# Мобилно приложение за регистрация на потребител в база, регистрацията включва освен основни данни на потребителя, изображения и други файлове.

1. Изгледът на приложението на мобилното устройство
   1. Цветовете на клиентското приложение
   2. Иконите и исползваните ресурси за UI на цлиентското приложение
   3. Скитирането и изгледът на готов UI
2. Краткото въведение
   1. Средата за разработването
      1. Средата за разработката, Android Studio IDE
      2. GIT – Version control
      3. Работата с бази дани, PHPMYADMIN
   2. API, REST, JSON
3. Инсталирането и конфигурирането на сървъра
   1. Инсталирането на Apache HTTP Server
   2. Инсталирането на MYSQL
   3. Инсталирането на PHP и PHPMYADMIN
4. Разработването на REST API
   1. Създаването на MYSQL база и таблици
   2. Създаването на PHP проект
      1. Структурата на проекта
      2. Конфигурацията на API (Config.php)
      3. DB\_Connect class – source code
      4. DB\_Functions class – source code
      5. Registration endpoint – source code
      6. Начинът на комуникацията с API. Типове на JSON отговори.
   3. Начинън за тестирането.

# Цветовете на клиентското приложение

За да е модерно нашето приложение, ще е добре да започнем от избора на хубавите цветове. За тази цел няма да е коректно да изберем случайни цветове, ще е добре да използваме дизайнерските цветове, подходящи един спрямо друг. Първо трябва да прегледаме цветовия кръг (color wheel - <https://en.wikipedia.org/wiki/Color_wheel>), който представя цветови преходи, моделите на HVS (*Hue, Saturation, Value*). С други думи, това ни представя правило, което ще ни помогни да се ориентираме в пространството на цветовете, за да можем да създъдем нюанси на различните цветови модели.



Цветови кръг на [Иохан Иттен](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D1%82%D0%B5%D0%BD,_%D0%98%D0%BE%D1%85%D0%B0%D0%BD%D0%BD%D0%B5%D1%81) (1961)

Решението, върху което се спряхме, е възползвaнето на безплатения сервиз **paletton.com**, благодорение на което, имаме цветове от зададена цветова гама:

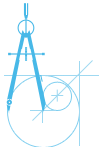


# Иконите и исползваните ресурси за UI на клиентското приложение

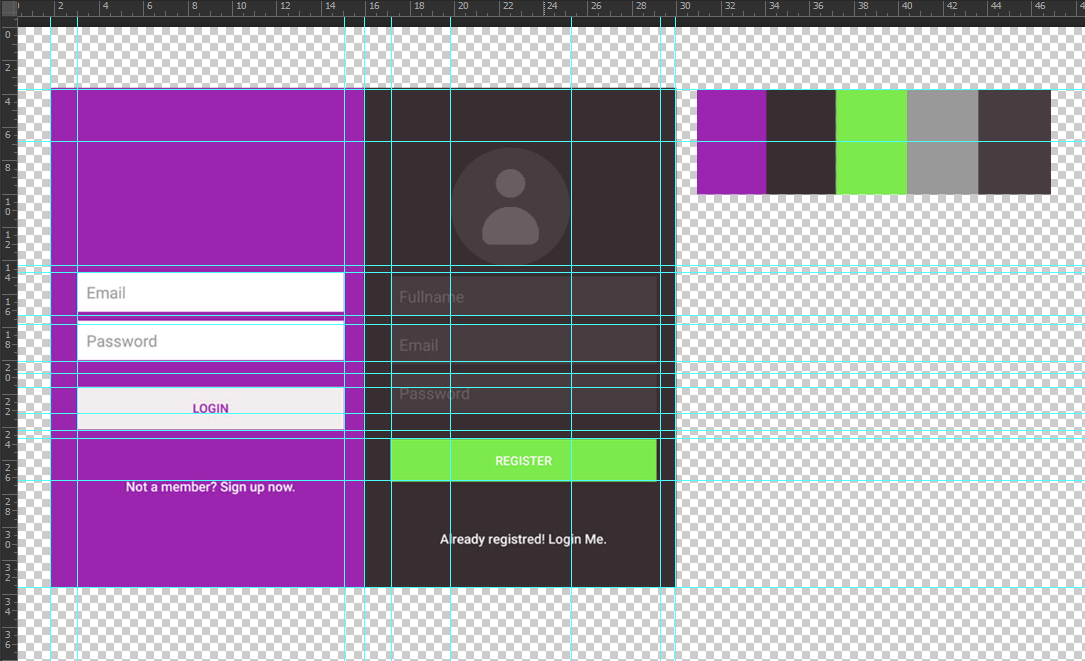


За разработването на графичения интерфейс често се използват различните икони, картинки и други ресурси, благодорение на които, управляването на приложението става по-лесно, по-удобно и изглежда много по-приятно. За да спестим времето, пак ще се възползваме от безплатения сервиз **iconfinder.com,** където ще потърсим необходимите ни файли с икони.

