Generic Appointment Application V1.0

Table of Contents

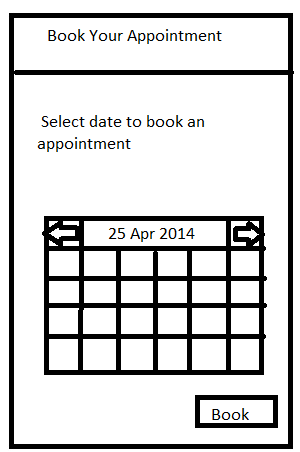
[1. Screen Designs Generic Appointment Application 1](#_Toc386884014)

[2. GIT Application Check in Initial Setup 3](#_Toc386884015)

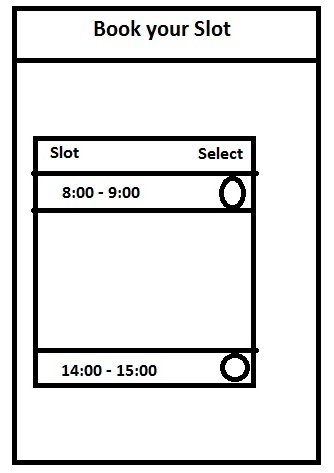
[3. Technology & Tools used 5](#_Toc386884016)

# 1. Screen Designs Generic Appointment Application

Screen 1:- Generic appointment booking application screen 1 Book Your Apointment to select today , future dates. Select date for appointment and click on book button.



Screen 2 :- Generic appointment booking application screen 2 Book Your Slot to select slot available.Select appointment slot and click on book button.

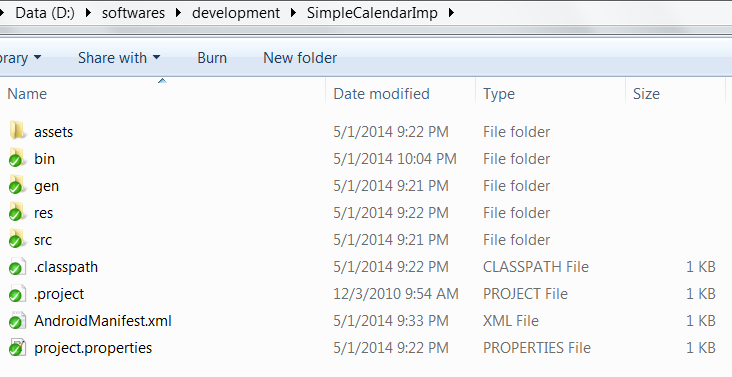


# 2. GIT Application Check in Initial Setup

When you are creating any application from scratch in your workspace and want to attach it to remote git repo below steps will help you to achieve this.

**Step 1:-** Go to your workspace root directory and run command > git init as shown below. After running init command your project will look like as shown in screen 2

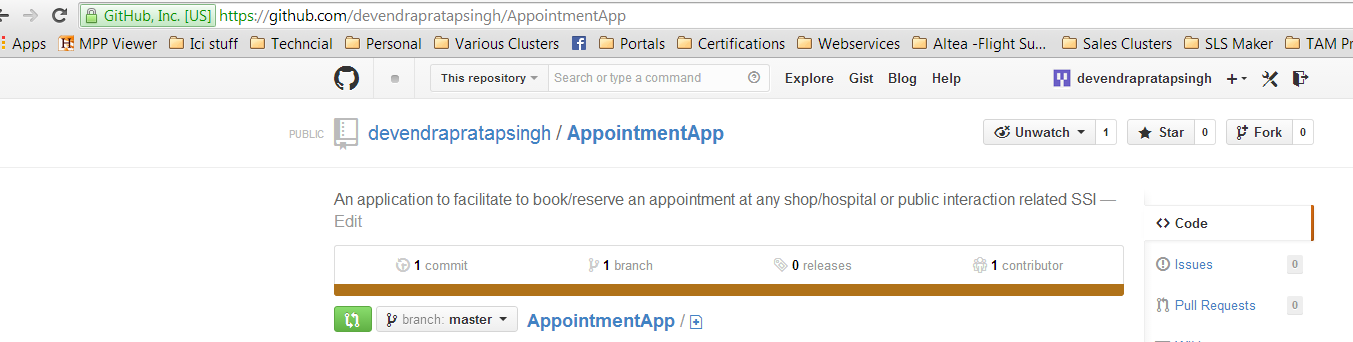




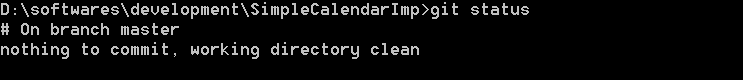
Step 2 :- You have initiated the git repository on your workspace now attach the local git repository to remote git repository by running below command & as shown .This will attach your local GIT repo to your remote repo created in github or bitbucket (which ever source repository you choose).In our case we are using github where we have created repository name “AppointmentApp” :-

