README

1. SERVER

I have used Perfect-Swift-Server for the backend, and MySQL for Database. There are the following prerequisites.

- Swift 3.0.2 and Xcode 8 to upper.
- MySQL 5.7 to upper
- macOS 10.12 to upper

Follow the following instructions to setup the server.

A. Setup Homebrew.

i. Install and setup Homebrew if you don't have already on your MAC. Go to and follow the instructions at https://brew.sh/ it set it up.

B. Setup MySQL

If you don't have mysql setted up on your mac via Brew/Homebrew then follow this section. Note that, you need to install MySQL via **Brew** only, not via MAMP or XAMPP.

If you have already MySQL setted up via MAMP or XAMPP or MySQL installer then probably Homwbrew version of MySQL it won't work. Because Homebrew version of MySQL may get confused with other instance of MySQL. In that case please take a backup of your database and remove all of the instances and installations of My SQL. Because homebres mysql server get confused and won't work.

Follow the instructions to create a fresh setup of MySQL via homebrews. Also you can check this link >> http://stackoverflow.com/a/38687225/2089253

i. Remove previous MySQL installations completely. Enter the following commands on your terminal.

```
killall -u username

brew remove mysql

brew cleanup

sudo rm /usr/local/mysql

sudo rm -rf /usr/local/var/mysql

sudo rm -rf /usr/local/mysql*

sudo rm ~/Library/LaunchAgents/homebrew.mxcl.mysql.plist

sudo rm -rf /Library/StartupItems/MySQLCOM
sudo rm -rf /Library/PreferencePanes/My*

launchctl unload -w ~/Library/LaunchAgents/homebrew.mxcl.mysql.plist

rm -rf ~/Library/PreferencePanes/My*

sudo rm -rf /Library/Receipts/mysql*

sudo rm -rf /Library/Receipts/MySQL*

sudo rm -rf /private/var/db/receipts/*mysql*
```

ii. Reinstall MySQL server with Homebrew. Enter the following command on the terminal

```
brew doctor

brew update

brew install mysql

unset TMPDIR

mysqld -initialize --verbose --user=whoami --basedir="$(brew --prefix mysql)" --
datadir=/usr/local/var/mysql --tmpdir=/tmp

mysql.server start

brew services start mysql
```

C. Connect to Database

Now it’s time to connect to database and create data schema. Enter to following commands on the MAC terminal after successfully starting MySQL server.

i. **mysql -u root -h 127.0.0.1 -p**

Here `root` is username and `127.0.0.1 ` is host

Now, if it asks for password, just hit ENTER because `root` doesn’t need any password in this case.

ii. **CREATE DATABASE crossover_database;**

It will create a database named `crossover_database ` for `root` user

iii. **INSERT INTO person_items VALUES ('UnitTestSSN','Test User','Male',9857286387,0);**

This insersion in mandatory to run the unit test cases and successfully pass.

D. Setup API Server

AT this point you will need to setup and run API Server

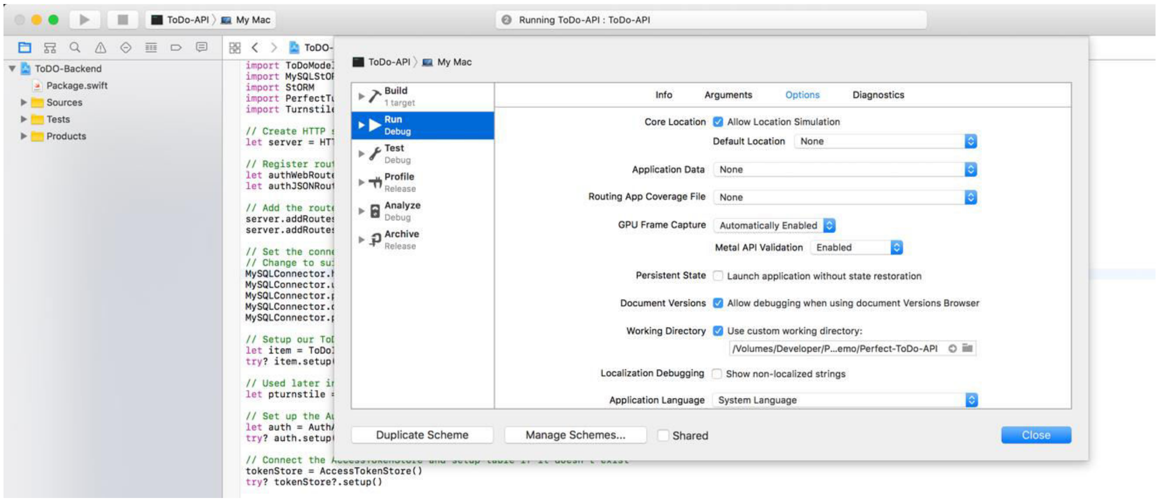
i. A t this point in time you will need to edit the **mysqlclient.pc** file located here:

