



Firestore auth Manipular usuários

Prof.me. Hélio Esperidião

Visão geral do projeto

Criação

- Authentication
- Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Liberar e monitorar

- Crashlytics
- Performance
- Test Lab
- App Distribution

Analytics

- Dashboard
- Realtime
- Events
- Conversions
- Audiences
- Funnels
- Custom Definitions
- Latest Release

Authentication

Users Sign-in method **Templates** Usage

Modelos

E-mail

Verificação de endereço de e-mail

Redefinição de senha

Alteração de endereço de e-mail

Configurações do SMTP

SMS

Verificação por SMS

Verificação de endereço de e-mail

Quando um usuário se inscreve usando um endereço de e-mail e uma senha, é possível enviar uma mensagem de confirmação para verificar o endereço de e-mail fornecido. [Saiba mais](#)

Nome do remetente De
não fornecido noreply@pdmfirebase-3c53c.firebaseio.com

Responder para
noreply

Assunto
Verify your email for %APP_NAME%

Mensagem

Hello %DISPLAY_NAME%,

Follow this link to verify your email address.

https://pdmfirebase-3c53c.firebaseio.com/_/auth/action?mode=action&oobCode=code

If you didn't ask to verify this address, you can ignore this email.

Thanks,

Your %APP_NAME% team

Idioma do modelo
inglês

Templates

E-mail de verificação

Verifica se o usuário confirmou o endereço de e-mail

```
fun VerificarUsuario(email:String, password:String) {
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                val user = auth.currentUser
                if(!user.isEmailVerified()) {
                    user.sendEmailVerification()
                        .addOnCompleteListener { task ->
                            if (task.isSuccessful) {
                                Log.d(TAG, "Email enviado.")
                            }
                        }
                }
                Toast.makeText(baseContext, "Logado com sucesso.", Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(baseContext, "Erro Login.", Toast.LENGTH_SHORT).show()
            }
        }
}
```

redefinir a senha de um usuário

```
/*  
* Para executar esse método o usuário deve estar logado.  
* apenas passe como parametro a nova senha  
* */  
fun redefinirSenha(senha:String){  
    val user = FirebaseAuth.getInstance().currentUser  
    user.updatePassword(senha)  
        .addOnCompleteListener { task ->  
            if (task.isSuccessful) {  
                Log.d(TAG, "Senha atualizada.")  
            }  
        }  
}
```

Enviar um e-mail de redefinição de senha

```
/*
 * Utilize o método sendPasswordResetEmail
 * Passe como parâmetro o email do usuario cadastrado
 * */
fun redefinirSenhaEmail(EmailUsuario:String){
    val auth = FirebaseAuth.getInstance()
    auth.sendPasswordResetEmail(EmailUsuario)
        .addOnCompleteListener { task ->
            if (task.isSuccessful) {
                Log.d(TAG, "Email enviado.")
            }
        }
    }
}
```

Excluir usuário.

```
/*chame o método delete()
O usuário deve estar logado.
* */
fun excluirUsuario(){
    val user = FirebaseAuth.getInstance().currentUser
    user.delete()
        .addOnCompleteListener { task ->
            if (task.isSuccessful) {
                Log.d(TAG, "usuario excluído.")
            }
        }
    }
}
```