Portal Templates

SAP Mobile Platform 3.0

DOCUMENT ID: DCxxxxx-01-0300-01
LAST REVISED: November 2013
Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.
Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary.
These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Please see *http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark* for additional trademark information and notices.

Contents

Application Services (Mobiliser) Portal Templates 1
Enabling the Mobiliser Portal Templates 1
Administration Portal 1
Logging in to the Administration Portal 1
Self Care 2
Customer Care 2
System Configuration 6
Bulk Processing 21
Coupon Admin 24
User Manager 27
Notification Manager 28
Reports 30
Audit 31
Consumer Portal 32
Getting Started 32
Self Care 34
Manage Accounts 36
Transactions 38
Bill Payment 40
Partner Portal 42
Logging in to the Partner Portal 42
Self Care 42
Customer Services 43
Bulk Processing 48
Manage Agents 50
Reports 53
Portal Development and Customization 55
Overview 55
Page Layout 56
Application Configuration 59
Application Customization 61

Style Guidelines 67
Localization Guidelines 76
Index 83

Application Services (Mobiliser) Portal Templates

The Mobiliser portal templates are reference Web applications, which you can customize to meet your mobile-banking needs. You can also develop new Web applications to communicate with SAP® Mobile Platform.

Enabling the Mobiliser Portal Templates

By default, the portal-template Web applications are disabled. You must enable the optional Mobiliser server features in the Management Cockpit.

1. Start SAP Mobile Platform Server.
2. Log in to the Management Cockpit.
3. On the Home page, select **Settings**, then select **Features**.
4. Select each of these features, and click **Enable**.
   - com.sap.mobile.platform.server.build.feature.mobiliser.feature.group
   - com.sap.mobile.platform.server.build.feature.mobiliser.web.mobile.feature.group
   - com.sap.mobile.platform.server.build.feature.mobiliser.feature.web.portal.feature.group
   - com.sap.mobile.platform.server.build.feature.mobiliser.odc

Administration Portal

Customer support agents can use the Administration Portal to manage customers and agents. For example, agents can register anyone as an administrator, an agent, a consumer, or a merchant. The tool includes functions for creating and managing coupons, and managing short message service (SMS) and e-mail notifications. Additionally, agents can configure system parameters, such as exchange rates, roles and privileges, and transaction fees and limits.

Logging in to the Administration Portal

Log in to the Administration Portal to manage customers, agents, notifications, and system parameters.

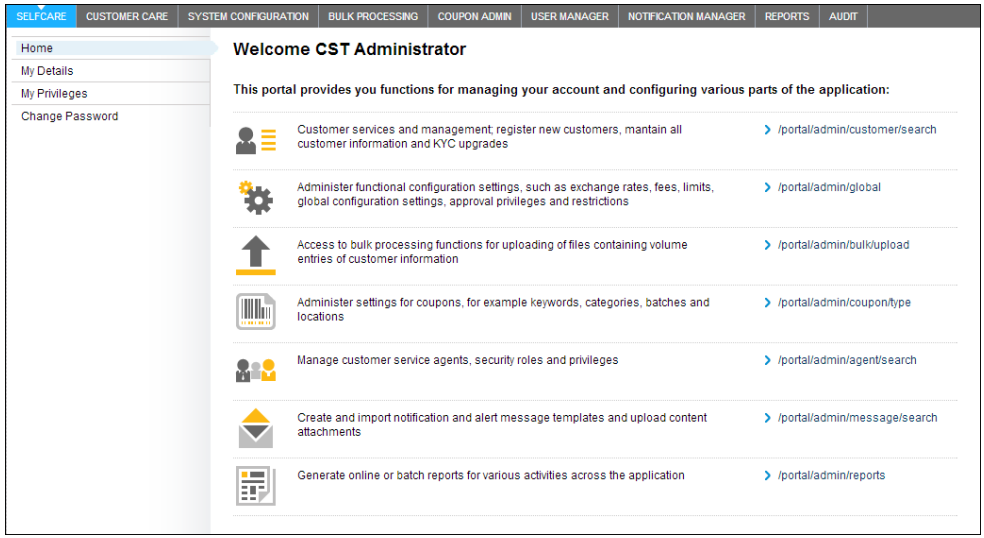
1. In a browser, navigate to: `http://www.`**serverName**`.8080/portal/`, where **serverName** is the name of the machine on which SAP Mobile Platform Server is running.
2. Enter the log-in credentials for the access you need:

User Name	Password	Access
cstfull	secret	Full administrative privileges
usermgr	secret	Manage agent accounts
notifmgr	secret	Manage notifications and alerts

3. Click **Login**.
   After logging in the first time, you are prompted to change the password.

Self Care

The Self Care option provides self-service functions for viewing your display name, login name, and privileges. You cannot edit your display name or login name using this option, but you can change your password. Depending on your assigned roles and privileges, you can change your display name using the Find/Edit Agents option.



Customer Care

Agents can use the Customer Care option to register new customers, agents, merchants, or administrators. They can search for customers who want to change their information, such as mobile phone numbers or postal addresses. Agents can also search for actions that are pending approvals, such as wallet entries, transactions, or customer registrations.

Registering Customers

Agents can use the Customer Registration option to register anyone as an administrator, an agent, a consumer, or a merchant.

Prerequisites

A valid mobile phone number, postal address, and e-mail address.

Task

1. Click **Customer Care**.
2. Select **Customer Registration**.
3. Enter all required information.
4. (Optional) Configure the individual fees and limits.
5. Click **Register**.
6. Confirm the customer's information and click **Continue**.
7. (Optional) Click **Reset Password** to generate a temporary password and click **OK**.
   A temporary password is sent to the consumer via e-mail, SMS, or both.

Find Customer

Agents can use the Find Customer option to search for customers who want to change their personal information, mobile phone numbers, or security questions and answers. Agents can also view customer transactions, and verify a customer's information for validation. You can search for customers by mobile phone number, last name, first name, display name, e-mail address, street, city, postal code, Identification OrgUnit, or Customer OrgUnit. All search parameters are optional.

Menu Item	Description
Standing Data	Customer information, such as name, date of birth, customer type, address, fees and limits, mobile phone number, and security information. You can make changes on the customer's behalf, and reset the customer's PIN, offline PIN, password, and mobile phone number. You can also deactivate a customer or set a black-list reason. If you deactivate a customer, you must select a cancellation reason; the customer is disabled and does not appear in the search results. **Note:** Deactivation is permanent. To block a customer temporarily, use the Black List feature.

Menu Item	Description
Accounts	Accounts in the consumer's wallet. You can add, edit, remove, or block (disable) payment instruments on behalf of the consumer. You can also transfer funds between accounts and set up balance alerts on behalf of the consumer. • Bank Account – add one or more bank accounts to a consumer's wallet, which makes them available for mobile transactions. • Credit Card – add one or more credit cards to a consumer's wallet. • External Account – similar to bank accounts and credit cards in a consumer's wallet, because an external account can be used to pay bills. However, you cannot add money to the customer's SVA from an external account, or transfer money from the consumer's SVA to an external account. • SVA – add funds to a consumer's SVA from a bank account or a credit card set up in their wallet. • Offline SVA – offline stored value accounts are similar to SVAs, but they are stored directly on mobile phones. You can perform the usual operations, such as adding and withdrawing funds, and viewing the current balance and transaction history. Unlike a regular SVA, you can have more than one offline SVA. You can reset the dedicated offline PIN that secures the offline SVA, which is stored on the mobile phone.
History	Tracks data changes, such as time zones and preferences, and unsuccessful login attempts.
Contacts	Agent's justification history, which explains reasons for viewing consumer information. You can close an open contact. You can also edit an open contact, but you cannot edit a closed contact.
Transactions	Consumer's transaction history.
Agent Transactions	Transactions made by the agent on the consumer's behalf.
Attachments	Consumer-related attachments, such as a scanned ID for proof of identification. You can add other attachments such as an invoice.
Bill Configuration	Current bills configured for the consumer. You can add, edit, or remove a bill configuration on the consumer's behalf.
Open Bills	All open bills for the consumer. You can pay a bill on the consumer's behalf.
Bill History	The consumer's bill history.
Messages	All SMS messages sent to the consumer, such as resetting a PIN; this does not include the registration one-time password (OTP).

Menu Item	Description
Contact Points	Contact points for the consumer, which can include e-mail addresses and mobile phone numbers. You can add, edit, or remove contact points on the consumer's behalf.
Mobile Alerts	Mobile alerts that are set up for the consumer's payment instruments. You can add, edit, or remove mobile alerts on the consumer's behalf.
Synchronization History	Synchronized results of all completed operations and pending operations.
Service Packages	Use this option to assign service packages to an individual banking consumer.

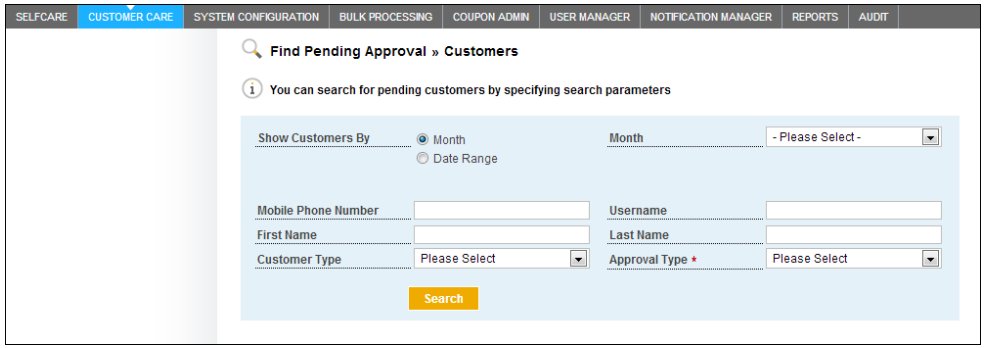
Find Pending Approval

Agents use the Find Pending Approval option to search for actions that are pending approval: consumer registrations, wallet entries, or transactions.

- Consumer search results display the consumers whose registrations are currently pending approval.
- Wallet search results display wallet entries that are currently pending approval. Depending on the payment instrument class, you may see specific information about the payment instrument that is pending.
- Transactions search results display any transactions that are currently pending approval.

You can leave the search fields blank, or you can enter search criteria to refine the search results. To approve or reject a pending registration, click the task ID, then select either **Approve** or **Reject** at the bottom of the details page.

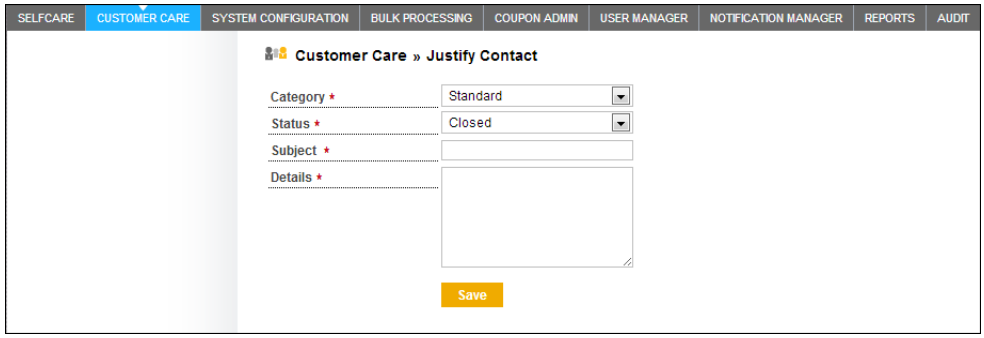
Note: You cannot edit the data on the details page of the search results.



Justify Contact

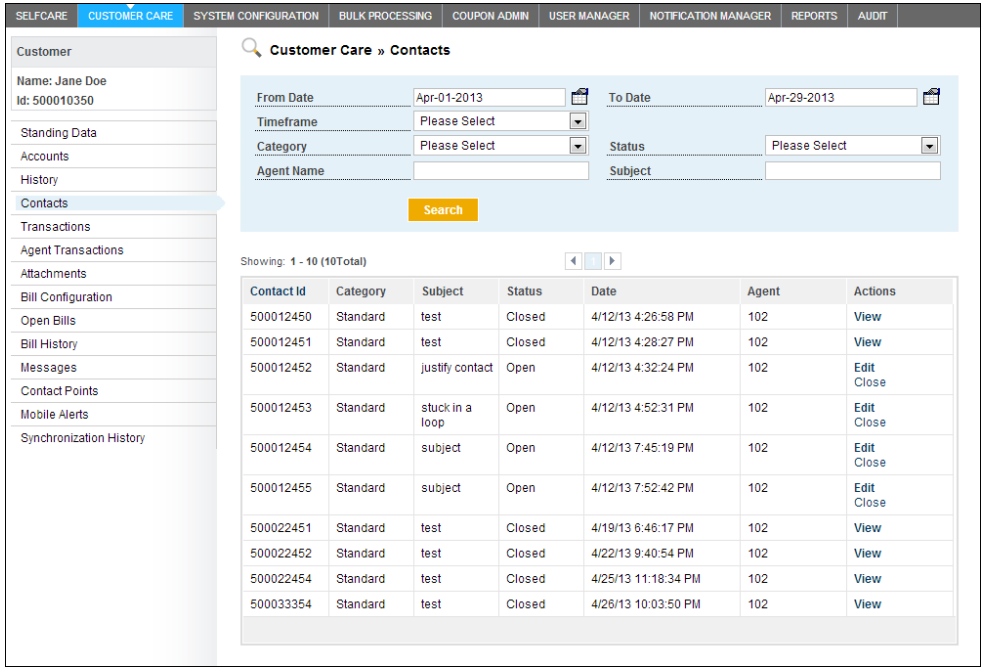
Each time you search for a consumer, you are asked to justify your reason for looking at the consumer's information. You can close a justification or keep it open. You can use the Show

Contacts feature to search for open justifications for a consumer. You cannot search for closed justifications.



Show Contacts

Agents can search for open consumer-contact justifications, and edit or close them.



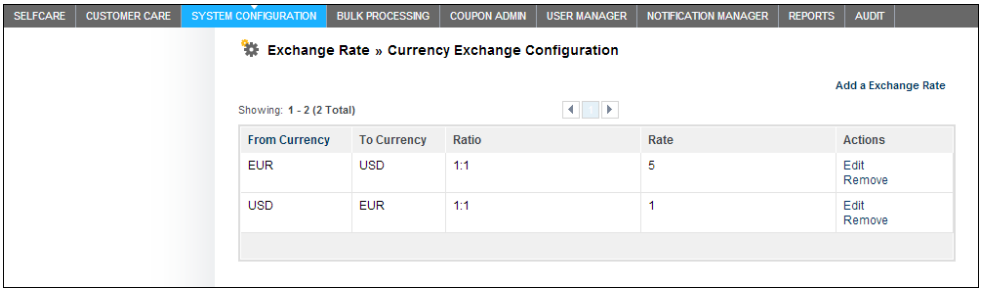
System Configuration

Agents use the System Configuration option to configure system parameters, such as fees, exchange rates, transaction restrictions, or terms and conditions.

Exchange Rate

Agents can use the Exchange Rate option to configure exchange rates between currencies that are set up during system installation, which provide a default rate for transactions. In addition to setting up exchange rates, agents can view previous exchange rates. For example, if you change the exchange rate from USD to EUR, you see the old exchange rate in the exchange-rate history search results.

Countries establish the ratio of their currencies and commit to maintaining those exchange rates. Different currencies have different values. When calculating the conversion ratio that is applied to your current conversion, you must account for conversion fees. Many banks and other converters calculate the fee as a percentage of your total conversion, so the more money you convert, the higher the fee. Only after accounting for the fee can you accurately calculate the conversion ratio that is used in the transaction.



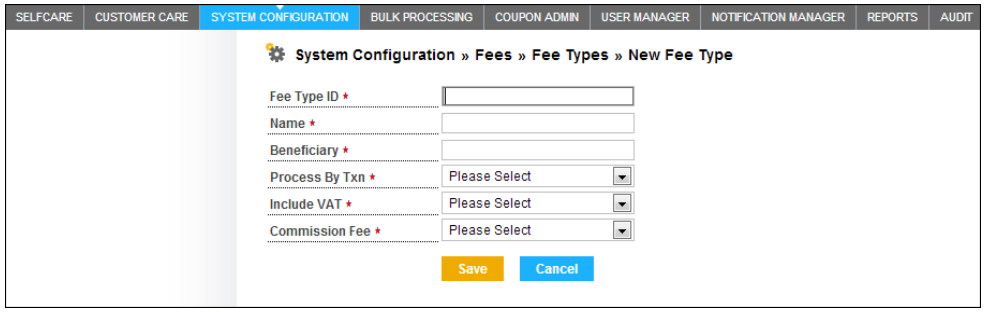
Fees

Fees and commissions are calculated for payers and payees in financial transactions, per subtransaction. However, fees and commissions are processed differently: fees are applied and charged during transaction processing; commissions are collected and settled offline in batch mode.

Fee Types

Agents use the Fee Types option to configure applicable fee types for payers and payees. For each financial subtransaction, fees are calculated, applicable fee types for payer and payee are

determined, and the scale used for the customer (payer or payee) is found. Scales are connected to fee types through fee sets.



Field	Description
Fee Type ID	A five-digit number that identifies the fee type.
Name	Short description of the fee type.
Beneficiary	The consumer who receives the fee. The beneficiary receives funds only via an offline settlement process.
Process By Txn	Indicates whether the fee should be deducted during transaction processing; if not, it is charged in a separate process.
Include VAT	Indicates whether the fee should include value added tax (VAT).
Commission Fee	Indicates whether the fee type is a commission.

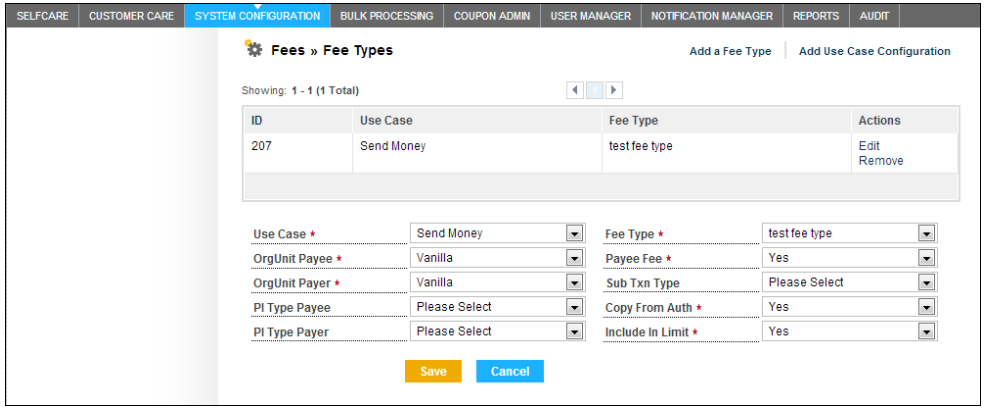
Adding Fee Types

Agents can set up fee types to specify beneficiaries who receive the fees. Agents can also specify whether to process the transaction, include VAT, and if there is a commission fee.

1. Click **System Configuration**, and select **Fees**.
2. Select **Fee Types**.
3. Click **Add a Fee Type**.
4. Enter all required information.
5. Click **Save**.

Fee Type Use Case Configuration

Use cases provide a set of requirements for processing financial transactions, such as merchant payments, bill payments, or commission settlements. A fee-type use case configuration identifies the fee type, based on organizational unit and payment instrument for both the payer and the payee. A use case is linked to one or more fee types.



Field	Description
Use Case	Use case to which the fee applies.
OrgUnit Payee	Organizational unit of the payee.
OrgUnit Payer	Organizational unit of the payer.
PI Type Payee	Indicates that the fee applies only to transactions in which the payee's payment instrument is of this type.
PI Type Payer	Indicates that the fee applies only to transactions in which the payer's payment instrument is of this type.
Fee Type	Fee type that applies to the transaction.
Payee Fee	Indicates whether the fee must be paid by the payee or the payer.
Sub Txn Type	Subtransaction type to which the fee is applied.
Copy from Auth	Indicates whether the fee should be copied from the authorization subtransaction.
Include in Limit	Indicates whether to use the fee when checking the consumer's limit.

Adding Use Case Configuration

Agents can set up a fee-type use case configuration to identify the fee type, based on the organizational unit and the payment instrument, for both the payer and the payee

Prerequisites

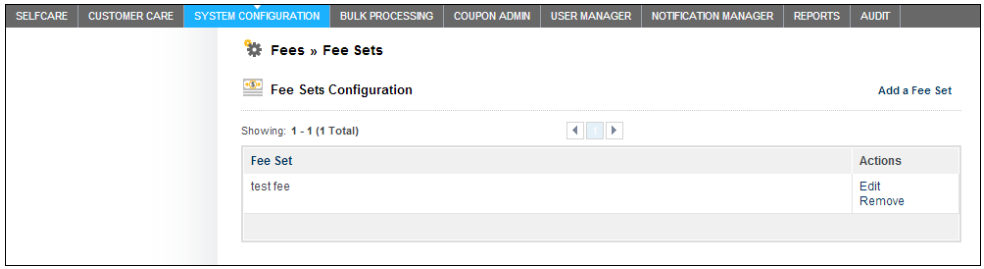
Add at least one fee type.

Task

1. Click **System Configuration** and select **Fees**.
2. Select **Fee Types**.
3. Click **Add Use Case Configuration**.
4. Enter all required information.
5. Click **Save**.

Fee Sets

Fee sets are incremental: agents can apply as many as three fee sets to a single customer, customer type, or organizational unit. For each fee type, fee sets that apply to a customer have the highest priority, followed by fee sets that apply to the customer type, and fee sets that apply to the organizational unit.



Limits

Financial limits are evaluated during transaction processing. During transaction initialization, payer and payee limits are checked individually.

Agents can configure limits through limit sets, which are attached to customers, payment instruments, and wallets.