/usr/local/lib/pkgconfig/mysqlclient.pc

Remove the occurrence of **"-fno-omit-frame-pointer"**. This file is read-only by default so you will need to change its permissions first.

E. Setup Xcode-Server project.

Goto …>>Server>> FamilyTree-Backend directory. Open “FamilyTree-Backend.xcodeproj” file. You MUST change to the executable target(To the BLACK one between these 2 target) AND setup a custom working directory for its scheme in order for the database to create and work properly.



```
import ToDoModel
import MySQLStOR
import StORM
import PerfectTu
import Turnstile

// Create HTTP s
let server = HTT

// Register rou
let authWebRoute
let authJSONRout

// Add the route
server.addRoutes
server.addRoutes

// Set the conne
// Change to sui
MySQLConnector.h
MySQLConnector.u
MySQLConnector.p
MySQLConnector.d
MySQLConnector.p

// Setup our ToD
let item = ToDoI
try? item.setup(

// Used later in
let pturnstile =

// Set up the Au
let auth = AuthA
try? auth.setup(

// Connect the
tokenStore = AccessTokenStore()
try? tokenStore?.setup()
```

Core Location ☑ Allow Location Simulation

Default Location: None

Application Data: None

Routing App Coverage File: None

GPU Frame Capture: Automatically Enabled

Metal API Validation: Enabled

Persistent State □ Launch application without state restoration

Document Versions ☑ Allow debugging when using document Versions Browser

Working Directory ☑ Use custom working directory:

