Logintap

Your Backend

Your Frontend

Current DB User Table

current set of user fields

ADD NEW DB fields to User Table or a NEW Table

.....
lt_objectuuid: varchar255 (Logintap's user ID)
lt_shortlinks: text (an array of unique links per channel)
lt_channel: varchar255 (channel selected by a user)
lt_channelstate: integer (active or non-active auth state)

shortlinks

2.1

2.3

Messenger Selection Form

Available as a ready made JS library for web shows received shortlinks in QR on desktop and as direct links on mob

Alive for 120 sec, then need to either POST "**refbots**", or POST "**createObject**", (depending on your with)

New user connection to auth starts when a user enters any shortlink.

LoginTap (LT) account settings (можно хранить в БД или в коде)

id : integer (your internal ID, if needed)
login: varchar255 (Logintap's account login)
password: varchar255 (Logintap's accounts pswrd)
accesstoken: varchar1000
refreshtoken: varchar1000 (lives 1 week)
tenant: varchar255 (hashed ID in UR of LT account)
appuuid: varchar255 (ID of LT project in your account)

accesstoken

0

LT's Library with API methods (PHP, C++, other)

auth: GET (returns access & refresh tokens)

createObject: POST (returns new LT user)
loginRequest: POST (returns hashed session & token + returns auth results to CallBack URL)
refsbots: POST (refresh unique bot links for a user)

User Recognition Mechanism

Use any variable on a Frontend to recognize who is going to need authentication. This allows sending an auth request into messenger before a user presses any Login forms.

Typically done with cookies on Web.

1.2

1.1

Auth Form (Login + Password)

1. If a user is "recognized" - show the spinner with text about expecting the mobile auth.

2. If not "recognized" - offer entering login and pressing Mobile Auth button.

3.1

Procedure checking the auth results

Compare data from the "LT-auth-sessions" table

With those received on the CallBack URL:

lt_sessionuuid = sessionuuid
lt_objectuuid = objectuuid
lt_tokenuuid = tokenuuid

If all tokens are identical + recieved loginState = 1
->
allow user entry on Frontend

ADD NEW DB Table : LT-auth-sessions

id : integer (incremental internal session number)
login: varchar255 (your internal user ID)
lt_sessionuuid: varchar255 (id of each auth session)
lt_loginState: integer (result of auth)
lt_tokenuuid: varchar255 (this session's token)
lt_objectuuid: varchar255 (user'd uuid)
lt_channel: varchar255 (channel used for this auth)

3.6

3.2

3.3

loginRequest:
objectuuid
appuuid

sessionuuid
tokenuuid

URL anywhere at your disposal

CallBack should receive POSTs from LT's IPs

3.5

3.4

sessionuuid
tokenuuid

objectuuid
loginState

2.2

objectuuid
channelState
channel

objectuuid
shortlinks

createObject:
appuuid

Authenication to make API call

Creating New User in LT

Connect User to Auth

Auth on each following Entry

Step 1 - Creating New User

Option 1 - for all users in your User Table

Option 2 - one by one, right when user connects Logintap auth

Step 2 - Connect User to Auth

Auto process starts on users entry into a bot channel from the Messenger Selection Form

channelState = 1, auth is connected
channel = name of the chosen channel

Step 3 - Entry Auth

Do loginRequest to LT and save received hashes for session ID and token.

Receive auth results to the Webhook and compare all hashes to allow entry.

Overall LoginTaps sequence of actions > > >