Oracle Banking Digital Experience

Data Protection Guide
Release 18.2.0.0.0

Part No. E97825-01

June 2018

Data Protection Guide

June 2018

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001
www.oracle.com/financialservices/

Table of Contents

1. Objective and Scope

1.1 Background

OBAPI is designed to help banks respond strategically to today's business challenges, while also transforming their business models and processes to reduce operating costs and improve productivity across both front and back offices. It is a one-stop solution for a bank that seeks to leverage Oracle Fusion experience across its core banking operations across its retail and corporate offerings.

OBAPI provides a unified yet scalable IT solution for a bank to manage its data and end-to-end business operations with an enriched user experience. It comprises pre-integrated enterprise applications leveraging and relying on the underlying Oracle Technology Stack to help reduce in-house integration and testing efforts.

In order to provide these services OBAPI needs to acquire, use or store personally identifiable information (PII). In some cases, OBAPI may be owner of the PII data and in some other cases OBAPI might just acquire and use this data for providing required services to the customer.

1.2 Objective

By the very nature of PII data, it is necessary for the Bank to be aware of the information being acquired or used or stored by OBAPI. This knowledge will enable the Bank to take necessary measures and put apt policies and procedures in place to deal with PII data. In some of the geographies Bank might need to comply with local laws and regulations for dealing with PII data. This document attempts to provide necessary information so as to enable the Bank to do so.

1.3 Scope

This document is intended for technical staff of the Bank as well as administration users of the Bank and provides information about following aspects of the PII data.

- Identifies what PII data is acquired, used or stored in OBAPI
- Process to extract PII data from OBAPI
- Process to purge and delete the PII data from OBAPI

Out of scope

This document does not intend to suggest that OBAPI is out of box compliant with any local laws and regulations related to data protection. The purpose of this document is to provide information about PII data dealt with in the system so that the Bank can put in place appropriate processes to comply with laws and regulations of the land.

2. Personally Identifiable Information (PII)

Personally identifiable information (PII) is any data that could potentially identify a specific individual. Any information that can be used to distinguish one person from another and can be used to de-anonymizing anonymous data can be considered PII.

OBAPI needs to acquire, use or store some PII data of the customers of the Bank in order to perform its desired services. This section declares the PII data captured by OBAPI so that the Bank is aware of the same and adopts necessary operational procedures and checks in order to protect PII data in the best interest of its customers.

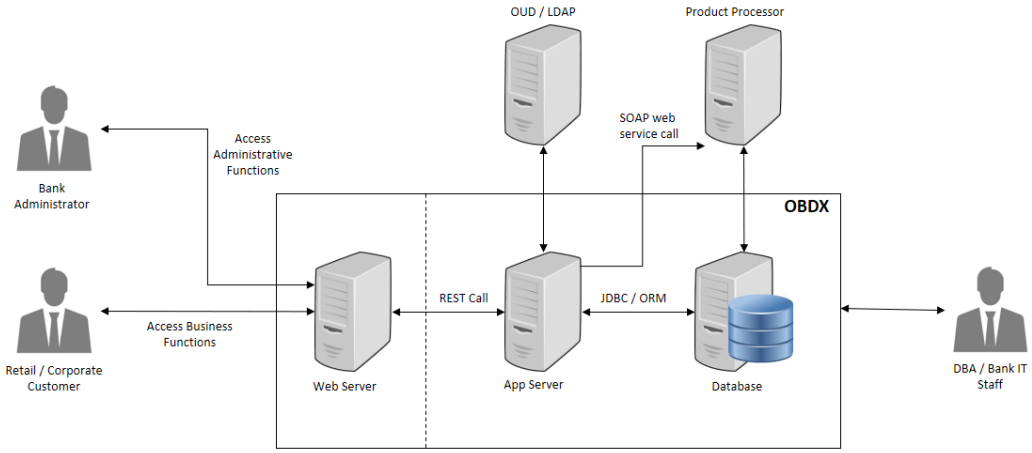
Fields	OBAPI 18.2
Bank account information	Yes
Beneficiaries	Yes
Biometric records	No
Birthplace	No
Bonus	No
Country, state, or city of residence	Yes
Credit card numbers	Yes
Criminal record	No
Date of birth	Yes
Digital identity	No
Disability leave	No
Driver's license number	Yes
Education history	No
Email address	Yes
Emergency contacts	No
Employee ID	Yes
Ethnicity	No
Financial information and accounts	Yes
Fingerprints	No
Full name	Yes
Gender	Yes

Fields	OBAPI 18.2
Genetic information	No
Health information (including conditions, treatment, and payment)	No
Healthcare providers and plans	No
Personal/office telephone numbers	Yes
IP address	No
Job title	Yes
Login name	Yes
MAC address	Yes
Marital status	Yes
Military rank	No
Mother's maiden name	No
National identification number	Yes
Passport number	Yes
Performance evaluation	No
Personal phone number	Yes
Photographic images	No
PIN numbers	Yes
Political affiliations	No
Property title information	No
Religion	No
Salary	Yes
Screen name	No
Sexual life	No
Social security number	Yes
Taxpayer information	Yes

Fields	OBAPI 18.2
Union membership	No
Vehicle registration number	Yes
Work telephone	Yes
Citizenship Number	No
Geo-Location	No
Product has Customer defined fields	No
Mobile Subscriber Identifier (IMSI)	No
Surname	Yes
First name	Yes

3. Flow of PII Data

This section depicts the flow 'personally identifiable information' (PII) within the OBAPI system in the form of a data flow diagram.



The Bank Administrator is Bank's employee who is performing administrative functions using OBAPI. As part of these, he will be dealing with PII data. An example is that the Administrator creates Retail and Corporate users in OBAPI and while creating users he/she enters user information such as first name, last name, email address, mobile number, correspondence address etc.

Retail / Corporate Customer is Bank's customer who is accessing the online banking features. As part of this he/she will be able to see his/her accounts, balances, beneficiaries, transactions, profile details etc. Note that OBAPI also supports onboarding of new users. The system captures some user information such as first name, last name, email address, mobile number, correspondence address and financial information such as income profile.

DBA / Bank IT Staff is Bank's employee who is not a user of OBAPI but has access to the database that stores OBAPI bank end data or the server environments on which OBAPI is deployed.

Web server typically contains static web content such as styling information (CSS), Javascript resources, images, static HTMLs etc. Web server passes the REST service calls to Application server.

Application (App) Server is the server on which OBAPI services are deployed. This server performs required processing on the service calls. It does use the database for retrieval or storage of data. It can also connect to external user credential store (such as OUD or Open LDAP). It can also connect to core product processor to enquiring CIF or Account related data or for posting any transactions initiated by the Retail or Corporate customer.

Database is the persistence store for OBAPI. It can contain master configuration data, user data and transactional data.

OUD / LDAP represents the external user credentials store. OBAPI does not maintain user credentials locally but depends on external specialized software to do that. An example can be Oracle Unified Directory (OUD) or Open LDAP.

Product Processor is the core banking solution which actually processes actual banking transactions. OBAPI connects to the product processor to fetch data such as CIFs or Accounts or transactions. It also connects to the product processor to post new transaction initiated by Retail or Corporate customer.

4. Administration of PII Data

This section provides information about doing administrative tasks on PII data. This includes retrieval, modification, deletion or purging of such data.

4.1 Extracting PII data

OBAPI stores some PII data in its database and it also accesses data stored or owned by external systems such as OUD / LDAP or product processor.

4.1.1 Data stored in OBAPI

This section provides information about the tables that store PII data. This information is useful for the Bank to extract PII information.

PII Data	Table
Bank account information	DIGX_AC_ACCOUNT_NICKNAME DIGX_AM_ACCOUNT_ACCESS DIGX_AM_ACCOUNT_EXCEPTION
Beneficiaries	DIGX_PY_PAYEEGROUP DIGX_PY_PAYEE DIGX_PY_DOMESTIC_UK_PAYEE DIGX_PY_INTERNAL_PAYEE DIGX_PY_DEMANDDRAFT_PAYEE DIGX_PY_INTNATNL_PAYEE_BNKDTLS DIGX_PY_DOMESTIC_INDIA_PAYEE DIGX_PY_PEERTOPEER_PAYEE DIGX_PY_INTERNATIONAL_PAYEE DIGX_PY_DOMESTIC_SEPA_PAYEE DIGX_PY_GLOBAL_PAYEE
Country, state, or city of residence	DIGX_OR_APPLICANT, DIGX_OR_APPLICANT_ADDRESS DIGX_UM_USERPROFILE
Credit card numbers	DIGX_CD_CREDITCARD_MASTER DIGX_CD_SUPP_CARD_RELATION
Date of birth	DIGX_OR_APPLICANT DIGX_UM_USERPROFILE
Driver's license number	DIGX_OR_APLT_IDNT

PII Data	Table
Email address	DIGX_OR_APPLICANT_CONTACT DIGX_OR_EMAIL_VERIFICATION (used only for email verification, data is purged once email is verified) DIGX_UM_USERPROFILE
Employee ID	DIGX_OR_APLT_EMPT
Financial information and accounts	Only financial information(Income, Asset, expense, Liability) DIGX_OR_APLT_FIN_INCM, DIGX_OR_APLT_FIN_AST, DIGX_OR_APLT_FIN_EXP, DIGX_OR_APLT_FIN_LIB
Full name	DIGX_OR_APPLICANT DIGX_UM_USERPROFILE
Gender	DIGX_OR_APPLICANT
Personal/office telephone numbers	DIGX_OR_APPLICANT_CONTACT DIGX_UM_USERPROFILE
Job title	DIGX_OR_APLT_EMPT DIGX_UM_USERPROFILE
Login name	DIGX_UM_USERAPPDATA DIGX_UM_USERPARTY_RELATION USERS GROUPMEMBERS DIGX_UM_USERPROFILE DIGX_AM_ACCOUNT_ACCESS

PII Data	Table
MAC Address	DIGX_AUDIT_LOGGING
Marital status	DIGX_OR_APPLICANT
National identification number	DIGX_OR_APLT_IDNT
Passport number	DIGX_OR_APLT_IDNT
Personal phone number	DIGX_OR_APPLICANT_CONTACT
PIN numbers	DIGX_OR_APPLICANT_ADDRESS
Salary	DIGX_OR_APLT_FIN_INCM, DIGX_OR_APLT_EMPT
Social security number	DIGX_OR_APLT_IDNT
Taxpayer information	DIGX_OR_APLT_IDNT
Vehicle registration number	DIGX_OR_APLT_IDNT
Work telephone	DIGX_OR_APPLICANT_CONTACT
Surname	DIGX_OR_APPLICANT DIGX_UM_USERPROFILE
First name	DIGX_OR_APPLICANT DIGX_UM_USERPROFILE

Please note that OBAPI provides user interface to access most of this data. The data will be accessible to you only if you have required roles and policies mapped to your OBAPI login. For example, an Administrator user can see retail user's profile only if he is entitled by a policy to access this information.

