sendbird

SendBird User Access Guide

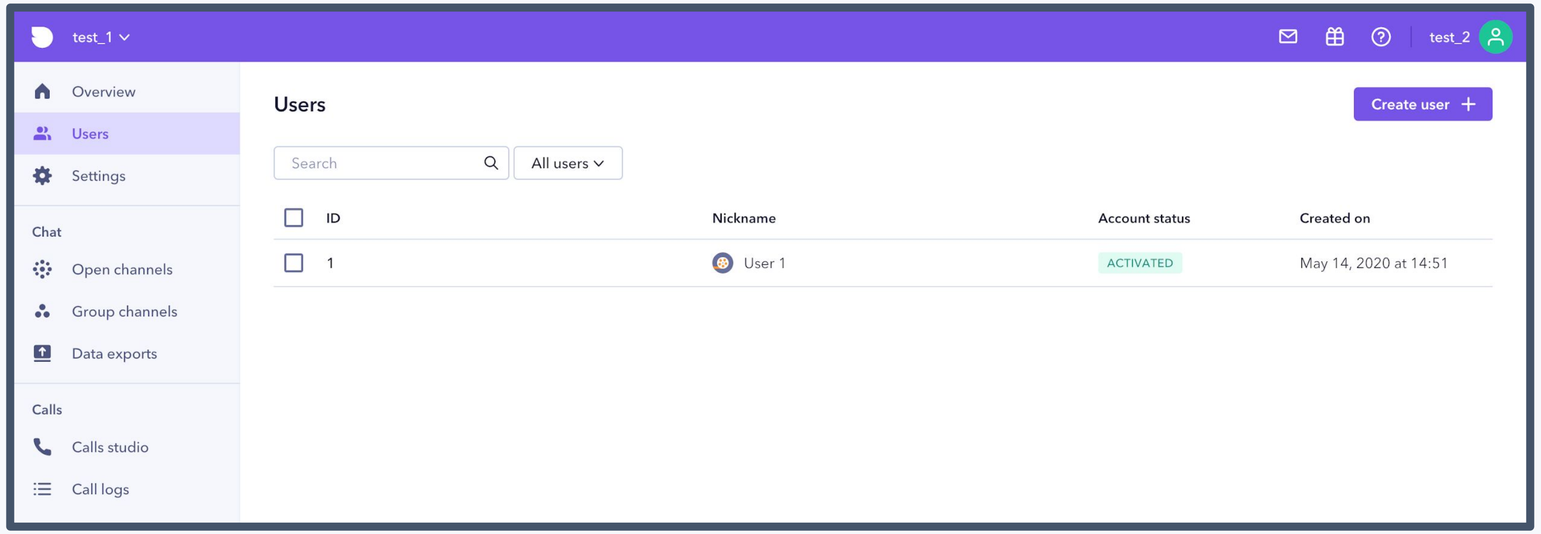
Updated 2020.05.05

Step 1 Understanding Users

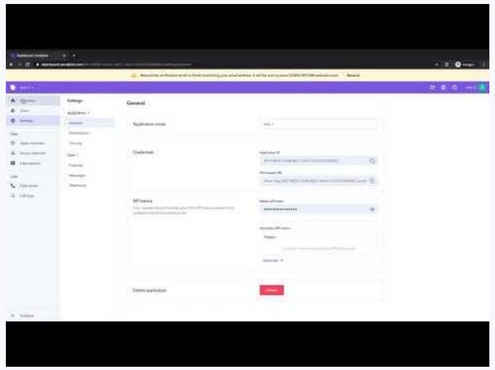
Users: Overview

User → The entity that connects to sendbird Chat or Calls via sendbird's client side SDK.

- Sendbird Application Users can be created manually when first starting development
- Users can be created via Platform API
- All User Dashboard operations can be performed programmatically, via the Platform API.
- User metadata can be added to the User object, via Platform API only.
- Users should be created one by one.
- Users connect to the sendbird Application using a sendbird SDK.



