Flexible Authentication feature APIs (For InterAPP)

**A] Normal APP related APIs**

**1) Normal App user registration**

*URL:* http://localhost:9090/RelyingParty/service/public/appUserRegistration

*Method:* POST

*Input:*

{

"username": "jasmira",

"password": "abc",

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

"phoneNumber": "8907674534"

}

*Output:*

{

"accountid": "1112"

}

**2) Normal App user login/authentication**

*URL:* http://localhost:9090/RelyingParty/service/public/rpUserLogin

*Method:* POST

*Input:*

{

"username": "jasmira",

"password": "abc"

}

*Output:*

{

"status": "SUCCESS/FAILURE",

"token": "jasmira (same as the username; is currently set as token)"

}

**3) Normal App user logout**

*URL:* http://localhost:9090/RelyingParty/service/public/dologout/{accountid}

*Method:* GET

*Output:*

{

"status": "SUCCESS/FAILURE"

}

**B] FIDO Registration related APIs for APP.**

**1) FIDO App registration**

*URL:* http://localhost:9090/RelyingParty/service/public/appFidoRegistration

*Method:* POST

*Input:*

{

"rpDisplayName": "PSLBANK",

"displayName": "jasmira",

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

"accountId": "1112""

}

*Output:*

{

"registrationResponse": "688664"

}

**2) API to inform PSL BANK server about the authenticators that are FIDO registered via FIDO client.**

***NOTE: FIDO client will make this call at the end of FIDO registration process, after FIDO server has sent reply to regResponse API as SUCCESS for each authenticator. This is to inform the PSL Bank App’s backend server about the authenticators registered and store the AAIDs for those authenticators in DB on PSL Bank server side.***

*URL:* http://localhost:9090/RelyingParty/service/public/registeredAuthenticators

*Method:* POST

*Input:*

{

"accountId": "111222",

"aaids": [

"PIN0#0001",

"TCH0#0001"

]

}

*Output:*

{

"status": "SUCCESS/FAILURE"

}

**B] FIDO Authentication related APIs for APP.**

**1) App Authentication Request to get the enforced authenticators based on context.**

***NOTE: User logs into the PSL Bank App using FIDO. On click of Login button it will fire below API on PSL Bank server, which will return the authenticator AAIDs allowed for the sent user context based on above table(In future logic will be stored in Risk Based and Federated Identity Management systems). It will also internally call a FIDO server API to notify the server about the AAIDs being enforced on the selected context of the user:***

*Description*: Authentication Request to get the Authenticators to be used for logging in.

*URL*: http://localhost:9090/RelyingParty/service/public/authRequest

*Method*: POST

*Input*:

{

"accountId": "111222",

"location": "Home/Office/Other"

}

*Output*:

{

"location": "Home/Office/Other ",

"authenticatorsAllowed": "PIN0#0001, TCH0#0001",

"fidoNotificationStatus": "notified/ error\_in\_update/ error "

}

**2) FIDO Client Authentication Request API to get the policy from FIDO server.**

***NOTE: This is not PSL BANK App API. This API will be fired by FIDO Client APP to FIDO server directly. The Change here is that, FIDO Client App makes the FIDO server’s authRequest call by passing the AccounId of the User and gets the policy created in return. This policy JSON will include only those AAIDs which are enforced and are to be used during authentication.***

*Description*: FIDO Client Authentication Request to FIDO server.

*URL*: http://localhost: 8080/fidouaf\_test/v1/public/authRequest/flexiblePolicy/{accountid}

*Method*: GET

*Output*:

As Expected. The Policy JSON will include only those authenticator AAIDs which were selected and returned during API 1 above.

**3) FIDO Client Get signed context request from FIDO Server.**

***NOTE: After the AuthResponse API is a success for each Authenticator AAID, the PSL Bank App should confirm that this verification was done by the FIDO server himself and not by some imposter. To achieve this, the FIDO client App, will call the below API which will get the signed context from FIDO server, signed using the SSL private key generated specifically for this purpose. The public key pair of this private key is already shared and stored with the PSL Bank server for future communication processing.***

*Description*: FIDO Client Get Signed Context Request to FIDO server.

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

*Method*: POST

*Input*:

{

"rpaccountid": "151"

}

*Output*:

{

"signedContext": "sIUt4WF079hKh7W/jjXJQ2c+x79t73fdDcQoD9adS8AD58SCx65LmTCc973RyXbPw//mJbNrgsSZSW6hZKRBbBM8lGu2Gou6Kz+XcOPQkAt/5W9ys5XsrmWhdHq7GnTPK4h15C4Lyj8908BgCSaxpGTI0Oq7xj1Rdkt3dIaKKhYHrLEYfvxPFIm795GcwzXUK7r3cO/GRl6bEryCmASGU3GkcO/0sxTZ0SAK7FmQNQO2FU4wqe7D8VDR+6amSUrExSY0XRKDvyjYkawrml7R7729guQm3Wrqw8roVtL8BH6vbDcih4rNyQ9JFPsGscqKxeDlEjpbYSjkiz8zHAK5uQ=="

}

**4) APP Verify signed context request to PSL Bank server, to verify the signed context sent by FIDO server using the public key.**

***NOTE: Once the FIDO client app received the signed context from FIDO server as a result of above API, it job is to forward it to the PSL Bank App, which will then forward it to its PSL Bank Server for verification using the public key. To achieve this, below API needs to be fired from PSL Bank App.***

*Description*: Verify signed context Request to get the privately signed context data and verify it using public keys shared by FIDO server exclusively for this purpose.

*URL*: http://localhost:9090/RelyingParty/service/public/appContextVerification

*Method*: POST

*Input*:

{

"accountId": "151",

"signedContext": "sIUt4WF079hKh7W/jjXJQ2c+x79t73fdDcQoD9adS8AD58SCx65LmTCc973RyXbPw//mJbNrgsSZSW6hZKRBbBM8lGu2Gou6Kz+XcOPQkAt/5W9ys5XsrmWhdHq7GnTPK4h15C4Lyj8908BgCSaxpGTI0Oq7xj1Rdkt3dIaKKhYHrLEYfvxPFIm795GcwzXUK7r3cO/GRl6bEryCmASGU3GkcO/0sxTZ0SAK7FmQNQO2FU4wqe7D8VDR+6amSUrExSY0XRKDvyjYkawrml7R7729guQm3Wrqw8roVtL8BH6vbDcih4rNyQ9JFPsGscqKxeDlEjpbYSjkiz8zHAK5uQ=="

}

*Output*:

{

"status": "VERIFIED"

}

***OLD Context validation API (Not in use anymore):***

*Description*: Validate Contexts Request to get the privately signed context data signed and verified using public keys of each authenticator sored on FIDO server side.

*URL*: http://localhost:9090/RelyingParty/service/public/rpFidoContextValidation

*Method*: POST

*Input*:

{

"rpaccountid": "111222",

"contextDetails": [

{

"aaid": "TCH0#0001",

"signature": "zvzdvzdvdvadvdavadvavadvadvad",

"signedData": "dzvmdlvmdlkvnlakfnvlkamvlknvlksnvlkadnvlkanvklandl"

},

{

"aaid": "PIN0#0001",

"signature": "csdcmdklsfmcadklfmclkadmfclkm",

"signedData": "ccdksclafmdkfmkldmkmfckasmcdkncdkndknknkndvdllcdsv"

}]

}

*Output*:

{

"status": "VERIFIED/ NOT\_VERIFIED"

}