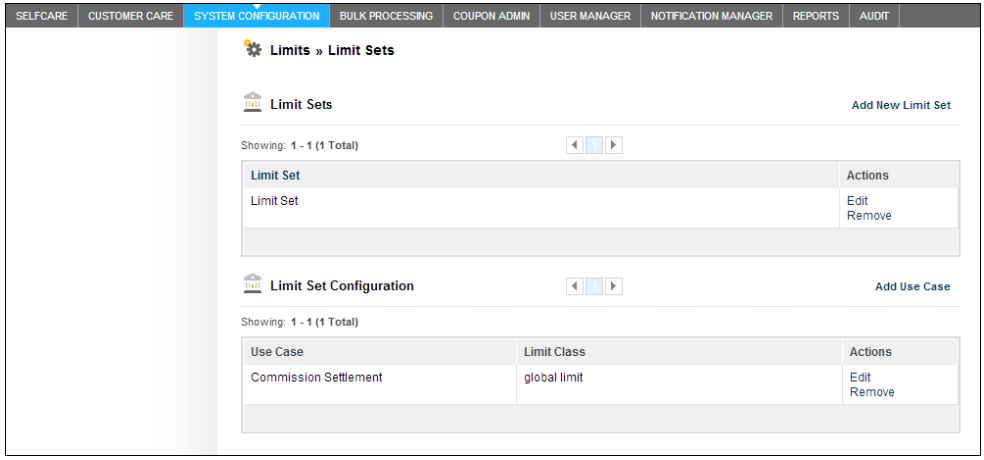
Limit Sets

The Limit Sets option lets agents link limit sets to consumers, payment instruments, and wallet entries. Agents can define a limit set in which there is a global spending limit for a consumer, for example, $300 per day, independent of the payment instrument being used. Defining limit sets for consumers complies with many banking regulations and reduces risk.

Agents can define the risk category for a consumer type. Risk categories are rules that determine limit sets and roles for a consumer type. You can set the consumer type when you register the consumer.

- If no limit set is defined for a consumer, the limit set that is defined for the consumer's risk category is used.
- If no risk category is configured for the consumer, the risk category that is defined for the consumer type is used.

- If no limit set is defined for the risk category, the limit set that is defined for the consumer's organizational unit is used.



Adding Limit Sets

Agents can link limit sets to consumers, payment instruments, and wallet entries.

Prerequisites

Add at least one limit class.

Task

1. Click **System Configuration** and select **Limits**.
2. Select **Limit Sets**.
3. Click **Add New Limit Set**.
4. Enter the name of the limit set.
5. Click **Save**.
6. In the Limit Set Configuration section, select a use case.
7. Select a limit class.

Adding a Use Case to an Existing Limit Set

Use cases are linked to limit sets.

Prerequisites

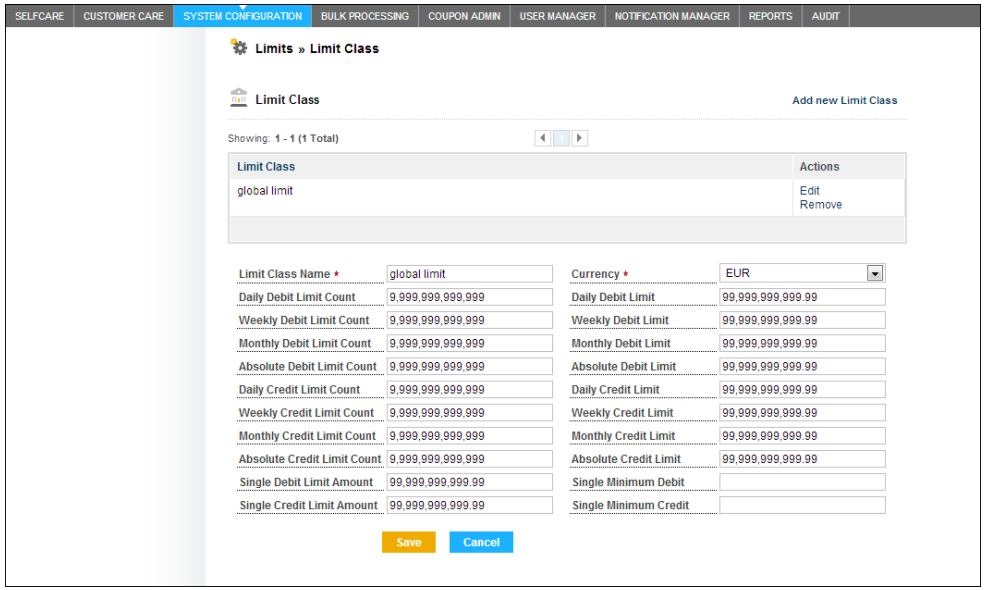
Add at least one limit set and one limit class.

Task

1. Click **System Configuration**, and select **Limits**.
2. Select **Limit Sets**.
3. In the Actions column, click **Edit** for the limit set you want to add.
4. Click **Add Use Case**.
5. Select a use case.
6. Select a limit class.

Limit Classes

Agents can use the Limit Class option to configure daily, weekly, or monthly limits, and total debits and credits. Agents can set limits on the number of transactions, on the total sum of transaction amounts, and they can configure a minimum and a maximum amount for a single transaction.



Field	Description
Limit Class Name	Name of the limit class.
Daily Debit Limit Count	Maximum number of debits per day.
Weekly Debit Limit Count	Maximum number of debits per week.

Field	Description
Monthly Debit Limit Count	Maximum number of debits per month.
Absolute Debit Limit Count	Maximum number of total debits.
Daily Credit Limit Count	Maximum number of credits per day.
Weekly Credit Limit Count	Maximum number of credits per week.
Monthly Credit Limit Count	Maximum number of credits per month.
Absolute Credit Limit Count	Maximum number of total credits.
Single Debit Limit Amount	Maximum debit amount per single transaction.
Single Credit Limit Amount	Maximum credit amount per single transaction.
Currency	The currency for which limits are defined.
Daily Debit Limit	Maximum debit amount per day.
Weekly Debit Limit	Maximum debit amount per week.
Monthly Debit Limit	Maximum debit amount per month.
Absolute Debit Limit	Maximum absolute debit amount.
Daily Credit Limit	Maximum credit amount per day.
Weekly Credit Limit	Maximum credit amount per week.
Monthly Credit Limit	Maximum credit amount per month.
Absolute Credit Limit	Maximum absolute credit amount.
Single Minimum Debit	Minimum debit amount per single transaction.

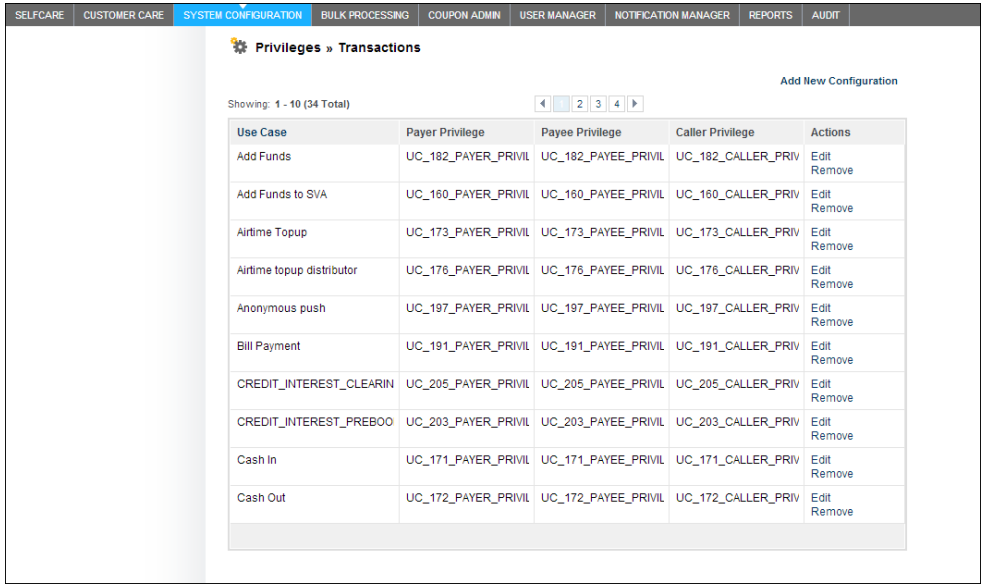
Field	Description
Single Minimum Credit	Minimum credit amount per single transaction.

Privileges

Use the Privileges option to define the required privileges for dual approval. For example, use the Create Customers option to create a new customer-type approval configuration. Depending on the customer type, you can configure maker-checker privileges.

Use these options to configure privileges:

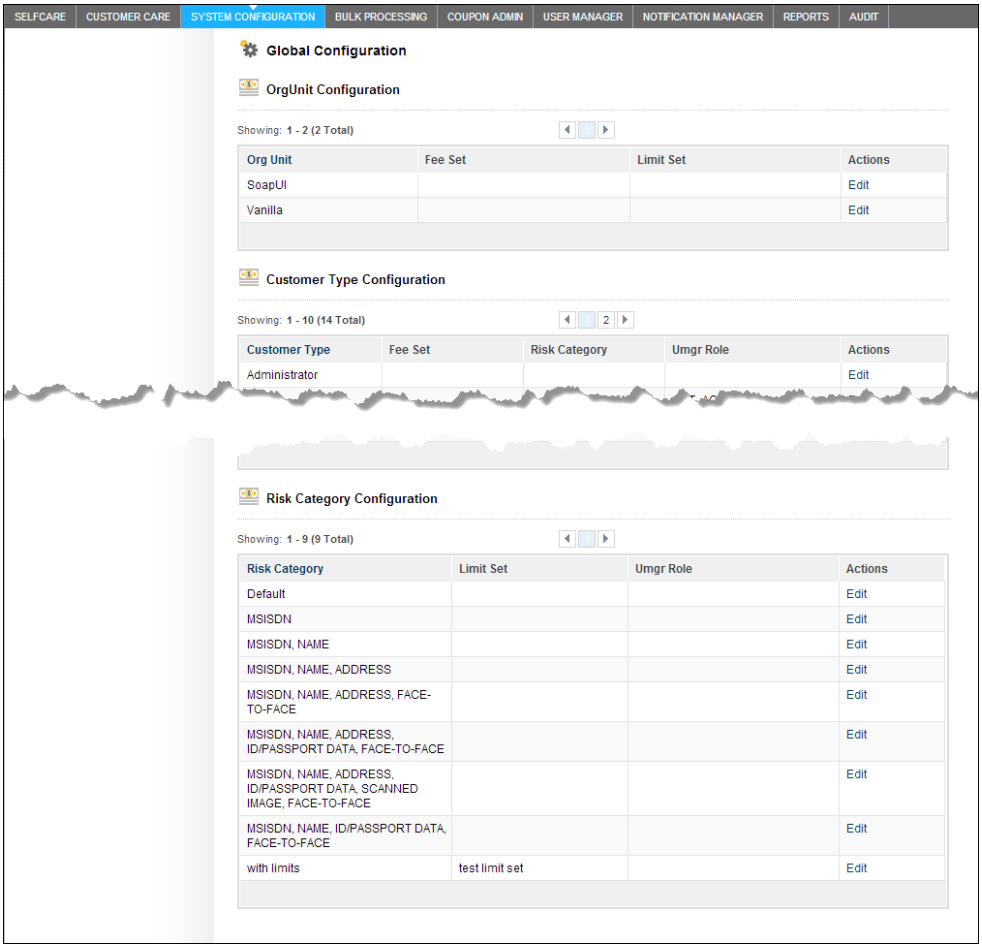
- **Create Customers –** create new customer-type approval configuration.
- **Add Wallet Entries –** create a new wallet approval configuration. For security reasons, consumers and agents might not have permission to directly activate a new payment instrument. In this case, you can either search for, then approve or reject, pending add-to-wallet operations, or see the list of pending wallet operations when you are editing a customer.
- **Transactions –** create a new transaction approval configuration that depends on the transaction type configured.
- **File Types –** create a new file-type approval configuration that depends on the file type configured. For example, you can configure maker-checker validation for deactivating a customer file type, or for cancelling transactions.



Global Configuration

The Global Configuration option displays the current configuration settings for the organizational unit, customer type, and risk category. Global configurations are created during

initial setup. You cannot create additional configurations; you can only edit existing configurations.



Dual Approval

Dual approval is configured during the initial system setup; however, agents can change or add a new configuration anytime. Dual approval is a system-level configuration, so changes can affect many dual-approval processes, such as customer registration, wallet entries, and transactions. Use the Find Pending Approval option to see any pending tasks associated with the dual-approval configuration.

Configuring dual approval might include creating new privileges, assigning privileges to certain roles, and using these privileges to configure dual approval. For each service that supports dual approval, the system defines the MAKER privilege to start or initiate the service, and the CHECKER privilege to complete or approve the service. Additionally, there is an

option that allows users to directly execute services, bypassing the dual-approval process; this requires the EXECUTE privilege.

Restrictions and Restriction Groups

Use the Restrictions option to configure rules that restrict transaction processing for payers and payees, based on their transaction history. A restriction first determines whether it applies to the current transaction, then it accumulates historical transaction data and checks for restriction violations.

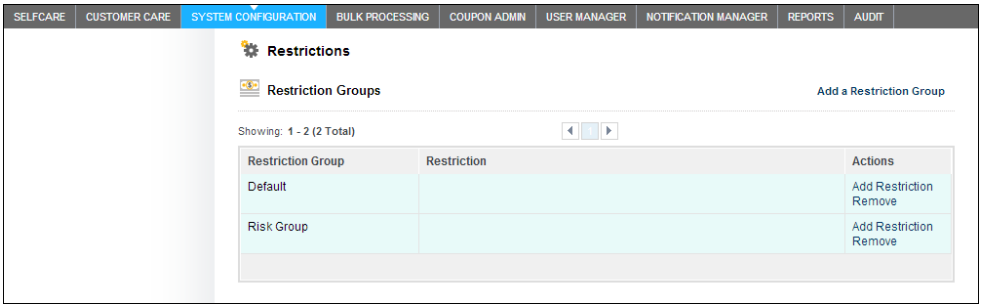
Agents can configure restrictions that apply to all customers in an organizational unit and risk category, but not for individual customers; for example, restrict customers to only one person-to-person payment per day.

Each restriction has a set of rules that define it:

- Minimum and maximum transaction amounts for the current restriction
- Time frame such as minutes, hours, days, weeks, and years
- Maximum number of transaction per time frame
- Maximum transaction sum per time frame
- Currency rules for converting currencies from the standard that is used by one country to the standard that is used by another country

Restriction rules are linked to restriction groups, and always include the currency.

Note: If a restriction group does not exist, you must create one before adding a restriction and a restriction rule to it.



Adding Restriction Groups

Agents can create restriction groups, assign restrictions to them, and apply them to organizational units and risk categories.

1. Click **System Configuration**.
2. Select **Restrictions**.
3. Click **Add a Restriction Group**.
4. Enter the required information.

5. Click **Save**.

Adding Restrictions

Add restrictions to restrict transaction processing for payers and payees.

Prerequisites

Add at least one restriction group.

Task

1. Click **System Configuration**.
2. Select **Restrictions**.
3. In the Actions column, click **Add Restrictions**.
4. Enter the required information.
5. Click **Save**.

Adding Rules

Agents can configure rules that restrict transaction processing for payers and payees, based on their transaction history.

Prerequisites

Add at least one restriction group.

Task

1. Click **System Configuration**.
2. Select **Restrictions**.
3. In the Actions column, click **Add Restrictions**.
4. Click **Add a Rule**.
5. Select the currency.
6. Click **Save**.

Service Packages and Terms and Conditions

Service packages enable features for a specific customer type or class. Customers can inherit roles and privileges through service packages, as well as through their customer type and risk category. A service package is valid for an organizational unit and customer type. Service packages are assigned when a consumer is registered, either self-assignment in the portal, or through a call-center agent. Some service packages can be assigned multiple times to a single customer.

Service packages are attached to terms and conditions, which are either optional, such as a newsletter, or mandatory. Terms and conditions can be added to a service package; therefore, customers must accept a service package to proceed. Each term and condition has an optional

date range. Customer must accept terms and conditions that have a date range, which may include the current date, or be opened ended.

Credential Policies

Credential policies define the rules for composing credentials, such as the minimum and maximum length of a password or PIN. Credential policies also define the number of times a credential can be used, the number of allowable authentication failures, and the time range during which the credential is valid. You can add new policies, and edit and remove existing policies.

Field	Description
Customer Type Policy	
Customer Type	The type of customer: consumer, merchant, or support agent.
Credential Type	The type of credential: password or PIN.
Credential Policy	Defines the rules for how a credential must be composed, such as the minimum and maximum length of a password or PIN.
Policy	
Name	Name of the policy.
Min	Minimum number of characters allowed for the credentials.
Max	Maximum number of characters allowed for the credentials.
Block Size	Number of consecutive characters or numbers that cannot be used, such as 1234, aaaa, abcd, and so forth.
Block Timeout (minutes)	Number of minutes the account is locked when the block threshold is exceeded.
Block Threshold	Number of unsuccessful log-in attempts.
Password Retention	Password history, such as the number of passwords that are stored and how frequently old passwords can be reused.
Expiry (days)	Number of days before a password must be changed.
Temporary Expiry (days)	Number of days before the system generated credentials expire. If left blank, temporary credentials use the value of Expiry (above).
Unused (days)	Number of days before an unused credential expires. Once it expires, it cannot be used.
Credential Pattern	
Regular Expression	Sequence of text characters that specifies a set of strings. Typical expressions are `/d+` for digit-only credentials.
Match	Indicates whether the credential must match the regular expression.

Field	Description
Message Template	Defines the message template key that can be used to inform the customer when an entry fails the new credential.

Adding a Customer Type Policy

Agents can add a customer-type policy to map to a credential policy.

1. Click **System Configuration**.
2. Select **Credential Policies**.
3. Click **Add Customer type policy**.
4. Select the required information from the menus.
5. Click **Save**.

Adding a Credential Policy

Add a credential policy to define the rules for how to compose credentials, such as the length of a password or PIN, and how long credentials are valid.

1. Click **System Configuration**.
2. Select **Credential Policies**.
3. Click **Add policy**.
4. Enter the required information.
5. (Optional) Click **Add Credential Pattern**, enter the required information, and click **Save**.
6. Click **Save**.

Adding a Credential Pattern

Agents can add one or more credential patterns to further restrict a credential policy.

Prerequisites

Add at least one credential policy.

Task

1. Click **System Configuration**.
2. Select **Credential Policies**.
3. In the Credential Policies section, click **Edit** for the policy to which you want to add a credential pattern.
4. Click **Add Credential Pattern**.
5. Enter the required information.
6. Click **Save**.

Interest Configuration

Interest Configuration is used to apply interest to balances on stored value accounts (SVAs), which is similar to how interest is calculated for regular bank accounts. Interest is calculated for each SVA, based on the balance at the end of a calculation period. You can configure various interest schemes, and group them into products; products can be mapped to an SVA, based on the organizational unit, customer type, and payment instrument type.

Interest is calculated for SVAs based on the balance-tracking method. The interest is calculated on interest-balance snapshots, which use the calculation method and scales that are configured in the interest product. For each calculation, two transactions are created that move money between the clearing accounts, in preparation for the clearing of the interest. Therefore, interest calculations are cleared by triggering the transaction to move the interest from the clearing accounts to the customer's SVA.

If an SVA is created during an interest-calculation period, and it has an applicable interest product, its balance snapshots are created for the complete calculation period. For example, if the calculation period is one month, and a new SVA is created on the 15th of the month, entries for the new SVA are created from the 1st of the month up to the current snapshot date. Therefore, 14 entries with a balance of zero are created, and 1 entry with the current balance is created.

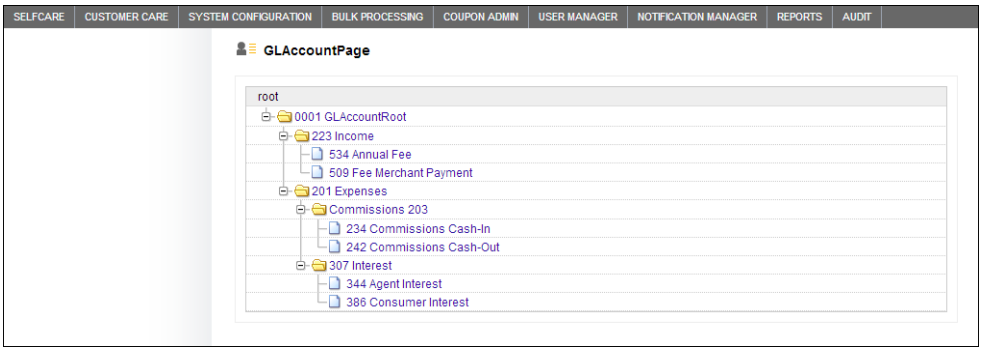
GL Account Tree

The general ledger (GL) account tree can replicate the GL account structure of an enterprise.

Agents can use different GL account types to form a GL account tree:

- **Parent only –** use for aggregation, and to create the required structure. Do not assign this account type to payment instruments, fee beneficiaries, or interest calculation accounts. Other GL account types can also act as parent accounts.
- **Regular –** use for standard accounting/reporting purposes, and assign to a consumer SVA.
- **Expense/Income –** use when the GL account reflects both an internal GL account and a revenue or expense account that is settled via an account management system—that is, a fee account. This account type has a bank account attached to it, and is used for settlement transactions.

The Administration Portal is a flexible way of creating a GL account structure. You can add nodes at any level, except the root, and the tree can have any number of levels. You can delete a node only if it is unassigned, but you can deactivate a node anytime.



Bulk Processing

You can use the Bulk Processing option to upload customer data and location-based points of interest (POI) into either comma-separated value (CSV) or XML file formats. Depending on your assigned privileges, files are either uploaded for confirmation or processed immediately.

Your assigned privileges also determine the bulk-file formats they are allowed to upload:

- **Customer registration –** contains the equivalent data supported by the Customer Registration option.
- **Customer deactivation –** is supported by listing customer identifications. This is the same file used for registering customers; however, you must provide the actual IDs of the customers being deactivated.
- **Points of Interest - CSV format –** allows you to upload or replace multiple points of interest or addresses for a consumer into a comma-separated values file.
- **Points of Interest - XML format –** allows you to upload or replace multiple points of interest or addresses for a consumer in XML format.

The Confirm File option allows you to search for pending bulk processes to approve or reject. You can approve or reject a pending file by clicking the file name to view the details of the file. Use the History option to search for files that have been approved or rejected. All fields are optional, and if any are blank, you see the full list of pending files. You can also download files for further inspection.

Customer XML File Preparation

Uploading customer files in bulk allows you to upload or replace multiple customers.