/Volumes/Developer/P...emo/Perfect-ToDo-API

Localization Debugging □ Show non-localized strings

Application Language: System Language

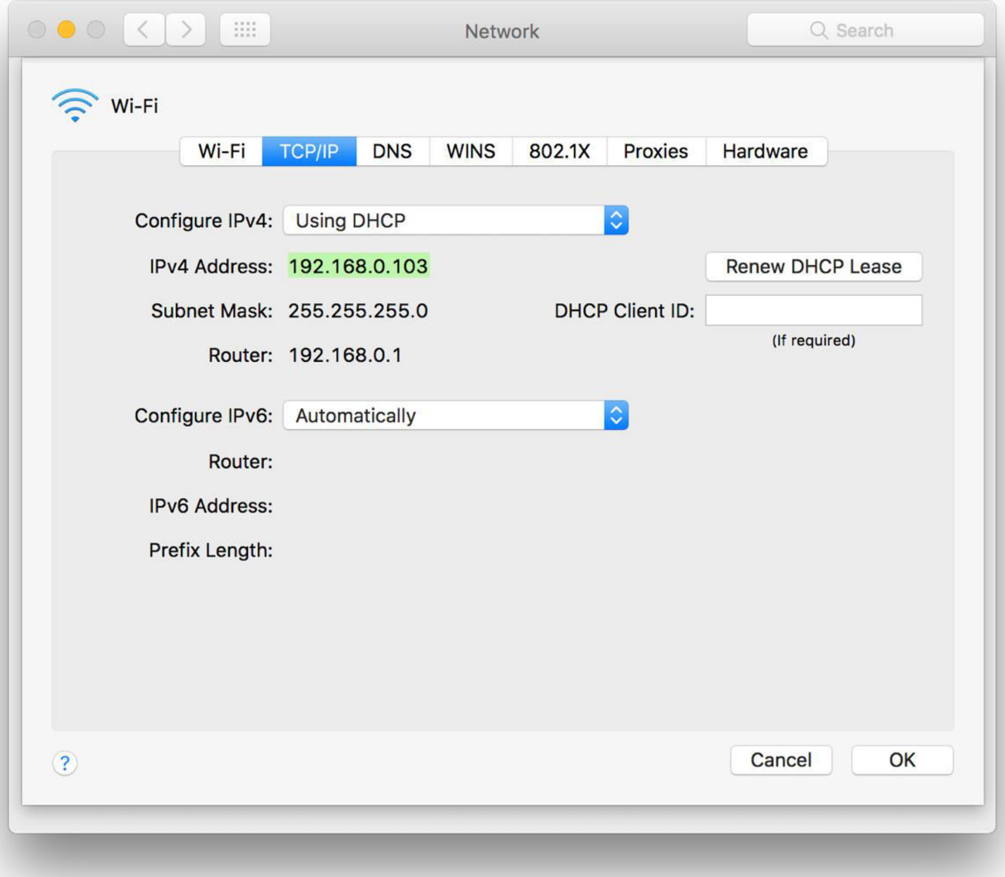
□ Shared

F. Configure API Server Xcode Project

On your MAC, go to System Preferences >> Network >> Select your connected ISP/connection >> Click “Advanced” tab >> Click “TCP/IP” tab >> And look for and copy IPv4Adresses.

This is gonna be your HTTP API server address.

Copy and pest this IPv4 address into **Utils.swift >> struct HTTP_Server_Config >> serverAddress** field.



Configure IPv4: Using DHCP

IPv4 Address: 192.168.0.103

Subnet Mask: 255.255.255.0

DHCP Client ID:

Router: 192.168.0.1

Configure IPv6: Automatically

Router:

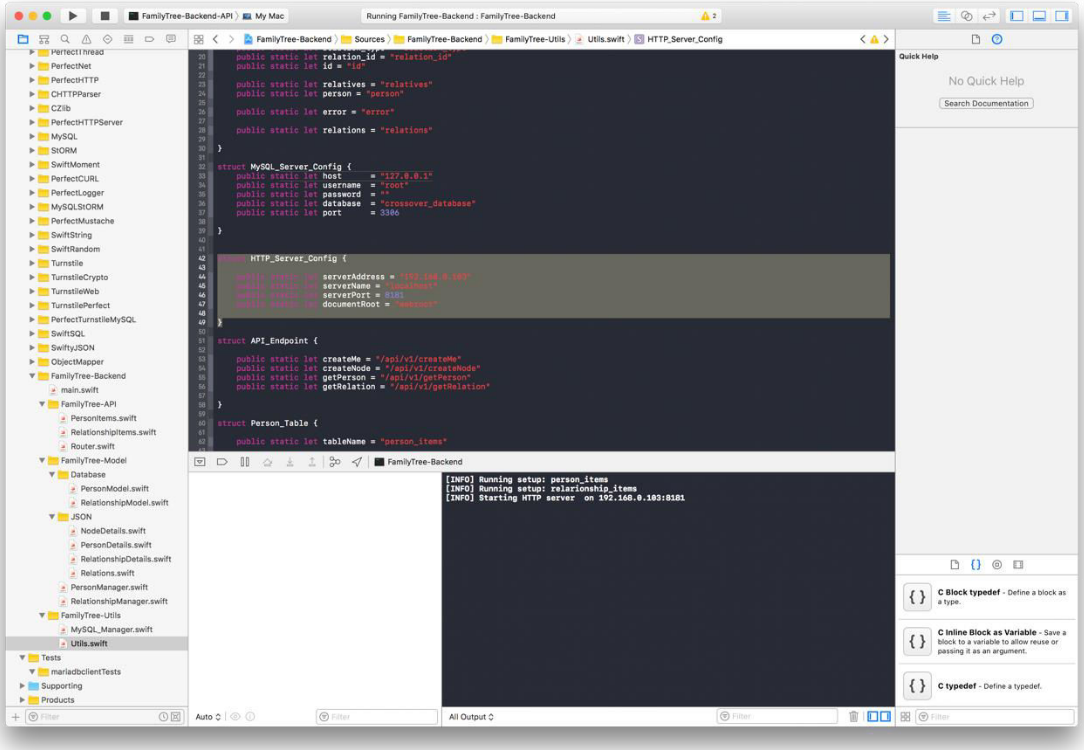
IPv6 Address:

Prefix Length:

G. Start API server

Run the target named “**FamilyTree-Backend-API**”.

Upon successful running the server you will see something like the following screenshot on the console.



```swift
    public static let relation_id = "relation_id"
    public static let id = "id"

    public static let relatives = "relatives"
    public static let person = "person"

    public static let error = "error"

    public static let relations = "relations"

}

struct MySQL_Server_Config {
    public static let host       = "127.0.0.1"
    public static let username   = "root"
    public static let password   = ""
    public static let database   = "crossover_database"
    public static let port       = 3306

}


struct HTTP_Server_Config {

    public static let serverAddress = "192.168.0.103"
    public static let serverName = "localhost"
    public static let serverPort = 8181
    public static let documentRoot = "webroot"

}

struct API_Endpoint {

    public static let createMe = "/api/v1/createMe"
    public static let createNode = "/api/v1/createNode"
    public static let getPerson = "/api/v1/getPerson"
    public static let getRelation = "/api/v1/getRelation"

}

struct Person_Table {

    public static let tableName = "person_items"
```

```
[INFO] Running setup: person_items
[INFO] Running setup: relarionship_items
[INFO] Starting HTTP server  on 192.168.0.103:8181
```

2. IOS APP(SWIFT)

I have used Swift for the writing iOS App. There are the following prerequisites.

- Swift 3.0.2 and Xcode 8 to upper.
- iOS 8 to upper
- macOS 10.12 to upper
- Cocoa pods Latest

A. Installing Cocoa pods and its dependencies.

i. If you don’t have CocoaPods installed already then follow this link to install it.

ii. Now you need to install the dependencies for the project.

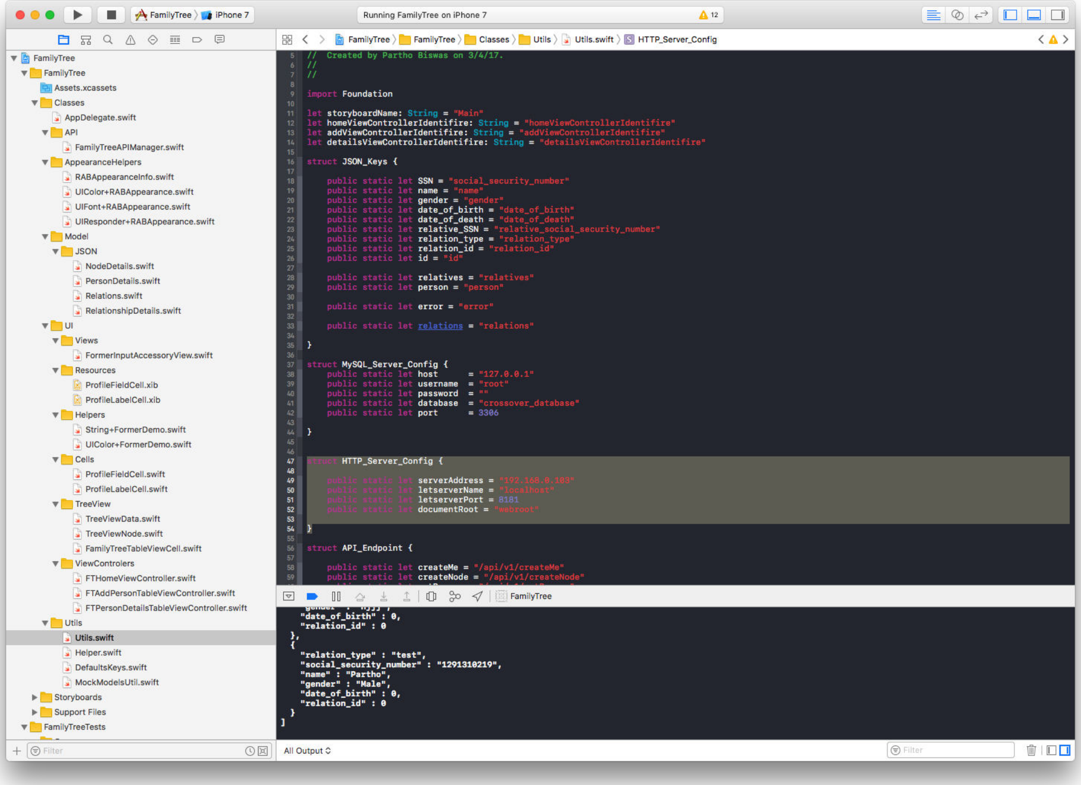
iii. To do this Open the terminal and and navigate to the project root directory where “FamilyTree.xcodeproj” file resides.