3.7 Application View - Users: Walkthrough



Step 2
Understanding Authentication

Authentication: Tokens

Sendbird provides two types of authentication tokens:

- Access tokens
- Session tokens

- Advantages of access tokens are that you don't need to reissue tokens since they don't expire. However, if the token is ever compromised, you will need to manually update the user to issue a new token (up to 10 tokens).
- Advantages of session tokens are that they are more secure due to expiration date. However, this requires caching the session token on the client or at least in your server-side and handling fetching a new session token when an outdated token expires (up to 100 active tokens).

	Access token	**Session token**
Used for	**Stateless** authentication	**Stateful** authentication
Work as	Permanent credential to the system	Temporary credential to the system
Valid or active until	Revoked	Timestamp set when issued (default: the next **7 days** from now)
Identification for	The user account	The user's current session
Tokens per user	Up to **10** (valid)	Up to **100** (active)
If exceeded the limit	The oldest token is revoked and the new one is added to the list.	The oldest and active token is revoked and the new one is added to the list.
Scopes	Not limited	Not limited
Auto-revocation	No	Yes (by default the system revokes the expired tokens)

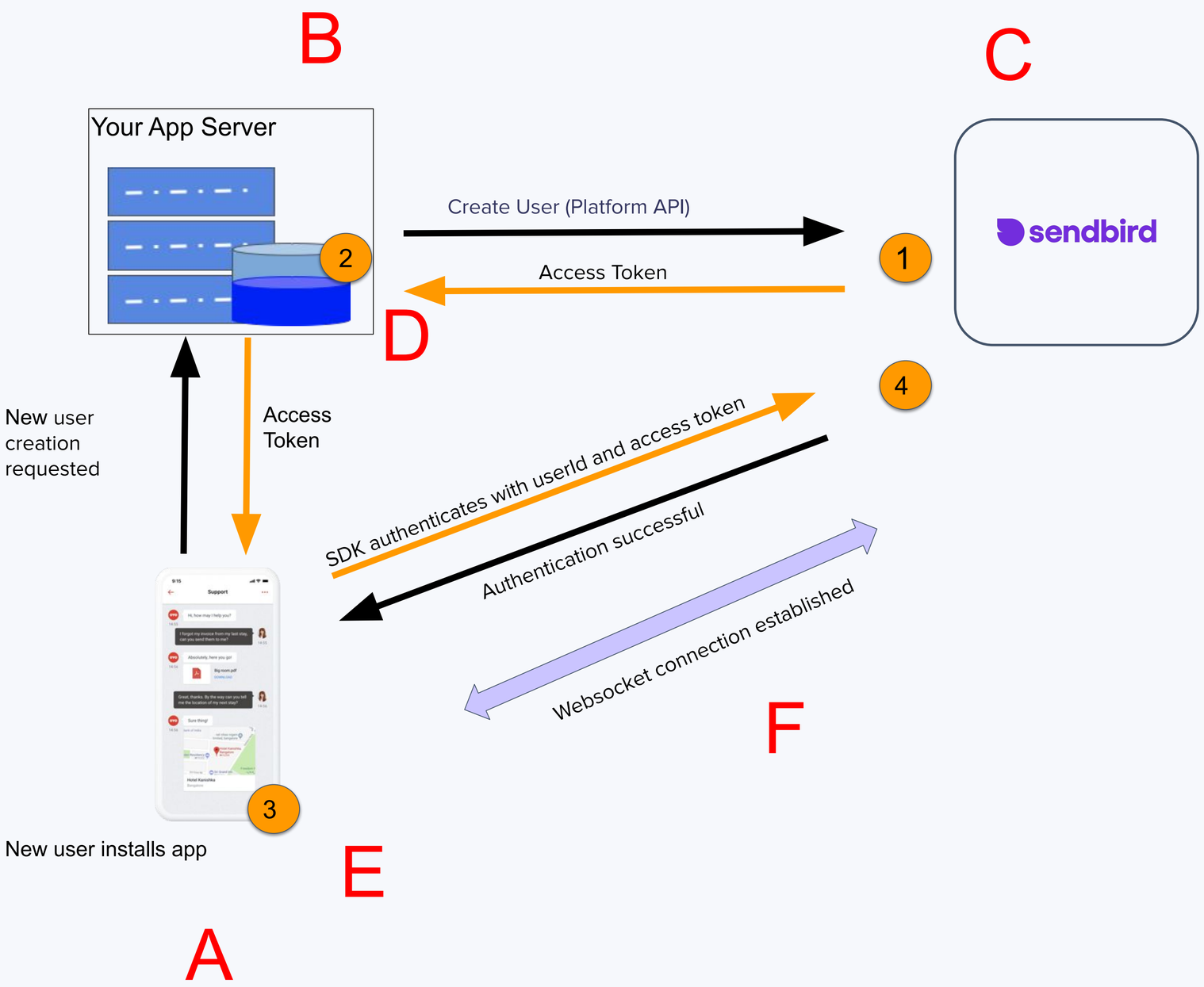
- SendBird SDKs all support the option of authenticating users with Access or Session tokens
- Tokens are only issued via the Platform API
- Tokens are only used for authentication, so even if a token is revoked or expires an already authenticated user will still have an active connection until disconnected.
- https://docs.sendbird.com/platform/user#3_create_a_user_4_access_token_vs_session_token