4.1.2 Data stored outside OBAPI

OBAPI can store user information in external systems such as OUD or LDAP. OBAPI provides screens for fetching this data. Please refer to the 'User Management' section of the Core user manual of OBAPI. Web help is also available at https://docs.oracle.com/cd/E97825_01/WebHelp/OBAPI.htm#obdx/core/coreintro.htm

Also note that the data can be accessed directly from the external system i.e. OUD, Open LDAP or the Product Processor. These details are outside the scope of this document. Please refer to the manual of corresponding software for more details.

4.2 Deleting or Purging PII data

There are two ways in which PII data can be deleted or purged from the system.

4.2.1 Using User Interface

The information created in (or owned by) OBAPI can be deleted from its user interface. For example, a retail user can delete the beneficiaries he/she has maintained. Please refer to 'Manage Payee' section of following user manual for more details.

https://docs.oracle.com/cd/E97823_01/webhelp/OBDX.htm#obdx/retail/payments/managepayees.htm

Note that user's data such as CIF or account number is not owned by OBDX and hence it cannot be deleted from OBAPI. However information such as account access granted to a particular user can be modified or deleted by the bank administrator. Please refer to 'Party Account Access' and 'User Account Access' sections of the Core user manual for more details.

https://docs.oracle.com/cd/E97825_01/webhelp/OBAPI.htm#obdx/core/accountaccess/acctaccesintro.htm

4.2.2 Using scheduled purge procedures

OBAPI provides some out of the box purge procedures that can be scheduled to purge the data. Otherwise the DBA / IT staff can prepare similar procedures to purge required data. However note that it is not recommended to purge or delete any data stored in OBAPI tables without doing detailed impact analysis. Please also note that the scheduled purge jobs are useful typically for purging old data. They may not be useful for purging data of a specific customer.

4.2.3 Manual truncation of data from backend

In scenarios where OBAPI does not have user interface to remove customer data and scheduled purge option is not useful, then data needs to be purged using SQL scripts. Below section provides some queries that can be used for such a purging. This option must be used with utmost care and proper impact analysis must be done before using these scripts.

PII Data	Table	Script
For modules other than Origination: **Personal information of user including Country, state, or city of residence, Date of birth, Email address, Employee ID, Full name, Gender, Personal/office telephone numbers, Login name, Work**	USERS GROUPMEMBERS DIGX_UM_USERPROFILE DIGX_UM_USERAPPDATA DIGX_UM_USERPARTY_RELATION	delete from digx_um_userparty_relation where user_id = '<USER IDENTIFIER>'; delete from digx_um_userappdata where id = '<USER IDENTIFIER>'; delete from DIGX_UM_USERPROFILE where U_NAME = '<USER IDENTIFIER>'; delete from GROUPMEMBERS where G_MEMBER = '<USER IDENTIFIER>'; delete from USERS where U_NAME = '<USER IDENTIFIER>';

PII Data	Table	Script
telephone, First Name, Surname		
Bank Account Information	DIGX_AC_ACCOUNT_NICKNAME DIGX_AM_ACCOUNT_ACCESS DIGX_AM_ACCOUNT_EXCEPTION	delete from DIGX_AC_ACCOUNT_NICKNAME where USER_ID = <USER IDENTIFIER>; delete from DIGX_AM_ACCOUNT_EXCEPTION where ACCOUNT_ACCESS_ID in (select ACCOUNT_ACCESS_ID from DIGX_AM_ACCOUNT_ACCESS where ACCESS_LEVEL = 'USER' and USERID = <USER IDENTIFIER>); delete from DIGX_AM_ACCOUNT_ACCESS where ACCESS_LEVEL = 'USER' and USERID = <USER IDENTIFIER>;

Beneficiaries	DIGX_PY_PAYEEGROUP
	DIGX_PY_PAYEE
	DIGX_PY_DOMESTIC_UK_PAYEE
	DIGX_PY_INTERNAL_PAYEE
	DIGX_PY_DEMANDDRAFT_PAYEE
	DIGX_PY_INTNATNL_PAYEE_BNKDTLS
	DIGX_PY_DOMESTIC_INDIA_PAYEE
	DIGX_PY_PEERTOPEER_PAYEE
	DIGX_PY_INTERNATIONAL_PAYEE
	DIGX_PY_DOMESTIC_SEPA_PAYEE
		delete from DIGX_PY_INTERNAL_PAYEE where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>);
		
		delete from DIGX_PY_DOMESTIC_UK_PAYEE
		where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>);
		delete from DIGX_PY_DEMANDDRAFT_PAYEE
		where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>);
		delete from DIGX_PY_INTNATNL_PAYEE_BNKDTLS

PII Data	Table	Script
		where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>); delete from DIGX_PY_INTERNATIONAL_PAYEE where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>); delete from DIGX_PY_DOMESTIC_INDIA_PAYEE where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>); delete from DIGX_PY_PEERTOPEER_PAYEE where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>); delete from DIGX_PY_DOMESTIC_SEPA_PAYEE where PAYEE_ID in (select PAYEE_ID from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>); delete from DIGX_PY_PAYEE where CREATED_BY = <USER IDENTIFIER>; delete from DIGX_PY_PAYEEGROUP where CREATED_BY = <USER IDENTIFIER>;
Credit Card Information	DIGX_CD_CREDITCARD_MASTER DIGX_CD_SUPP_CARD_RELATION	delete from DIGX_CD_SUPP_CARD_RELATION where PRIMARYCARDNUMBER in (select ID from DIGX_CD_CREDITCARD_MASTER

PII Data	Table	Script
		where PARTY_ID = <PARTY IDENTIFIER>); delete from DIGX_CD_CREDITCARD_MASTER where PARTY_ID = <PARTY IDENTIFIER>;
Party/User Information in Originations	DIGX_OR_APPLICANT DIGX_OR_APPLICANT_ADDRESS DIGX_OR_APLT_IDNT DIGX_OR_APPLICANT_CONTACT DIGX_OR_EMAIL_VERIFICATION DIGX_OR_APLT_EMPT DIGX_OR_APLT_FIN_INCM DIGX_OR_APLT_FIN_AST DIGX_OR_APLT_FIN_EXP DIGX_OR_APLT_FIN_LIB	delete from DIGX_OR_APLT_FIN_INCM where APPLICANT_ID = '<APPLICANT IDENTIFIER>'; delete from DIGX_OR_APLT_FIN_AST where APPLICANT_ID = '<APPLICANT IDENTIFIER>'; delete from DIGX_OR_APLT_FIN_EXP where APPLICANT_ID = '<APPLICANT IDENTIFIER>'; delete from DIGX_OR_APLT_FIN_LIB where APPLICANT_ID = '<APPLICANT IDENTIFIER>'; delete from DIGX_OR_APLT_EMPT where APPLICANT_ID = '<APPLICANT IDENTIFIER>'; delete from DIGX_OR_APLT_IDNT where APPLICANT_ID = '<APPLICANT IDENTIFIER>'; delete from DIGX_OR_APPLICANT_CONTACT where APPLICANT_ID = '<APPLICANT IDENTIFIER>'; delete from DIGX_OR_EMAIL_VERIFICATION where SUBMISSION_ID = '<SUBMISSION IDENTIFIER>'; delete from DIGX_OR_APPLICANT_ADDRESS where APPLICANT_ID =

PII Data	Table	Script
		'<APPLICANT IDENTIFIER>'; delete from DIGX_OR_APPLICANT where PARTY_ID = '<PARTY IDENTIFIER>';

4.3 Masking of PII data

OBAPI framework provides a facility to mask user sensitive information before showing on the screen. Masking is a process in which only some portion of the data is displayed to the user while remaining portion of the data is either skipped or is replaced with hash characters such as '*'. Main purpose of masking is to avoid a possibility of 'over the shoulder' stealing of sensitive information. However it is also used so that the clear text sensitive information is not logged in system logs.

A typical example of masking is the account numbers. When OBAPI is invoked that contains Account number is the response, the API will always give masked value. So complete clear text account number is never displayed on the screen.

OBAPI provides masking for following fields out of the box.

Sr. No	Field Name
1	Party Identifier
2	Account Number (Includes current account, saving account, deposit, loan account)
3	Debit Card Number
4	Credit Card Number
5	Mobile/phone number
6	E-mail ID
7	Social Security Number
8	Submission Identifier
9	Application Identifier

OBAPI framework also provides a provision in which any field other can the ones mentioned in above table can also be masked as per the requirement. This can be achieved by following steps:

1. Create a complex datatype in OBAPI. This datatype must extend com.ofss.digx.datatype.complex. MaskedIndirectedObject
2. Define a ‘masking qualifier’ and a ‘masking attribute’
3. Configure this masking qualifier and masking attribute in DIGX_FW_CONFIG_ALL_B. An example of the configurations for account number mask is given below

INSERT INTO digx_fw_config_all_b (PROP_ID, CATEGORY_ID, PROP_VALUE, FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS, OBJECT_VERSION_NUMBER)

VALUES ('*.account_id', 'Masking', 'AccountNumberMasking<', 'Y', null, null, 'ofssuser', sysdate, 'ofssuser', sysdate, 'A', 1);

INSERT INTO digx_fw_config_all_b (PROP_ID, CATEGORY_ID, PROP_VALUE, FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS, OBJECT_VERSION_NUMBER)

VALUES ('AccountNumberMasking', 'MaskingPattern', 'xxxxxxxxxxxxNNNN', 'Y', null, null, 'ofssuser', sysdate, 'ofssuser', sysdate, 'A', 1);

With above steps, the OBAPI framework will make sure to mask the data of this data type during serialization phase in the REST tier.