iv. Now enter the “pod install" on the terminal and hit ender to install the dependencies.

v. Then open “FamilyTree.xcworkspace” file

B. Running the app

Now just need to set Server IP. Copy and pest that(the one, user to start the HTTP server while setting up the server) IPv4 address into **Utils.swift >> struct HTTP_Server_Config >> serverAddress** field.



```
//  Created by Partho Biswas on 3/4/17.
//
//

import Foundation

let storyboardName: String = "Main"
let homeViewControllerIdentifire: String = "homeViewControllerIdentifire"
let addViewControllerIdentifire: String = "addViewControllerIdentifire"
let detailsViewControllerIdentifire: String = "detailsViewControllerIdentifire"

struct JSON_Keys {

    public static let SSN = "social_security_number"
    public static let name = "name"
    public static let gender = "gender"
    public static let date_of_birth = "date_of_birth"
    public static let date_of_death = "date_of_death"
    public static let relative_SSN = "relative_social_security_number"
    public static let relation_type = "relation_type"
    public static let relation_id = "relation_id"
    public static let id = "id"

    public static let relatives = "relatives"
    public static let person = "person"

    public static let error = "error"

    public static let relations = "relations"

}

struct MySQL_Server_Config {
    public static let host      = "127.0.0.1"
    public static let username  = "root"
    public static let password  = ""
    public static let database  = "crossover_database"
    public static let port      = 3306

}


struct HTTP_Server_Config {

    public static let serverAddress = "192.168.0.103"
    public static let letserverName = "localhost"
    public static let letserverPort = 8181
    public static let documentRoot = "webroot"

}

struct API_Endpoint {

    public static let createMe = "/api/v1/createMe"
    public static let createNode = "/api/v1/createNode"
```

```
    "date_of_birth" : 0,
    "relation_id" : 0
  },
  {
    "relation_type" : "test",
    "social_security_number" : "1291310219",
    "name" : "Partho",
    "gender" : "Male",
    "date_of_birth" : 0,
    "relation_id" : 0
  }
]
```

