NedoChat

AUTHOR
Версия 1.0.0
Пн 17 Дек 2018

Иерархический список классов

Иерархия классов

Иерархия классов.

Алфавитный указатель классов

Классы

Классы с их кратким описанием.

Пакет com.example.firebasechat

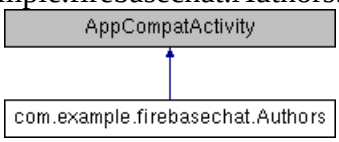
Классы

- class Authors
  Класс для отображения страницы с информацией о создателях
- class ChatRoom
  Данное Activity представляет собой комнату обмена зашифрованными сообщениями
- class MainActivity
  Activity регистрации и авторизации пользователей
- class Message
  Класс сообщений пользователей
- class SplashActivity
  Заставка приложения
- class User
  Класс пользователя
- class ViewHolder
  Класс holder отображения сообщений в RecyclerView.

Классы

Класс com.example.firebasechat.Authors

Класс для отображения страницы с информацией о создателях
Граф наследования:com.example.firebasechat.Authors:



Защищенные члены

- void onCreate (Bundle savedInstanceState)
  Создание activity Authors.

Подробное описание

Класс для отображения страницы с информацией о создателях

См. определение в файле Authors.java строка 13

Методы

void com.example.firebasechat.Authors.onCreate (Bundle *savedInstanceState*) `[protected]`

Создание activity Authors.

Аргументы:

savedInstanceState	

См. определение в файле Authors.java строка 22

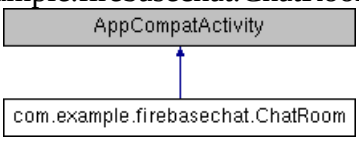
Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/Authors.java

Класс com.example.firebasechat.ChatRoom

Данное Activity представляет собой комнату обмена зашифрованными сообщениями

Граф наследования:com.example.firebasechat.ChatRoom:



Открытые члены

- void displayChat ()
  Функция отображения сообщений
- void get_keys ()
  Данная функция предназначена для получения ключей
- void Admin_actrivity ()
  Данная функция предназначена для обработки публичных ключей администатором
- boolean onCreateOptionsMenu (Menu menu)
  Функция создания кнопки выхода
- boolean onOptionsItemSelected (MenuItem item)
  Обработка нажатия на кнопку выхода

Открытые атрибуты

- FirebaseListAdapter< Message > adapter
  Адаптер для отображения сообщений с сервера
- DatabaseReference Admen = FirebaseDatabase.getInstance().getReference("Admen")
  База данных с обработанными Администратором публичными ключами пользователей
- DatabaseReference messages = FirebaseDatabase.getInstance().getReference("messages")
  База данных с сообщениям пользователей
- DatabaseReference pkeys = FirebaseDatabase.getInstance().getReference("Public_Key")
  Базад данных публичных ключей

Статические открытые данные

- static int MAX_MESSAGE_LENGTH = 300
  Максимальная длина сообщения

Защищенные члены

- void onCreate (Bundle savedInstanceState)

Подробное описание

Данное Activity представляет собой комнату обмена зашифрованными сообщениями

См. определение в файле ChatRoom.java строка 55

Методы

void com.example.firebasechat.ChatRoom.Admin_actrivity ()

Данная функция предназначена для обработки публичных ключей администатором

Слушатель данных из базы данных

Аргументы:

dataSnapshot	снимок обекта из базы данных

См. определение в файле ChatRoom.java строка 236

void com.example.firebasechat.ChatRoom.displayChat ()

Функция отображения сообщений

Данная функция обращается к базе данных сообщений пользователей и, используя специальный адаптер, отображает их

См. определение в файле ChatRoom.java строка 141

void com.example.firebasechat.ChatRoom.get_keys ()

Данная функция предназначена для получения ключей

Контроль добавления данных в базу данных

Аргументы:

dataSnapshot	снимок обекта из базы данных

См. определение в файле ChatRoom.java строка 182

void com.example.firebasechat.ChatRoom.onCreate (Bundle *savedInstanceState*) `[protected]`

Создание activity ChatRoom

Аргументы:

savedInstanceState	

См. определение в файле ChatRoom.java строка 80

boolean com.example.firebasechat.ChatRoom.onCreateOptionsMenu (Menu *menu*)

Функция создания кнопки выхода

Аргументы:

menu	

Возвращает:

См. определение в файле ChatRoom.java строка 286

boolean com.example.firebasechat.ChatRoom.onOptionsItemSelected (MenuItem *item*)

Обработка нажатия на кнопку выхода

Аргументы:

item	

Возвращает:

