



Бесплатная электронная книга

УЧУСЬ

# firebase-authentication

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#firebase-  
authenticati

on

.....	1
<b>1: firebase</b> .....	<b>2</b>
.....	2
Examples.....	2
.....	2
.....	<b>2</b>
<b>Firestore v3</b> .....	<b>4</b>
.....	4
<b>2: Google Plus Android</b> .....	<b>6</b>
Examples.....	6
Google Plus.....	6
.....	<b>10</b>

---

# Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [firebase-authentication](#)

It is an unofficial and free firebase-authentication ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official firebase-authentication.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# глава 1: Начало работы с аутентификацией firebase

## замечания

В этом разделе приведен обзор того, какая проверка подлинности firebase и почему разработчик может захотеть ее использовать.

Следует также упомянуть о любых крупных объектах в рамках проверки подлинности firebase и ссылки на связанные темы. Поскольку документация для проверки подлинности firebase является новой, вам может потребоваться создать начальные версии этих связанных тем.

## Examples

### Установка или настройка

---

## КОНСПЕКТ

Полнофункциональная демо Firebase v3 Web аутентификации видимую [здесь](#) . Войдите в Facebook, Github, Google, Twitter, пароль и анонимные аккаунты. Код, [доступный в Github](#) , легко читается и соблюдается и хорошо документирован. Основное внимание уделяется полностью функциональной системе аутентификации.

Пользователям на основе паролей отправляется ссылка для проверки. Они также могут изменять свой адрес электронной почты и пароль - оба этих события отправляют электронное письмо с подтверждением в качестве дополнительной меры безопасности.

Наконец, продемонстрирована разница между аутентификацией, авторизацией на стороне клиента и авторизацией на стороне сервера, обеспеченной с помощью правил безопасности базы данных Firebase Realtime.

### 1. Предпосылки

1. Веб-проект Firebase. *СВОБОДНО!*
2. IDE. Что такое IDE? Попробуйте [Cloud9](#) . *СВОБОДНО!*
3. Аккаунт Github, Google, Facebook и Twitter. *СВОБОДНО!*
4. Две учетные записи электронной почты. *СВОБОДНО!*

### 2. Настройка среды IDE

2. Создайте проект HTML5.
3. Установите Firebase Tools. `npm install -g firebase-tools`
4. Используя командную строку Firebase Tools, войдите в свой проект Firebase. `firebase login --no-localhost`
5. Используя командную строку Firebase Tools, настройте проект Firebase в текущем каталоге. `firebase init`
6. Клонирование этого набора файлов и папок в вашу среду IDE. `git clone https://github.com/rhroyston/firebase-auth.git`
7. Используя командную строку Firebase Tools, нажмите проект IDE в проект Firebase. `firebase deploy`
8. Просмотрите проект Firebase в своем браузере. Какие-либо сломанные изображения или ошибки? Легкое исправление ниже.
9. Возможно, вам потребуется обновить `href`, `src` и `background: url` во всех JS, CSS и всех HTML-файлах в зависимости от структуры вашей веб-хостинга.
  1. Используйте функцию поиска, чтобы искать как `href` и `src` и обновлять по мере необходимости.
  2. Браузерная консоль отобразит все оставшиеся ошибки неправильных путей.
  3. Примечание. Script.js строка 781 `privateLink.href = "../firebase-auth/private" ..` кажется, требуется.
  4. После того, как все страницы правильно отображаются из Firebase Hosting (без разбитых изображений или ошибок консоли), продолжайте.

### 3. Настройка Firebase

1. Включить все 6 форм аутентификации. Следуйте инструкциям по настройке параметров сайта в социальных сетях.
2. Настройте URL-адрес обработчика сообщений электронной почты, чтобы указать URL-адрес вашего веб-приложения Firebase + '/ ack', например `https://my-app-1234/ack .`

### 4. Вход в веб-приложение

1. Войдите в систему с помощью поставщика OAuth.
2. В командной строке браузера используйте открытый `demo.update('mynode', 'myKey', 'myValue')` чтобы добавить безопасную разметку в вашу базу данных реального времени.
  1. Сообщение об успехе появится в консоли вашего браузера.
  2. Возможно, вам потребуется обновить путь `href` чтобы он соответствовал вашей структуре папок.

```
demo.update("markup", "secureData", "<div class=\"mdl-card__title\"> <h1 class=\"mdl-card__title-text mdl-color-text--white\">Secured Data</h1> </div><div class=\"mdl-card__supporting-text mdl-typography--headline\"> <p>This is a secure card. The HTML markup that renders this card is secured in the Realtime Database. Access is determined server side so no matter what you do with JavaScript on your browser you will not be able
```

```
to view this card unless you are authorized to.</p><p>Secured data can be markup, JSON, strings, numbers, etc. Your imagination is the limit!</p></div><div class=\"mdl-card__actions mdl-card--border intro-card-actions\"> <a class=\"mdl-button mdl-button--colored mdl-js-button mdl-js-ripple-effect\" href=\"../firebase-auth/\">Home</a></div>");
```

# Firestore v3 Прохождение аутентификации и авторизации

## 1. Войти с использованием каждого поставщика OAuth

1. Обратите внимание, что обновление настроек адреса электронной почты или пароля отсутствует на странице вашего аккаунта.
2. Обратите внимание на дополнительные ссылки в ящике бокового меню?
3. Попробуйте удалить свою учетную запись. Что происходит?