The masking pattern can contain following characters

1. N – Original character in the data will be retained
2. H – Original character in the data will be skipped
3. * (Or any other placeholder character) – Original character in the data will be replaced with this character

5. Access Control for Audit Information

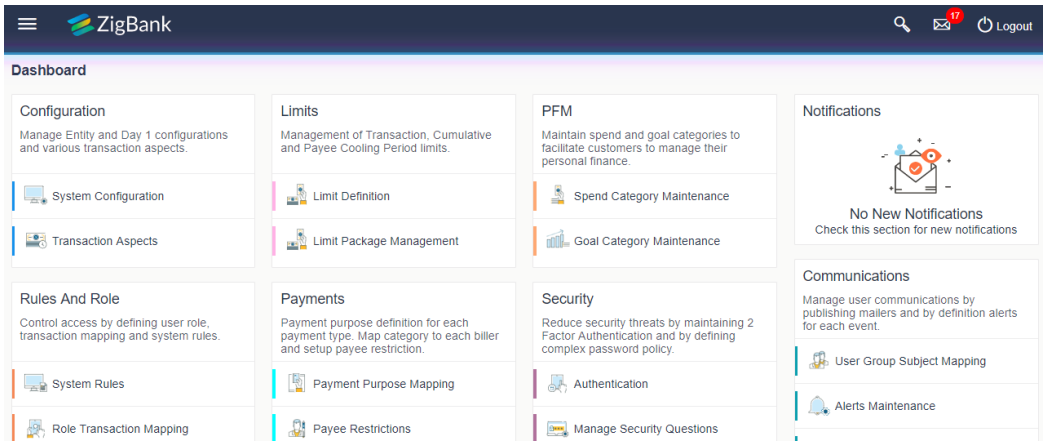
OBAPI provides mechanism for maintaining audit trail of transactions / activities done by its users in the system. This audit trail is expected to be used for customer support, dispute handling. It can also be used for generating some management reports related to feature usage statistics etc.

From a data protection perspective it is worth noting that the audit trail contains PII data in the form of transactional data as well as usage trends or statistics. Hence it is necessary for the Bank to put in place appropriate access control mechanisms so that only authorized Bank employees get access to this data. OBAPI provides comprehensive access control mechanism that the Bank can leverage to achieve this.

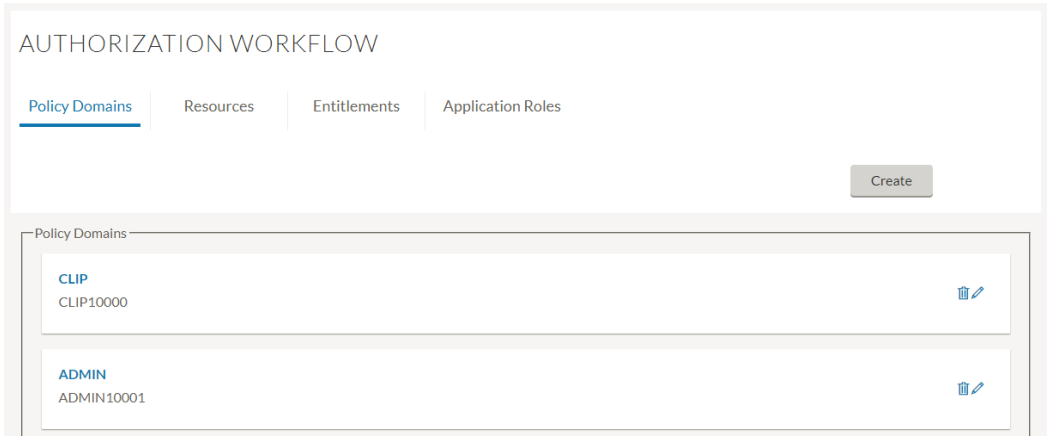
This access control can be achieved using the role based transaction mapping. This section focuses specifically from data protection aspect. You are requested to go through the user manual for 'Role Transaction Mapping' before reading further in this section. As an example, we have considered a use case where the Bank wants to restrict access to 'Audit Log' feature so that only the permitted set of administration users will be able to access audit of the users. Please note that same process can be applied to other services that deal with PII data. For example, same process can be used for restricting access to user management functions.

Check the 'out of box' access granted

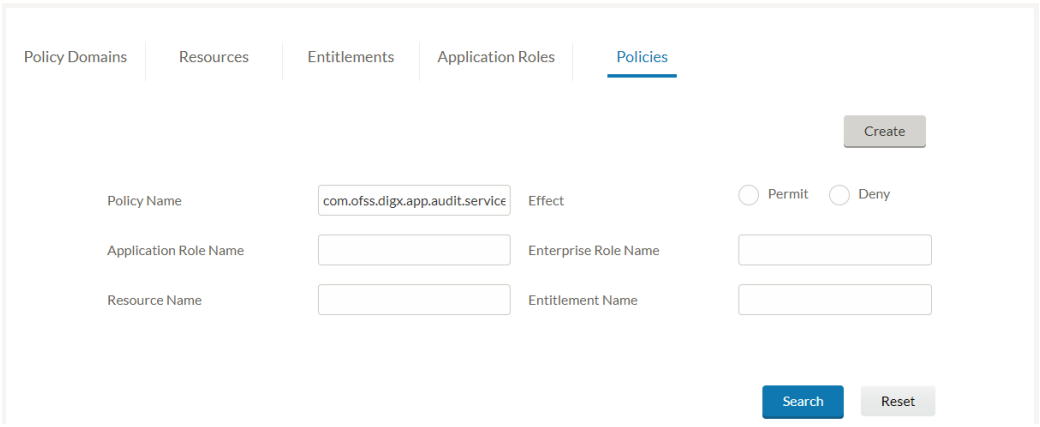
1. Login to OBAPI as Administrator



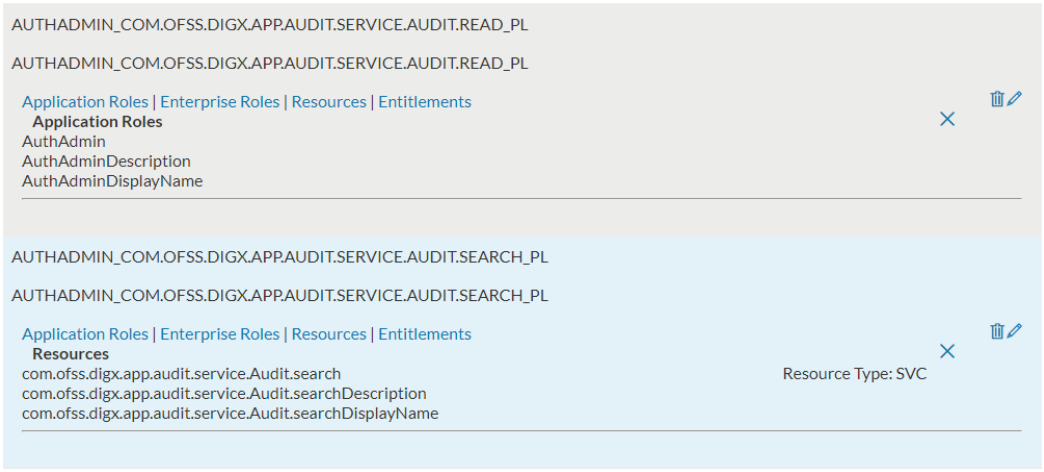
2. On dashboard, click on 'Role Transaction Mapping'



3. Click on Policy Domain ‘Admin’.



4. In ‘Resource Name’ field, input the Audit service name com.ofss.digx.app.audit.service.Audit and click on ‘Search’

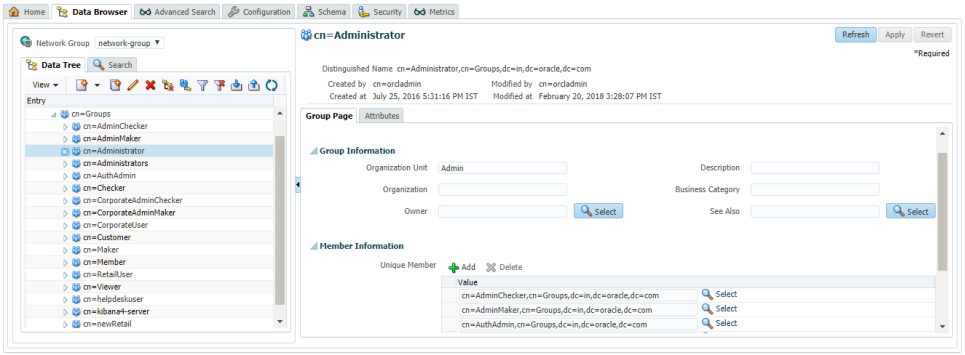


5. You will see a list of policies. Check the Application roles mapped in each of these policies. The users of these roles will get access to View Audit Log feature.
6. You can delete these policies or remove the required application roles from these policies to restrict access of ‘View Audit Log’ feature from users of those roles. Please refer to the ‘Edit Policy’ and ‘Delete Policy’ sections of the Core user manual https://docs.oracle.com/cd/ E97823 _01/PDF/UserManual/User%20Manual%20Oracle%20Banking%20Digital%20Experience%20Core.pdf for more details.