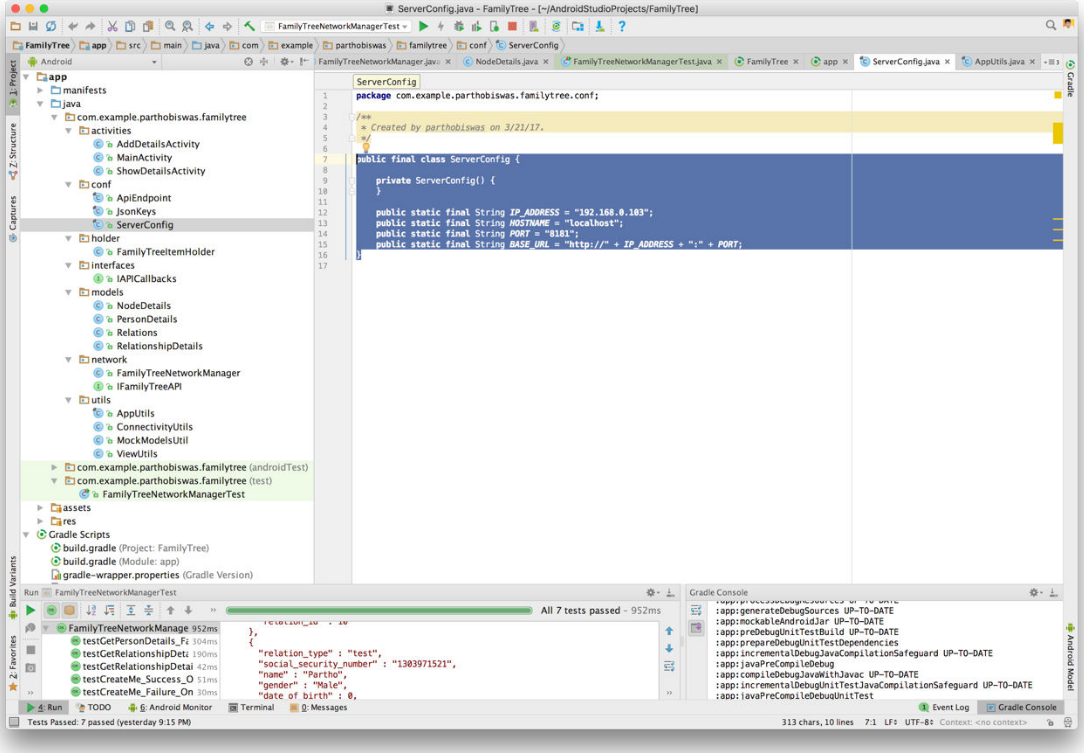
How just hit and run the app

3. ANDROID APP

I have used Java for the writing Android App. There are the following prerequisites and tools and it's version which I have used.

- Android Studio 2.3 to upper.
- Gradle 2.3 to upper.
- macOS 10.12 to upper
- Java 8
- Minimum SDK version 21
- Target and Compile SDK version 25

There is not much things to setup to run Android app. Just open the project with android studio. It will resolve the dependencies automatically and prompt you to download necessary tools. Once finished, now need to set Server IP. Copy and pest that (the one, user to start the HTTP server while setting up the server and for iOS app) IPv4 address into **ServerConfig.java >> public static final String IP_ADDRESS** field.



ServerConfig.java - FamilyTree - [~/AndroidStudioProjects/FamilyTree]

```
package com.example.parthobiswas.familytree.conf;

/**
 * Created by parthobiswas on 3/21/17.
 */

public final class ServerConfig {

    private ServerConfig() {
    }

    public static final String IP_ADDRESS = "192.168.0.103";
    public static final String HOSTNAME = "localhost";
    public static final String PORT = "8181";
    public static final String BASE_URL = "http://" + IP_ADDRESS + ":" + PORT;
}
```

Run: FamilyTreeNetworkManagerTest — All 7 tests passed - 952ms

```
FamilyTreeNetworkManage 952ms
  testGetPersonDetails_Fa 304ms
  testGetRelationshipDeta 190ms
  testGetRelationshipDetai 42ms
  testCreateMe_Success_O 51ms
  testCreateMe_Failure_On 30ms
```

```
  },
  {
    "relation_type" : "test",
    "social_security_number" : "1303971521",
    "name" : "Partho",
    "gender" : "Male",
    "date of birth" : 0,
```

Gradle Console

```
:app:generateDebugSources UP-TO-DATE
:app:mockableAndroidJar UP-TO-DATE
:app:preDebugUnitTestBuild UP-TO-DATE
:app:prepareDebugUnitTestDependencies
:app:incrementalDebugJavaCompilationSafeguard UP-TO-DATE
:app:javaPreCompileDebug
:app:compileDebugJavaWithJavac UP-TO-DATE
:app:incrementalDebugUnitTestJavaCompilationSafeguard UP-TO-DATE
:app:javaPreCompileDebugUnitTest
```

Tests Passed: 7 passed (yesterday 9:15 PM)

How just hit and run the app on your connected device.