All records in a XML file must be for the same object. When processing XML files:

- To create a record, verify that all required fields exist.
- Files must be in UTF-8 format.

Bulk customer registration and deactivation files contain assembled service requests; one complete request per line. The following example creates a consumer; it has been reformatted for readability:

```
<cus:CreateFullCustomerRequest <cus:CreateFullCustomerRequest xmlns:cus=
"http://mobiliser.sybase365.com/money/contract/v5_0/customer"
```

```
origin="SOAPUI" traceNo="1234">

<customer>
  <orgUnitId>0000</orgUnitId>
  <blackListReason>0</blackListReason>
  <active>true</active>
  <test>false</test>
  <displayName>Test Full Customer</displayName>
  <riskCategoryId>0</riskCategoryId>
  <customerTypeId>2</customerTypeId>
  <cancellationReasonId>0</cancellationReasonId>
  <txnReceiptModeId>0</txnReceiptModeId>
</customer>

<identifications>
  <type>0</type>
  <identification>+49987654329</identification>
</identifications>

<addresses>
  <addressType>0</addressType>
  <addressStatus>0</addressStatus>
  <firstName>Test</firstName>
  <lastName>Customer 0815</lastName>
  <street1>Test Street</street1>
  <houseNumber>1</houseNumber>
  <city>Test City</city>
  <zip>12345</zip>
  <country>DE</country>
</addresses>
</cus:CreateFullCustomerRequest>
```

The following example deactivates two customers :

```
<cus:DeleteCustomerRequest xmlns:cus="http://mobiliser.sybase365.com/
money/contract/v5_0/customer" origin="SOAPUI"
traceNo="1234">
  <customerId>500005459</customerId>
</cus:DeleteCustomerRequest>

<cus:DeleteCustomerRequest xmlns:cus="http://mobiliser.sybase365.com/
money/contract/v5_0/customer" origin="SOAPUI"
traceNo="5678">
  <customerId>500005458</customerId>
</cus:DeleteCustomerRequest>
```

Points of Interest CSV File Preparation

Uploading points of interest (POI) files in bulk allows you to create or replace multiple addresses for a customer. The first row in a comma-separated value (CSV) file lists the field names for the object that you are processing. Each subsequent row corresponds to a record in the database. A record consists of fields that are delimited by commas. A CSV file can contain multiple records and constitutes a batch.

All the records in a CSV file must be for the same object. When you process CSV files:

- Bulk processing does not support any delimiter except a comma.
- Verify that all required fields exist.
- Files must be in UTF-8 format.

The POI data header must include:

```
customerid,location type,purge
```

For example:

```
1234567,7,true
```

This adds all the addresses to customer 1234567; addresses are created with a location type of 7, and existing addresses with location type 7 are removed.

In addition to the header, the body of the bulk file should contain:

```
longitude,latitude,name,"extra"
```

name represents the name of the company, for example, Walmart. The value of *extra* is optional; it may include details of the location, such as cross streets or an exact address. For example:

```
-85.206844,35.008629,"Walmart","Skurlock Rd & Cornelison Rd"
-85.217847,34.998767,"Parkridge East Hospital","941 Spring Creek Rd"
```

Note: You can download preconfigured CSV files that contain coordinates for points of interest from *http://www.poi-factory.com/*.

Points of Interest XML File Preparation

Uploading points of interest (POI) files in bulk allows you to create or replace multiple points of interest or addresses for a customer.

All the records in a XML file must be for the same object. When you process XML files:

- Verify that all requried fields exist.
- Files must be in UTF-8 format.

The POI data header must include:

```
<PoiDataHeader customerId="12345467" locationType="7"
purgeExisting="true">
```

This adds an address for customer 1234567; the address is created with a location type of 7, and existing addresses with location type 7 are removed.

In addition to the header, the body of the bulk file should contain:

```
<Poi>
   <shortName>Starbucks Dublin</shortName>
   <longitude>123.309484</longitude>
   <latitude>123.309484</latitude>
   <accuracy>10</accuracy>
   <name>blah</name>
   <street1>blah</street1>
   <street2>blah</street2>
   <zip>blah</zip>
```

```
   <city>blah</city>
   <state>blah</state>
   <country>blah</country>
   <phone>blah</phone>
   <email>blah</email>
   <url>blah</url>
</Poi>
```

Coupon Admin

Use the Coupon Admin option to create and manage coupon types, create and assign coupons, and upload coupon batches. Each coupon has a unique code that identifies the coupon or the coupon type. Agents can upload coupons as a batch, or manually create them in the user interface. If a coupon has a purchase price, you can assign it to a consumer, who must use a payment instrument to purchase the coupon before using it.

Coupon Types

Every coupon has a coupon type. Therefore, before you can create or upload any coupons, you must create a coupon type. A coupon type defines the coupon issuer, name, category, and description.

Menu	Description
Main	Defines the coupon issuer, issue dates, price, code type, and other attributes.
Description	Details of the coupon type in multiple languages and formats. You can edit or delete a description.
Keyword	Searchable tags assigned to the coupon type. **Note:** If the Public check box is not selected, the Keyword menu option is not visible.
Categories	Categories assigned to the coupon type
Generate	Delivers a code that is rendered on a smartphone as a bar code, or delivered as a binary image. A coupon code is checked upon redemption; the code is directly bound to the coupon. Coupons always have a coupon code and a serial number.
Batches	Uploads a batch of coupons to a CSV file format. Each batch must contain one or more coupons, which must belong to the same coupon type. You must manually activate a new coupon batch before you can use it.
Locations	A list of geolocation coordinates for each customer type. Use these geolocations, along with radius information, to prioritize search results for a user. **Note:** If the Public check box is not selected, the Locations menu option is not visible.

Menu	Description
Assign	Assigns coupons to one or more customers, depending on the coupon quantity. Users can search for public coupons using their mobile phones. **Note:** A coupon is not assigned to a customer when the coupon is created.

Creating Coupon Types

Create a coupon type to define categories, labels, and destinations that allow coupons to be searched or browsed by consumers.

1. Click **Coupon Admin** and select **Coupon Types**.
2. Click **Add a Coupon Type**.
3. In the main pane, enter the required information.
   a) (Optional) Click **Active** to activate the coupon for use.
   b) (Optional) Click **Public** to make the coupon available in searches via a mobile device. If Public is not selected, the Keyword and Locations menu options are not visible.
4. Click **Save**.
5. In the left pane, click **Description**.
6. Click **Add Description**, and enter a brief description of the coupon type.
7. Click **Save**.
8. In the left pane, click **Keyword**, and enter comma-separated keywords for the coupon type, then click **Save**.
9. In the left pane, click **Categories**, and assign one or more categories to the coupon type.

Generating Coupons

Coupons can be delivered as a code that is rendered on a smartphone as a bar code, or delivered as a binary image. A coupon code is checked upon redemption; the code is directly bound to the coupon. Coupons always have a coupon code and a serial number.

1. Click **Coupon Admin** and select **Coupon Types**.
2. Click **Edit** for the coupon type you want to generate.
3. In the left pane, click **Generate**.
4. Enter the required information.
5. Click **Save**.

Uploading Batch Coupons

Agents can upload a batch of coupons into a comma-separated value (CSV) file format. A batch must contain one or more coupons, and all coupons in a batch must belong to the same

coupon type. You must manually activate a new coupon batch before consumers can use the coupons.

1. Click **Coupon Admin**, and select **Coupon Types**.
2. Click **Edit** for the coupon type.
3. In the left pane, click **Batches**.
4. Click **Add a Coupon Batch**.
5. Browse for the file, and click **Save**.
6. Click **Main**.
7. Click **Activate** to select it.

Adding Coupon Locations

Agents can maintain a list of location coordinates for each coupon type. Use these locations, along with radius information, to prioritize search results for a coupon.

1. Click **Coupon Admin**, and select **Coupon Types**.
2. Click **Edit** for the coupon type to which you want to add a location.
3. In the left pane, click **Locations**.
4. Enter the longitude and latitude.
5. (Optional) Enter the radius.
6. Click **Save**.

Assigning Coupons

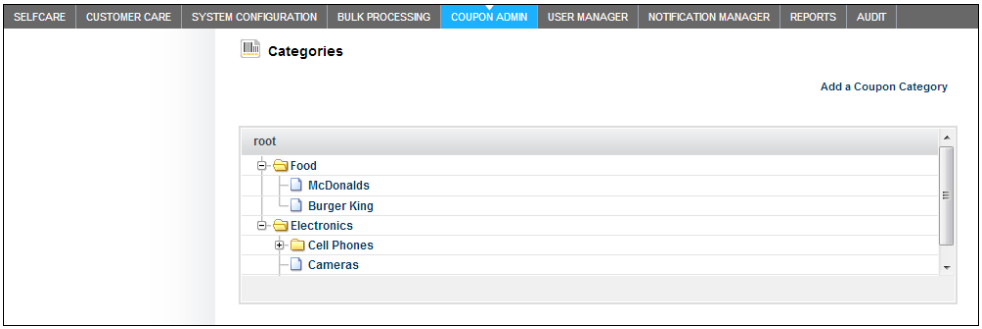
Typically, when agents create a coupon, they do not assign it to consumers. You can assign coupons to one or more consumers, depending on the coupon quantity. Consumers can search for public coupons using their mobile devices.

1. Click **Coupon Admin**, and select **Coupon Types**.
2. Click **Edit** for the coupon type you want to assign.
3. In the left pane, click **Assign**.
4. Select one of these identification types:
   - Mobile Phone Number
   - Customer ID
   - User Name
5. Enter the customer's identification.
6. Click **Save**.

Categories

Consumers can browse coupon categories on their mobile devices to purchase and redeem coupons. Coupon categories are managed globally in a hierarchy. Coupon categories without a parent category are root categories and are listed first. All other categories are shown when

you drill further down into the hierarchy. Agents can add or edit coupon categories, but cannot remove them.



Coupon Search

Use the Coupon Search option to search for coupons that are assigned to a consumer ID and a coupon serial number.

User Manager

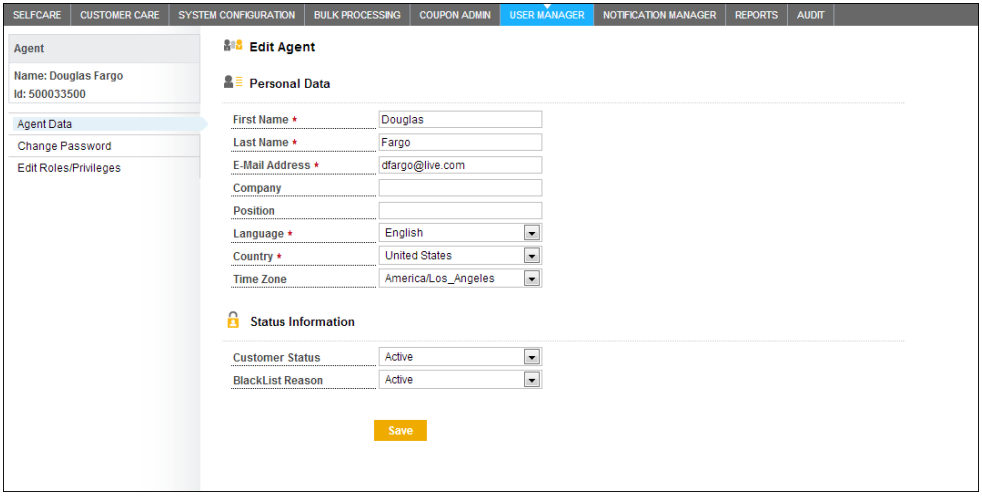
The User Manager option manages back-office or customer-support agents. Agents can add new agents or edit agents' personal data, password, roles, and privileges. This option is available only to those agents who have the user manager role and its respective privileges.

Agents

Agents can use the Agents option to manage back-office and customer-support agents: edit agents' personal data, passwords, roles, privileges, and status information. However, you cannot change an agent's user name, or remove agents from the system. You can grant roles and privileges to other agents. For example, if an agent with the CSTFULL role grants roles and privileges to agent A, agent A can assign these same roles and privileges to other agents.

If you deactivate (remove) an agent, the agent does not appear in the search results. You must select a reason when you deactivate an agent.

Note: Deactivation is permanent. If you want to block an agent temporarily, use the Black List feature.



Roles and Privileges

Privileges control access to functions in the portals. Agents can have individual roles and privileges. Depending on the agent type and assigned roles and privileges, only parts of the portal may be visible.

For example:

- A first-level agent who is authorized to read customer core data can search for customers and view their information.
- An agent who is assigned the role of notification manager can see only the Notification Manager menu.

You can add, edit, and remove roles and privileges. If a role is mapped to a privilege, you must remove the role before you can remove the privilege. Conversely, if a privilege is mapped to a role, you must remove the privilege first. You cannot edit role names or descriptions.

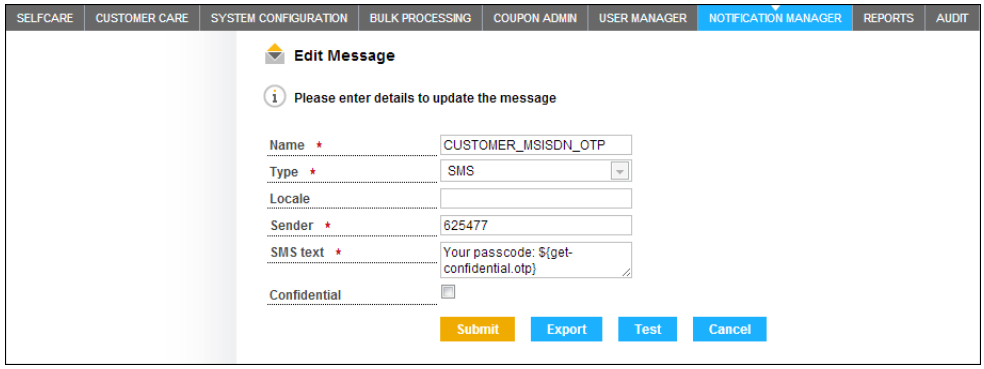
Notification Manager

Use the Notification Manager option to manage SMS and e-mail messages. Notifications of important events are sent to consumers and back-office agents. For example, a message can be sent to consumers when their account balance drops below a given threshold. You can add, import, export, edit, or remove messages.

Messages

Dedicated syntax is used for message placeholders. For example, if you enter `${get.PARAMETER}` in message text, it is treated as a placeholder. If you use `get-confidential` in a message, it is assumed that the message contains sensitive information. Messages that have confidential placeholders are not visible in clear text on the server, and message content is not written to log files; the messages are stored encrypted in the database.

For example, `${get-confidential.otp}` in a one-time password (OTP) notification marks a message as confidential.



You can import or export messages that are in XML format. XML files contain message names, senders' information, template types, and message content. If you export more than one message, the messages are saved in one XML file. If you export a message and then try to import it, the existing file cannot be overwritten.

After you have created, edited, or imported a message, you can test it to verify that it can be delivered. When testing a message with placeholders, provide a replacement value for each placeholder in the message text.

Attachments

You can add or edit attachments, but you cannot remove attachments. If you edit an attachment, it must be the same file type as the original. You cannot view attachments. You can search for an attachment, edit the character set, or attach a new file.

Transaction Template

Use the Transaction Template option to send receipts of financial transactions. Transaction receipts are sent to payers and payees of transactions via e-mail or SMS messages, based on individual consumer preferences and the data in the template.

Field	Description
Use Case	Use case for this configuration.
Template	Template name available in the message gateway.
Payer	Indicates whether this configuration is for the payer or the payee of the transaction.
Active	Indicates whether the configuration is active.
Org Unit	Organizational unit of the consumer who participated in the transaction.

Field	Description
Transaction Status	Status of the transaction about which to send a message.
Test	Indicates whether this template is for testing, or for live transactions.
Error Code	Error code to include in the message. You can use three different values: • 0 – success messages. • *N* – message for a particular error. • -1 – undefined error codes.

Reports

Agents can use the Reports option to generate reports in a variety of formats: PDF, CSV, Microsoft Excel, SAP® Crystal Reports, or RTF.

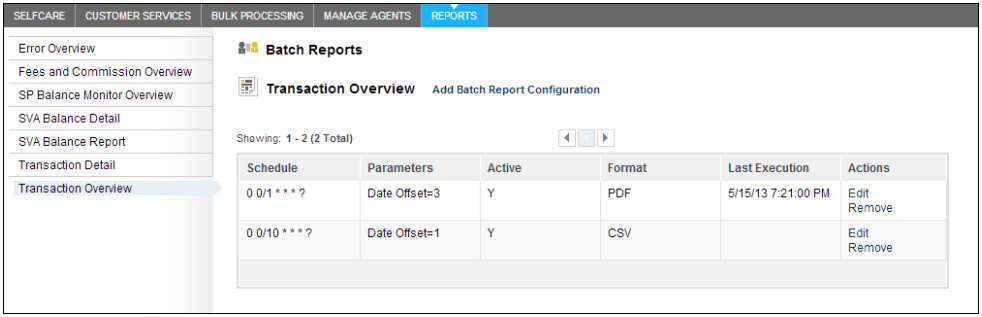
Report Type	Description
Commission	Commission details within a date interval.
Daily Transaction	All transactions performed in a single day.
Error Overview	Failed transactions.
Fees and Commissions Overview	Fees and total commissions in a single day.
SP Balance Monitor Overview	End-of-day balance, aggregated on SVA type.
SVA Balance Detail	End-of-day balance for all SVA accounts in the system.
SVA Balance Report	Balance snapshot of the SVA for the current user.
Transaction	Transaction summary for the currently logged-in agent.
Transaction Detail	Details of all successful transactions from the previous day.
Transaction Overview	All successful transactions from the previous day, aggregated by use case.

By default, report data is generated for the previous day; however, you can use the day offset field to generate data for dates earlier than the previous day. For example, a day offset of 2

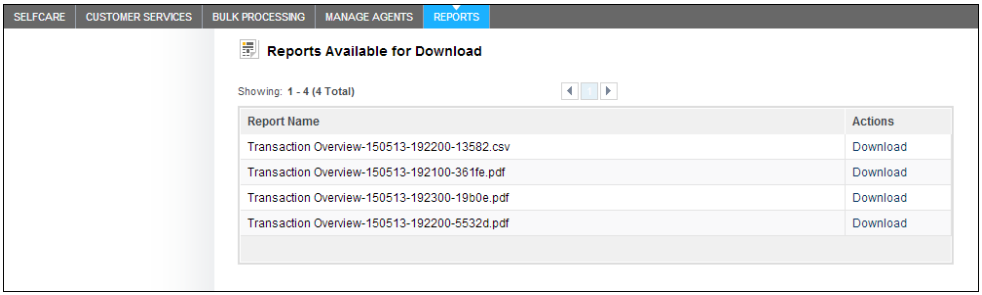
produces a report for the day before yesterday. For some reports, you must specify a date range to generate the data.

You can run online reports and view them immediately. The report data generated in online reports is for the current session only. After viewing the report, you can export it to PDF, CSV, RTF, Crystal Reports, or Microsoft Excel. You cannot edit or remove online reports.

You can configure batch reports to run based on a specified schedule, and in a CSV, PDF, or RTF format. You can also configure multiple batch reports for the same report type. In addition to adding batch reports, you can edit and remove batch reports. You can also disable a batch report by selecting **NO** from the Active list.



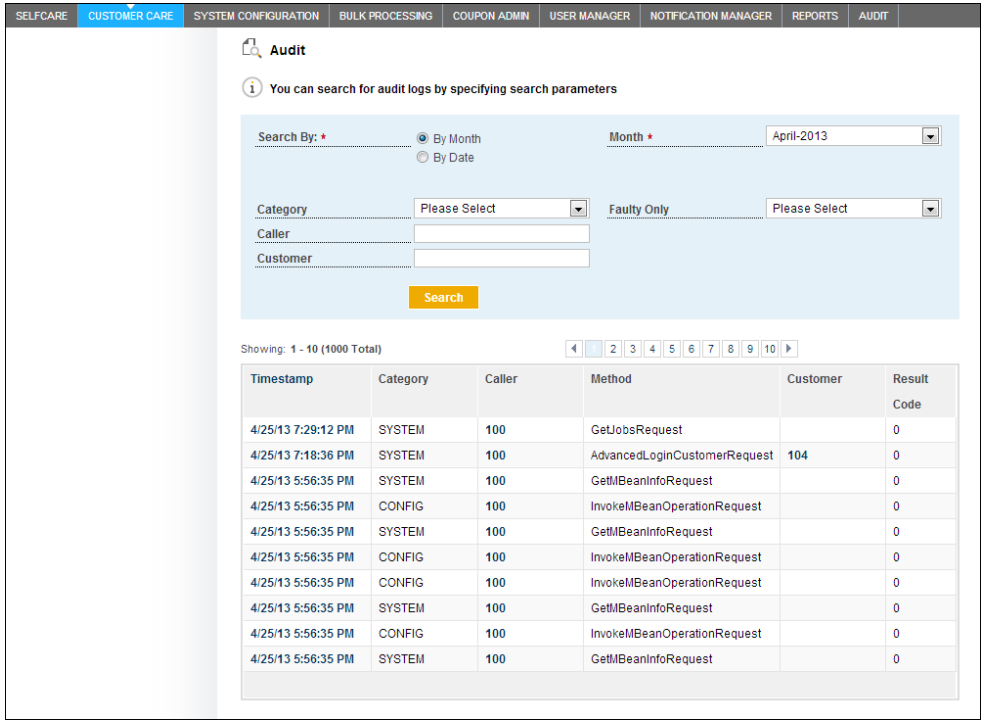
The report names used for batch reports consist of the report type, date, time, and a unique identifier. After a batch report runs, click **Download** in the Actions column to view it.



Audit

Use the Audit option to search for logged versions of security-related events, data read access, and configuration changes. The security audit logs include events that impact the confidentiality, integrity, and availability of the system, for example, failed logins and authorization checks, and changes in critical system parameters. The data read access audit logs include all attempts to access sensitive personal data or personally identifiable information (PII). The configuration change audit logs include changes to any configuration data, such as creating a batch report or removing a limit set.

Click the time stamp to view the details of an audit record.



