Table of Contents

You can store values fully encrypted in your backend/database - transparently, securely, and without ever exposing plaintext on the server. All you need is a client-side token that lives exclusively in secure storage on the device. The backend and the database only ever see encrypted bytes. When the data is read, it's automatically decrypted on the client. In other words: your app never has to persist unencrypted data again.

This is a highly specialized capability, but incredibly powerful. Passwords never appear in plaintext on the server - even if someone were to gain direct database access due to a security flaw, they'd only see encrypted blobs.

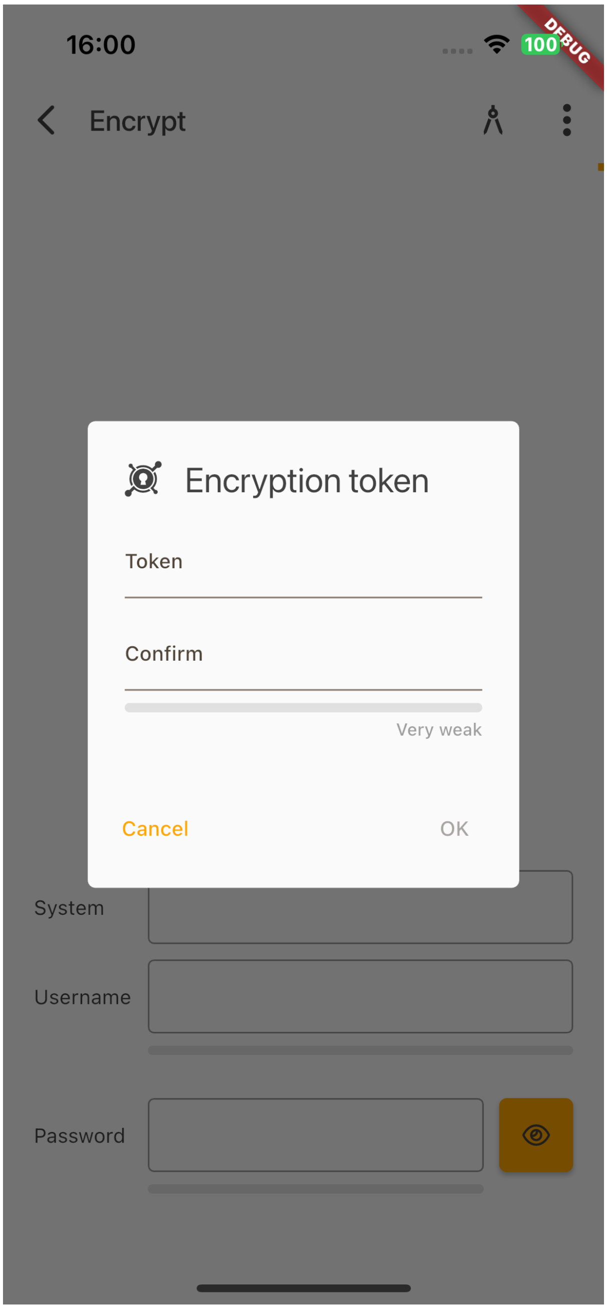
To use this feature, simply mark the columns you want encrypted:

```
book.getRowDefinition().addColumnDefinition(
  new ColumnDefinition("PASSWORD", new EncodedBinaryDataType())
);
```

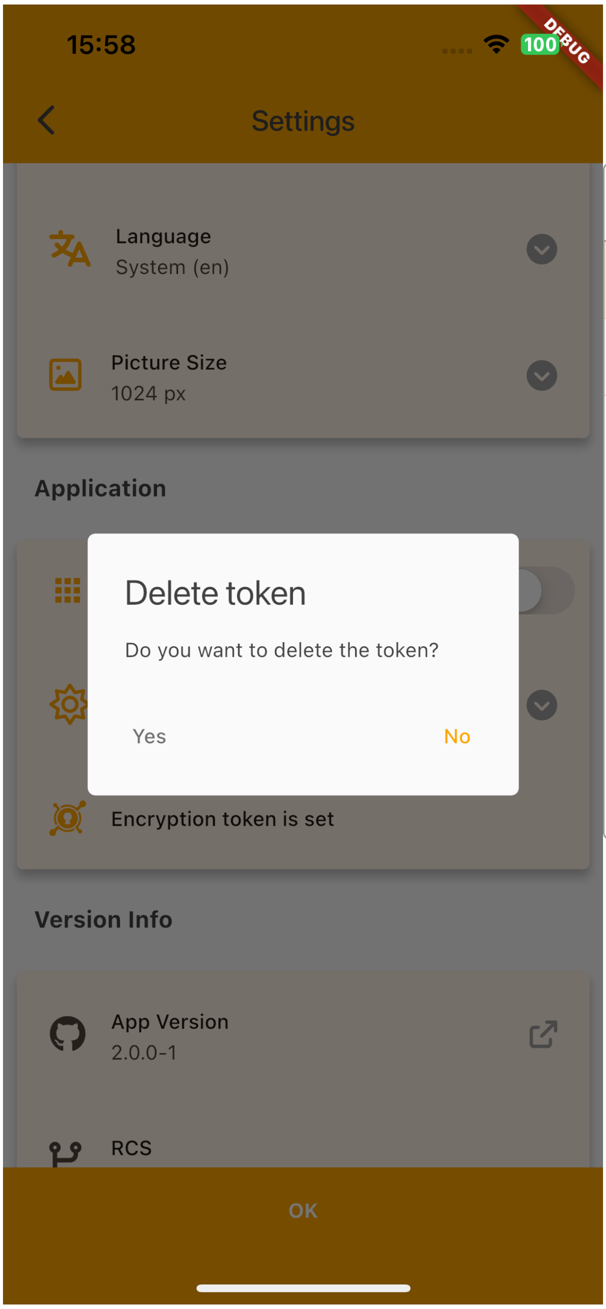
The corresponding database column must use a binary data type (not text).