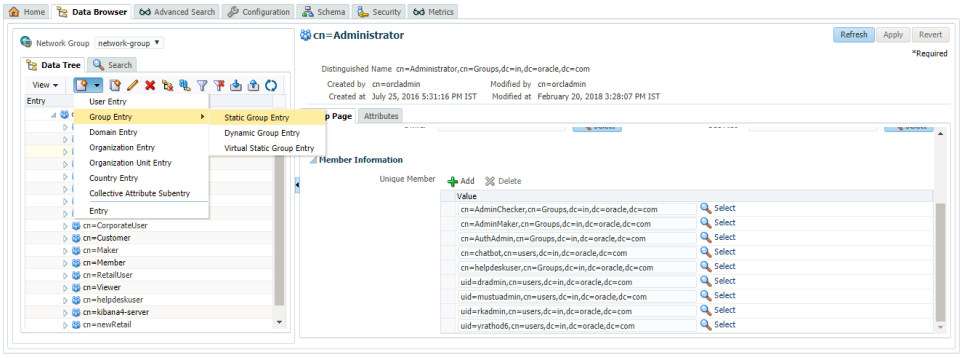
Below sections describe the steps required to grant the audit log access to restricted set of users

Create Enterprise Role

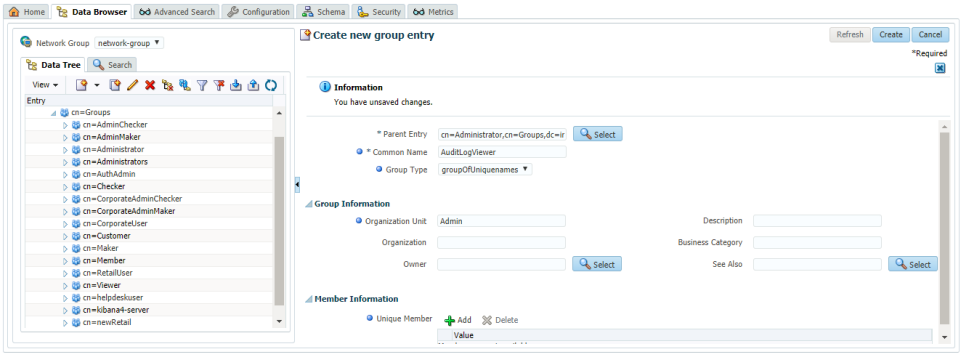
1. Login to OUD and navigate to Data Browser. In the left panel, expand Groups section. This will show you the existing enterprise roles in the system. For example, below screen shows that the ‘Administrator’ group has ‘AdminMaker’, ‘AdminChecker’ and ‘AuthAdmin’ as member groups.



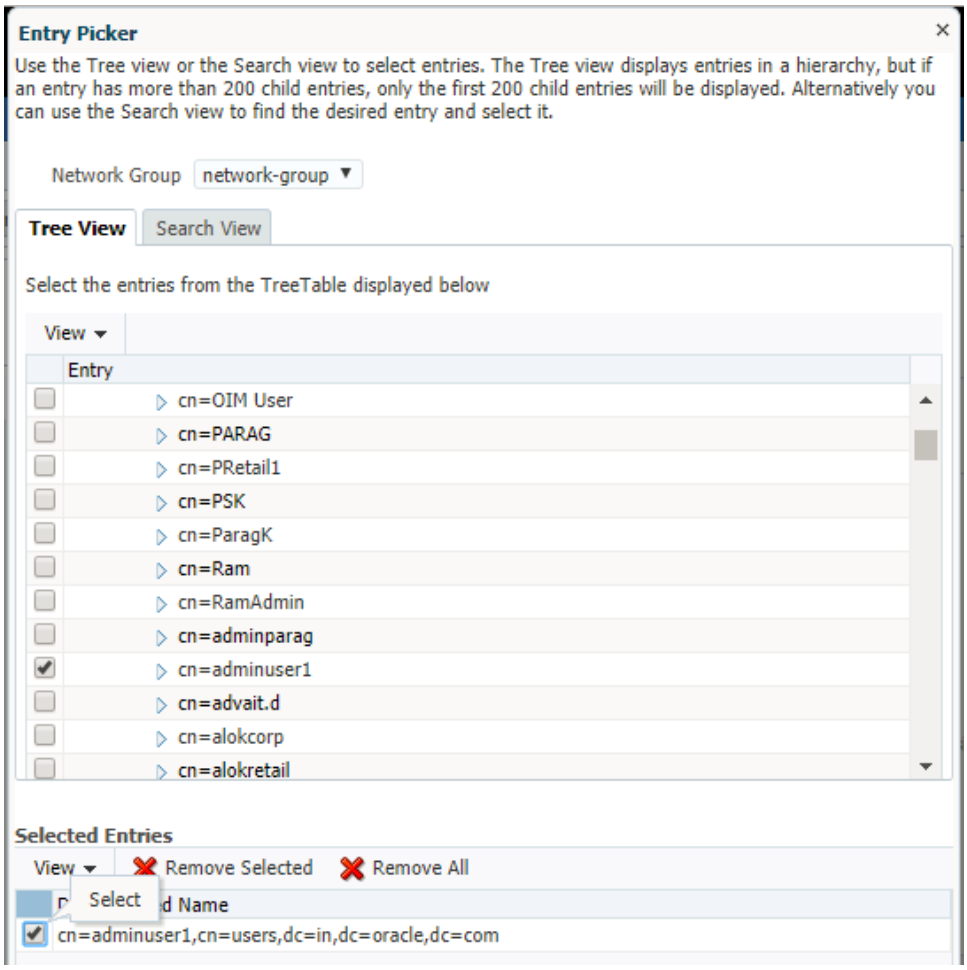
2. Create new Group by clicking on the menu option in left panel as shown in below screen

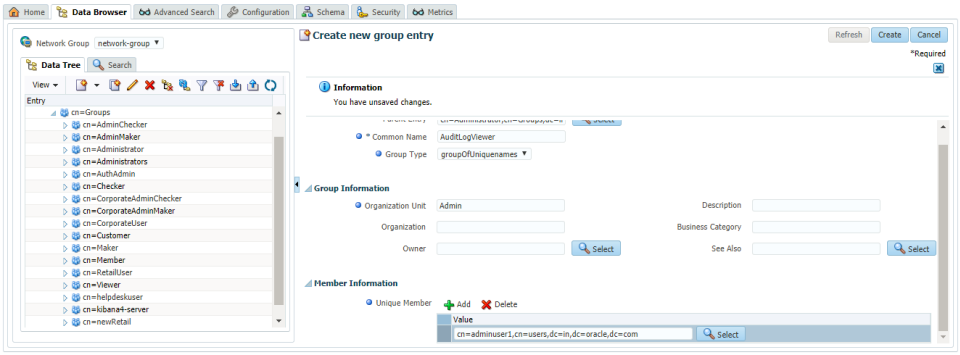


3. Provide name to the new group. Below image shows a group ‘AuditLogViewer’ created.

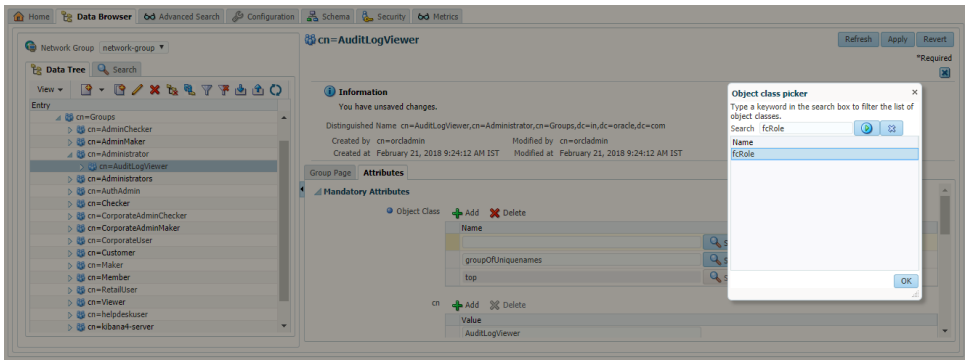


4. Under ‘Member Information’ section, click on the ‘Add’ icon to add required users to this group. Below screen shows a user ‘adminuser1’ added to the group

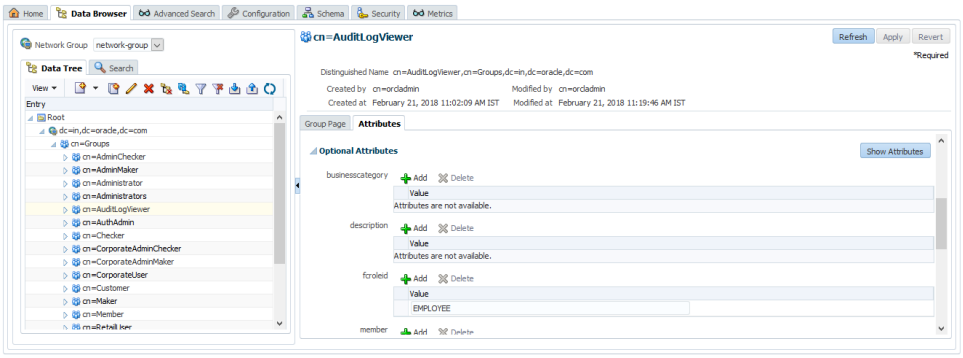




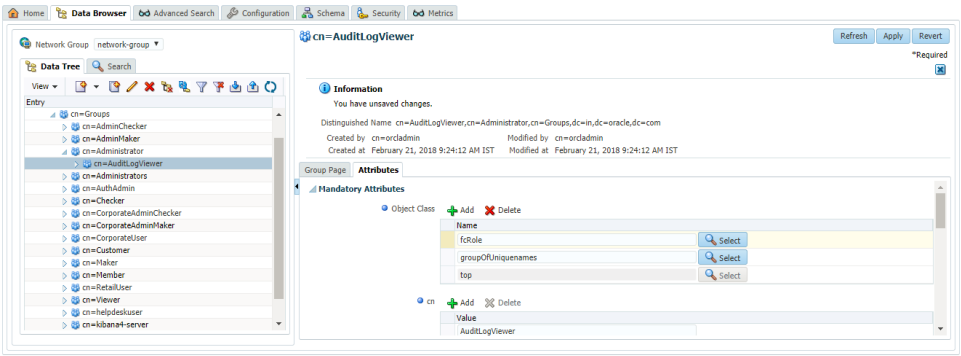
5. Click on ‘Create’ button present at top right corner of the screen to create this group
6. Click on ‘Attributes’ section of the group. Under ‘Mandatory Attributes’, click on the ‘Add’ button for ‘Object Class’. Object class picker window will appear. Select the ‘fcRole’ object class from this window.