Consumer Portal

The Consumer Portal includes functions for managing your mobile-banking accounts. For example, consumers can manage their accounts (also known as wallets) with multiple payment instruments, such as bank accounts, credit cards, stored value accounts (SVA), and Offline SVAs. Additionally, you can pay bills, send money to family or friends, and top up airtime.

Getting Started

Get started with the Consumer Portal by creating a consumer account and logging in to the portal.

Signing Up for a Consumer Account

On the Consumer Signup page you can create a consumer account. You need not have a bank account.

Prerequisites

A valid mobile phone number, postal address, and e-mail address.

Task

1. In a browser, navigate to: `http://www.`**serverName**`.8080/portal/`, where **serverName** is the name of the machine on which SAP Mobile Platform Server is running.
2. On the Login page, click **Consumer Signup**.
3. Choose one of the consumer types:
   - Money – a mobile wallet that is used for financial transactions, such as sending money, paying bills, and topping up airtime.
   - mBanking – a mobile wallet with the same features as a Money account that also lets you monitor payments and transfers, using a mobile browser and smart-client channels.
4. Enter all required information.

   The default time zone is Europe/Berlin.

   Note: If you call customer support, you must know your security question.
5. Enter the distorted (CAPTCHA) characters.

   CAPTCHA is an acronym for Completely Automated Public Turing test to tell Computers and Humans Apart.
6. Accept the terms and conditions, then click **Continue**.
7. Accept the terms of the license agreement, and click **Next**.
8. Review your information, and click **Continue**.
   A one-time passcode (OTP) is sent to your mobile phone, but depending on your Do-Not-Disturb settings and geographic location, you might not receive it.
9. (Optional) If you do not receive an OTP, click **Resend**.
   If resend fails, contact customer support.
10. Enter the OTP.
11. To finalize your registration, click **Continue**.
12. To return to the Login page, click **Continue** again.

Logging in to the Consumer Portal

Log in to the Consumer Portal so you can manage your accounts.

Prerequisites

Sign up for a consumer account.

Task

1. Enter your user name and password.
2. (Optional) If you forget your password, click **Forgot Password**, and enter the required information.

Note: A temporary password is sent to your registered e-mail address. After you log in using the temporary password, you are prompted to change it immediately.

3. Click **Login**.

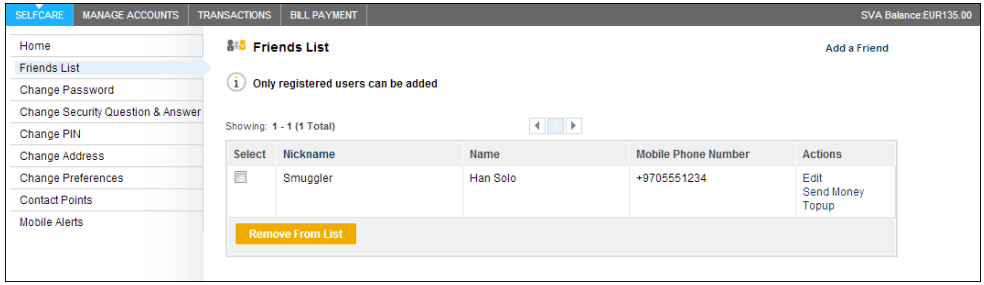
Self Care

The Self Care option provides self-service functions for managing your personal information, such as your security question and answer, password, PIN, address, contact points, mobile alerts, and preferences. You can also manage a friends list, which lets you send money to, and top up airtime for, people on the list.

Friends List

The Friends List option lets you add, edit, and remove other consumers, to whom you can send money or airtime, either online or from your phone. For example, you can pay back a friend or give weekly allowances. Once you set up your friends list, you can send money or airtime immediately from your primary payment instrument to the primary payment instrument of anyone on the list.

Note: You can add registered consumers to your friends list.



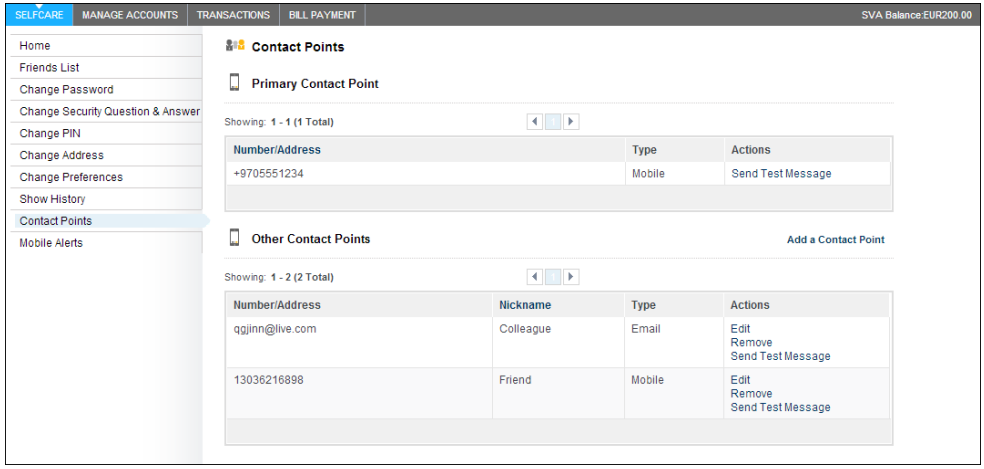
Preferences

The Preferences option lets consumers choose whether they prefer to receive marketing notifications by e-mail, SMS, both, or not at all. You can also specify your time zone.

Contact Points

The Contact Points option lets you add, edit, and remove contact points, and send test messages to any contact point. If you set up mobile alerts, you must choose one or more contact points.

You can have only one primary contact point, which is the registered mobile phone number of your account. You cannot modify or remove the primary contact point; however, you can register more than one mobile device, fax, or e-mail address for the same account, or across multiple accounts. Joint account owners can customize alerts and other features for their devices.



Mobile Alerts

The Mobile Alerts option lets you add one or more alerts from a predefined list, and manage existing alerts. For example, you can receive daily or weekly alerts if your bank account balance falls below a defined threshold. Additionally, you can define quiet periods (also known as Do Not Disturb) within a calendar period, or disable messaging for specific days of the week, for example, weekends.

Note: Mobile alerts are for bank accounts only.

Adding Mobile Alerts

You can add mobile alerts for a bank account.

Prerequisites

A registered user name.

Task

1. Click **SELFCARE**.
2. In the left pane, click **Mobile Alerts**.
3. Click **Add an Alert**.
4. In the Actions column, click **Add** for the alert you want to add.
5. Choose the account for the alert.
6. Select one or more contact points.
7. Click **Save**.

Scheduling Do Not Disturb Times

You can add, edit, or remove Do Not Disturb settings for an existing alert.

Prerequisites

Create at least one mobile alert.

Task

1. Click **SELFCARE**.
2. In the left pane, click **Mobile Alerts**.
3. In the Actions column, click **Edit** for the alert to which you want to add a Do Not Disturb schedule.
4. Enter the required information.
5. Select the Do Not Disturb days and times.
6. Click **Add**.
7. Click **Save**.

Manage Accounts

The Manage Accounts option provides a wallet, which you can use to set up electronic-payment instruments, such as bank accounts and credit cards. Neither a bank account nor a credit card is required. When you create an account, a stored value account (SVA) is set up automatically as your primary payment instrument. However, you can select either a bank account or a credit card as your primary payment instrument. You can use any payment instrument in your wallet to pay bills, but your primary payment instrument is used for person-to-person payments.

If you do not have a bank account or a credit card, funds can be added to your account:

- If another consumer sends money to you using the Send Money or Friends List features.
- Through a pickup code.
- By depositing cash into the account (cash-in) at a distribution center.

If you have a bank account or credit card, you can:

- Add funds from either one.
- Withdraw funds from your SVA and transfer them to your bank account.

Customer support agents can deactivate payment instruments. When a payment instrument is inactive, it appears in the list of accounts, but you cannot edit the information. You can remove an inactive payment instrument from your list of accounts.

Bank Accounts

The Bank Accounts option lets you add one or more bank accounts to your wallet. You can select any bank account in your wallet as your primary payment instrument. You can also edit or remove a bank account.

You can set up mobile alerts for bank accounts in your wallet to notify you of balance summaries, transaction summaries, or threshold summaries. You can also make your bank accounts available for mobile transactions.

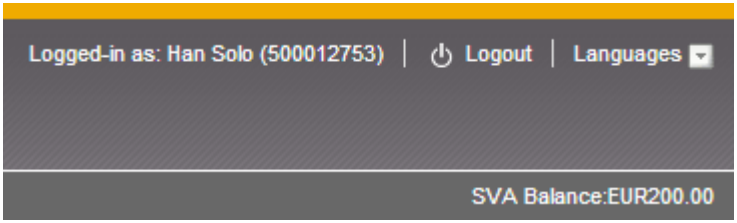
Credit Cards

Use the Credit Cards option to add one or more credit cards to your wallet. You can add and remove credit cards from your wallet, but only a customer-service representative can make changes to your credit card information.

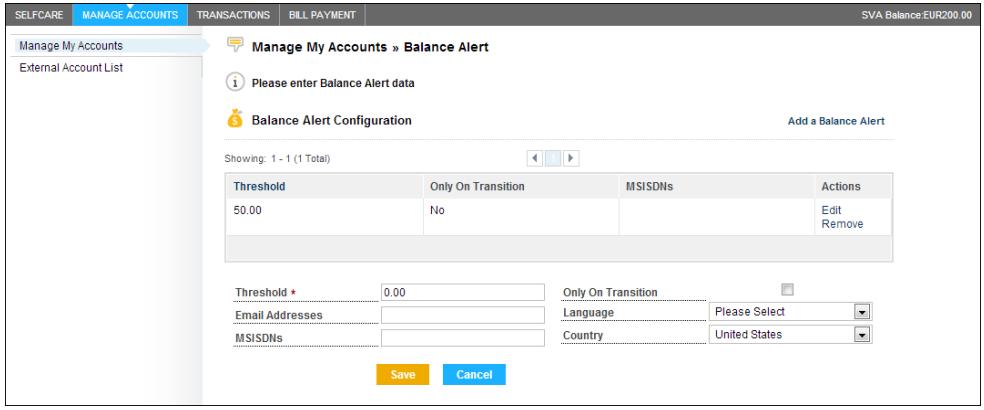
Stored Value Account

Use the Stored Value Account option to add funds to your SVA from a bank account or a credit card in your wallet. You can also transfer funds from your SVA to a bank account in your wallet.

By default, your SVA is set as your primary payment instrument, and the balance starts at zero. The balance of your SVA appears in the upper right of the navigation bar, and in View Transactions under the Transaction menu.



To be notified when the balance falls below a defined threshold, send alerts to an e-mail address, a mobile phone number, or both. If you select Only On Transition, you receive an alert only the first time the threshold is reached; otherwise, you receive an alert each time you make a transaction and your balance is below the threshold.



Offline Stored Value Accounts

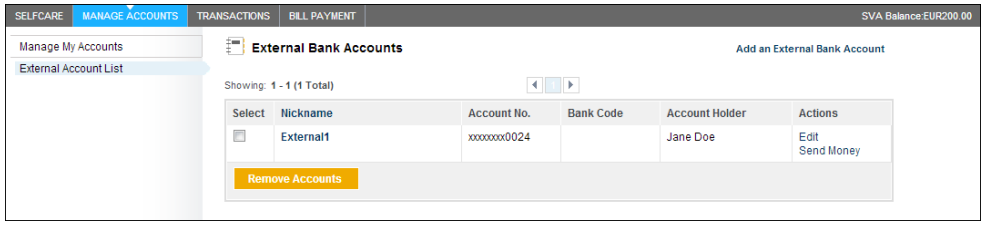
Offline stored value accounts are similar to SVAs, but stored directly on the mobile phone. You can perform the usual operations, such as adding and withdrawing funds, and viewing the current balance and transaction history. Unlike a regular SVA, you can have more than one offline SVA.

Note: By default, your offline SVA serves as the global fall-back account for merchants who own SVAs.

External Accounts

Use the External Accounts option to add this type of payment instruments to your wallet. You can send money from external accounts to third-party bank accounts that you do not own. If an external account is your primary payment instrument, you can send money from it to any third-party account holders.

External accounts are similar to the bank accounts and credit cards in your wallet, because you can use external accounts to pay bills. However, you cannot add money to your SVA from an external account, or transfer money from your SVA to an external account. You can edit and remove an external account from your wallet.

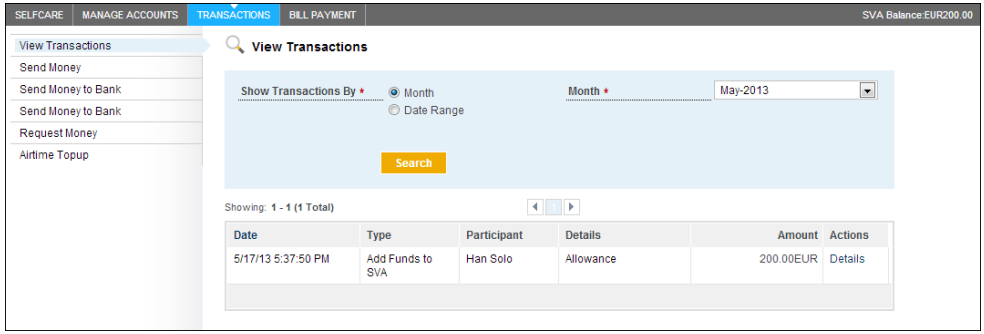


Transactions

The Transactions option lets you view all your transactions, send money, top up airtime for, and request money from, another consumer.

View Transactions

The View Transactions screen displays all airtime topup, bill payment, and SVA transactions. The SVA balance appears anytime money is transferred into or out of the SVA. You can view all money transfers that are requested or sent between other consumers. To view your transactions, search either by month or in a date range.



Send Money

Use the Send Money option to instantly transfer money to another consumer's primary payment instrument. This is similar to the Send Money feature in your friends list; however, in a Send Money transaction, you must enter the recipient's mobile phone number.

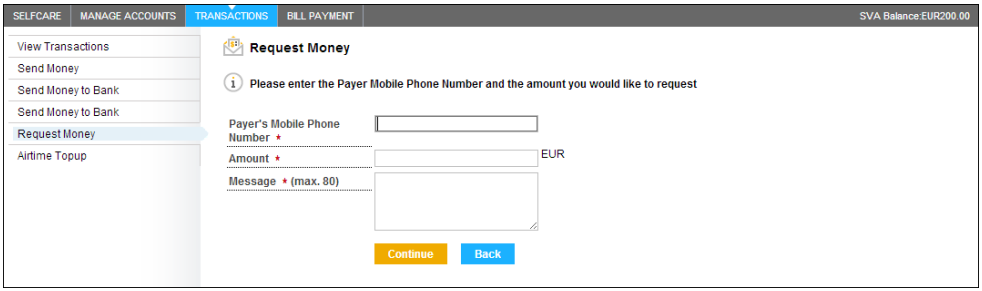
You can also send money to unregistered customers. If you send money to a unregistered customer, a pickup code is sent to them via SMS. The customer can then visit an agent location, and receive the money by providing the pickup code and mobile phone number. The agent may ask the customer to register.

Send Money to a Bank

Use the Send Money to a Bank option to send money from your primary payment instrument to a registered third-party bank account that is not set up as an external account. To send money to a bank, you must know the account number, bank code of the account holder, and the account holder's name.

Request Money

Use the Request Money option to request money from other consumers; they must use the Open Bills feature to send the money. The funds are withdrawn from the sender's primary payment instrument.



Airtime Topup

Use the Airtime Topup option to add airtime minutes to either your mobile phone or another consumer's mobile phone. Purchase airtime minutes using your primary payment instrument. You can see all airtime topup transactions in both the View Transactions and Bill History search results.

To add airtime minutes, select the invoice type (Operator), and enter the amount. If you are adding airtime minutes to another consumer's phone, enter the mobile phone number. If you do not enter a mobile phone number, you add airtime minutes to your own mobile phone.

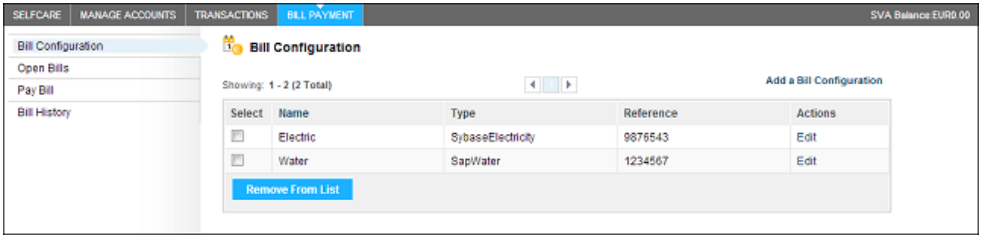
Bill Payment

Bill payments are implemented with invoice types, bill configuration, and bills. The invoice issuer (biller) configures the invoice type. A bill configuration is specific to both the consumer and the invoice type, which creates a personalized reference to the invoice type. The actual bills include the amount due and a reference number, which the biller provides.

Bill Configuration

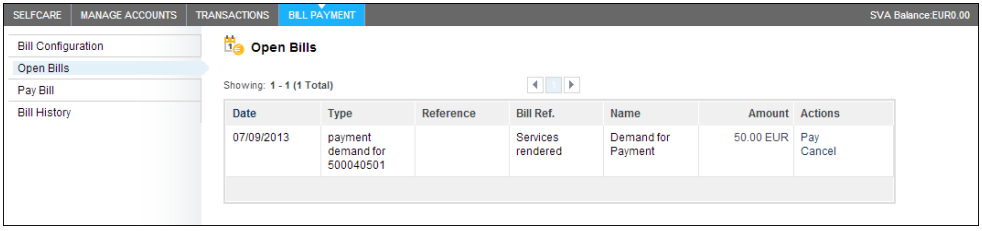
The Bill Configuration option lists your current bill configurations, which are linked to invoice types that store default configuration data for the bills. A bill configuration includes at least one reference number, which the biller provides. The reference number identifies a consumer's account, for example, your account number at the electric company.

To make payments on demand, add a bill configuration using the available invoice types. You can also edit or remove a bill configuration. An invoice issuer can create multiple bills; for example, one issuer can provide bills for electricity and water, and the customer pays separately for each service.



Open Bills

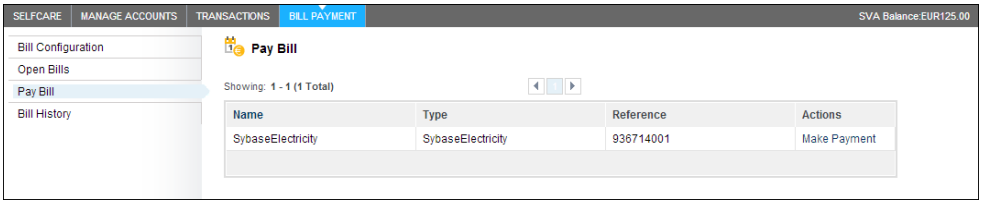
The Open Bills option displays a consumer's active bills, and money that other consumers are requesting. You can pay open bills with any of the payment instruments in your wallet. To pay an open bill, choose the payment instrument you want to use. After paying an open bill, it no longer appears in the list. You can also cancel an open bill at any time. Paid bills appear under Bill History.



Pay Bill

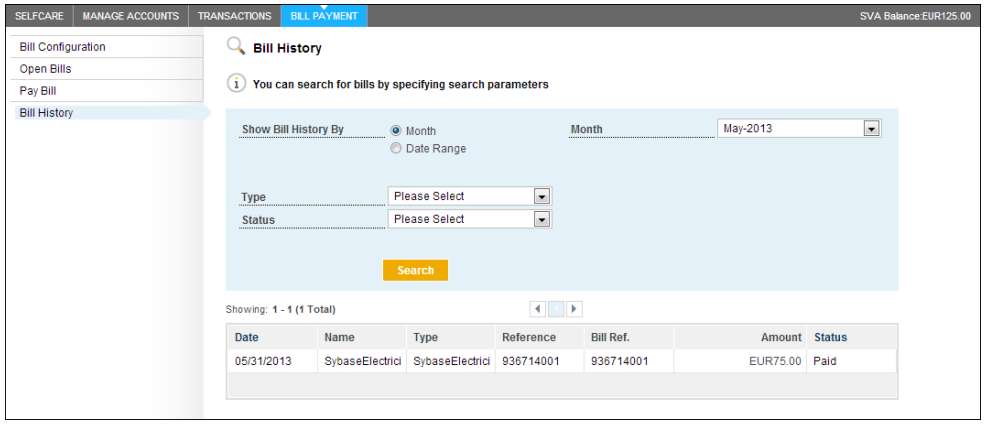
The Pay Bill option displays the bills that are set up under Bill Configuration. You can pay bills immediately, but you cannot set up future payments or edit a payment. To pay a bill, enter the reference number that is defined on the invoice, the amount to pay, and select the payment instrument. The payment is immediately removed from your payment instrument.

The issuer assigns a unique reference number for each bill.



Bill History

Bill History displays bill payments, airtime topup transactions, and money requests. Each bill you have paid appears in Bill History. You can search for bills by month, date range, type, or status; all search parameters are optional.



Partner Portal

Anyone who sells to consumers on behalf of a system provider can use the Partner Portal. Distribution partners are also known as agents. In the portal, agents can manage existing consumers, register new consumers, top up airtime, and validate pending consumer registrations. Agents can also settle commissions and run a number of reports. Depending on assigned roles and privileges, agents can create agent hierarchies.

Logging in to the Partner Portal

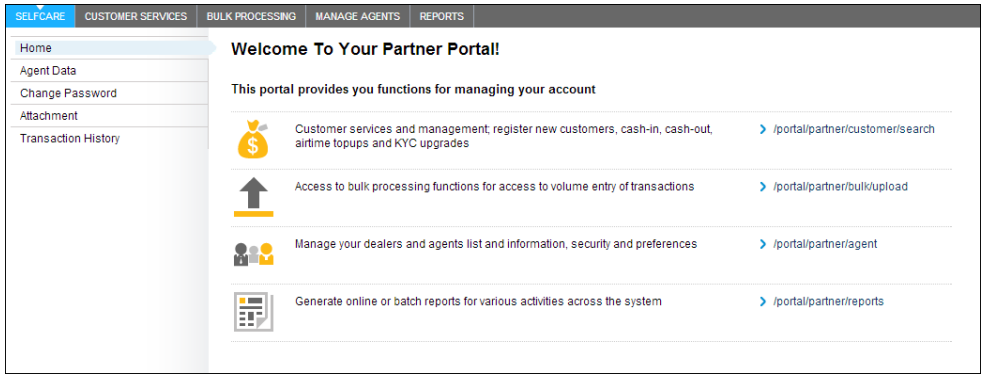
Log in to the Partner Portal to service customers, settle commissions, and run reports.