Step 3
Authentication Architecture

Access token: New user

In summary authentication for sendbird requires a

A. A new user installs the application and a request for a new profile in your system is made.
B. Your server creates a new user profile within your applications user database.
   a. Additionally your app server contacts sendbird requesting for a new user to be created, in the same call is a request for an access token.
   b. Often the newly created user app id from your app server can be used as the user id for creating a new user in sendbird.
C. Sendbird creates the new user and passes back an access token to your app server.
D. Your server securely stores the access token, and also passes it down to the user's client.
E. The user's client receives the access token and attempts to connect to sendbird by sending a user id and the access token.
F. If successful a websocket connection is established and real time communications begin.



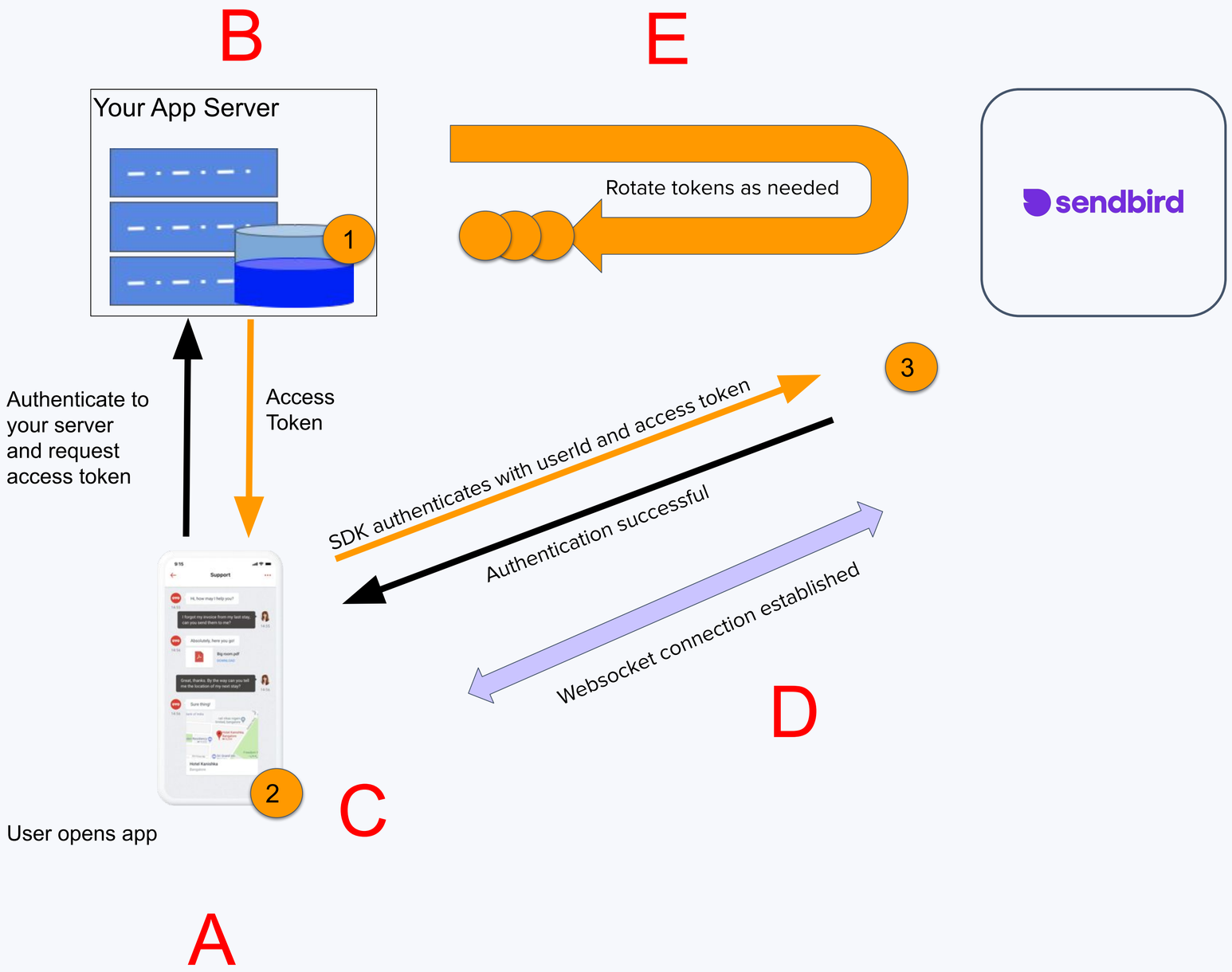
Access token: Existing user

In summary authentication for sendbird requires a

A. An existing user opens the application
B. The application authenticates the user's profile and then returns the access token stored securely for that user.
C. The user's client receives the access token and attempts to connect to sendbird by passing a user id and the access token.
D. If successful a websocket connection is established and real time communications begin.
E. Regular access token rotation is advised.

Please note, all access tokens remain valid until there are 10 in total. Then when number 11 is created it replaces number 1 (which becomes revoked) token number 12 replaces token number 2 and so on.

