**FIDO v0.2 scenario:**

**A] FIDO Registration**

1) User installs app.

2) User opens the app and a pop-up for "enable push notification" is shown. User clicks on "yes". following URI is fired:

Enable Push Notifications

**URL:** http://localhost:8080/fidouaf\_test/v1/public/enablePushNotifications

**Input:**

{

"deviceid" :"",

"devicetoken": "",

"phoneNumber": ""

}

**Output**:

{

"deviceDataUpdateStatus": "true/false"

}

Above device details are stored in DB.

3) User logs into Relying Party website using his normal creds and clicks on "Register with FIDO", which will fire the below API:

**URL:** http://localhost:8080/fidouaf\_test/v1/public/rpRegRequest

**Input:**

{

"rpDisplayName": "HDFC",

"displayName": "Amogh",

"email": "amogh\_tarcar@persistent.com",

"accountId": "11123",

"phoneNumber": "9923411825"

}

**Output:**

{

"registrationResponse": "<otp>"

}

Above OTP + details like: rpDisplayName(VendorName), displayName(username) and accountId will be combined and showed together as the QR code in the following format:

eg. PSLBANK.amogh.688789.57593ff29b9d6848171247cb

[Format is: rpDisplayName.displayName.otp.accountId]

4) User will scan the QR code, extract the above 4 things seperated by the dot(.) and fire the below API, to verify the QR content and send the unique deviceId(instance id) for linking the RP and User:

**URL:** http://localhost:8080/fidouaf\_test/v1/public/verifyQRContents

**Input:**

{

"rpDisplayName": "HDFC",

"displayName": "Amogh",

"email": "amogh@gmail.com",

"otp": "<otp>",

"accountId": "11123",

"deviceId": "<instanceid>"

}

**Output:**

{

“QRCodeVerification”: “true/false”

}

5) User then fires the Registration request API (/public/regRequest), to get the policy.

6) Finally user fire the Registration response API (/public/regResponse), which will store the public key generated in DB and complete the registration process.

**B] FIDO Authentication**

The regular process which was present for FIDO 1.0 version will follow.