1. In a browser, navigate to: `http://www.`**serverName**`.8080/portal/`, where **serverName** is the name of the machine on which SAP Mobile Platform Server is running.
2. Enter:
   - User Name – `headquarter`
   - Password – `secret`
3. Click **Login**.

Self Care

The Self Care option includes self-service functions that agents can use to view their assigned user name, and attachments to their user profile. Additionally, agents can view a history of transactions they have run, including those they ran on behalf of other agents, such as cash-out

and top-up airtime transactions. Agents can change their passwords, but cannot edit their user names, which are assigned by administrators.



Customer Services

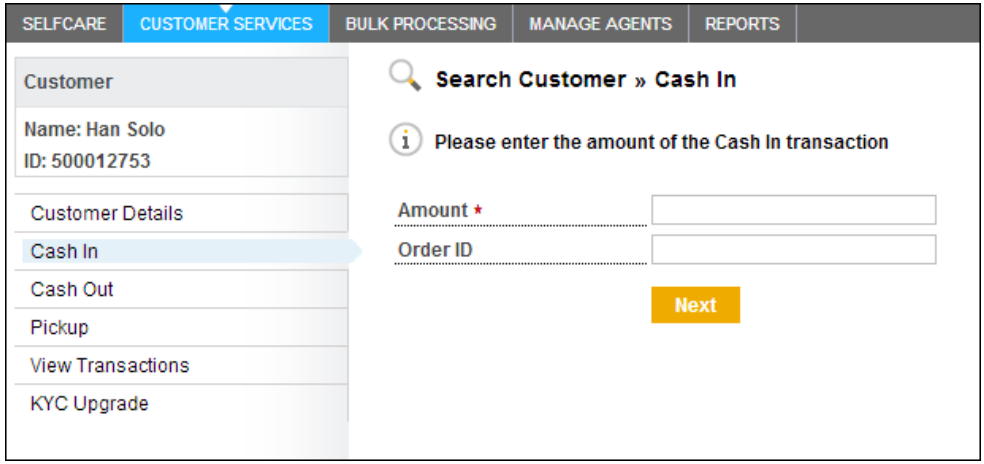
Agents use the Customer Services option to search for consumers who want to transfer or withdraw money through a distribution partner, to view a consumer's transaction history, and to upgrade know-your-customer (KYC) information. Additionally, agents can register new consumers and add airtime minutes. Agents cannot edit consumers' personal data unless they are validating consumers information through the KYC Update feature.

Search Customer

Agents use the Search Customer option to search for consumers who want to transfer money to their stored value account (SVA), withdraw cash from their SVA through a distribution partner, or retrieve a payment through a distribution partner. Agents can view consumers transactions and verify consumers' information for validation.

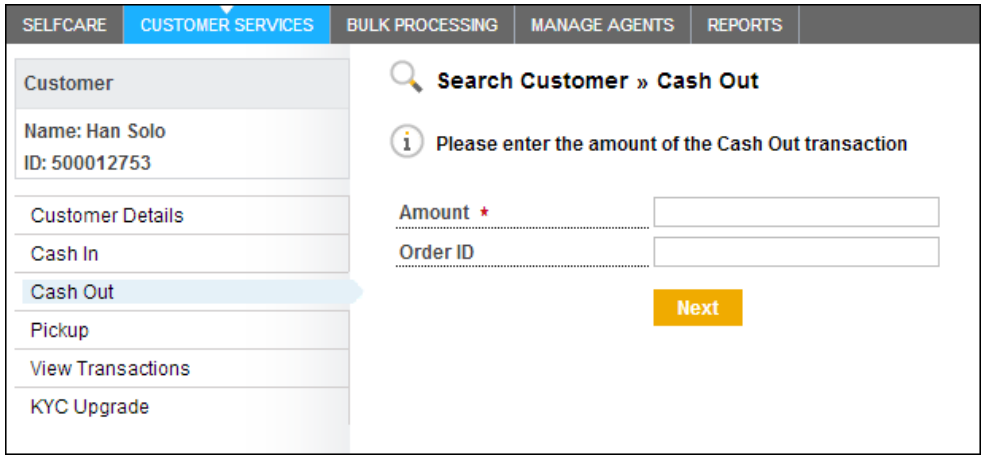
Cash In

The Cash In option lets consumers transfer money to their SVA through a distribution partner. Agents can earn commissions by performing cash-in transactions.



Cash Out

The Cash Out option lets consumers withdraw cash from their SVA through a distribution partner.

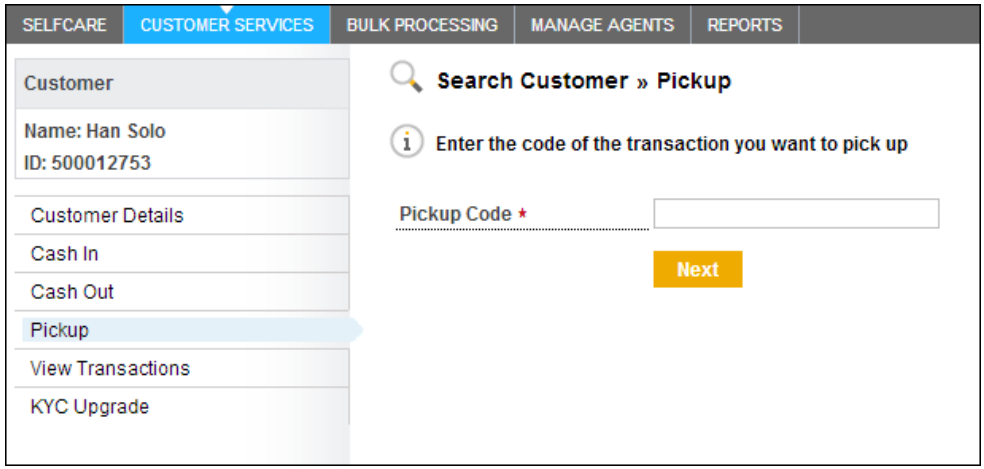


Pickup

Agents use the Pickup option on behalf of unregistered customers who are receiving money from a resistered consumer. A customer receives a pickup code via short message service (SMS). The customer can then visit an agent location, and pick up the money by providing the

pickup code and mobile phone number. When the customer picks up the money, they can register as a consumer.

Note: If the money is not picked up by the expiration date, it is returned to the sender.

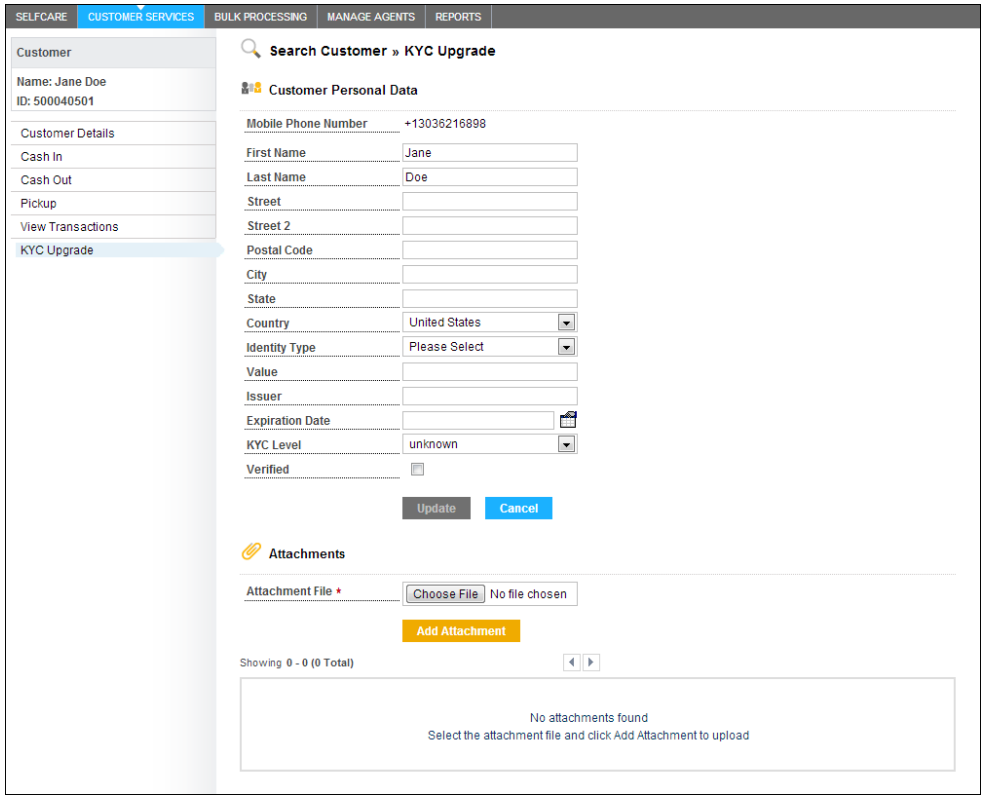


View Transactions

Agents can use the View Transactions option to see consumer transaction details, such as airtime top-up amounts, bill payments, and SVA transactions. You can search by month, or specify a date range.

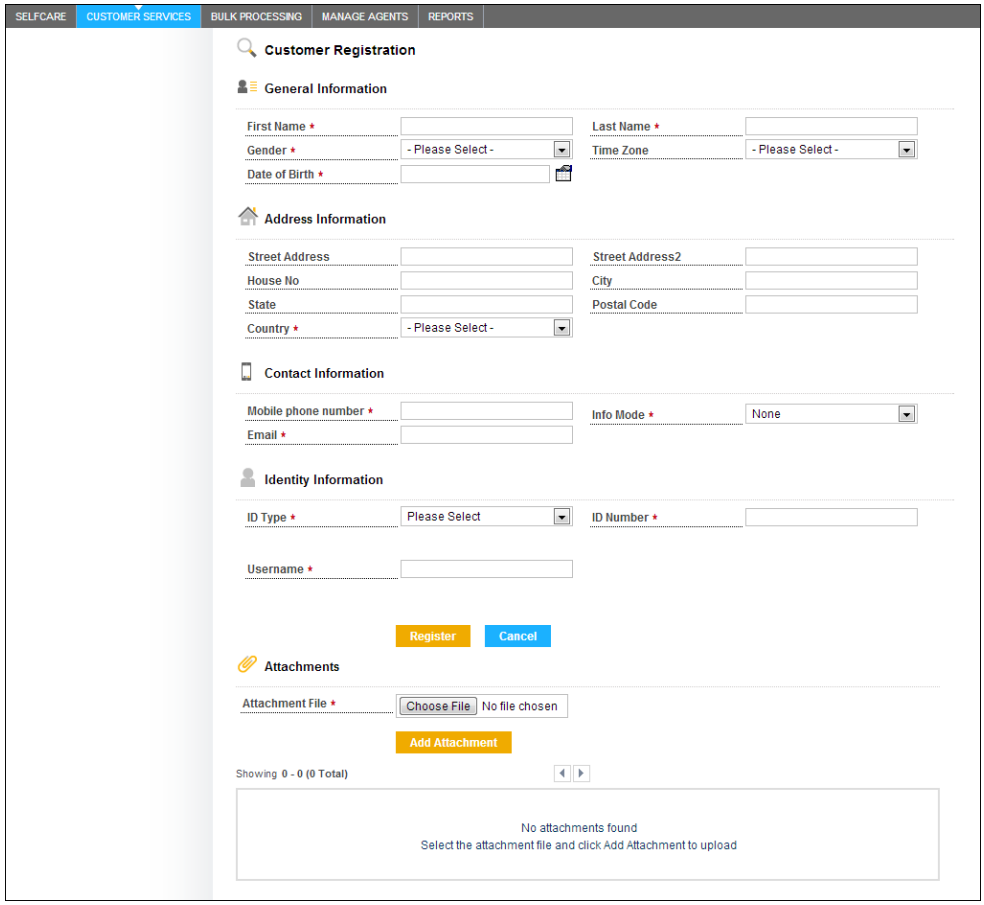
KYC Upgrade

As part of the consumer registration process, additional validation of KYC information might be required before a consumer can be activated in the system or participate in remittance transactions. A back-office agent with sufficient privileges can verify consumer details.



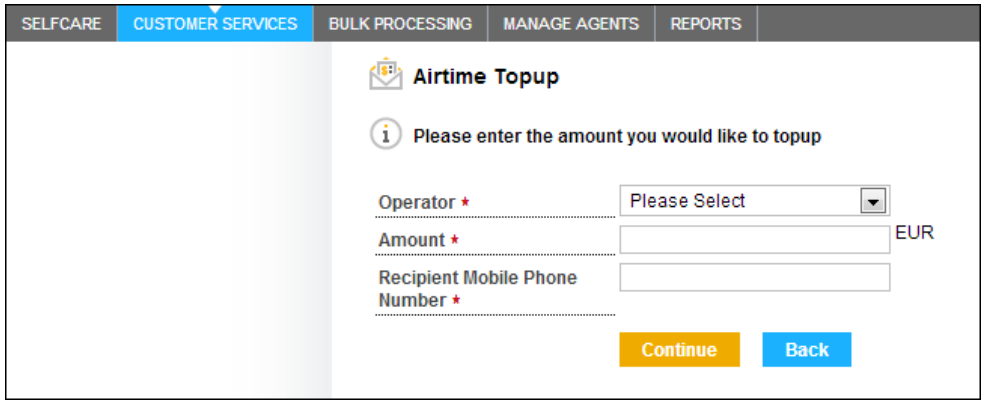
Customer Registration

Customers can go to an agent location and provide a valid ID to register as a Consumer Portal user. A copy of the customer's ID is uploaded and attached to the customer details in the system.



Airtime Topup

Consumers can go to an agent location to top up airtime for either themselves or another consumer. A back-office agent with sufficient privileges can top up airtime for consumers. Airtime is directly credited on a consumer's mobile device.



Bulk Processing

You can use the Bulk Processing option to upload transaction data into XML file formats. Depending on your assigned privileges, files are either uploaded for confirmation or processed immediately.

Your assigned privileges also determine the bulk-file formats they are allowed to upload:

- **Transaction authorization –** contains the equivalent data that is supported by the authorize-transaction Web service. The transaction payer must be either the consumer or the parent-agent.
- **Transaction cancellation –** cancels transactions that either the consumer or the parent-agent initiated.

The Confirm File option allows you to search for pending bulk processes to approve or reject. You can approve or reject a pending file by clicking the file name to view the details of the file. Use the History option to search for files that have been approved or rejected. All fields are optional, and if any are blank, you see the full list of pending files. You can also download files for further inspection.

Transaction XML File Preparation

If agents upload transaction files in bulk, they can authorize or cancel transactions.

All the records in a XML file must be for the same object. When processing XML files, verify that:

- All required fields exist for creating a record.

- Files are in UTF-8 format.

The bulk customer registration and deactivation files contain assembled service requests; one complete request per line. The following example authorizes transactions:

```
<tran:Authorisation xmlns:tran="http://mobiliser.sybase365.com/money/
contract/v5_0/transaction"
    origin="SOAPUI" traceNo="1234" repeat="false" autoCapture="false"
orderChannel="0"
    test="false" usecase="193">

<Payer>
  <identifier type="1">500005460</identifier>
  <paymentInstrumentType>0</paymentInstrumentType>
</Payer>

<Payee>
  <identifier type="1">500005461</identifier>
  <paymentInstrumentType>0</paymentInstrumentType>
</Payee>

<Amount currency="EUR" vat="0">100</Amount>
</tran:Authorisation>

<tran:Authorisation xmlns:tran="http://mobiliser.sybase365.com/money/
contract/v5_0/transaction"
 origin="SOAPUI" traceNo="5678" repeat="false" autoCapture="true"
orderChannel="0"
 test="false" usecase="193">

<Payer>
  <identifier type="1">500005461</identifier>
  <paymentInstrumentType>0</paymentInstrumentType>
</Payer>

<Payee>
  <identifier type="1">500005460</identifier>
  <paymentInstrumentType>0</paymentInstrumentType>
</Payee>

<Amount currency="EUR" vat="0">200</Amount>
</tran:Authorisation>
```

The following example cancels transactions :

```
<tran:AuthorisationCancel xmlns:tran="http://mobiliser.sybase365.com/
money/contract/v5_0/transaction"
     origin="SOAPUI" traceNo="1234">
  <ReferenceTransaction systemId="500007104" type="101">A6WRVN</
ReferenceTransaction>
</tran:AuthorisationCancel>

<tran:CaptureCancel  xmlns:tran="http://mobiliser.sybase365.com/money/
contract/v5_0/transaction"
     origin="SOAPUI" traceNo="5678">
  <ReferenceTransaction systemId="500007105" type="111">ACKZX8</
ReferenceTransaction>
```

```
  <Amount currency="EUR" vat="0">100</Amount>
</tran:CaptureCancel>
```

Manage Agents

Agents can use the Manage Agents option to create new agents and subagents, edit existing agents, and activate or delete pending agents. They can also view agent-subagent relationships in a hierarchy. Agents' privileges determine which portal features they can access.

Agents

Agents act on behalf of merchants or dealers who are creating consumer transactions, for example, cash-in or cash-out transactions. Agents have capital stored in their SVA that is used for consumer transactions. For example, when a consumer withdraws money (a cash-out transaction), an agent’s SVA is credited the same amount.

You can build an agent hierarchy by setting up distribution partner agents or outlet agents, or by creating subagents with assigned agent roles. Subagents work for an agent and can act on the agent’s behalf. Subagents do not have their own payment instrument, and must use their agents' payment instruments. Agents can also hire or acquire other distribution partners to further build their agent hierarchy.

Agents can have individual roles and privileges. Depending on an agent's type and assigned roles and privileges, only parts of the portal may be visible. For example, a merchant agent who has the standard merchant role and no privileges, sees only some features under Customer Services, but all features under Bulk Processing.

You can edit an agent’s personal data, credit and debit limits, password, and status. You cannot change an agent’s role or user name. Active agents cannot be removed from the system. However, customer-service representatives can deactivate agents, which removes them from search results.

Creating Agents

Agents can create other agents to act on behalf of merchants or dealers who create consumer transactions.

1. Click **Manage Agents**, and select **Create Agent**.
2. Enter the required personal information.
3. Choose the **Role ID**:
   - Merchant – can create other agents and dealers. This role is created with an SVA, and offers services such as registration, cash-in and cash-out transactions, and airtime top-up.
   - Merchant agent – can act only on behalf of the merchant. Agents with this role do not have SVAs, cannot create subagents, and can be neither payers nor payees in a transaction.

- Merchant dealer – can create only merchant agents and subagents. Agents with this role do not have SVAs, but can use the merchant's SVA. Daily and monthly limits can be applied to a merchant dealer role, and commissions can be calculated for them.

4. (Merchants only) Set a balance configuration:
   a) Click **Add a Balance Alert**.
   b) Enter the required information for the balance alert.
   c) Click **Save**.
5. Enter the login data.
6. Click **Create**.
7. Add the roles to grant to the agent.
8. Add the privileges to grant to the agent.
9. In the left pane, click **Commission Management**.
10. Enter the required information.
11. Click **Save**.

Creating Subagents

Agents can create subagents to act on their behalf. Subagents can use the agent's payment instruments.

Prerequisites

Create at least one agent.

Task

1. Click **Manage Agents**, and select **Find Agent**.
2. Enter the search criteria, and click **Search**.
3. Select an agent that has the Merchant role.
4. In the left pane, click **Create Sub Agent**.
5. Enter the required personal information.
6. Choose the **Role ID**:

   - Merchant – can create other agents and dealers. This role is created with an SVA, and offers services such as registration, cash-in and cash-out transactions, and airtime top-up.
   - Merchant agent – can act only on behalf of the merchant. Agents with this role do not have SVAs, cannot create subagents, and can be neither payers nor payees in a transaction.
   - Merchant dealer – can create only merchant agents and subagents. Agents with this role do not have SVAs, but can use the merchant's SVA. Daily and monthly limits can be applied to a merchant dealer role, and commissions can be calculated for them.

7. (Merchant only) Set a balance configuration:
   a) Click **Add a Balance Alert**.
   b) Enter the required information for the balance alert.
   c) Click **Save**.
8. Enter the login data.
9. Click **Create**.
10. Add the roles to grant to the subagent.
11. Add the privileges to grant to the subagent.
12. In the left pane, click **Commission Management**.
13. Enter the required information.
14. Click **Save**.

Adding a Balance Alert

Agents can add balance alerts for merchant agents.

1. Click **Manage Agents**, and select **Find Agent**.
2. Enter the search criteria, and click **Search**.
3. Select an agent from the list.
4. Click **Add a Balance Alert**.
5. Enter the required information.
6. Click **Save**.

Activate Agent

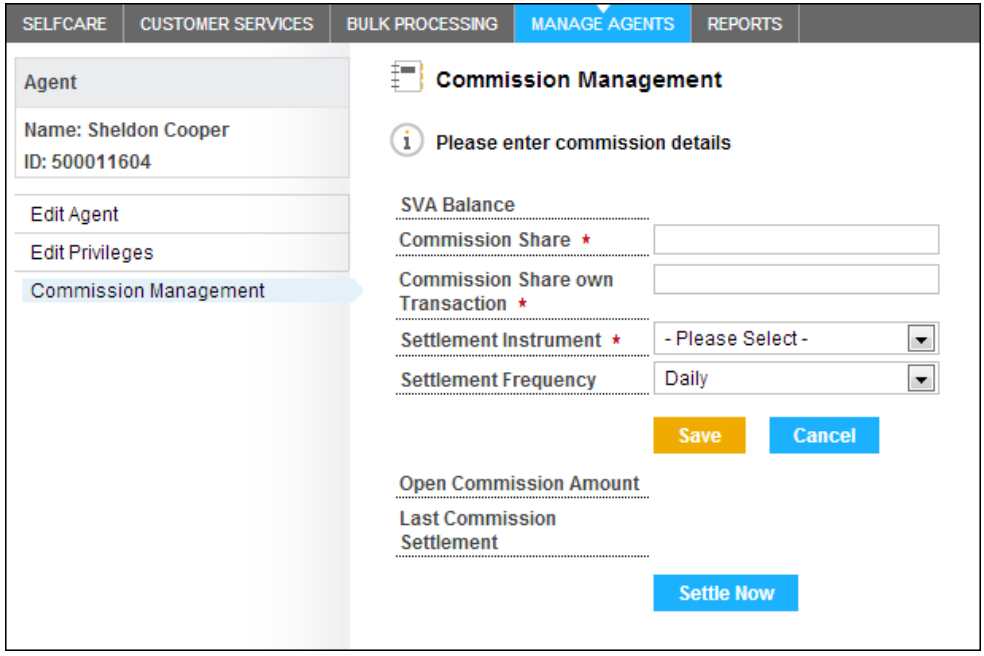
Use the Activate Agent option to activate a pending agent registration. If an agent's status is blocked, or if a merchant dealer created the agent, you must activate the agent. Agents can also delete an agent from the system or edit its information. If no agent registrations are pending, no agents are listed.