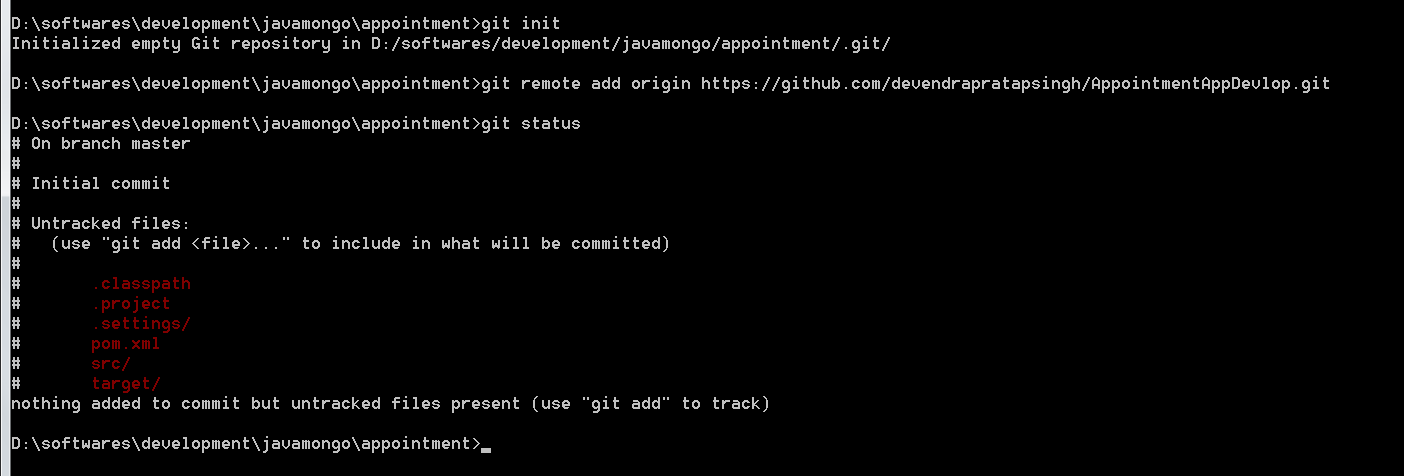
D:\softwares\development\SimpleCalendarImp>git remote add origin <https://github.com/devendrapratapsingh/AppointmentApp.git>





Step 3 :- After successfully creating local git repo and attaching it to remote github repo check the status of your git repo with command :- git status



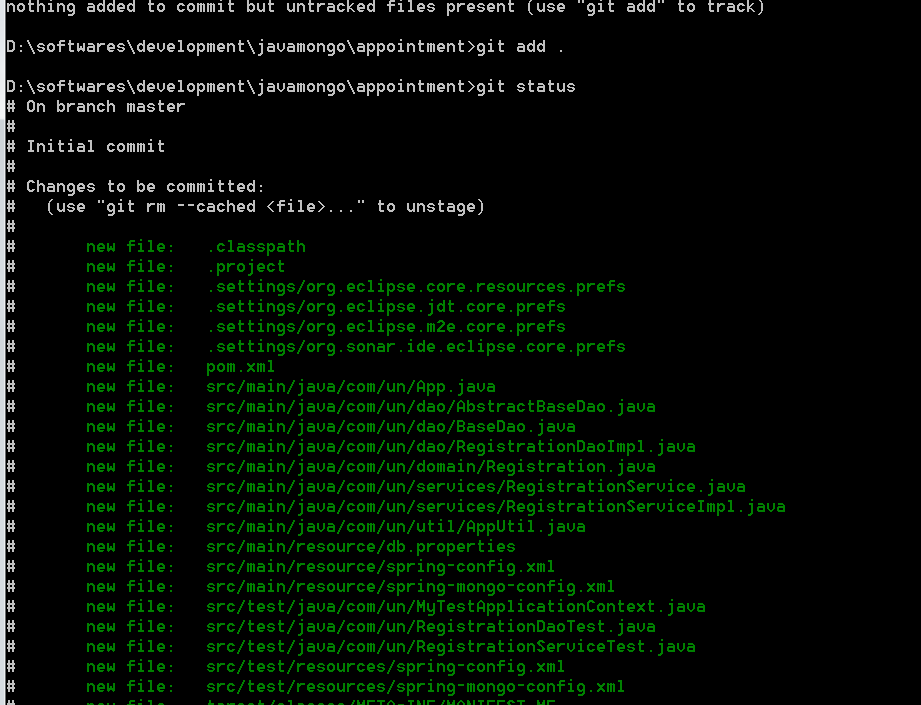


Step 4 :- Now as it is your first time check in blank git repo you should add all files in your local git repo to git with below command as shown below :- git add .

Note :- .(dot) will add all your files in present directory to git.