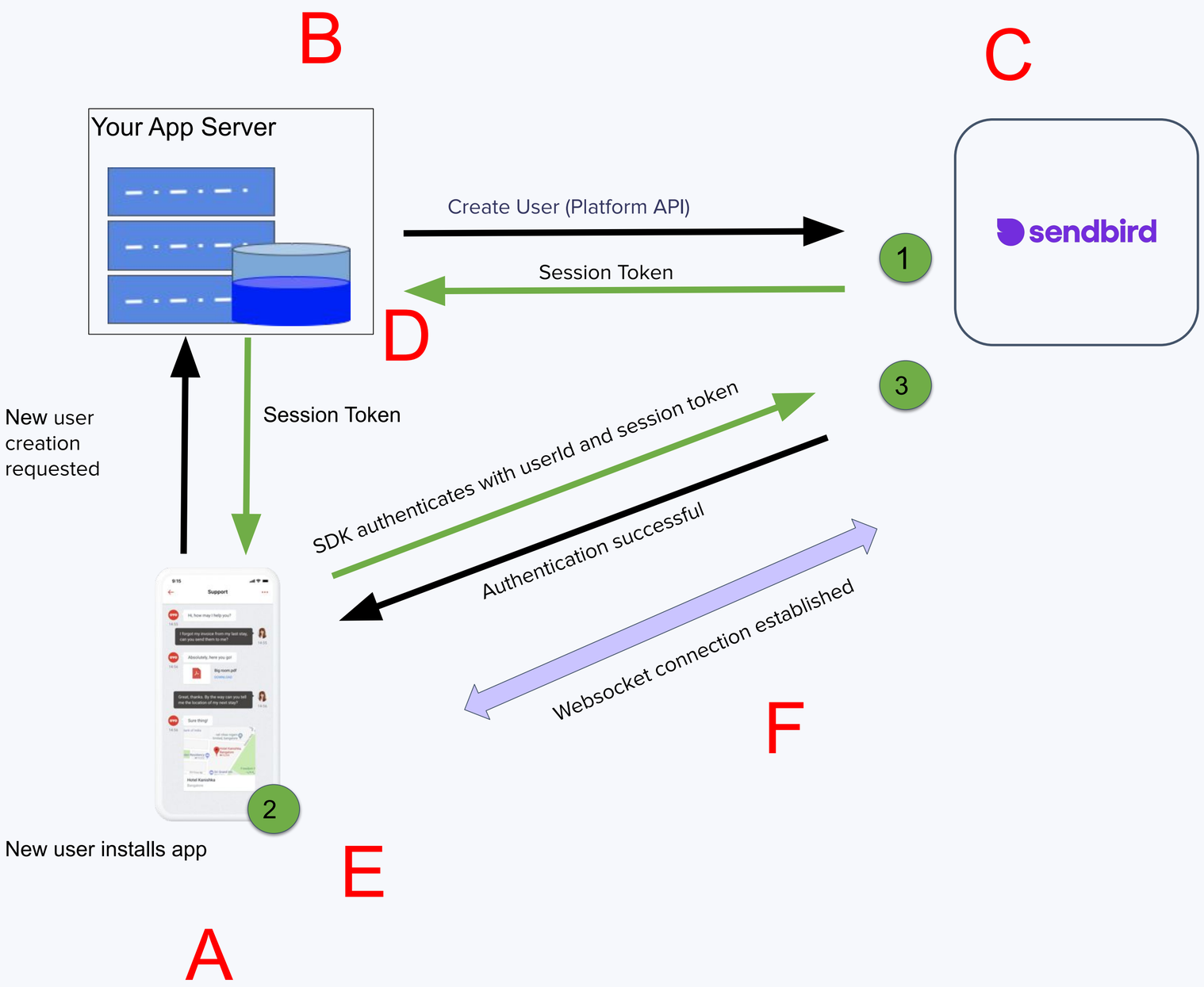
In summary, there is an access token list with a maximum length of 10. Once it fills up, any new tokens start adding at the beginning of the list replacing the tokens already there, sequentially.



Session token: New user

In summary authentication for sendbird requires a

A. A new user installs the application and a request for a new profile in your system is made.
B. Your server creates a new user profile within your applications user database.
   a. Additionally your app server contacts sendbird requesting for a new user to be created, in the same call is a request for an access token.
   b. Often the newly created user app id from your app server can be used as the user id for creating a new user in sendbird.
C. Sendbird creates the new user and passes back a session token with an expires at timestamp to your app server.
D. Your server passes the session token and expires at timestamp down to the user's client.
E. The user's client receives the session token and expires at timestamp and **stores them locally.** Then the client attempts to connect to sendbird by sending a user id and the session token.
F. If successful a websocket connection is established and real time communications begin.