Commission Management

The Commission Management option enables distribution partners to pay commissions to their acquired distribution partners (subagents). Commissions are based on transactions that their respective outlets or agents generate. For example, you can earn commissions on cash-in transactions. Commissions are paid either at fixed intervals or on demand.

Commissions are calculated per transaction, and are settled at daily or weekly intervals. Agents can also settle a commission immediately by clicking **Settle Now**. Agents can view the accrued commission amounts and the last commission settlement.

Note: Commission Management is based on the fee-type configuration in the Administration Portal. You must configure the commission fee types in the Administration Portal before commissions can be applied to transactions, and settled in the Partner Portal.



Field	Description
Commission Share	Commission an agent receives from subagents.
Commission Share own Transaction	Commission amount to be passed on to the parent agent. If an agent has no parent, the commission remains with the agent.
Settlement Instrument	Payment instrument used for the commission payout.
Settlement Frequency	Interval at which commissions are settled.

Reports

Agents can use the Reports option to generate reports in a variety of formats: PDF, CSV, Microsoft Excel, SAP® Crystal Reports, or RTF.

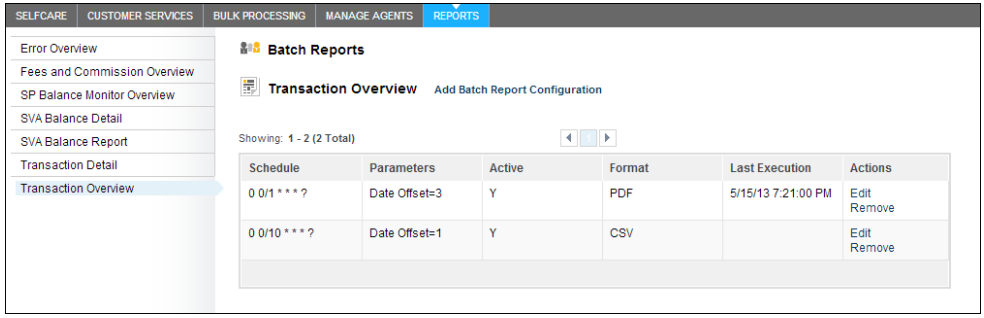
Report Type	Description
Commission	Commission details within a date interval.
Daily Transaction	All transactions performed in a single day.
Error Overview	Failed transactions.

Report Type	Description
Fees and Commissions Overview	Fees and total commissions in a single day.
SP Balance Monitor Overview	End-of-day balance, aggregated on SVA type.
SVA Balance Detail	End-of-day balance for all SVA accounts in the system.
SVA Balance Report	Balance snapshot of the SVA for the current user.
Transaction	Transaction summary for the currently logged-in agent.
Transaction Detail	Details of all successful transactions from the previous day.
Transaction Overview	All successful transactions from the previous day, aggregated by use case.

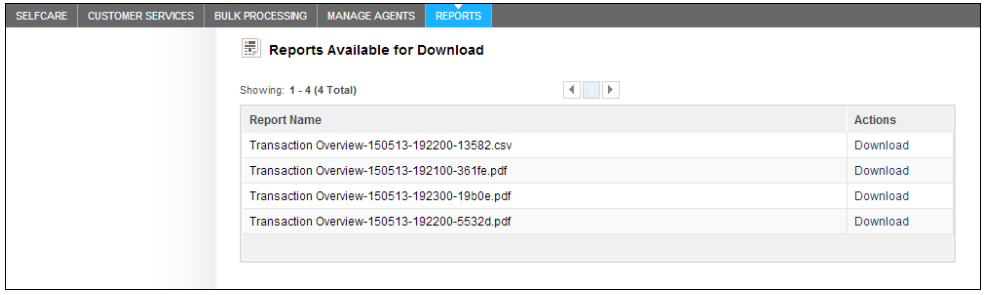
By default, report data is generated for the previous day; however, you can use the day offset field to generate data for dates earlier than the previous day. For example, a day offset of 2 produces a report for the day before yesterday. For some reports, you must specify a date range to generate the data.

You can run online reports and view them immediately. The report data generated in online reports is for the current session only. After viewing the report, you can export it to PDF, CSV, RTF, Crystal Reports, or Microsoft Excel. You cannot edit or remove online reports.

You can configure batch reports to run based on a specified schedule, and in a CSV, PDF, or RTF format. You can also configure multiple batch reports for the same report type. In addition to adding batch reports, you can edit and remove batch reports. You can also disable a batch report by selecting **NO** from the Active list.



The report names used for batch reports consist of the report type, date, time, and a unique identifier. After a batch report runs, click **Download** in the Actions column to view it.



Portal Development and Customization

You can develop new Web applications to interface with SAP Mobile Platform Server, and customize the portal-template Web applications.

Overview

You can develop new Web applications, and customize the portal-template applications to support your mobile-banking needs. To build Web applications, you should understand the basics of Web page design and the Apache Wicket Web page framework, and have a good working knowledge of HTML formatting.

A unified Web application allows you to share page layouts and styles across multiple Web applications. All applications are consolidated into a single Web application archive file (WAR), which:

- Builds on the Apache Wicket framework, focusing on reuse
- Simplifies development and customization
- Increases flexibility and standardization

A single unified Web application allows you to use a consistent approach that is understood and shared across Web applications. For customizing Web applications, it is important to understand:

- Page layout – available types of page layouts and general principals to apply to all pages.
- Application configuration – how to add Web applications to the unified environment.
- Application customization – basic steps for including customized functions in the unified environment.
- Style guidelines – how to apply individual style elements to common HTML components.
- Localization guidelines – how to organize text resources related to localization, and best practices for managing text labels.

An aggregated WAR manages a single Wicket application, which defines an application sign-on page and an application home page.

- If an authenticated Wicket Web session does not exist, users are directed to the application sign-on page, which is defined by `ApplicationLoginPage.class`.
- Once users are authenticated, they are directed to the application home page, which is defined by `ApplicationHomePage.class`.

Because Wicket defines an application-specific home page, the base-application home page identifies roles that are associated with the logged-in user, and redirects the user to the appropriate application home page.

Page Layout

Web application page layout structures are based on underlying Apache Wicket templates.

Web application pages are either unauthenticated—sign-on pages or other pages that are open to anyone—or authenticated application pages.

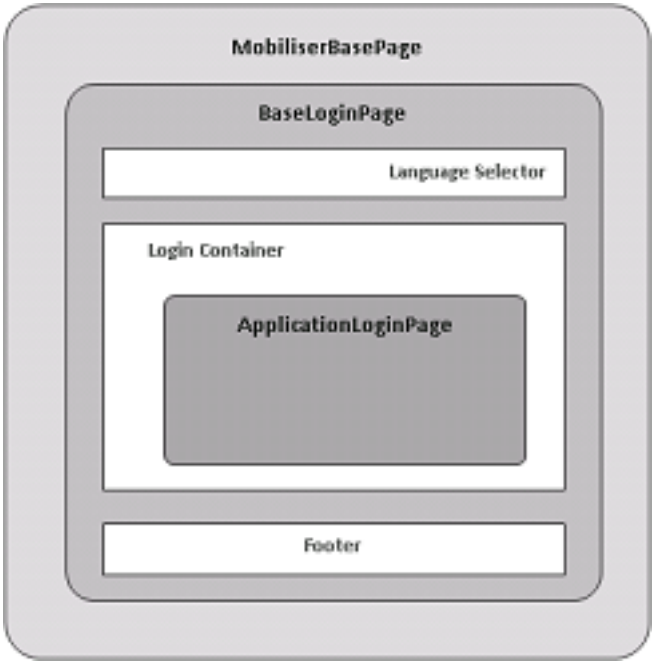
Unauthenticated Pages

Unauthenticated pages present login options.

An unauthenticated sign-on page displays options and information for users to:

- Provide credentials for authentication
- Assign a temporary password
- Chang temporary passwords after successful logins, and before entering authenticated applications
- View global application error pages

Figure 1: Structure of Unathenticated Sign-on Page



The hierarchy of unauthenticated pages mimics that of Wicket pages. The base page for all Wicket Web UI applications is defined by the `MobiliserBasePage` class. The `BaseLoginPage` class extends `MobiliserBasePage`, and provides:

- A language selector for changing UI text labels
- An extension point for the unauthenticated page type
- A standard unauthenticated footer

The classes that define the unauthenticated application pages, and extend `BaseLoginPage` are:

- `ApplicationLoginPage`
- `ForgotPasswordPage`
- `ChangeTempPinPage`
- `TechnicalErrorPage`

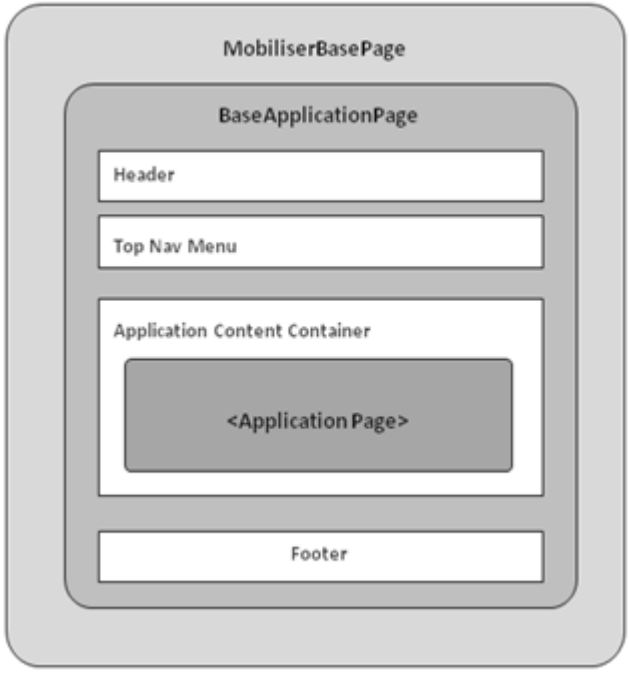
Authenticated Pages

Authenticated pages display options and information about applications.

A consumer registration page is an authenticated page, and is a separate application type. However, to access consumer registration pages, you need have only a standard access role that is granted to all users upon entry to the login page. Consumer registration pages have the same structure as other application pages, without a specific authenticated user Web session.

The hierarchy of an authenticated page replicates how all Wicket pages are formed.

Figure 2: Structure of Authenticated Page



The base page for all Wicket Web UI applications is defined by the `MobiliserBasePage` class. The `BaseApplicationPage` class extends `MobiliserBasePage` and provides:

- Standard header area, which displays application graphics, the application name, and optionally, the logged-on user's information, such as a name. It also contains a logout link and a standard language selector.
- Top navigations menu, which includes global navigation items for all applications. The items shown to users are application specific, and based on users' roles.
- Container for the main application page content.
- Standard authenticated footer that includes session timeout information.

The classes that define authenticated application pages extend `BaseApplicationPage`, and set the basic structure for the application-specific content:

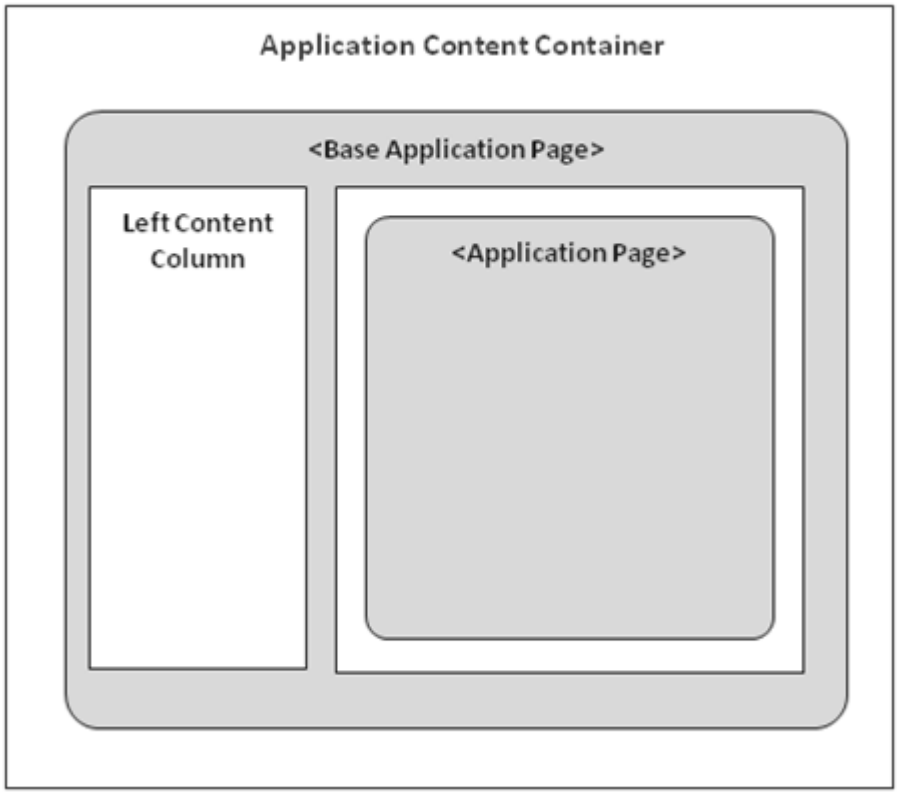
- `BaseConsumerPage` – application content structure for both consumer registration and the Consumer Portal.
- `BaseDistributorPage` – application content structure for the Partner Portal.
- `BaseCstPage` – application content for the Administration Portal.
- `BaseDashboardPage` – base content for the Operations Dashboard Web application.

Application pages allow you to apply specific layouts to their innermost pages:

- An area on the left side of the page (left content column) displays page-context specific information and menus.
- The right side of the page displays the application page content.

Note: Typically, Web applications have maintained the concept of a left content column, which is enforced by application-specific pages. If you write a new application that does not require this layout, the `BaseApplicationPage` class need not enforce this design. The hierarchy of Wicket applications allows application pages to define their own content layout within the application content container.

Figure 3: Structure of Page with Application Content Container



Application Configuration

Configure applications in the runtime environment to expose their functions.

Basic contributions that applications must provide to an aggregated environment are:

- Bookmarkable pages – pages that can be referenced through a name in the Web page URL, rather than the standard Wicket URL format, which is not unique.
- Login privilege (optional) – if an application is an authenticated type, access to it is determined by an exclusive and unique login privilege that is assigned to a logged in user.
- Home page – the initial point of access into the application.

Configure each application as a Spring bean, and define the information above, using a single Spring application-context configuration file. After you declare application beans, list them as relevant properties for the base application bean, for registration. If an application need not be registered in the runtime environment, you may leave it off the list.

Application Configuration Example

In this example, the Spring configuration for an instance of the Web UI restricts access to the Consumer Signup and the Consumer Portal applications only.

```
 ...
<!-- MOBILISER CONTROLLING APPLICATION -->
<bean id = "mobiliserApplication"
  class =
"com.sybase365.mobiliser.web.application.MobiliserApplication">
  <property name = "authenticatedApplications">
   <util:list>
    <ref bean = "consumerPortal" />
   </util:list>    </property>
  <property name = "unauthenticatedApplications">
   <util:list> </util:list>
  </property>
  <property name = "signupApplications">
   <util:list>
    <ref bean = "consumerSignup" />
   </util:list>
  </property>
</bean>
...
<!-- CONTRIBUTING APPLICATIONS -->
<bean id = "consumerPortal"
 class =
"com.sybase365.mobiliser.web.consumer.pages.portal.ConsumerPortalApplica
tion">
 <property name = "bookmarkablePages">
  <util:map>
   <entry key = "/consumer" value =
"com.sybase365.mobiliser.web.consumer.pages.portal.selfcare.Consumer" \>

  </util:map>
 </property>
 <property name="loginPrivilege">
  <util:constant static-
field="com.sybase365.mobiliser.web.util.Constants.PRIV_CONSUMER_LOGIN" /
>
 </property>
 <property name = "homepage"
 class =
```

```
"com.sybase365.mobiliser.web.consumer.portal.pages.selfcare.ConsumerHome
Page" />
</bean>
...
```

Application Objects

Use `com.sybase365.mobiliser.web.application.model` and abstract classes to ensure that applications conform to the Web UI contributing applications object hierarchy.

- `IMobiliserApplication` – interface that represents the basic information required for an application.
- `IMobiliserAuthenticatedApplication` – extended interface for applications that use the common application sign-in page.
- `IMobiliserSignupApplication` – extended interface for the single application providing the user sign-up functions.
- `IMobiliserUnAuthenticatedApplication` – extended interface for applications that do not require authentication.
- `AbstractMobiliserApplication` – base class that allows you to configure bookmarkable pages, and assign a class that represents the application home page.
- `AbstractMobiliserAuthenticatedApplication` – extended base class for applications that require authentication. This class allows you to specify a unique login privilege that represents basic access to the application. The privilege is assigned to individual users, and each user is exclusively assigned to one application.

Application Customization

You can customize existing functions in the Web UI.

The Web UI, which runs outside the OSGi container, uses service client interfaces to access the container. Service clients provide the interface for all services.

Integrating a new service interface requires:

- Spring bean configuration for the refreshable client interface
- Spring bean configuration for the client logic, the service caller
- Methods that access the new functions being used

Different service-client interfaces are provided for back-end authentication, via the built-in system user (client beans defined in the `system-clients-context.xml` file), or by using authentication for the currently logged in user (beans defined in the `self-clients-context.xml` file). The application code determines which service-client interface bean to use, `systemAuthTransactionClientLogic` or `selfAuthTransactionClientLogic`.

You can also customize the application top menu (main navigation) items and left-menu items (context specific menu items) for existing portal applications, or provide new top-menu and left-menu items. To customize menu items:

- Create new top-menu beans that identify text labels, target page Java classes, privileges required to view top-menu items, and whether the items are active.
- Modify the existing `topMenu` Spring bean, by referencing new or modified top-menu item beanss
- Modify existing or create new lists of left-menu beans, and identify text labels, target page Java classes, privileges required to view the left-menu items, and whether the items are active. These `java.util.List` beans are injected into existing Wicket pages by references to the list bean names.

Services Interface – Spring Configuration

The Web UI provides service client interfaces through Spring configuration.

- Context beans identify service access points and the implemented service interfaces.
- Client logic beans provide the caller logic for accessing various end-point interface services.

Below is an example of a refreshable SOAP client Spring bean for the transaction service client context and logic beans..

`application-common-context.xml`:

```
 ...
<!-- MOBILISER SERVICE CONFIGURATION - GENERAL -->

<bean id = "soapClientFactory"
 class =
"com.sybase365.mobiliser.util.tools.clientutils.soap.SoapClientFactory"
/>
<bean id="serviceConfigurationBase"

class="com.sybase365.mobiliser.util.tools.wicketutils.services.Configura
tion">
 <property name="preferences" ref="prefsNode" />
</bean>

<bean id="serviceConfiguration" parent="serviceConfigurationBase" />

...
```

`system-clients-context.xml`:

```
<bean id="clientTargetSourceParent" abstract="true"
class="com.sybase365.mobiliser.util.tools.clientutils.api.Refreshab
leClientTargetSource">
<property name="clientFactory" ref="soapClientFactory" />
</bean>
...
<!-- TRANSACTION CONTEXT CLIENT -->
<bean id="systemAuthTransactionClientSource"
parent="clientTargetSourceParent">
  <property name="clientInterfaces" ref="transactionInterfaces" />
```

```
  <property name="endpointSuffix" value="/transaction" />
  <property name="configuration">
<bean parent="serviceConfigurationBase">
  <property name="qualifier" value=".transaction" />
</bean>
</property>
</bean>
<bean
class="com.sybase365.mobiliser.util.prefs.util.RegisterChangeListen
erFactoryBean">
  <constructor-arg ref="prefsNode" />
  <constructor-arg ref="systemAuthTransactionClientSource" />
</bean>
<bean id="systemAuthTransactionContextClient"
class="org.springframework.aop.framework.ProxyFactoryBean">
  <property name="targetSource"
ref="systemAuthTransactionClientSource" />
  <property name="interfaces" ref="transactionInterfaces" />
</bean>
<bean id="systemAuthTransactionClientLogic"
class="com.sybase365.mobiliser.web.application.clients.TransactionC
lientLogic">
  <property name="wsAuthEndpoint"
ref="systemAuthTransactionContextClient" />
  <property name="wsPreAuthEndpoint"
ref="systemAuthTransactionContextClient" />
...
```

For service clients that match multiple service endpoints, you can use Spring aliases to differentiate between interfaces. For example:

```
<alias name="systemAuthTransactionContextClient"
alias="systemAuthTransactionCancelClient" />
<alias name="systemAuthTransactionContextClient"
alias="systemAuthConfirmVoucherClient" />
<alias name="systemAuthTransactionContextClient"
alias="systemAuthStartVoucherClient" /> ...
```

Services Interface – Client Logic Implementation

All services should be exposed through their own *xyz*ClientLogic class, and created as a Spring Bean. An *xyz*ClientLogic class should be a subclass of the `BaseClientLogic` class, so it inherits the standard client logic methods.

