဖစ္ပုိ ့ျပီးေမးလုိ ့ရပါတယ္။ က်ြန္ေတာ္လုပ္ေနတဲ့အရာနဲ ့ပတ္သက္ျပီးအၾကံဥာဏ္ေပးဖုိ ့ေ၀ဖန္ဖို ့ကုိလည္းၾကိဳဆုိပါတယ္။