См. определение в файле ChatRoom.java строка 297

Данные класса

FirebaseListAdapter<Message> com.example.firebasechat.ChatRoom.adapter

Адаптер для отображения сообщений с сервера

См. определение в файле ChatRoom.java строка 57

DatabaseReference com.example.firebasechat.ChatRoom.Admen = FirebaseDatabase.getInstance().getReference("Admen")

База данных с обработанными Администратором публичными ключами пользователей

См. определение в файле ChatRoom.java строка 63

int com.example.firebasechat.ChatRoom.MAX_MESSAGE_LENGTH = 300 `[static]`

Максимальная длина сообщения

См. определение в файле ChatRoom.java строка 56

DatabaseReference com.example.firebasechat.ChatRoom.messages = FirebaseDatabase.getInstance().getReference("messages")

База данных с сообщениям пользователей

См. определение в файле ChatRoom.java строка 64

DatabaseReference com.example.firebasechat.ChatRoom.pkeys = FirebaseDatabase.getInstance().getReference("Public_Key")

Базад данных публичных ключей

См. определение в файле ChatRoom.java строка 65

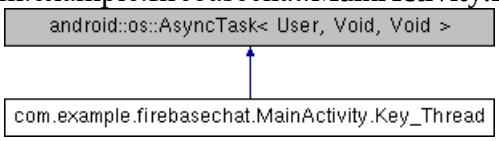
Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/ChatRoom.java

Класс com.example.firebasechat.MainActivity.Key_Thread

Поток для обработки генерации ключей

Граф наследования:com.example.firebasechat.MainActivity.Key_Thread:



Защищенные члены

- void onPreExecute ()
- Void doInBackground (User ... current_user)
- void onPostExecute (Void result)

Подробное описание

Поток для обработки генерации ключей

См. определение в файле MainActivity.java строка 272

Методы

Void com.example.firebasechat.MainActivity.Key_Thread.doInBackground (User ... *current_user*) `[protected]`

См. определение в файле MainActivity.java строка 284

void com.example.firebasechat.MainActivity.Key_Thread.onPostExecute (Void *result*) `[protected]`

См. определение в файле MainActivity.java строка 303

void com.example.firebasechat.MainActivity.Key_Thread.onPreExecute () `[protected]`

См. определение в файле MainActivity.java строка 275

Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/MainActivity.java

Класс com.example.firebasechat.MainActivity

Activity регистрации и авторизации пользователей

Граф наследования:com.example.firebasechat.MainActivity:

Классы

- class Key_Thread
  Поток для обработки генерации ключей

Открытые члены

- void onClick (View view)
- void signin (String email, final String password_)
  Функция, реализующая авторизацию пользователя по паролю и почте
- void registration (String email, String password)
  Функция, реализующая регистрацию пользователя по паролю и почте
- void keys_generation ()
  Переотправка ключей для пользователей, у которых нет их на данном устройстве
- void admin_auth (String password)
  Авторизация для Администратора
- void user_auth ()
  Авторизация для пользователей

Открытые атрибуты

- DatabaseReference pkeys = FirebaseDatabase.getInstance().getReference("Public_Key")
  Базад данных публичных ключей

Статические открытые данные

- static User current_user
  Пользователь

Защищенные члены

- void onCreate (Bundle savedInstanceState)
  Создание activity MainActivity.

Подробное описание

Activity регистрации и авторизации пользователей

Данное Activity предназначено для регистрации или авторизации пользователя, также содержит кнопку перехода на страницу информации об авторах

См. определение в файле MainActivity.java строка 47

Методы

void com.example.firebasechat.MainActivity.admin_auth (String *password*)

Авторизация для Администратора

Аргументы:

password	

См. определение в файле MainActivity.java строка 196

void com.example.firebasechat.MainActivity.keys_generation ()

Переотправка ключей для пользователей, у которых нет их на данном устройстве

См. определение в файле MainActivity.java строка 179

void com.example.firebasechat.MainActivity.onClick (View *view*)

См. определение в файле MainActivity.java строка 104

void com.example.firebasechat.MainActivity.onCreate (Bundle *savedInstanceState*) `[protected]`

Создание activity MainActivity.

Аргументы:

savedInstanceState	

См. определение в файле MainActivity.java строка 78

void com.example.firebasechat.MainActivity.registration (String *email*, String *password*)

Функция, реализующая регистрацию пользователя по паролю и почте

При успешной регистрации для пользователя генерируются ключи, которые после генерации отправляются на сервер

Аргументы:

email	Данный, который пользователь ввел в поле почты
password	Данный, который пользователь ввел в поле пароля

См. определение в файле MainActivity.java строка 162

void com.example.firebasechat.MainActivity.signin (String *email*, final String *password_*)

Функция, реализующая авторизацию пользователя по паролю и почте

Аргументы:

email	Данный, который пользователь ввел в поле почты
password	Данный, который пользователь ввел в поле пароля