To implement an *xyz*ClientLogic class, import the required service interface and bean (or alias):

```
import
com.sybase365.mobiliser.money.services.api.ITransactionEndpoint;

private ITransactionEndpoint wsTransactionEndpoint;

public void setWsTransactionEndpoint(ITransactionEndpoint
wsTransactionEndpoint) {
  this.wsTransactionEndpoint =
```

```
        wsTransactionEndpoint;
}
```

Next, provide an access method that uses the new member field:

```
public   List <SimpleTransaction> findTransactions (
  SearchTransactionCriteria criteria, Integer maxNumberToFetch )
  throws Exception {

  LOG.debug ("#  TransactionClientLogic.findTransactions ()");

  List <SimpleTransaction> transactions = new ArrayList <
SimpleTransaction >() ;

  if (PortalUtils.exists(criteria.getCustomerId()))
     request.setCustomerId(criteria.getCustomerId().longValue());


  request.setFromDate(criteria.getFromDateXml());
  request.setToDate(criteria.getToDateXml());
  request.setMerchantOrderIdFilter(criteria.getOrderID());
  request.setShowFaulty(criteria.getShowFaulty());
  request.setStatusFilter(criteria.getTxnStatus());
  request.setMaxRecords(maxNumberToFetch.intValue());
  request.setJoinedCustomerId(criteria.getJoinedCustomerId());
  request.setCustomerIsPayer(criteria.getConsumerIsPayer());
  request.setShowInitial(criteria.getShowInitial());
  request.setCaller(criteria.getCallerId());
  FindTransactionsResponse response = wsTransactionEndpoint.
findTransactions(request);

  if (!evaluateMobiliserResponse(response))
  {
   LOG.warn("# An error occurred while loading customer transactions");

  }
  return response.getTransactions();

 }
}
```

Using the `getNewMobiliserRequest` helper method ensures that all requests are formed with correct values for these properties:

- Origin – identifies the request as originating from the Web UI.
- Conversation Id – a random value.
- Trace No – a random value.
- Audit Data – contains the browser user agent (or first 80 characters) of the Web request and the remote IP address of the sender.

```
public <Req extends MobiliserRequestType>
   Req getNewMobiliserRequest(Class<Req> requestClass)
   throws Exception {}
```

Using the `evaluateNewMobiliserResponse` helper method ensures that all responses are verified:

- Valid Response – a status response code value of 0 indicates that processing can continue.
- Session Ended Response – response codes 352 and 353 indicate that the session has been closed or expired; you are automatically redirected to the login page.
- Another Error Response – other response codes require the method to look up the localized error message associated with the code.

```
public <Resp extends MobiliserResponseType>
   boolean evaluateMobiliserResponse(Resp response){ }
```

Using a Spring Bean for the *xyz*ClientLogic class, defined in the section above, you can inject the client logic handler into a page to use, for example:

```
...
public class AddFunds extends BaseManageAccountsPage {
  private static final long serialVersionUID = 1L;
  @SpringBean(name = "systemAuthTransactionClientLogic")
  public TransactionClientLogic transactionClientLogic;
...
```

Application Top Menus

You can declare menus using Spring XML configuration.

For the top menus (main navigation items) you need only modify the existing `topMenu` Spring bean.

In the `application-common-context.xml` file:

```
 ...
<property name="topMenu">
<bean id="topMenu"

class="com.sybase365.mobiliser.util.tools.wicketutils.menu.SybaseMe
nu">
<property name="entries">
  <util:list>
   <!-- SELFCARE TOP MENUS -->
   <ref bean="consumerSelfCare" />
   <ref bean="manageAccount" />
   <ref bean="viewTransactions" />
   <ref bean="billConfiguration" />
   <ref bean="myNewTopMenuItem" />
...
```

In the `my-menus-context.xml` file:

```
 ...
<bean id="myNewTopMenuItem"

class="com.sybase365.mobiliser.util.tools.wicketutils.menu.MenuEntr
y">
 <property name="name" value="top.menu.my.new.menu" />
 <property name="page"
```

```
value="com.sybase365.mobiliser.web.dashboard.pages.home.MyMenuPage"
/>
 <property name="privilege" value="UI_DASHBOARD_LOGIN" />
 <property name="active" value="true" />
</bean>
...
```

Application Left Menus

Application left-menu items are specific to individual applications and pages.

Declare a Spring bean list of the possible left-menu options, for example:

```
 ...
<util:list id="myLeftMenuList" value-type="java.util.LinkedList" list-
class="java.util.LinkedList">
<bean id="myLeftMenu1"
class="com.sybase365.mobiliser.util.tools.wicketutils.menu.MenuEntry">

<property name="name" value="dashboard.leftmenu.home.1" />
<property name="page"
value="com.sybase365.mobiliser.web.dashboard.pages.home.MyFirstPage"/>

<property name="privilege" value="UI_DASHBOARD_LOGIN" />
</bean>

<bean id="myLeftMenu2"
class="com.sybase365.mobiliser.util.tools.wicketutils.menu.MenuEntry">

<property name="name" value="dashboard.leftmenu.home.2" />
<property name="page"
value="com.sybase365.mobiliser.web.dashboard.pages.home.MySecondPage"/>

<property name="privilege" value="UI_DASHBOARD_PREFS" />
</bean>
</util:list>
...
```

The left menu list bean is injected into the page where the menus are relevant and used to build the left menu items.

```
 ...
@AuthorizeInstantiation(Constants.PRIV_DASHBOARD_LOGIN)
public class MyPagePage extends BaseDashboardPortalPage {
private static final long serialVersionUID = 1L;
@SpringBean(name = "myLeftMenuList")
private LinkedList<IMenuEntry> myLeftMenu;
public void setMyLeftMenu(LinkedList<IMenuEntry> myLeftMenu) {
   this.myLeftMenu = myLeftMenu;
}
...
@Override    public LinkedList<IMenuEntry> buildLeftMenu() {
   LOG.debug("#buildLeftMenu()");
   setActiveTopMenuByName("top.menu.dashboard.mymenu");
   return this.myLeftMenu;     }
...
```

Web Pages

When designing and developing Web pages, or changing existing pages, use the generic page layout as a guide, and follow the style guidelines.

Although designers may use HTML rendering and mock-up, they should be familiar with the current Web layout; otherwise, they may need to extensively modify the underlying application structure. Web developers need not be expert in Wicket application development techniques, especially with respect to integrating customized functions and HTML templates that use Wicket components.

When you customize Web page templates, SAP recommends you follow these Wicket best practices:

- Reuse code at either the panel or the component level.
- Use lists to provide standard access to aggregates through a shared `DataProvider` class, for each data object.
- Follow the existing localization guidelines for consistent application maintenance and control.

Style Guidelines

A generic implementation is provided as an example of how you can customize the user interface.

Page Width and Height

Standards for page width and height allow pages to expand both horizontally and vertically; this caters to large screens, while also maintaining the visual appearance of the page at minimum sizes.

Standard Web page dimensions are:

- A minimum width of 1050 pixels (CSS: #content)
- A minimum height of 550 pixels (CSS: #content)

By default, Web pages do not include content in the main application content pages that expand horizontally when you resize the window: only the header and footer items that are right aligned move when you resize the window. Typically, all content is left aligned. To center the main content, rework the base page structures and styles.

Page Components

The standard Web page components include a header, a footer, top-menu items, and left-menu content.

Header

The standard authenticated user header maintains:

- An application or site graphic – (CSS: #newHeader background).
- Dynamic application name – set for each application by overriding this `BaseApplicationPage.java` method:

  ```
  protected abstract String getApplicationName();
  ```
- Context information – for the user who is logged in.
- The language selector – defaults to English and German; you can add other languages to the header.

Footer

The standard authenticated-user footer maintains:

- Footer container – static copyright text (CSS: #newFooter).
- Blank links and session timeout information – the session timeout counts down to the session termination time.
- Small footer graphic.

Top-Menu Items

The top menu navigation items identify the set of pages that are globally accessible to users who are logged in, based on their roles (privileges). The selected menu item text appears in the breadcrumbs of the main content page. This allows the text of the top navigation menu items (and by implication the horizontal space taken up by the items) to be fixed. The standard authenticated top menu items maintain:

- The selected menu item is tagged with a "notch" – the active item is set by the menu code (CSS: ul.dropdown li.selected).
- Subitems – you cannot select a menu item that has subitems; you can select the subitems.

Left Menu Content

In different parts of an application, the left-menu content column contains:

- Context-specific information – such as a selected customer or agent name.
- Context-specific menu options – relevant to either the chosen top menu item or the context item. The background of the active menu is highlighted (CSS: ul.submenu li.selected).

Note: It is valid to have an empty left-content column.

To set the active left menu item, override this `BaseApplicationPage.java` method:

```
protected abstract Class getActiveMenu();
```

Application Page Components

Every application page maintains basic characteristics and guidelines for content.

Content Containers

Content is enclosed in a set of standard `<div>` containers; this allows for standard alignment and content padding in the main-content column.

```
<div id = "centerContent" class = "content center">
   <div class = "module clear">

       [ page content here ]

   </div>
</div>
```

Breadcrumbs

Every application page must manage a breadcrumb trail that defines the position of the page within the selected top menu or left-menu item. Breadcrumbs provide a textual representation of the page in the overall set of available functions.

Note: Breadcrumbs should be the first text item on each page—error or information messages must appear below the breadcrumb text.

Enclose breadcrumb text in HTML tags:

```
<h1 class = "[type]">
<wicket:message key = "[breadcrumbtextkey]" />
</h1>
```

where *type* is one of:

- home – for application home pages only.
- signup – for application sign-up page only.
- phone – when the page relates to a mobile phone.
- lock – when the page relates to a security option.
- house – when the page relates to address information.
- search – when the page is a search or find operation.
- attachment – when the page is an attachment operation.
- bank – when the page relates to a banking operation.
- friends – when the page relates to a personal or friend operation.
- coins – when the page relates to a general money operation.
- wallet – when the page relates to a wallet oriented operation.
- payments – when the page relates to a payment.

Breadcrumb trails must expand when users follow actions that take them to other pages. To separate elements of a breadcrumb, use a standard separator.

```
<h1 class = "[type]">
  <wicket:message key = "[breadcrumb1textkey]" />
  <wicket:message key = "application.breadcrumb.separator" />
  <wicket:message key = "[breadcrumb2textkey]" />
</h1>
```

Your Friends » Add Friend to List » Add Complete

Note: Breadcrumb text must relate to a short page name, not to an instruction or help text.

Help Text

(Optional) Pages may contain a short text message that describes the required input.

The help text must be immediately below the breadcrumb text. To display help text, enclose it in HTML tags:

```
<h3 class = "help" >
  <wicket:message key = "[helptextkey]" />
</h3>
```

Error Messages and Information

All pages display messages using a standard feedback component.

The feedback component must be at the top of the page content, between breadcrumbs, or help text, if it exists, and the main page content. It is typically enclosed in a page `<form>`, if there is one.

```
div id = "centerContent" class = "content center">

  <div id = "module wrap">

  <h1 class = "[type]">
     <wicket:message key = "[ breadcrumbtextkey ]" />
  </h1>

  <h3 class = "help">
    <wicket:message key = "[helptextkey]" />
  </h3>

  <div wicket:id = "errorMessages"> </div>

   [ rest of page content here ]

  </div>
</div>
```

Success Messages

Web pages may implement messages that indicate successful operations. This differs from the information that appears in the feedback panel. An `Operation Succeeded` message appears in the page format.

Operations that involve multiple steps implement success messages at the end of the operation. Messages may include visual feedback and related data, such as an authorization code or fee details.

```
<ul class = "messages">
  <li>
    <wicket:message key = "[successmsgkey]" />
```

```
  </li>
</ul>
```

Success messages should appear after breadcrumbs and help text, but before any related data or other page content.

Form Page Components

Form pages maintain style guidelines for user-input controls in forms.

Form Input

Every form input field has an associated descriptive label and an entry field.

- Form fields must be enclosed in a `<div class="input">` container.
- The text label is a page-specific text key that describes the input value.
- The `<span class="required">*</span>` container is necessary only for mandatory fields.
- The `<input>` tag must be assigned to a `class="text"`.
- Place tags that are enclosed in `<label>` tags on the same line as the `<label>` start and end tags.

```
<div class = "input">
  <label> <wicket:message key = "[labelkey]"/> <span class =
"required"> * </span></label>
  <input class = "text" wicket:id = "field" />
</div>
```

Note: You need not annotate an asterisk with the text "* indicates a required field."

Form Text

Form-field input is typically shown to users for confirmation.

- Form fields are enclosed in a `<div class="input">` container.
- Text labels are page-specific text keys that describe input values.
- The input `<div>` tag that represents the value must be defined as `class="text"`.
- Place tags that are enclosed in `<label>` tags on the same line as the `<label>` start and end tags.

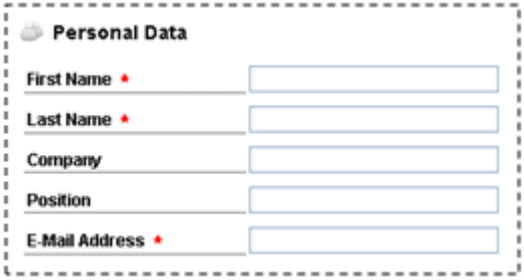
```
<div class = "input">
  <label><wicket:message key = "[labelkey]" /></label>
  <div class = "text" wicket:id = "field" />
</div>
```

One- and Two-Column Forms

You can set up forms so users enter data either in a single column of fields, or in a series of fields in two columns.

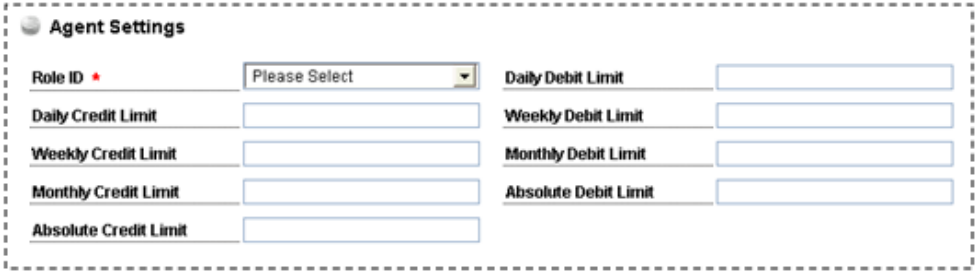
Complex data entry pages can contain a combination of both one-column and two-column data fields. Separate the form entry values into sections. Form-entry fields in a single column are included in a standard `<div>` container, for example:

```
<div class = "column">

  <div class = "input">
    <label><wicket:message key = "[labelkey1]" /></label>
    <div class = "text" wicket:id = "field1 " />
  </div>

  <div class = "input">
    <label><wicket:message key = "[labelkey2]" /></label>
    <div class = "text" wicket:id = "field2" />
  </div>
</div>
```



Place form-entry fields that appear in two columns into two different `<div class = "input">` containers. To ensure the correct layout, use the `twoColumn` CSS class.

```
<!-- left column -->
<div class = "column twoColumn">

<div class = "input">
  <label><wicket:message key = "[labelkey1]" /></label>
  <div class = "text" wicket:id = "field1" />
</div>

</div>

<!-- right column -->
<div class = "column twoColumn">

<div class = "input">
  <label><wicket:message key = "[labelkey2]" /></label>
  <div class = "text" wicket:id = "field2" />
</div>

</div>
```



Form Sections

On a form with multiple sets of entry values, you can group entry fields for visual efficiency, or to link related data items. A form may comprise multiple sections, each of which includes its own header to describe the data type.

Use sectioned forms only when there is more than one section. Simple forms with only one section use breadcrumbs as the page header.

Each form section is identified with a header tag:

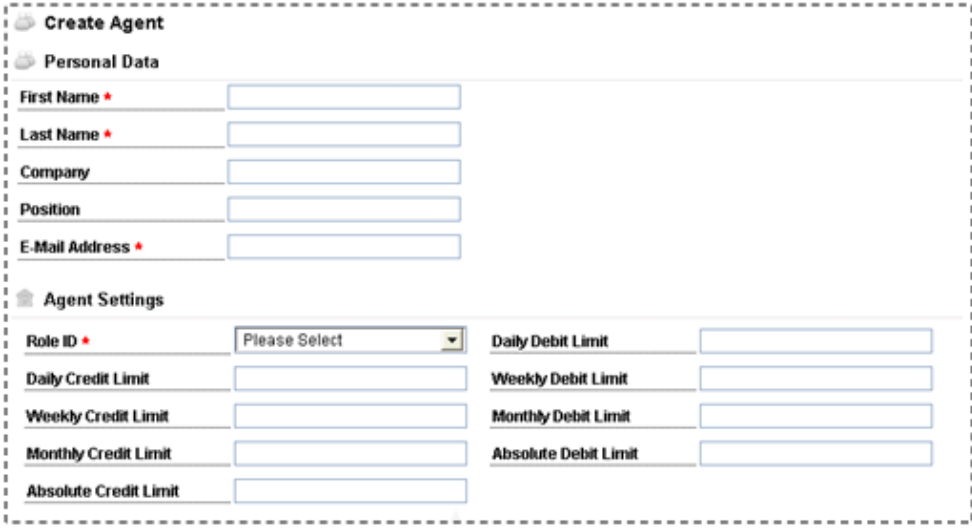
```
<h2 class = "section [type]>
  <wicket:message key = "[headerkey]" />
</h2>
```

where *type* is one of:

- phone – when the section relates to a mobile phone.
- lock – when the section relates to a security option.
- house – when the section relates to address information.
- search – when the section is a search or find operation.
- attachment – when the section is an attachment operation.
- bank – when the section relates to a banking operation.
- friends – when the section relates to a personal or friend operation.
- coins – when the section relates to a general money operation.
- wallet – when the section relates to a wallet-oriented operation.
- payments – when the section relates to a payment.

To improve readability, separate form sections with white space using the following `<div>` tag:

```
<div class = "column buffer"></div >
```



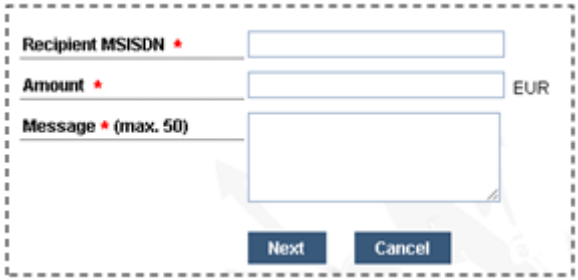
Form Buttons

Place form buttons that trigger actions either below the last entry field at the end of the form, or if there are multiple forms for each section, below the form entry fields in the section.

The form button layouts are:

- Form buttons that are associated with entry fields – standard and most typical forms with labels and entry fields. The layout for these forms aligns the buttons vertically with the entry fields. Create these form buttons using code similar to:

```
<div class = "column">
 <div class = "formAction">
 <input type = "submit" wicket:id = "submit" wicket:message =
"value:Save" />
 <input type = "submit" class = "submitForm" wicket:id = "save"
wicket:message = "value:Save" />
 </div>
</div>
```



- Form buttons that are associated with nonstandard entry forms – forms that may not have entry fields, for example, buttons that are aligned at the bottom of a list table:

```
<div class = "column">
 <div class = "buttonWrap">
 <input type = "submit" wicket:id = "submit" wicket:message =
"value:Save" />
 <input type = "submit" class = "submitForm" wicket:id = "save"
wicket:message = "value:Save" />
 </div>
</div>
```

When considering form button order, place buttons that have positive actions first, followed by negative or other actions. For example, place a Save button to the left of a Cancel button. This allows the positive action to be associated with the natural next step.

Use standard text labels for form buttons, such as Continue or Cancel.

Form Field Validation

Form fields may have a series of validation steps that are enforced by Wicket validation behavior. Wicket attempts to validate, then uses the standard feedback panel on the page to report errors.

In many cases, Wicket automatically produces validation text; however, these messages may not be user friendly or descriptive. All form fields must have a required-validation message associated with it in the text resources file. You can share these messages across pages if they relate to common form fields.

All pattern-validated form fields should have a pattern-validation message that relates to the field type; you can share these messages across pages.

For example, a field labelled "Recipient MSISDN" on a Send Money page has:

- A label – `sendMoney.recipient` (Recipient MSISDN).
- An entry field name received by the Java code – **recipient**.
- A shared required-validation message – `recipient.Required` (Recipient is required).
- A shared pattern-validation message – `recipient.PatternValidation` (Input is not a valid MSISDN).

Tab Index

The sequence of focus in a form between input fields is typically the logical order in which the input fields are referenced in the page. If a form is complex and contains two column layouts, the logical tab index may not apply.

In complex forms, it is important to maintain a tab index for the data being entered, so each form entry field must define a **tabindex** attribute.

Table Page Components

Table page components on a application page display lists of items and associated actions.

List Tables

To enable you to better visualize tables before generating them, the Web UI uses standard Wicket list views to populate HTML tables that are designed in the source HTML template pages. You can design the pages independent of the data population at runtime. Each table must include a number of components related to paging, headers, footers, and table items.

List Table Paging

All tables display standard paging options at the top of the table.

List Table Header

Each column in the list must have header text. Each column should be appropriately sized for the type of data it contains. Assign column sizes using the CSS classes that specify widths for the table header `<th>` tag.

```
<table>
 <thead>
 <th class = "selected"><wicket:message key = "selected" /></th>
 ...
</thead>
```

List table headers may contain a header that is styled as a link, which indicates the order of the list items in the table.

Best Practices for Table Design

Adding items – an option to add a new item to a list table must appear as a link in the table or page heading (CSS: additionalLink).

```
<h1 class = "friends">
  Your Friends
  <input class = "additionalLink last" type = "submit" name =
"addFriend" d = "addFriend24" value = "Add a Friend">
</h1>
```

Visibility of empty tables – if you have a table list without any items, SAP suggests that you show the table outline but not the table header and footer, if it is possible to do so.

No items in table – if there are no items in a table list, show an information message inside the table. If it is possible to do so, the message should indicate how users can add items to the table.

Localization Guidelines

You can localize the Web UI for individual environments.

Locale Configuration

Web UI users' locale settings define the language, country-specific text, currency, dates and times, and other locale-specific codes and text.

Browser locale settings for unauthenticated application sessions are based on:

- The selected language configuration, or,
- A user's selection from the language list in the Web UI.

Browser locale settings for authenticated application sessions are based on:

- The authenticated customer's locale (MOB_CUSTOMERS.ID_LANGUAGE) configuration, or,
- A user's selection from the language list in the Web UI.