## 2. Зарегистрируйтесь как пользователь на основе пароля

1. Вы получили электронное письмо с подтверждением?
2. Можете ли вы просмотреть личные данные, пока не нажмете ссылку подтверждения?
3. Вы можете изменить свой пароль?
4. Можете ли вы изменить свой адрес электронной почты?
5. Можете ли вы отменить изменение электронной почты, щелкнув ссылку на уведомление об изменении электронной почты электронной почты?

## 3. Выйти

1. Какие ссылки присутствуют в ящике бокового меню?
2. Можете ли вы получить доступ к личным данным?
3. Вы можете просматривать личные данные?

## Как создать пользователя на основе пароля

Вы можете использовать Firebase Authentication, чтобы позволить вашим пользователям проходить аутентификацию с помощью Firebase, используя их адреса электронной почты и пароли, а также управлять учетными записями вашего приложения на основе пароля.

В этом примере мы используем эти шаги, чтобы настроить его для нашего Android-проекта, основанного на JavaScript.

Но прежде чем это сделать, это то, что нужно сделать раньше:

1. Добавьте Firebase в свой проект JavaScript.
2. Если вы еще не подключили свое приложение к проекту Firebase, сделайте это с

консоли Firebase.

3. Включение входа в Email / Password: в консоли Firebase откройте раздел Auth. На вкладке «Метод входа» включите метод входа в систему «Эл. Почта / пароль» и нажмите «Сохранить».

Для создания пользователя на основе пароля с помощью displayName требуется 2 метода auth, а именно .createUserWithEmailAndPassword и .updateProfile. Я вложил последнее и создал одну функцию, которая запускает оба этих метода для удобства использования.

```
function registerPasswordUser(email, displayName, password, photoURL) {
  var user = null;
  //NULLIFY EMPTY ARGUMENTS
  for (var i = 0; i < arguments.length; i++) {
    arguments[i] = arguments[i] ? arguments[i] : null;
  }
  auth.createUserWithEmailAndPassword(email, password)
  .then(function () {
    user = auth.currentUser;
    user.sendEmailVerification();
  })
  .then(function () {
    user.updateProfile({
      displayName: displayName,
      photoURL: photoURL
    });
  })
  .catch(function(error) {
    console.log(error.message, 7000);
  });
  console.log('Validation link was sent to ' + email + '.');
}
```

Прочитайте Начало работы с аутентификацией firebase онлайн:

<https://riptutorial.com/ru/firebase-authentication/topic/4744/начало-работы-с-аутентификацией-firebase>

---

# глава 2: Аутентификация Google Plus с Android

## Examples

### Аутентификация входа в систему Google Plus

#### Аутентификация пользователя с помощью входа Plus

##### OnCreate

```
GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestScopes(new Scope(Scopes.PLUS_LOGIN))

    .requestEmail()
    .build();
mGoogleApiClient = new GoogleApiClient.Builder(this)
    .enableAutoManage(this /* FragmentActivity */, this /* OnConnectionFailedListener */)
    .addConnectionCallbacks(this)
    .addOnConnectionFailedListener(this)
    .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
    .addApi(Plus.API)
    .build();
```

##### OnStart ()

```
protected void onStart() {
    super.onStart();
    mGoogleApiClient.connect();
    mAuth.addAuthStateListener(mAuthListener);
}

protected void onStop() {
    super.onStop();
    if (mAuthListener != null) {
        mAuth.removeAuthStateListener(mAuthListener);
    }
    if (mGoogleApiClient.isConnected()) {
        mGoogleApiClient.disconnect();
    }
}

@Override
public void onConnected(Bundle bundle) {
    mSignInClicked = false;
    getProfileInformation();
    //getGoogleOAuthTokenAndLogin();
    Toast.makeText(this, "User is connected! (in onConnected
MActivity)", Toast.LENGTH_LONG).show();
```



```

// Get user's information
// Update the UI after signin
// updateUI(true);
}

@Override
public void onConnectionSuspended(int i) {

    mGoogleApiClient.connect();
    //updateUI(false);
}

@Override
public void onConnectionFailed(ConnectionResult connectionResult) {

    if (!connectionResult.hasResolution()) {
        GooglePlayServicesUtil.getErrorDialog(connectionResult.getErrorCode(), this,
            0).show();
    } else if (connectionResult.hasResolution()) {
        mConnectionResult = connectionResult;
        resolveSignInError();
    }
}

private void resolveSignInError() {
    Log.e("pavan", "User RESOLVE SIGN IN ERROR CALLED OUT OF IF");
    if (mSignInClicked) {
        if (mConnectionResult.hasResolution()) {
            try {
                Log.e("pavan", "User RESOLVE SIGN IN ERROR CALLED OUT OF IF TRY");
                mIntentInProgress = true;
                mConnectionResult.startResolutionForResult(this, GOOGLE_SIGNIN);
            } catch (IntentSender.SendIntentException e) {
                Log.e("pavan", "User RESOLVE SIGN IN ERROR CALLED OUT OF IF CATCH");
                mIntentInProgress = false;
                mGoogleApiClient.connect();
            }
        }
    }
}
}