См. определение в файле MainActivity.java строка 137

void com.example.firebasechat.MainActivity.user_auth ()

Авторизация для пользователей

См. определение в файле MainActivity.java строка 242

Данные класса

User com.example.firebasechat.MainActivity.current_user `[static]`

Пользователь

См. определение в файле MainActivity.java строка 67

DatabaseReference com.example.firebasechat.MainActivity.pkeys = FirebaseDatabase.getInstance().getReference("Public_Key")

Базад данных публичных ключей

См. определение в файле MainActivity.java строка 69

Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/MainActivity.java

Класс com.example.firebasechat.Message

Класс сообщений пользователей

Открытые члены

- Message (String textMessage, String author)
  Конструктор класса Message.
- Message ()
- String getTextMessage ()
- void setTextMessage (String textMessage)
  Setter для текста сообщения
- String getAuthor ()
  Getter для автора
- void setAuthor (String author)
  Setter для авора
- String getTimeMessageString ()
  Getter для времени сообщения в формате String.
- long getTimeMessage ()
  Getter для времени сообщения
- void setTimeMessage (long timeMessage)
  Setter для времени сообщения

Открытые атрибуты

- String textMessage
  Текст сообщения
- String author
  Автор сообщения
- long timeMessage
  Время отправки сообщения

Подробное описание

Класс сообщений пользователей

См. определение в файле Message.java строка 7

Конструктор(ы)

com.example.firebasechat.Message.Message (String *textMessage*, String *author*)

Конструктор класса Message.

Аргументы:

textMessage	Текст сообщения
author	Автор сообщения

См. определение в файле Message.java строка 18

com.example.firebasechat.Message.Message ()

См. определение в файле Message.java строка 24

Методы

String com.example.firebasechat.Message.getAuthor ()

Getter для автора

Возвращает:
Автор
См. определение в файле Message.java строка 47

String com.example.firebasechat.Message.getTextMessage ()

Getter для текста сообщения

Возвращает:
Текст сообщения
См. определение в файле Message.java строка 31

long com.example.firebasechat.Message.getTimeMessage ()

Getter для времени сообщения

Возвращает:
Время сообщения
См. определение в файле Message.java строка 71

String com.example.firebasechat.Message.getTimeMessageString ()

Getter для времени сообщения в формате String.

Возвращает:
Время сообщения
См. определение в файле Message.java строка 63

void com.example.firebasechat.Message.setAuthor (String *author*)

Setter для авора

Аргументы:

author	Автор

См. определение в файле Message.java строка 55

void com.example.firebasechat.Message.setTextMessage (String *textMessage*)

Setter для текста сообщения

Аргументы:

textMessage	Текст сообщения

См. определение в файле Message.java строка 39

void com.example.firebasechat.Message.setTimeMessage (long *timeMessage*)

Setter для времени сообщения

Аргументы:

timeMessage	время сообщения

См. определение в файле Message.java строка 77

Данные класса

String com.example.firebasechat.Message.author

Автор сообщения

См. определение в файле Message.java строка 10

String com.example.firebasechat.Message.textMessage

Текст сообщения

См. определение в файле Message.java строка 9

long com.example.firebasechat.Message.timeMessage

Время отправки сообщения

См. определение в файле Message.java строка 11

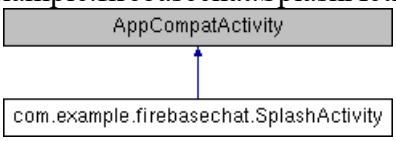
Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/Message.java

Класс com.example.firebasechat.SplashActivity

Заставка приложения

Граф наследования:com.example.firebasechat.SplashActivity:



Защищенные члены

- void onCreate (Bundle savedInstanceState)
  Создание activity SplashActivity.

Подробное описание

Заставка приложения

См. определение в файле SplashActivity.java строка 10

Методы

void com.example.firebasechat.SplashActivity.onCreate (Bundle *savedInstanceState*) [protected]

Создание activity SplashActivity.

Аргументы:

savedInstanceState	

См. определение в файле SplashActivity.java строка 17

Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/SplashActivity.java

Класс com.example.firebasechat.User

Класс пользователя

Открытые члены

- User ()
  Конструктор класса User.
- void setSessionPair_ (String[] sessionPair_)
  Установка сессионных ключей
- String getUserPubKeyEncStr ()
  Getter для публичного ключа
- String getUserPrivateKeyEncStr ()
  Getter для приватного ключа
- void setUserPubKeyEncStr (String userPubKeyEncStr)
  Setter для публичного ключа
- void setUserPrivateKeyEncStr (String userPrivateKeyEncStr)
  Setter для приватного ключа
- void EncryptSession (String[] result)
  Функкция обрабоки пользователем данных, полученных от администратора Пользователь получает данные администратора, генерирует общий секрет и получаает сессионные ключи
- String [] DHGenerateAdmin (String userPubKeyEncStr)
  Функция Администратора, где обрабатывается публичный клч пользователя и создаются необходимые ключи
- String encrypt (String value)
  Функция шифрования сообщения
- String decrypt (String encrypted)
  Функция расшифровывает сообщения с сервера