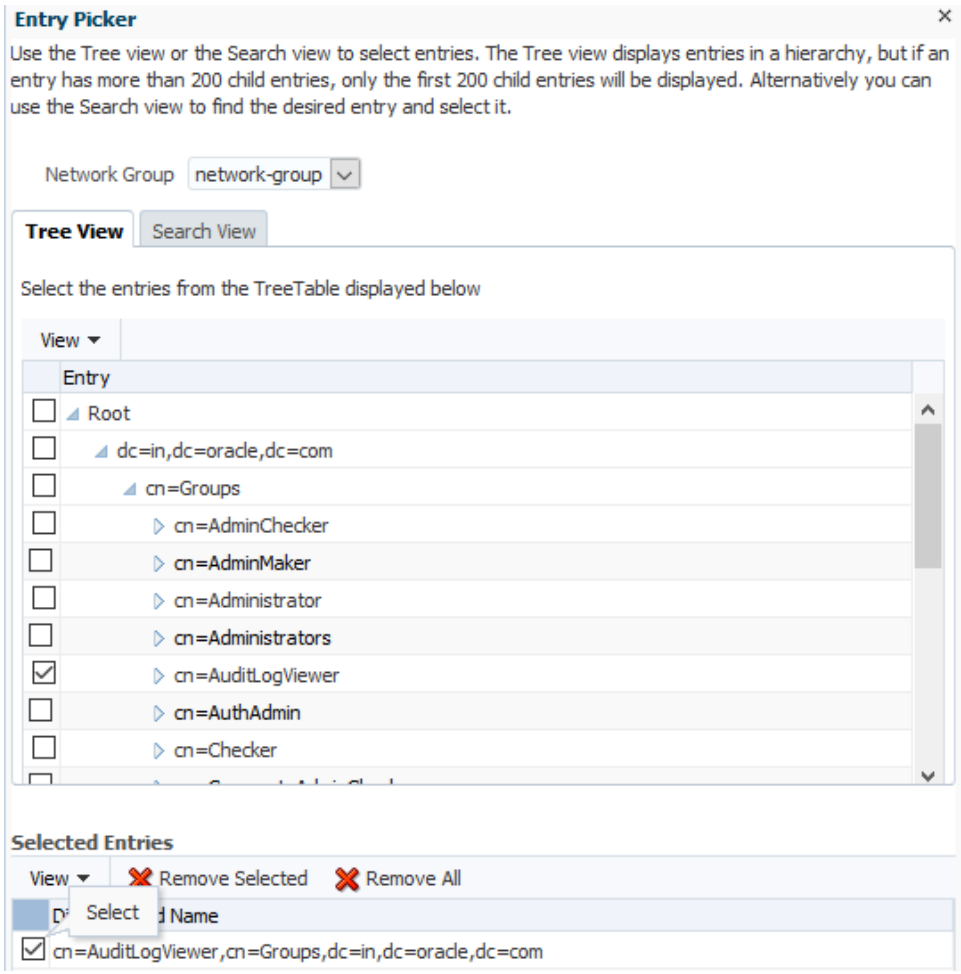
7. Under ‘Optional Attributes’, add a new attribute with name ‘fcroleid’ and value as ‘EMPLOYEE’.



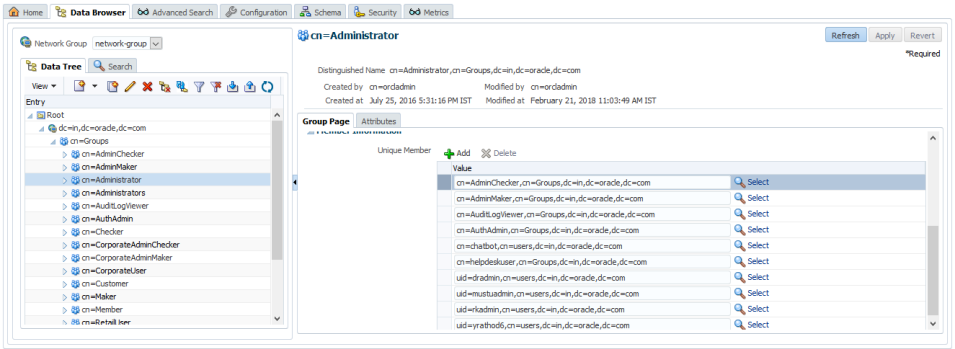
8. Click on ‘Ok’ button



9. Click on ‘Apply’ button.
10. Add the newly created group as member of the ‘Administrator’ group. For this, select the ‘Administrator’ group from left panel. Go to the ‘Member Information’ section and click on ‘Add’ icon. Now select the newly created group from the available list and click on ‘Select’



11. Click on ‘Apply’ on the main screen.

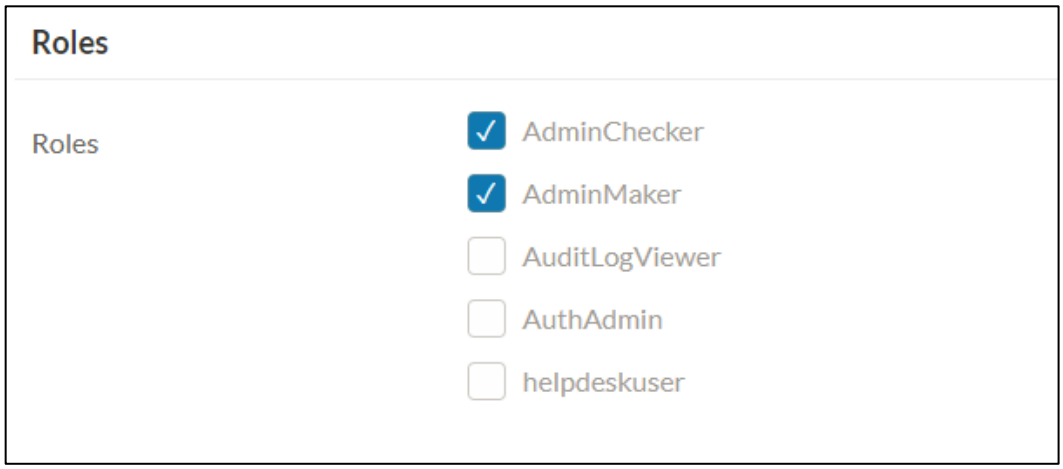


Map users to Enterprise Role

Users can be mapped to the new Enterprise role from OUD as mentioned in point 4 in previous section. Alternatively, OBAPI also provides screens for creating users and modifying them which can be used to map the newly created role to users. Please refer to 'User Management' section of following document for more details on user management.

https://docs.oracle.com/cd/E97825_01/WebHelp/OBAPI.htm#obdx/core/usermgmnt/usermgmnt.htm

Here the newly created role 'AuditLogViewer' will be visible to you for mapping to user as shown in below screen.



Create Application Role

Next step is to create an application role and map newly created enterprise role to this application role.

Please refer to the https://docs.oracle.com/cd/E97825_01/webhelp/OBAPI.htm#obdx/core/roletxnmapping/applicationrole.htm for more details on the same.

Create Resource

A resource represents specific function performed by the system. A resource is identified by fully qualified service and its method name. For example, the resource for searching audit log is com.ofss.digx.app.audit.service.Audit.search. If this resource is not already present in the system then, it can be created by the administrator. Please refer to the 'Application Resource' section of https://docs.oracle.com/cd/E97825_01/webhelp/OBAPI.htm#obdx/core/roletxnmapping/applicatnresource.htm for more details on the same.

Create Policy

Once the application role and resource are created then they can be added to an existing policy or can be added to a new policy. Please refer to the 'Authorization – Policy' section of https://docs.oracle.com/cd/E97825_01/webhelp/OBAPI.htm#obdx/core/roletxnmapping/policydomain.htm for more details on the same.

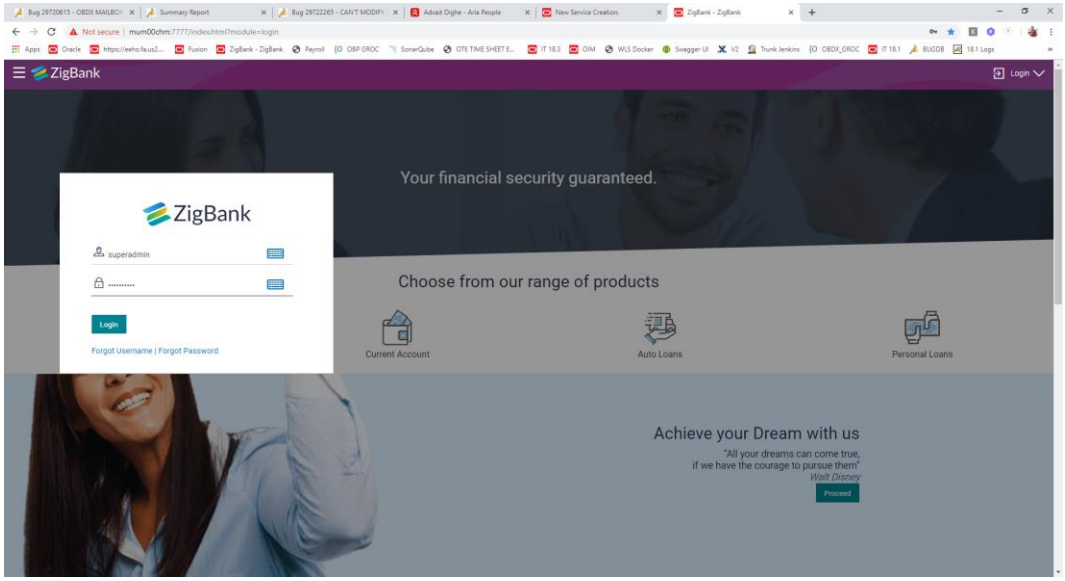
Once policy is created, users who are mapped to the application role get access to the resource mapped under the same policy.