Session token: Existing user

In summary authentication for sendbird requires a

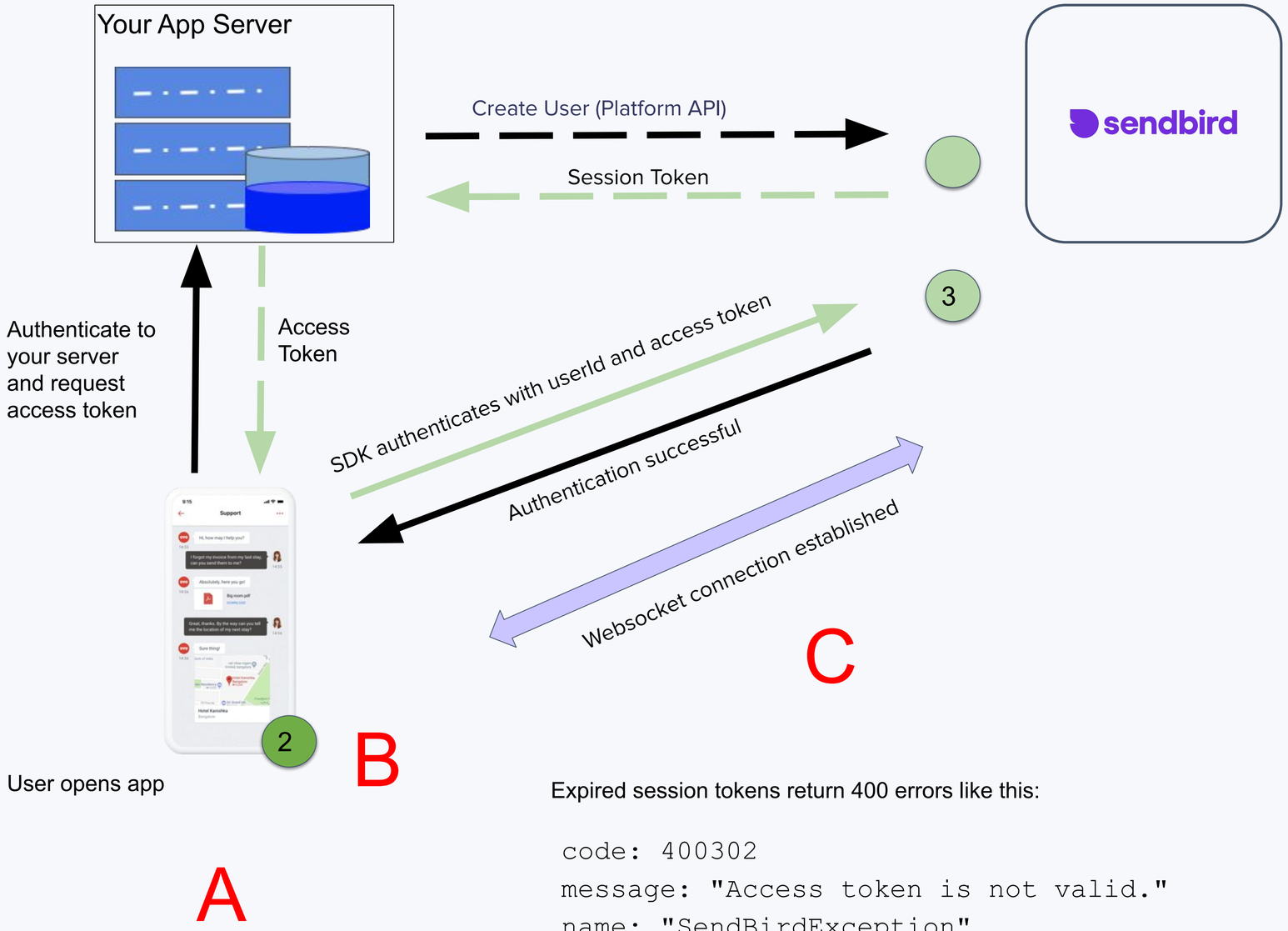
A. An existing user opens the application. The application authenticates the user's profile. The user's client checks the **locally stored session token's** expired at timestamp. if the session token has expired then the client should request to your app server for a new session token. Which is to be passed back to the user's client and **stored locally**.

B. If a session token has not expired it can be used to connect to sendbird by passing in the user's id and the session token when attempting to connect to sendbird.

C. If successful a websocket connection is established and real time communications begin.

D. If a user's device clock is wrong, the session token might already be expired. Sendbird will return an error. Therefore, account for any expired session tokens by requesting for a new session token.

Please note, a user can have up to 100 valid session tokens at one time. If session token 101 is created then session token 1 will no longer be valid. If session token 102 is created then session token 2 will not longer be valid and so on.



Expired session tokens return 400 errors like this:

```
code: 400302
message: "Access token is not valid."
name: "SendBirdException"
```