Открытые атрибуты

- String userPubKeyEncStr
- String userPrivateKeyEncStr
- KeyPair userKpair

Статические открытые данные

- static String [] sessionPair_ = new String[2]
  Сессионная пара ключей пользователя

Подробное описание

Класс пользователя

Класс пользоваателя выполняет основные функции, необходимые для обмена ключами между пользователем и администратором, а также функции шифрования и расшифрования сообщения Со стороны пользователя генерируется пара ключей, а после обрабатываются данные, полученные от администратора Со стороны администратора обрабатываются данные, полученные пользователем

См. определение в файле User.java строка 22

Конструктор(ы)

com.example.firebasechat.User.User ()

Конструктор класса User.

См. определение в файле User.java строка 50

Методы

String com.example.firebasechat.User.decrypt (String *encrypted*)

Функция расшифровывает сообщения с сервера

Аргументы:

encrypted	Зашифрованное сообщение

Возвращает:

Расшифрованное сообщение

См. определение в файле User.java строка 249

String [] com.example.firebasechat.User.DHGenerateAdmin (String *userPubKeyEncStr*)

Функция Администратора, где обрабатывается публичный клч пользователя и создаются необходимые ключи

Аргументы:

userPubKeyEncStr	Публичный ключ пользователя

Возвращает:

Результат обрабоки данных пользователя

См. определение в файле User.java строка 172

String com.example.firebasechat.User.encrypt (String *value*)

Функция шифрования сообщения

Аргументы:

value	Сообщение

Возвращает:

Зашифрованное сообщение

См. определение в файле User.java строка 228

void com.example.firebasechat.User.EncryptSession (String [] *result*)

Функкция обрабоки пользователем данных, полученных от администратора Пользователь получает данные администратора, генерирует общий секрет и получаает сессионные ключи

Аргументы:

result	Данные, который пользователь получает от администратора

См. определение в файле User.java строка 125

String com.example.firebasechat.User.getUserPrivateKeyEncStr ()

Getter для приватного ключа

Возвращает:

Приватный ключ

См. определение в файле User.java строка 100

String com.example.firebasechat.User.getUserPubKeyEncStr ()

Getter для публичного ключа

Возвращает:

Публичный ключ

См. определение в файле User.java строка 92

void com.example.firebasechat.User.setSessionPair_ (String [] *sessionPair_*)

Установка сессионных ключей

Аргументы:

*sessionPair_*	сессионная пара

См. определение в файле User.java строка 84

void com.example.firebasechat.User.setUserPrivateKeyEncStr (String *userPrivateKeyEncStr*)

Setter для приватного ключа

Аргументы:

userPrivateKeyEncStr	Приватный ключ

См. определение в файле User.java строка 116

void com.example.firebasechat.User.setUserPubKeyEncStr (String *userPubKeyEncStr*)

Setter для публичного ключа

Аргументы:

userPubKeyEncStr	Публичный ключ

См. определение в файле User.java строка 108

Данные класса

String [] com.example.firebasechat.User.sessionPair_ = new String[2] `[static]`

Сессионная пара ключей пользователя

См. определение в файле User.java строка 45

KeyPair com.example.firebasechat.User.userKpair

См. определение в файле User.java строка 39

String com.example.firebasechat.User.userPrivateKeyEncStr

См. определение в файле User.java строка 33

String com.example.firebasechat.User.userPubKeyEncStr

См. определение в файле User.java строка 27

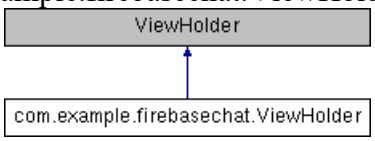
Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/User.java

Класс com.example.firebasechat.ViewHolder

Класс holder отображения сообщений в RecyclerView.

Граф наследования:com.example.firebasechat.ViewHolder:



Открытые члены

- ViewHolder (View ItemView)

Подробное описание

Класс holder отображения сообщений в RecyclerView.

См. определение в файле ViewHolder.java строка 10

Конструктор(ы)

com.example.firebasechat.ViewHolder.ViewHolder (View *ItemView*)

См. определение в файле ViewHolder.java строка 14

Объявления и описания членов класса находятся в файле:

- C:/Games/BackUp/FireBasechat/app/src/main/java/com/example/firebasechat/ViewHolder.j