6. User exporting the PII data

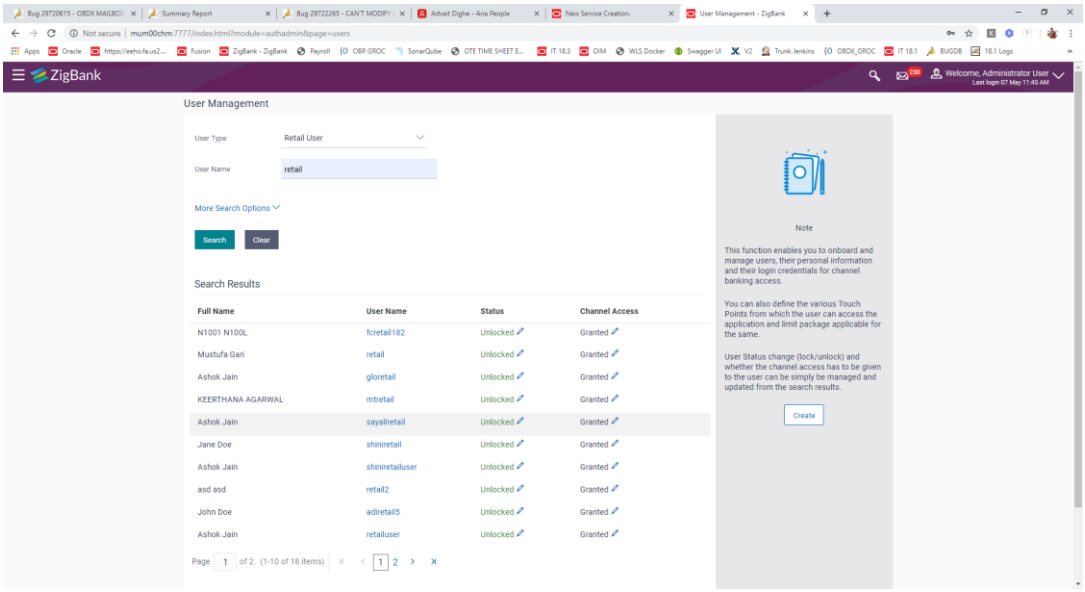
This functionality will allow to download of user wise PII in CSV formats.

6.1 Administrator

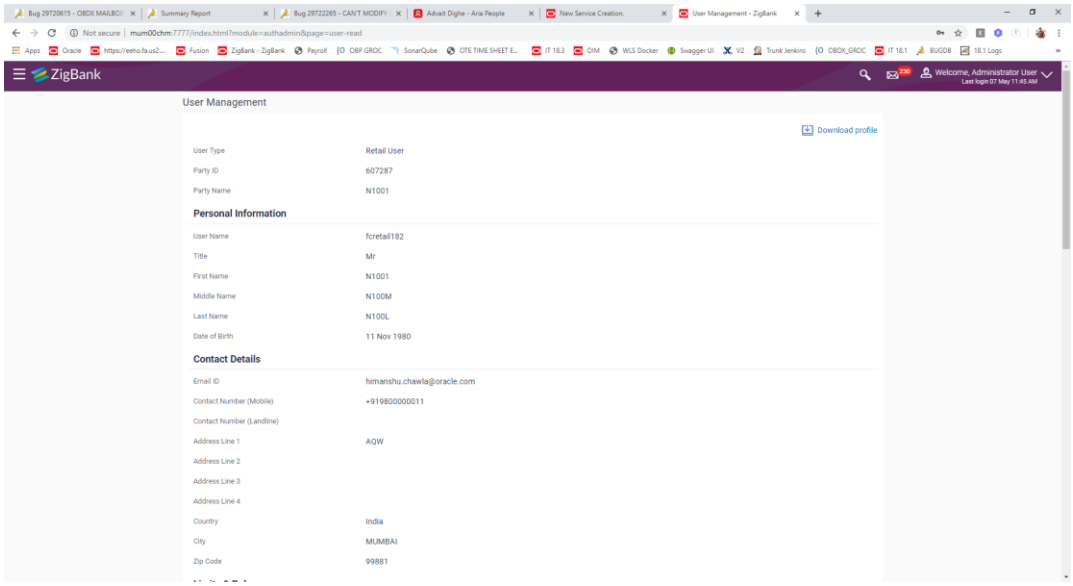
1. Login as administrator



2. Click on “User Management” and search for any user (Corporate User/ Administrator / Retail User) then clicked on the any “User Name” from the list of search users.

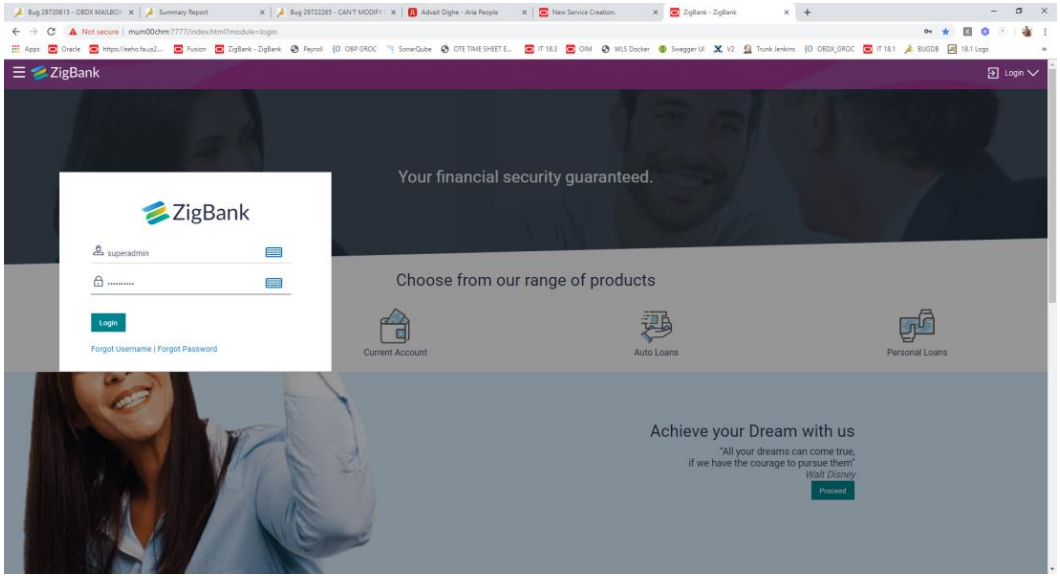


3. Clicked on the “Download profile” link.

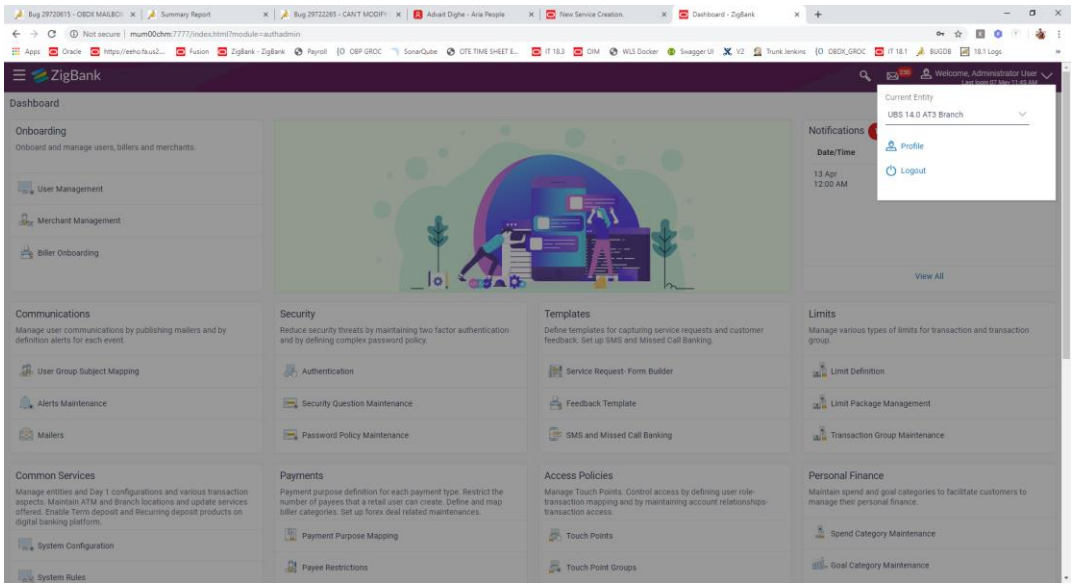


6.2 Business User

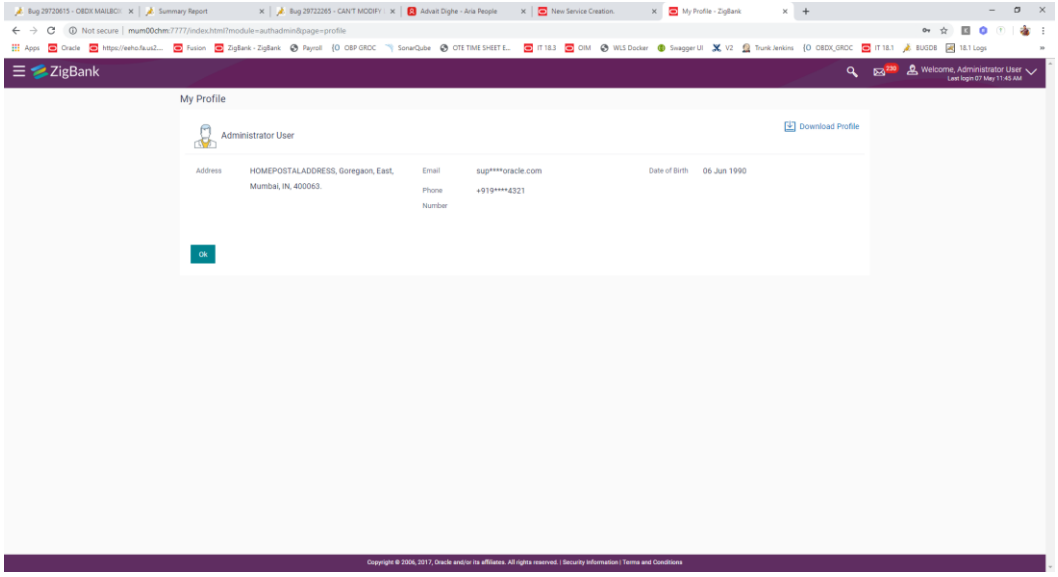
1. Login as Business User (Retail/Corporate/Admin)



2. Clicked on “Profile”



3. Clicked on “Download Profile”



7. Third Party Consents

This option enables the user to manage the access provided to third party application(s). The user can define the fine-grained entitlements i.e. account level access along with a set of transactions for the third party. The user can disable the access for a specific third party application whenever required.