When the user opens the screen in your app, they'll be prompted for a token. This token is stored securely on the device, never displayed, and used solely for encryption and decryption. Each application can use its own token. You can also delete the token to force re-entry whenever needed.

When you open a screen containing encrypted values, you will be prompted to enter a password (token), either immediately or once you start entering data.



If you have deleted the password (token), you will not be able to modify encrypted values:



15:58

Settings

Language
System (en)

Picture Size
1024 px

Application

Delete token

Do you want to delete the token?

Yes No

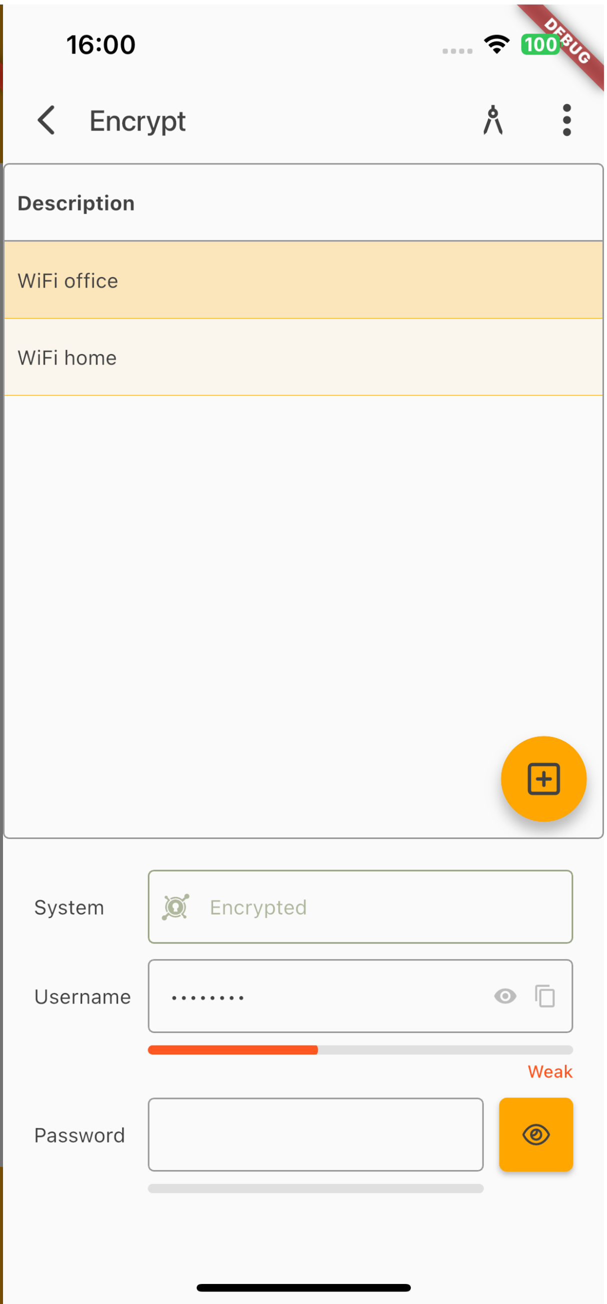
Encryption token is set

Version Info

App Version
2.0.0-1

RCS

OK



16:00

Encrypt

Description
WiFi office
WiFi home

System Encrypted

Username ••••••••

Weak

Password

Values that cannot be decrypted are marked.