For all sessions, if the configured locale is not defined in the Web UI environment, English is used in the generic Web UI.

Note: The generic Web UI only partially supports right-to-left (RTL) content rendering and associated locales. You must customize the Web UI to enable full RTL content-rendering support.

Language Selector

Use the Web UI language selector to override language settings. If you introduce additional languages, update the language selector to incorporate them.

Language selector HTML templates and Java actions are defined in these files:

- `html\com\sybase365\mobiliser\web\application\pages\BaseLoginPage.html`
- `html\com\sybase365\mobiliser\web\application\pages\BaseApplicationPage.html`

Sample HTML settings are:

```
<div id = "languageSelector">
  <script type = "text/javascript"> // <![ CDATA [
     function _switch() {
       \ $j ( "\# languageSelector").toggle("medium");
       return   false ;
     }
  //]]></script>
  <ul id = "language-select">
   <li>
     <a onclick = "_switch()" wicket:id = "changeToEnglish">
     <wicket:message key = "application.header.language.english"/>
     </a>
   </li>
   <li>
     <a onclick = "_switch()" wicket:id = "changeToGerman">
     <wicket:message key = "application.header.language.german"/>
```

```
      </a>
    </li>
  </ul>
</div>
```

Sample Java:

```
protected void initOwnPageComponents() {
...
   add(new Link("changeToEnglish") {
     @Override
     public void onClick() {
       getWebSession().setLocale(getUpdatedLocale("en_US"));
       getSession().info (
       getLocalizer().getString("language.change.success", this));
       setResponsePage(getApplication().getHomePage());
    }
   };

   add(new Link("changeToGerman") {
     @Override
     public void onClick() {
       getWebSession().setLocale(getUpdatedLocale("de_DE"));

getSession().info(getLocalizer().getString("language.change.success,
this));
       setResponsePage(getApplication().getHomePage());
     }
   };
...
}
```

Text Messages and Labels

For successful localization, define localized-text labels for all static text. To ensure all labels are localized, the Web UI uses a Wicket resource.

Resource Loading

Wicket implements resource loading with a fallback strategy using the Java class hierarchy of the associated Web page.

For example, the Wicket markup to identify a static-text label is defined in the `html\com\sybase365\mobiliser\web\consumer\pages\portal\billpayment\PayBillPage.html` file.

```
<html xmlns:wicket = "http://wicket.apache.org/dtds.data/wicket-
xhtml1.4-strict.dtd">
<body>
<wicket:extend>
   <div id = "centerContent" class = "content">
     <div class = "module clear">
       <h1 class = "payments">
         <wicket:message key = "payBill.title" />
       </h1>
       <div wicket:id = "billConfigListForPaymentPanel"></div>
```

```
    </div>
  </div>
</wicket:extend>
</body>
</html>
```

Based on the class hierarchy defined by the associated Java object:

- `com.sybase365.mobiliser.web.consumer.pages.portal.billpayment.PayBillPage` extends
- `com.sybase365.mobiliser.web.consumer.pages.portal.billpayment.BaseBillPaymentPage`, which extends
- `com.sybase365.mobiliser.web.consumer.pages.portal.BaseConsumerPortalPage`, which extends
- `com.sybase365.mobiliser.web.consumer.pages.portal.BaseConsumerPage`, which extends
- `com.sybase365.mobiliser.web.consumer.pages.portal.BaseApplicationPage`, which extends
- `com.sybase365.mobiliser.web.consumer.pages.portal.MobiliserBasePage`, which extends
- `com.sybase365.mobiliser.util.tools.wicketutils.components.BasePage`, which extends
- `org.apache.wicket.markup.html.WebPage`

Text resources are searched in the following order. If a resource file is not found, the next resource file in the hierarchy is checked. If a match to the associated key—for example, "payBill.title"— is found in a resource file, the search is terminated and the text label is returned.

- `xml\com\sybase365\mobiliser\web\consumer\portal\billpayment\PayBillPage.xml`
- `xml\com\sybase365\mobiliser\web\consumer\pages\portal\billpayment\BaseBillPaymentPage.xml`
- `xml\com\sybase365\mobiliser\web\consumer\pages\portal\BaseConsumerPortalPage.xml`
- `xml\com\sybase365\mobiliser\web\consumer\pages\BaseConsumerPage.xml`
- `xml\com\sybase365\mobiliser\web\application\pages\BaseApplicationPage.xml`
- `xml\com\sybase365\mobiliser\web\application\pages\MobiliserBasePage.xml`
- `xml\com\sybase365\mobiliser\util\tools\wicketutils\components\BasePage.xml`

Text labels should be unique and specific to the pages and panels in which they are used. For efficient maintenance, you may want to aggregate them, and associate them with a template.

For example, all the Consumer Portal text labels are aggregated in the `xml\com\sybase365\mobiliser\web\consumer\pages\portal\BaseConsumerPortalPage.xml` file.

Put application-wide labels in the resource file for the top-level application class. Put environment-wide labels—for example, shared form button labels—in the top-level environment resource file.

Note: Panel templates are associated with a general package name, because they are shared across multiple pages, and they can access text labels from a resource file, which is specific to either the panel or the environment. In either case, the resource file in which the text label exists may be outside of the package hierarchy where the panel is used.

Text Messages from Java

You can access text labels in Java code from standard Wicket localizers for a Web page, then access the text string using a key and a class name. The class name allows you to search the class resource hierarchy, which is the same as searching static Wicket text labels in HTML templates.

For example:

```
...
} catch (Exception e) {
   LOG.error("# An error occurred while updating the consumer from friends
list", e);
    error(getLocalizer().getString("update.friend.error", this));
}
...
```

Wicket Dynamic Validation Text

When Wicket applications use Wicket built-in validators for form-entry fields, Wicket references a set of implicit resource text strings that are related to the field name being validated and the validation type.

SAP recommends that you use explicit text strings, rather than using the implicit Wicket validation text strings. Using explicit text strings ensures that:

- All text strings can be translated for localization.
- Text strings are user-friendly.

The following example shows a user-friendly validation message, which Wicket uses when the pattern validation fails for **amountString**:

```
<properties>
...
  <entry key = "amountString.PatternValidator"> '\ ${input}' is not a
valid Amount </entry>
</properties>
```

Currency Amounts

Locales ensure that currency amounts are entered and appear correctly for the default currency that is assigned to applications and applied to transactions.

The following methods are defined in `MobiliserBasePage.java`, and provide convenient access to the appropriate `FormatUtils` class:

```
public long convertAmountToLong(String strAmount) throws
ParseException
public String convertAmountToString(long amount)
public String convertAmountToString(MoneyAmount amount)
public String convertAmountToStringWithCurrency(long amount)
public String convertAmountToStringWithCurrency(long amount, String
currency)
public String convertAmountToStringWithCurrency(MoneyAmount amount)
```

Note: If the currency is not supplied in the method call, you must configure it (through Preferences) for the application. If the locale is not supplied in the method call, the user's Web session locale is used.

Dates and Timestamps

Dates are localized as customers enter them, and always use their time zone.

The generic Web UI uses the jQuery datepicker plug-in for date selection. This plug-in supplies a series of internationalized text labels for months, days, and input-display formats. The plug-in provides most language variants, and some localization exists for languages that have not been completely localized across the environment. For example, the datepicker plug-in provides some Arabic localization, but full Arabic language support and right-to-left rendering is not provided in the generic Web UI.

A sample datepicker plug-in localization file is `scripts\jquery\i18n \jquery.datepicker-af.js`:

```
/* Updated for SMS Builder Web UI by msw@sybase.com */
/* monthNames and monthNamesShort changed to match JODA text so that
validation and */
/* conversion works with Wicket PatternDateConverter */
/* Note: If 'en' month names are being used, it is because there are no
JODA-specific */
/* values for this locale, and that is what it is expecting */
/* Afrikaans initialisation for the jQuery datepicker plugin. */
/* Written by Renier Pretorius. */

jQuery (function($) {
  $.datepicker.regional ['af'] = {
    closeText:'Selekteer',
    prevText:'Vorige',
    nextText:'Volgende',
    currentText:'Vandag',
    monthNames:
['January','February','March','April','May','June','July','August','Sept
```

```
ember’,’October’,’November’,’December’],
    monthNamesShort:
[’Jan’,’Feb’,’Mar’,’Apr’,’May’,’Jun’,’Jul’,’Aug’,’Sep’,’Oct’,’Nov’,’Dec’
],
    dayNames:
[’Sondag’,’Maandag’,’Dinsdag’,’Woensdag’,’Donderdag’,’Vrydag’,’Saterdag’
],
    dayNamesShort:[’Son’,’Maa',’Din’,’Woe’,’Don’,’Vry’,’Sat’],
    dayNamesMin:[’So’,’Ma’,’Di’,’Wo’,’Do’,’Vr’,’Sa’],
    weekHeader:’Wk’,
    dateFormat:’dd/mm/yy’,
    firstDay:1,
    isRTL:false,
    showMonthAfterYear:false ,
    yearSuffix:’’};
    $.datepicker.setDefaults($.datepicker.regional[’af’]);
 }
);
```

Localization files have been modified slightly from their original versions, as noted in the code comments. Dates that are generated by the datepicker plug-in are passed to Wicket for validation and conversion to internal Wicket date/time objects. Wicket uses the Java open source Joda-time library classes for this. Joda-time localized month names are not the same as the datepicker standard (or Java standard) month names. If a month name is not known by Joda-time classes, conversion and validation fail. Therefore, month names have been modified for all localized datepicker plug-in files to match the Joda-time equivalent values.

Lookup Lists and Codes

You can use generic Web UI Java classes to automatically populate localized versions of lists and codes.

- `com.sybase365.mobiliser.web.common.components.LocalizableLookupDropDownChoice.java`
- `com.sybase365.mobiliser.web.common.components.LocalizableLookupDropDownMultiChoice.java`

To enable these classes to display localized key-value pairs:

- Have an entry in an `*.xml` properties file that maps the lookup item values to localized strings. The resource key must use the format `lookup.[`*`lookupName`*`].[`*`lookupValue`*`]`.
- Include a lookup query in the database, which can be addressed by the resource loader. The lookup name must be the primary key for the lookup query in the database table.
- If there is no lookup query in the database, the key-value pairs must be in the `BasePage.xml` file, in the above format.

Index

A

account structures
    general ledger 20
accounts
    managing 36
    offline SVA 38
activating
    agents 52
adding
    agents 27
    balance alerts 52
    bill configurations 40
    coupon locations 26
    credential patterns 19
    credential policies 19
    customer-type policies 19
    do-not-disturb schedule 36
    fee types 8
    friends 34
    limit sets 11
    mobile alerts 35
    restriction groups 16
    restriction rules 17
    restrictions 17
    roles 28
    use case configuration 9
    use cases to limit sets 11
    wallet entries, approvals 14
Administration Portal 1
    logging in 1
agent hierarchy 50
agents
    activating 52
    adding 27
    blacklisting 27
    blocking 27
    creating 50
    creating subagents 51
    deactivating 27
    editing 27
    hierarchy 50
    managing 50
    privileges 50
    removing 27
    roles 50
    roles and privileges 28
    subagents 50
airtime topup 40, 48
alerts
    balance, adding 52
aliases, Spring 62
application menus
    left, customizing 66
    top, customizing 65
application page components
    breadcrumbs 69
    content containers 69
    error messages 70
    feedback 70
    help text 70
    success messages 70
applications
    configuration 59
    configuration, example 60
    customization 61
    objects 61
approving
    wallet entries 14
assigning
    coupons 26
attachments 42
    customer care 3
    notification manager 29
auditing 31
authenticated pages 57
authorizing
    transactions 21, 48

B

balance alerts
    adding 52
bank accounts
    adding 37
    editing 37
    mobile alerts 37
    mobile transactions 37
    payment instruments 37
banks
    sending money to 39
batch coupons, uploading 25

batch reports 30, 53
bill configurations 40
bill payment 40
bill payments
    open bills 41
bills
    paying 41
    payment history 42
blacklisting
    agents 27
    customers 3
blocking
    agents 27
    customers 3
breadcrumbs 69
bulk processing
    CSV files 22
    customer deactivation 21
    customer registration 21
    pending processes 21, 48
    transaction authorization 48
    transaction cancellation 48
    XML file preparation 21, 23, 48
buttons
    form 74

C

cancelling
    open bills 41
cash in 44
cash out 44
categories
    coupons 26
changing
    mobile phone number 3
    passwords 2, 42
    security questions 3
CHECKER privilege 15
client interfaces 62
commission management 52
commissions
    settling 7
components
    form input 71
    form page 71
configurations
    bills 40
configuring
    applications 59
    client interfaces 62
    do-not-disturb schedule 36
    dual approval 15
    fee-type use cases 9
    fees 3
    interest on SVAs 20
    limits 3, 12
    localization 77
consumer
    limit sets 10
consumer accounts
    signing up 32
consumer portal
    getting started 32
Consumer Portal 32
    forgot password 33
    logging in 33
consumer types
    limit sets 10
consumers
    cash in 44
    topping up airtime 48
    viewing transactions 45
contact justifications
    searching for 6
contact points
    sending test messages 34
contacts
    justify 5
containers
    content 69
content containers 69
coupon admin 24
    adding coupon locations 26
    assigning coupons 26
    categories 26
    coupon types 24
    creating coupon types 25
    generating coupons 25
    search 27
    uploading batch coupons 25
coupon types 24
    creating 25
coupons
    adding locations 26
    assigning 26
    categories 26
    generating 25
    searching 27

uploading batches 25
creating
agents 50
consumer accounts 32
coupon types 25
customer approvals 14
messages 28
subagents 51
credential patterns
adding 19
credential policies
adding 19
adding a credential pattern 19
adding customer-type policies 19
credential pattern 18
customer type 18
credit cards
adding 37
payment instruments 37
removing 37
CSV files
preparing for upload 22
currency
exchange rates 7
currency amounts 81
customer care 2
changing mobile phone numbers 3
changing security questions 3
configuring fees 3
configuring limits 3
customer registration 3
find pending approval 5
finding customers 3
registering 3
resetting passwords 3
restting PINs 3
show contacts 6
customer registration 47
customer registrations
pending approval 5
customer services 43
airtime topup 48
cash in 44
cash out 44
customer registration 47
KYC upgrade 46
pickup 44
search customer 43
view transactions 45
customer types 14
fee sets 10
global configuration 14
customer XML file preparation 21
customer-type policies
adding 19
customers
blacklisting 3
blocking 3
deactivating 3
fee sets 10
registering 3
registering, bulk processing 21
searching for 3
customizing
application left menus 66
application top menus 65
applications 61
Mobiliser Portals 55

D

DataProvider class 67
dates and timestamps 81
deactivating
agents 27
customers 3
developing
Mobiliser Portals 55
distribution partners
commission management 52
do not disturb 35
do-not-disturb schedule
configuring 36
downloading
reports 30, 53
dual approval 14
configuring 15
privileges, creating 15

E

editing
agents 27
bill configurations 40
do-not-disturb settings 36
friends list 34
messages 28
roles 28

enabling
    portal templates 1
error messages 70
evaluateNewMobiliserResponse helper method 63
exchange rates 7
EXECUTE privilege 15
exporting
    messages 28
external accounts
    adding 38
    editing 38
    payment instruments 38
    removing 38
    sending money 38

F

fee sets 10
fee types
    adding 8
    configuring use cases 9
    use-case configuration 8
fees 7
    types 7
file types, securing 14
files
    uploading, bulk processing 21, 48
financial limits 10
finding
    actions pending approval 5
footer components 67
form input 71
form page components 71
    form input 71
    form text 71
    sections 73
form text 71
forms
    buttons 74
    field validation 75
    sections 73
    single and two-column forms 71
    tab index 75
friends list
    adding 34
    adding airtime topup 34
    editing 34
    removing 34
    sending money 34

G

general ledger
    account structures 20
generating
    coupons 25
    reports 30, 53
getNewMobiliserRequest helper method 63
GL account tree 20
global configuration
    customer type 14
    organizational unit 14
    risk category 14
groups
    restriction 16
guidelines
    currency amounts 81
    dates and timestamps 81
    language selector 77
    localization 76
    localization, configuration 77
    lookup lists and codes 82
    text messages and labels 78

H

header components 67
help text 70
history
    airtime topup 42
    bill payments 42
    bulk processing 21, 48

I

importing
    messages 28
information messages 70
interest configuration
    SVAs 20
interfaces
    service 63

J

Java, text messages from 80
justify contacts 5

K

KYC upgrade 46

L

language selector 77
left-menu contents 67
limit classes 12
limit sets 10
    adding 11
    adding use case to 11
limits
    configuring 12
limits, financial 10
locale configuration 77
localization
    language selector 77
localization guidelines 76
logging 31
logging in
    Administration Portal 1
    Consumer Portal 33
    Partner Portal 42
lookup lists and codes 82

M

MAKER privilege 15
manage accounts
    credit cards 37
managing
    accounts and payment instruments 36
    agent hierarchy 50
    agents 50
    bank accounts 37
    external accounts 38
    offline SVA 38
    privileges 50
    roles 50
    stored value account (SVA) 37
    subagents 50
marketing notifications
    preferences 34
merchant agents 50
    adding a balance alert for 52
merchant dealers 50
messages
    creating 28
    editing 28
    errors and information 70
    exporting 28
    importing 28
    placeholders 28
    success 70
mobile alerts
    adding 35
    adding a date/time schedule 36
    bank accounts 37
    do not disturb 35, 36
Mobiliser Portals
    Administration Portal 1
    Consumer Portal 32
    development and customization 55
    enabling 1
    Partner Portal 42
MobiliserBasePage.java 63
money
    requesting 40
    requests, history 42
    sending 39

N

notification manager 28
    attachments 29
    messages 28
    transaction template 29

O

objects
    application 61
offline stored value accounts (SVA) 38
offline SVA 3
online reports 30, 53
open bills
    cancelling 41
organizational units
    fee sets 10
    global configuration 14
    limit sets 10
overview
    Web applications 55

P

page components
    form input 71
pages
    authenticated 57
    components 67
    components, application 68
    form buttons 74
    table 76
    unauthenticated 56
    Web 67
    width and height guidelines 67
Partner Portal 42
    logging in 42
passwords
    changing 2
    forgot, Consumer Portal 33
    resetting 3
paying
    bills 40, 41
payment instruments 36
    bank accounts 37
    credit card 37
    external accounts 38
    offline SVA 38
    stored value account (SVA) 37
payments
    open bills 41
pickup 44
placeholders
    messages 28
POI (points of interest)
    CSV file preparation 22
    XML file preparation 23
points of interest (POI)
    bulk processing 21
    CSV file preparation 22
    XML file preparation 23
policies
    credential, adding 19
    credentials 18
    customer type 19
portal templates 1
    enabling 1
portals
    development and customization 55
preferences 34
privileges 50
    add wallet entries 14
    adding 28
    CHECKER 15
    create customers 14
    customer type 14
    editing 28
    EXECUTE 15
    file types 14
    MAKER 15
    removing 28
    transactions 14

R

rates
    currency exchange 7
registering
    customers 3
    customers, bulk processing 21
removing
    agents 27
    bill configurations 40
    do-not-disturb settings 36
    friends from list 34
    roles 28
replicating
    general ledger, enterprise 20
reports
    batch 30, 53
    online 30, 53
requesting
    money 40
resetting
    password 3
    PIN 3
resource loading 78
restriction groups 16
    adding 16
restriction rules
    adding 17
restrictions
    adding 17
    risk category 16
    rules 16
risk categories
    global configuration 14
    limit sets 10
roles 50
    adding 28
    editing 28
    removing 28

roles and privileges 28
rules
    restriction 16

S

samples
    application configuration 60
scales 7
search customer 43
    cash in 44
    cash out 44
    KYC upgrade 46
    pickup 44
    view transactions 45
searching
    contact justifications 6
    coupons 27
searching for customers 3
sections
    form 73
securing
    transactions 14
    wallet entries 14
security
    auditing 31
security questions and answers 3
self care 2
    attachments 42
    change address 34
    change password 34, 42
    change PIN 34
    change security answer 34
    change security question 34
    contact points 34
    friends list 34
    preferences 34
    time zone 34
    transaction history 42
sending
    money 39
    money to a bank 39
service interfaces 63
service packages 17
signing up
    consumer accounts 32
single and two-column forms 71
Spring configuration 62
stored value account (SVA)
    adding funds 37
    balance 37
    balance alerts 37
    offline 38
    payment instruments 37
    withdrawing funds 37
stored value accounts (SVA)
    configuring interest on 20
style guidelines 67
    application page components 68
    form page components 71
    page components 67
    page width and height 67
    table page components 76
subagents 50
    creating 51
success messages 70
SVA (stored value account) 37
    offline 38
SVA (stored value accounts)
    configuring interest on 20
synchronization history 3
system configuration 6
    credential policies 18
    exchange rates 7
    fee sets 10
    fee types 7
    fees 7
    GL account tree 20
    interest configuration 20
    limit classes 12
    limit sets 10
    limits 10
    service packages 17
    terms and conditions 17

T

tab index 75
table page components 76
templates
    transaction 29
terms and conditions 17
testing
    messages 28
text messages
    Java, from 80
    Wicket dynamic validation text 80
text messages and labels 78
    resource loading 78

time zone
    preferences 34
top-menu items 67
topping up
    airtime 40
topping up airtime 48
transaction files
    bulk procesing 48
transactions 38
    airtime topup 40
    authorizing, bulk processing 21, 48
    cancelling 48
    history 42
    pending approval 5
    request money 40
    securing 14
    send money 39
    send money to banks 39
    templates 29
    viewing 39

U

unauthenticated pages 56
uploading
    batch coupons 25
    files, bulk processing 21, 48
use cases
    adding to limit sets 11
    fee type 8
    fee type, configuring 9
user manager 27
    agents 27

V

validation
    form field 75
viewing
    transactions 39, 45

W

wallet 36
    credit card 37
wallet entries
    approvals 14
    pending approval 5
    securing 14
wallets
    external account 38
    offline SVA 38
    stored value account (SVA) 37
Web applications
    overview 55
    page layout 56
Web pages 67
Wicket
    applications 55
    best practices 67
    dynamic validation text 80

X

XML files
    bulk processing 21
    uploading in bulk 23