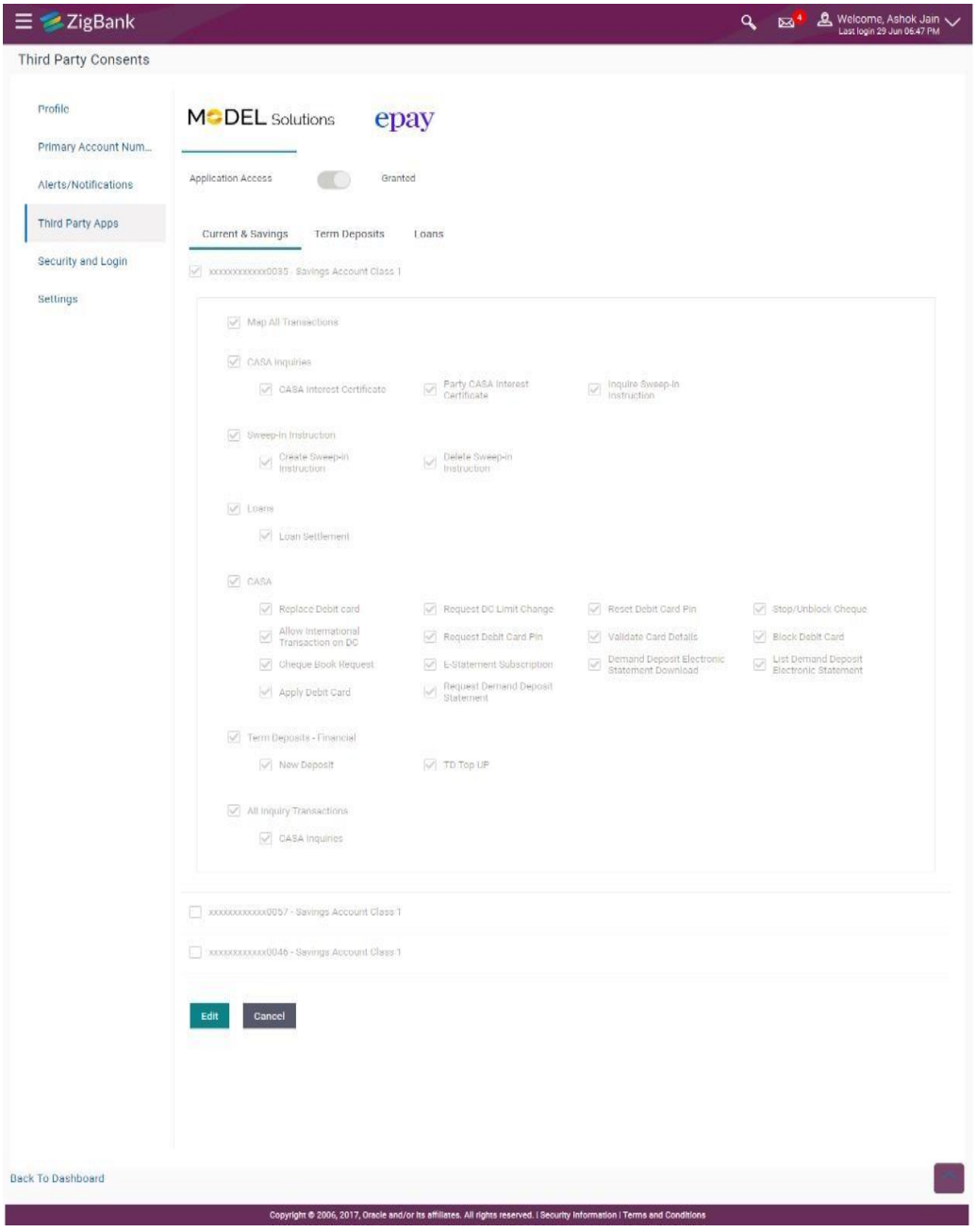
Note: Only those third party applications for which the user has registered and given rights to access his/her accounts for inquiries and transactions, will appear on this page.

How to reach here:

Dashboard > Toggle Menu > Account Settings > My Preferences > Third Party Application OR

Dashboard > My Profile > Profile > Third Party Application

Third Party Apps



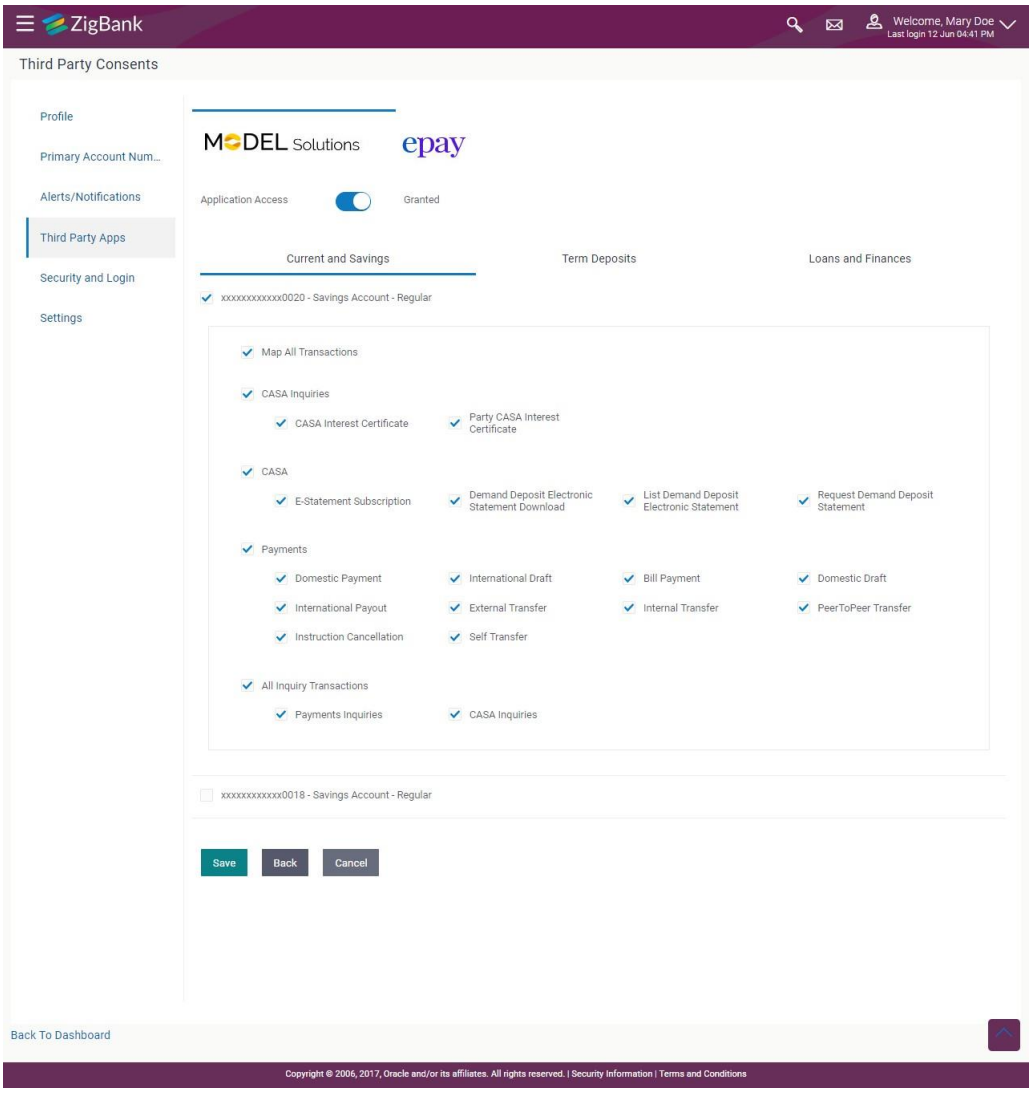
Field Description

Field Name	Description
Third Party Application Name	The names of the third party applications are displayed. Select a third party application to define access to the application.

Field Name	Description
Application Access	The option to define whether access for the application is to be provided or not. If access is granted, then the user can revoke access and if it was revoked, then the user can grant access whenever required.
Current and Savings/ Term Deposits/ Loans and Finances	Select a product to define account and transaction level access to the third party.

1. Select the third party application for which you wish to define fine grained access.
2. The system will display the list of accounts under each of the account types along with the transactions
3. Click **Edit** to modify account and transaction access. The **Third Party Consents – Edit**
4. Screen with values in editable form appears.
   OR
   Click **Cancel** to cancel the operation and to navigate back to the Dashboard.
   OR
   Click **Back** to Dashboard to go to the Dashboard.

Third Party Apps – Edit



Field Description

Field Name	Description
Third Party Application Name	The names of the third party applications are displayed. Select a third party application to define access to accounts and transactions.
Application Access	The option to define whether access for the application is to be provided or not.

Field Name	Description
Current and Savings/ Term Deposits/ Loans and Finances	Select a product to define account level access to the third party.
Accounts	All the accounts of the user are displayed under the respective account type.
Transactions	Once you select an account, all the transactions through which the account can be accessed are displayed. Select any or all transactions to provide account access for the transactions to the third party application.