# Скитирането и изгледът на готов UI

Разработването на дизайна включва в себе си рисуването на графичния интерфейс, с помощта на избраните цветове, подходящите изображенията, размерите и типовете на шривтовете и др. В същност, нашия интерфейс ще е с минимални ресурси, ще е лесен за употреба, интуитивно разбераем и няма да е сложен за реализация. То ест, няма да се използват всички възможни анимации и сложните елементи. Предполага се, че ще се реализира **минималистичният дизайн** с приятни за човешко око цветове и икони. Като рисуването ще се извърши в някакъв графичен редактор, примерно в **Adobe Photoshop**, ще се модулира предпологаемият краен резултат за всяка страница, лаяут (екран), която после ще се превърне в код.

Примерните скици:



# Средата за разработката - Android Studio IDE

Всеки един програмист се нуждае от комфортното разработване, което предполага – всичкото да е под ръка. Сега малко ще поговорим за основните неща, които ще ни помагат в разработването на дадения проект.

Средата за разработката, IDE (Integrated development environment) – софтуерното приложение, което предоставя цялостна среда на програмистите за разработване на софтуер. Средата обикновено състои от редактора на кода, компилатора и/или интерпретатора, линкера, дебъгера и други полезни неща. Android studio – това е средата за разработка на android платформа, анонсирана от Google. По подробно може да намерим тук: **wikipedia.org/wiki/Android\_Studio.**

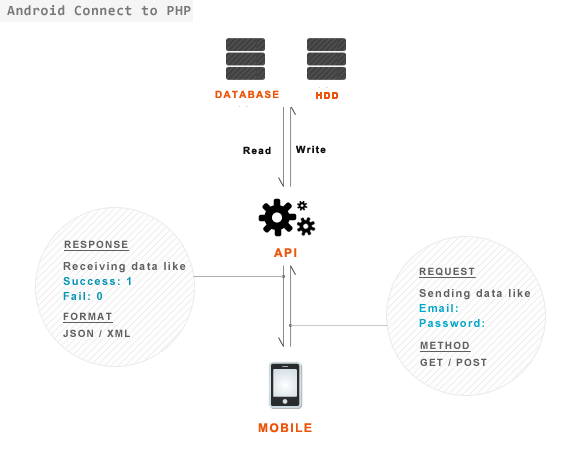
GIT - децентрализирана система за контрол на версиите на файлове. Създадена е от Линус Торвалдс за управление на разработката на Linux. Ще е полезна за работа като ще се запазват основните стъпки в разработване, като ни дава възможност да си върнем в състоянието на проекта, без да е нужно да се правят переодични бекапове. Също така, полезна е, защото ще е доста удобно за работа в екип, без да има нужда за чести срещи, като всеки ще си върши своята част на проекта отдалечено. Това ще стане възможно като си направим така нареченото дистанционно репо на **GITHUB**.

Oще една функция, която ще е доста полезна за разработването и проектирането на бази данни - PHPMYADMIN – уеб приложение с отворен код. То предоставя възможност на уеб интерфейса за администриране на MYSQL бази данни. С помощта на него може да се стартират SQL команди, да се преглеждат таблици и бази данни, без големи познания на SQL. По подробно: **wikipedia.org/wiki/PhpMyAdmin**.

# API - Application Programming Interface

За да се извърши взаимодействието на клиента с MYSQL база данни, ние ще изградим лесното API, малкото приложение, написаното на PHP. В същност, ще направим простото REST API, което в крайна сметка ще взаимодейства с клиента. По-долу показваме какво ще прави нашият интерфейс.

* Приема запроси от тип GET и POST
* Взаимодейства с MYSQL база дани, вмъква/извлича дани
* Взаимодейства с файловата система на сървъра. Качва в сървера файловете на клиента.
* Връща отговорите във формат JSON



Услугата REST (representation state transfer) представя разделителната системна рамка, базирана на уеб-протоколи и технологии (в нашия случай HTTP). Архитектурният модел REST включва взаимодействията между клиента и сървъра, осъществени по време на трансфера на данните.

JSON (JavaScript Object Notation) – текстово базираният отворен стандарт, създаден за човешки четим обмен данни. Подробности: **wikipedia.org/wiki/JSON.**

# Инсталирането на Apache HTTP Server

Apache HTTP Server – проект, разработан и поддържащ се от The Apache Software Foundation, в рамките на който се разработва кросплатформен HTTP-сървър с отворения исходения код. Влиза в пакет LAMP и XAMP.   
  