```

## Получить информацию о профиле

```

private void getProfileInformation() {
    try {
        if (Plus.PeopleApi.getCurrentPerson(mGoogleApiClient) != null) {
            Person currentPerson = Plus.PeopleApi
                .getCurrentPerson(mGoogleApiClient);
            String personName = currentPerson.getDisplayName();
            String personPhotoUrl = currentPerson.getImage().getUrl();
            String personGooglePlusProfile = currentPerson.getUrl();
            String email = Plus.AccountApi.getAccountName(mGoogleApiClient);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

## Аутентификация с помощью Firebase,

```
private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
//AuthCredential credential = GoogleAuthProvider.getCredential(tokenCredential, null);
AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(),null);
mAuth.signInWithCredential(credential)
    .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            Log.d("TAG", "signInWithCredential:onComplete:" + task.isSuccessful());

            // If sign in fails, display a message to the user. If sign in succeeds
            // the auth state listener will be notified and logic to handle the
            // signed in user can be handled in the listener.
            if (!task.isSuccessful()) {
                Log.w("TAG", "signInWithCredential", task.getException());
                Toast.makeText(MainActivity.this, "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
            }else{
                SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
                Log.e("SahajLOG", "Login PREF ISSSSSSSS ONACTIVITYRESULT
"+prefs.getBoolean("AuthByGplus", AuthByGplus));
                prefs.edit().putBoolean("AuthByGplus", true).commit();
                Log.e("SahajLOG", "Login PREF ISSSSSSSS ONACTIVITYRESULT IFTRUE..
"+prefs.getBoolean("AuthByGplus", AuthByGplus));
                Intent intent=new Intent(getApplicationContext(),MainActivity.class);
                startActivity(intent);
                finish();
            }
            // [START_EXCLUDE]
            // hideProgressDialog();
            // [END_EXCLUDE]
        }
    });
}
```

## onActivityResult

```
@Override
protected void onActivityResult(int requestCode, int responseCode,
    Intent intent) {

if (requestCode == GOOGLE_SIGIN) {
    if (responseCode != RESULT_OK) {
        mSignInClicked = false;
    }
    mIntentInProgress = false;
    if (!mGoogleApiClient.isConnecting()) {
        mGoogleApiClient.connect();
        if (mGoogleApiClient.isConnected()) {
            // getGoogleOAuthTokenAndLogin();
        }
    }
    GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(intent);
    if (result.isSuccess()) {
        // Google Sign In was successful, authenticate with Firebase
        GoogleSignInAccount account = result.getSignInAccount();
    }
}
```

```

        Log.e("SahajLOG", "account " + account.getIdToken());
        //getGoogleOAuthTokenAndLogin();
        firebaseAuthWithGoogle(account);
    } else {
        // Google Sign In failed, update UI appropriately
        // [START_EXCLUDE]
        Log.e("SahajLOG", "SIGN IN FAILED");
        // [END_EXCLUDE]
    }
}

public void signInWithGplus() {
    if (!mGoogleApiClient.isConnected()) {
        mSignInClicked = true;
        Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
        startActivityForResult(signInIntent, GOOGLE_SIGNIN);
    }
}
}

```

## Выйти

```

@Override
public void onLogout() {
    mAuth.signOut();
    if (mGoogleApiClient.isConnected()) {
        Plus.AccountApi.clearDefaultAccount(mGoogleApiClient);
        mGoogleApiClient.disconnect();
        mGoogleApiClient.connect();
        SignedInWithGoogle=false;
    }
    // Google sign out

    switchToLoginFragment();
}
}

```

Прочитайте Аутентификация Google Plus с Android онлайн: <https://riptutorial.com/ru/firebase-authentication/topic/8887/аутентификация-google-plus-с-android>

---

## кредиты

S. No	Главы	Contributors
1	Начало работы с аутентификацией firebase	<a href="#">Big Dude</a> , <a href="#">Community</a> , <a href="#">ErstwhileIII</a> , <a href="#">Ron Royston</a>
2	Аутентификация Google Plus с Android	<a href="#">Sahaj Rana</a> , <a href="#">ThunderStruct</a>