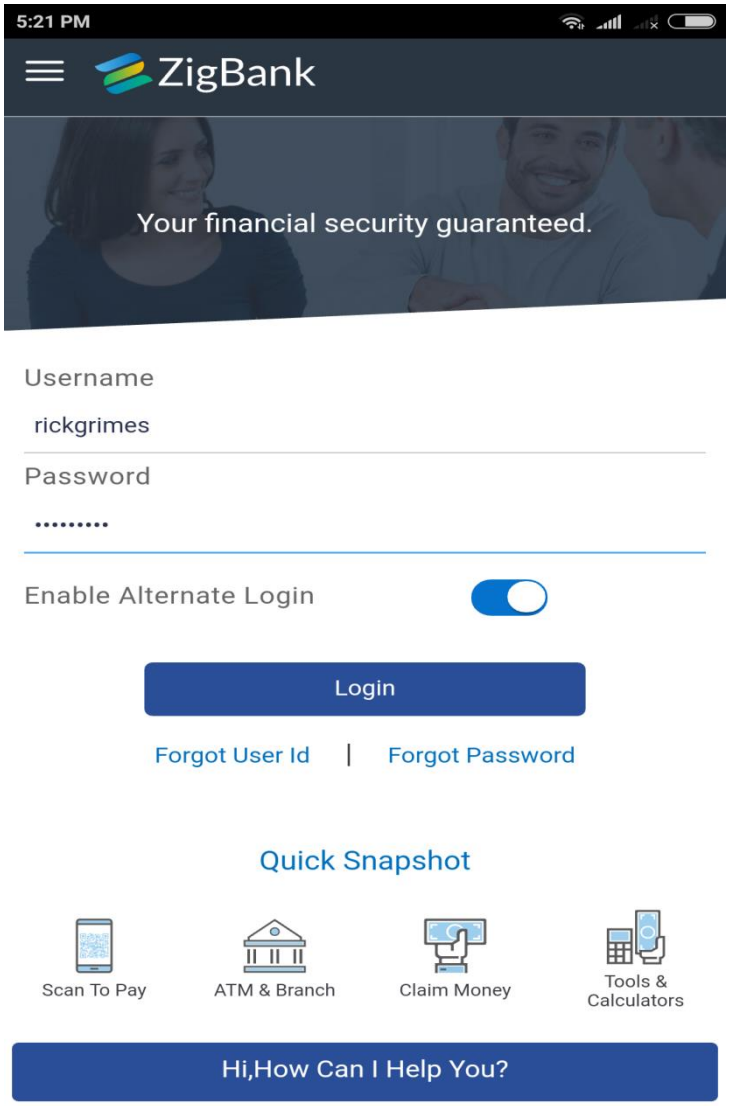
5. Click the **Application Access** button to enable / disable access for the third party application.
   a. If you select **Enable**,
      i. Click an account type.
      The account check boxes are enabled and you can select/deselect any check box to edit access of these accounts to the third party application
      ii. Select an account check box. The transactions for which the selected account can be accessed appear.
      iii. Select/Deselect all or any of the transaction checkboxes to define the transactions through which the selected account can be accessed.
6. Click **Save** to save the changes.
   OR
   Click **Back** to go back to previous screen.
   OR
   Click **Cancel** to cancel the operation and navigate back to 'Dashboard.
7. The Third Party Consents – Review screen appears. Verify the details, and click Confirm.
   OR
   Click Back to go back to the previous screen.
   OR
   Click **Cancel** to cancel the operation and navigate back to Dashboard.

8. The success message of third party consent setup appears along with the transaction reference number.
9. Click **OK** to complete the transaction and to navigate back to the Dashboard.

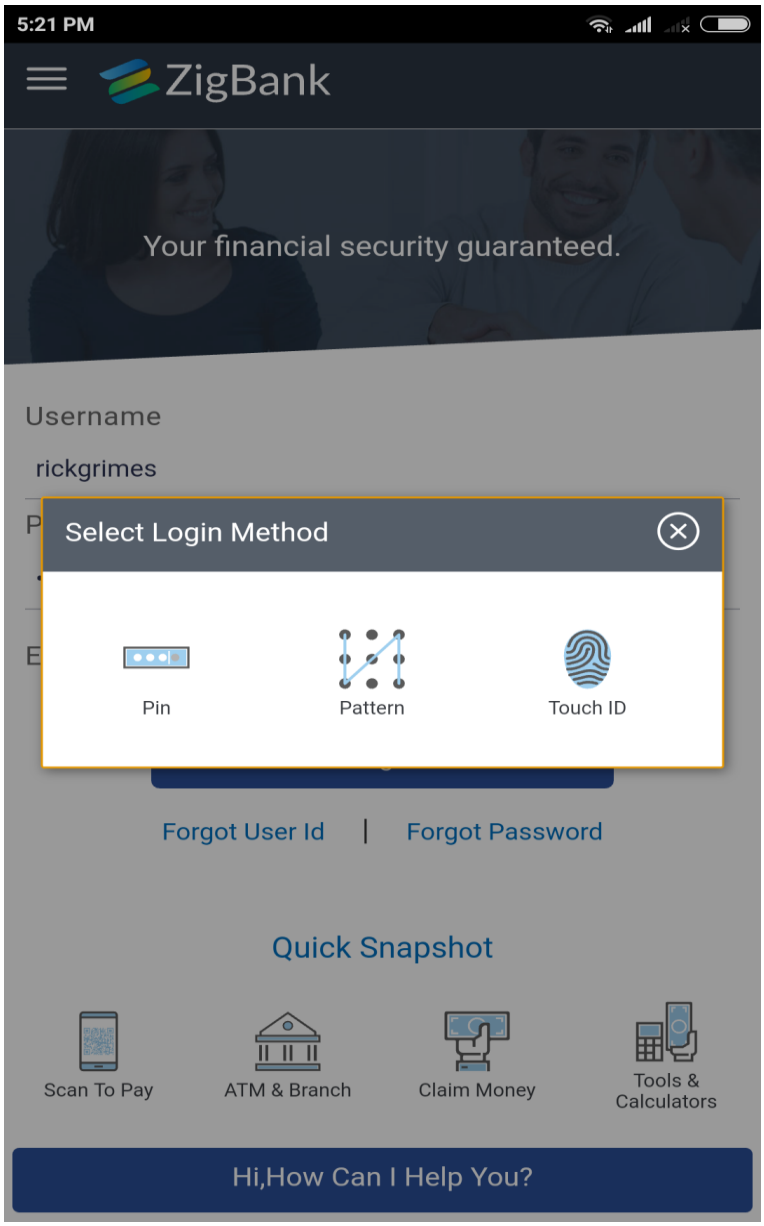
8. Device ID Consents

OBAPI framework provides a facility to enables the alternate login via Pin, pattern or touch ID.

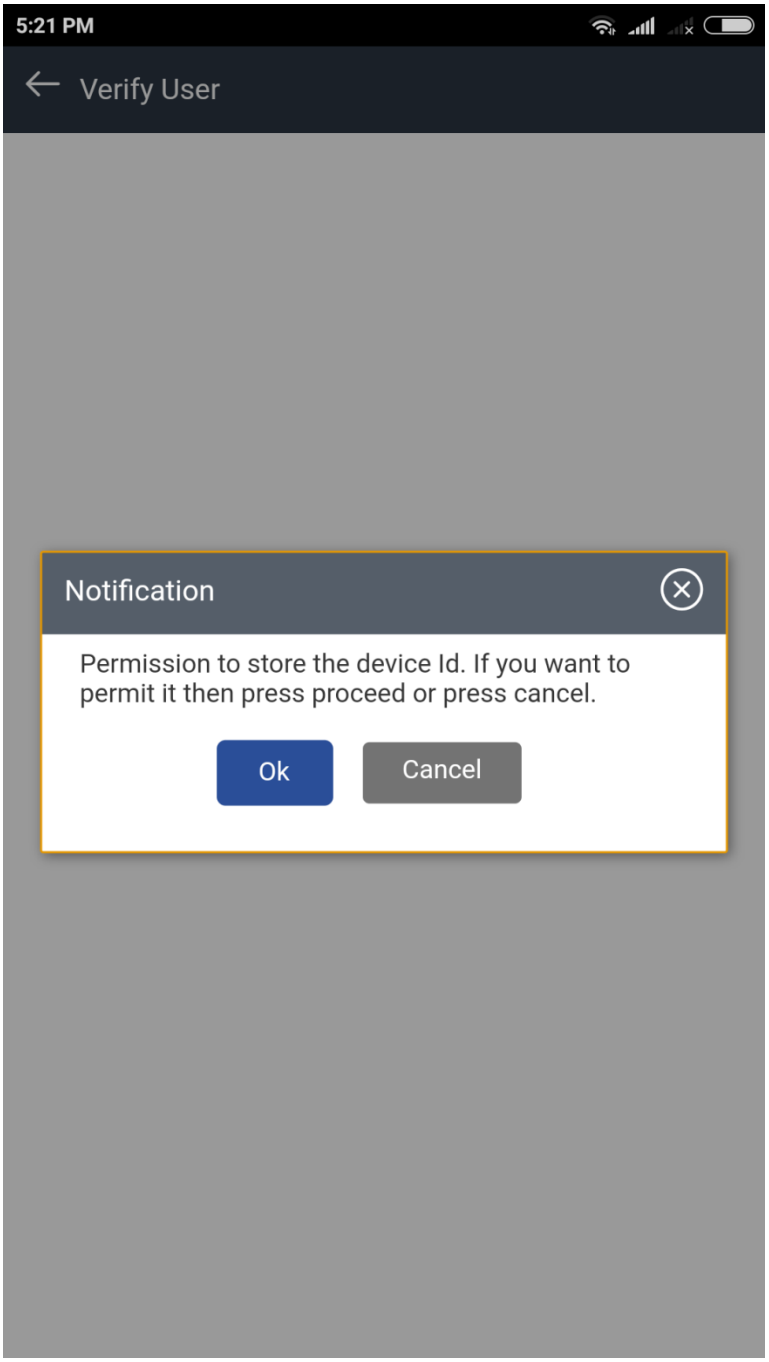
1. On the login page, user will get the “Enable Alternate login” functionality. User needs to enable this for alternate login as pin, pattern or touch ID.



2. Once user enables the functionality then, “Select Login Method” pop up will come from which user can select the alternate login method.



3. Once user will select the appropriate option, Notification of permission to store the device id message will display before setting up the alternate login method.



Unregister the Device ID

In the Settings page, user can disable the alternate login from all mobile devices.