Инсталирането в убунту:

**sudo** **apt-get** **install** apache2

Обикновено настройките му се пазят в /etc/apache2/conf-available -директория  
След промените, за да са достъпни настройките, е необходимо рестартирането на демон Apache.

**sudo** service apache2 restart

# Инсталирането на MYSQL

MYSQL – свободна СУБД за малки и средни приложения. Влиза в състав на LAMP и XAMP.

В Ubuntu за да се инсталира е необходимо изпълняването на командата:

**sudo** **apt-get** **install** mysql-server

По време на инсталирането, скрипта за инсталация ще поиска въвеждане паролата за администратора (root) на базите данни.

Ако е необходим конзолем-клиеент, инсталираме го с помощта на следната команда:

**sudo** **apt-get** **install** mysql-client

За настройки се използва файл: /etc/mysql/my.cnf

# Инсталирането на PHP и PHPMYADMIN

За инсталирането на PHP в Ubuntu извършваме:

**sudo** **apt-get** **install** php5-mysql

Обаче, преди инсталирането на PHPMYADMIN, ще ни е нужен достъп към базата, която я инсталирахме преди малко и самото инсталиране се извършва чрез командата:

sudo apt-get install phpmyadmin

# Създаването на MYSQL база и таблици

Отаваряме PHPMYADMIN и създаваме базата и таблиците. Тази стъпка е задължителна, заради това, че нашето API ще работи с данни от тази база. За да извършим задачата, е достатъчнп една таблица **users** с полета, съдържащи информацията за потребителя.

create database rest\_api /\*\* Creating Database \*\*/

use android\_api /\*\* Selecting Database \*\*/

create table users(

   id int(11) primary key auto\_increment,

   unique\_id varchar(23) not null unique,

   name varchar(50) not null,

   email varchar(100) not null unique,

   encrypted\_password varchar(80) not null,

   salt varchar(10) not null,

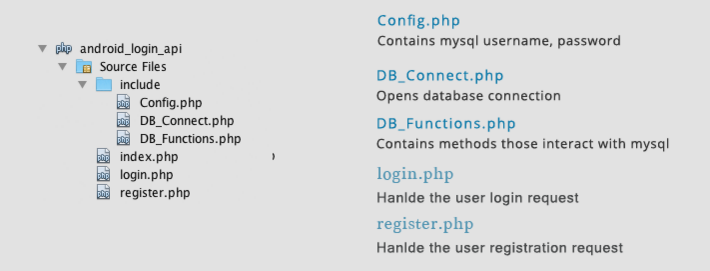
   created\_at datetime,

   updated\_at datetime null

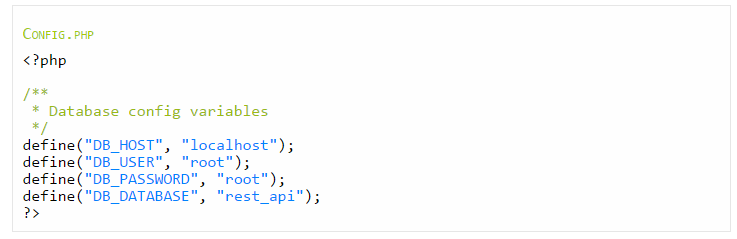
); /\*\* Creating Users Table \*\*/

# Създаването на PHP проект

## Структурата на проекта



1. В **/var/www/** директорията създаваме папка **android\_login\_api**. Тук ще е главната папка на проекта.
2. В **andoid\_login\_api** създаваме include папка. В нея ще слагаме всички вспомогателни класове.
3. Сега в **include** създаваме php-файл, който ще се казва **Config.php**, в него ще имаме информацията, за авторизация в MYSQL-база.



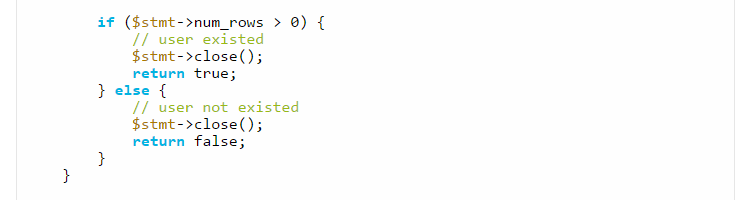
1. Създаваме **DB\_Connect.php** в **include**.