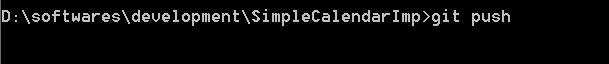
After adding file in local git repo if you do again git staus below status will be shown for added files as green :-



Step 5 :- As your all files are added in local git repo you can commit to local github repo with below commands :- git commit -m "first time added"



Step 6 :- After committing in git local repo your workspace changes you can push these changes to remote github repo with below command :- git push

****

# 3. Technology & Tools used

Installation for Phone gap follow below link :-

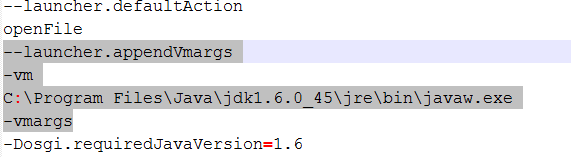
<http://phonegap.com/install/>

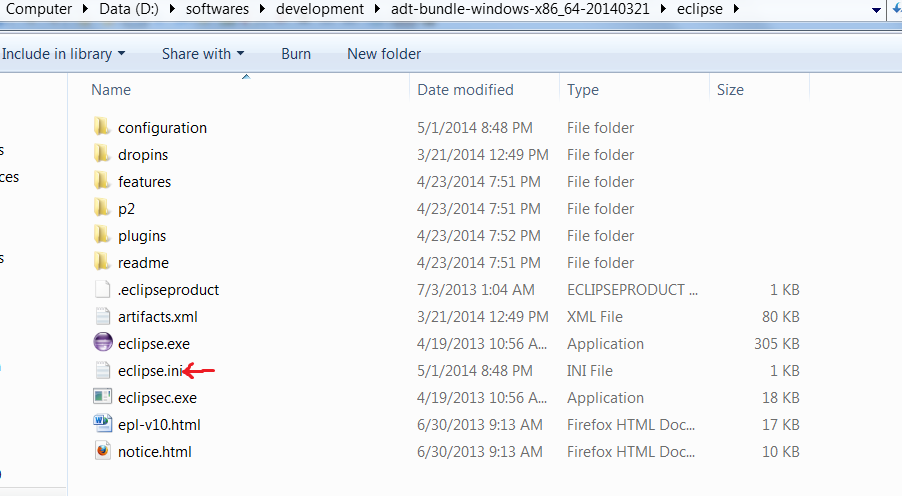
## Know Error in Setup

1. If you eclipse is not opening due to JDK issue please follow below steps to resolve it :-

Edit your SDK eclipse.ini file to include JDK 1.6 as shown below :-

Add in eclipse.ini file entry “–vm” in one line and on next line add “C:\Program Files\Java\jdk1.6.0\_45\jre\bin\javaw.exe”





2) Phone Gap commands gives below error “error an error occurred while listing android targets windows” . Set JAVA\_HOE to JDK 1.6 instead of JDK 1.5 and set in your path variable for

* android-sdk/tools
* android-sdk/platform-tools

.

See Answer 2 in <http://stackoverflow.com/questions/20953488/phonegap-run-android-phonegap-3-under-windows-cant-build-a-project>

3) How to run your first phone gap application

**Some important URL’s that we can use for reference:**

1. Setting up github

<https://help.github.com/articles/synchronizing-repositories/>

2. Calling Restful service from Android phone

<http://programmerguru.com/android-tutorial/android-restful-webservice-tutorial-how-to-create-restful-webservice-in-java-part-2>

3. Download Jersey Framework jar for Restful webservice

<http://www.vogella.com/tutorials/REST/article.html#installation_jersey>

Download advance rest client for testing webservice on Chrome

<http://programmerguru.com/android-tutorial/minimum-required-sdk-target-sdk-compile-with/Android>

Minimum SDK requirement and Android platform versions.

4. Asynchronous call back based http client

<http://loopj.com/android-async-http/>

5. Mongodb installation and starting a service

<http://docs.mongodb.org/v2.2/tutorial/install-mongodb-on-windows/>

6. Mongodb connectivity with Java progam using mongo.jar drivers

<http://www.tutorialspoint.com/mongodb/mongodb_java.htm>

7. Mongotemplate CRUD operations

<http://www.mkyong.com/mongodb/spring-data-mongodb-hello-world-example/>

8. Building rest web services

<http://spring.io/guides/gs/rest-service/>

9. Spring best practices

<http://howtodoinjava.com/2014/01/05/13-best-practices-for-writing-spring-configuration-files/>

<http://javapapers.com/spring/spring-component-service-repository-controller-difference/>

<http://javapapers.com/spring/spring-annotation-based-controllers/>

<http://javapapers.com/core-java/java-annotations/>

10. JUnit test in Spring (Autowiring and Application context)

<http://www.codesolid.com/2013/05/16/spring-unit-testing-using-junit/>

<http://www.javacodegeeks.com/2012/10/junit-testing-spring-service-and-dao.html>

<http://java.dzone.com/articles/junit-testing-spring-mvc>

<http://docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/testing.html>

**11. Android tutorial**

<http://www.raywenderlich.com/78574/android-tutorial-for-beginners-part-1>

<https://developer.android.com/training/basics/firstapp/index.html>

* End of Document --