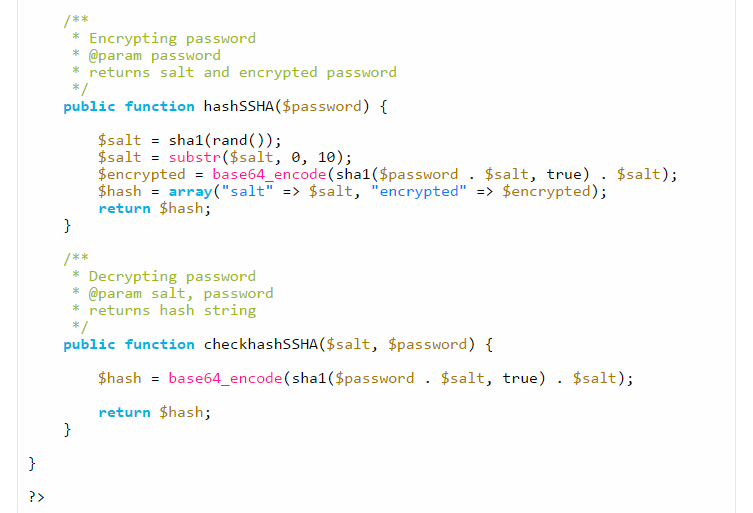


1. Съзздаваме **DB\_Functions.php** в **include** папката. Тук ще се съдържат функциите за работата с базата. Пример: isUserExist($email) ще проверява дали съществува потребител по заден email.



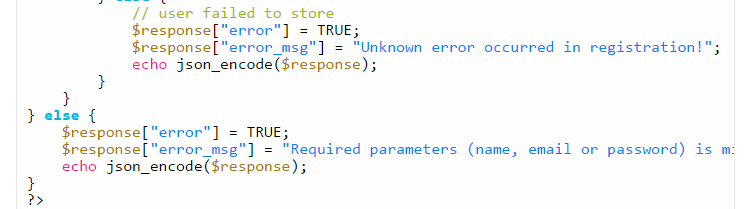






1. Регистрацията. Сега имаме всичко необходимо и завършващият етап - е създаването на начина на работата на API. Модулът за регистрацията ще получава няколко параметра, POST параметри от клиента и ще ги записва в базата.





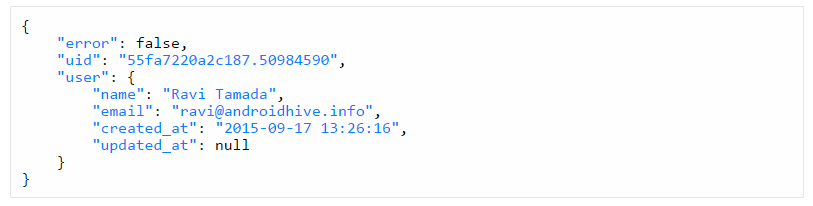
## Типове JSON отговори

Това API връща няколко вида отговори за регистрацията.

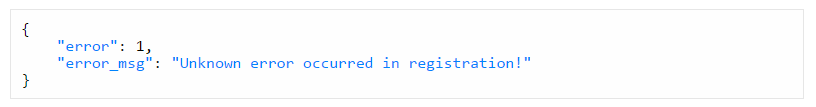
**URL**: <http://localhost/android_login_api/register.php>

**PARAMS**: name, email, password

Регистрацията мина успешно



Възникнаха грешки по време на записването в базата



Потребителят вече е регистриран



# Тестиране.

Би било добре да се тестира маправеното API. За тази цел има доста решения. Според мене, ще е удобно да се използва **CURL** и/или **Advanced Rest Client.**

**CURL** - представлява програма за командения ред, софтуерната библиотека (libcurl), която предлага възможността за прехвърлянето на данни през множество протоколи като FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS и FILE. Curl поддържа SSL-сертификатите, HTTP POST, HTTP PUT, FTP качване, качване на файлове с помощта на HTTP формите, прокситата, бисквитките, удостоверяването на самоличността (Basic, Digest, NTLM, Negotiate, kerberos...), подновяването на файловите трансфери, прокси тунелирането и много други функции.

Advanced rest Client – бесплатното допълнение (extension) на различни браузери. По подробно за него: [**http://wiki.scn.sap.com/wiki/display/SAPMOB/Using+Chrome's+Advanced+Rest+Client+to+test+SMP+3.0**](http://wiki.scn.sap.com/wiki/display/SAPMOB/Using+Chrome's+Advanced+Rest+Client+to+test+SMP+3.0)

**Ползвам Advanced Rest Client Extension защото той съвмещава различни платформи и възможности за работа чрез графичения интерфейс.**

други функции:

<http://www.androidhive.info/2012/01/android-login-and-registration-with-php-mysql-and-sqlite/>