



EngageOne Smart Bill

Developers' Guide for Product Versions 4.30

Document 602-00007-006
Published March 2021
US English Edition

Copyright © 2018, 2021 Precisely. All rights reserved.

The information contained in the documentation and/or disk is proprietary and is subject to all relevant copyright, patent, and other laws protecting intellectual property, as well as any specific agreement protecting EngageOne Digital Self Service's rights in the aforesaid information. Neither this document nor the information contained in the documentation and/or disk may be published, reproduced, copied, modified or disclosed to third parties, in whole or in part, without the express prior written permission. In addition, any use of this document, the documentation and/or the disk, or the information contained therein for any purposes other than those for which it was disclosed, is strictly forbidden. ALL RIGHTS NOT EXPRESSLY GRANTED ARE RESERVED.

Any representation(s) in the documentation and/or disk concerning performance of EngageOne Digital Self Service are for informational purposes only and are not warranties of product performance or otherwise, either express or implied. EngageOne Digital Self Service standard limited warranty, stated in its sales contract or order confirmation form, is the only warranty offered.

The documentation and/or disk is provided "AS IS" and may contain flaws, omissions, or typesetting errors. No warranty is granted nor liability assumed in relation thereto, unless specifically undertaken in EngageOne Digital Self Service's sales contract or order confirmation. Information contained in the documentation and in the disk is periodically updated, and changes will be incorporated in subsequent editions. If you have encountered an error, please notify the contacts within. All specifications are subject to change without prior notice.

Portions of this document are © Copyright 2015-2021, Sorriso Technologies, Ltd.

Contents

1	Introduction	4
2	EngageOne Smart Bill Tabs	4
3	Software Development Kit.....	4
3.1	Install the Software Development Kit.....	5
3.1.1	Step-by-step guide	5
4	EngageOne Smart Bill API Shortcuts	6
4.1	getBillSummary	6
4.2	getUsageCharges	7
4.3	getUsageUsed	7
4.4	getUsageDetails	7
4.5	getSummaryChart	8
4.6	shortcut getSummaryChart (returnSummaryChartResults)	8
4.7	getAssetChart.....	9
4.8	shortcut getAssetChart (returnAssetChartResults)	9
4.9	getUsageChart.....	9
4.10	shortcut getUsageChart (returnUsageChartResults)	9
5	EngageOne Smart Bill FreeMarker Templates	10
6	Container Templates (Java Injection)	11
7	Shortcut Templates (AJAX Injection)	15
8	EngageOne Smart Bill FreeMarker Internationalization	24
9	EngageOne Smart Bill Charts	25

1 Introduction

EngageOne Smart Bill is an online billing solution that allows online users to access account information, view digital bills, manage e-mail notifications, and designate personal profile preferences. Reliable, convenient, and easily customizable, this solution provides an intuitive digital experience.

The *EngageOne Smart Bill Developers' Guide* provides technical insight into the Application Programming Interface (API) - rules that allow the EngageOne Smart Bill application to use resources from other applications, sharing services and working collaboratively to provide the EngageOne Smart Bill functionality and page display.

EngageOne Smart Bill consists of three primary components:

- Customer Portal
- Agent Portal
- Core Platform

Once logged into the Customer portal, the EngageOne Smart Bill default page appears. This page comprises the **Billing & Usage**, **Notifications**, and **Profile** tabs.

2 EngageOne Smart Bill Tabs

The **Billing & Usage** tab is inclusive of the **Bill** and **Usage** subtabs, visible via a drop-down list box. The **Bill** subtab (default page) provides an account summary, bill alerts, and a graphical presentation of the billing history. Similarly, the **Usage** subtab reveals usage charges along with a graphical presentation of this information.

But what are the rules and resources that help generate the account detail evident throughout the EngageOne Smart Bill tabs, and how does this technology work?

Enter API shortcuts and Apache FreeMarker templates . . .

While API shortcuts are URLs that provide the rules and definitions for data display, Apache FreeMarker (hereafter referred to as FreeMarker) is a template engine that helps generate the associated HTML web pages. The templates, written in FreeMarker Template Language (FTL), determine what data to present and how to present it. More specifically, FreeMarker templates are used by EngageOne Smart Bill to transform the XML bill data into HTML. This HTML is returned in the shortcut calls defined previously and displayed on the page at the appropriate location (and according to the use case definition).

3 Software Development Kit

The [Sorriso] DSS Software Development Kit (SDK) allows customers and integrators to extend the standard products to support special customer features. With the SDK, customers, integrators, and [Sorriso] can:

- Implement completely new features as an integral part of the solution.

- Develop and test in the familiar Eclipse development environment and leverage all the features of that IDE. The SDK implements and Eclipse plug-in with full editing features such as autocompletion, suggestions, source highlighting, and source validation/compilation.

3.1 Install the Software Development Kit

These steps will install the SDK plugin for eclipse. This will allow you to edit Persona Scripting Language files in eclipse.

3.1.1 Step-by-step guide

Install JDK 11

1. (<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>)
2. Download the latest eclipse (<https://www.eclipse.org/downloads/>)
3. Run the eclipse installer to install eclipse. I would recommend installing "Eclipse IDE for Enterprise Java Developers".
4. While that is running update your source code to the latest (core_v1, archive_v1, and payment_v2).
5. Launch Eclipse, select your "core_v1" source directory for the workspace. (Let eclipse update your workspace).
6. Select Help->Install New Software...
7. Click "Add..."
8. Enter "Sorriso Technologies" for the name, and
"https://download.sorrisotech.com/eclipse/0.5.0/" for the Location,
Click "Add".
9. Check the checkbox for "Persona Usecase Category" and click "Next >". Complete the install of the software.
10. You don't need to restart yet.
11. Select Help->Install New Software...
12. Click "Add..."
13. Enter "Apache Ivy update site" for the name, and
"http://www.apache.org/dist/ant/ivyde/updatesite" for the Location,
Click "Add".
14. Check the checkbox for "Apache IvyDE Eclipse plugins" and click "Next >". Complete the install of the software.
15. Restart Eclipse.
16. Right click on the "common - jars" project select "Ivy -> Resolve".
17. Right click on the "library - core" project select "Ivy -> Resolve".
18. If you have payment you might need to update the payment projects to JAVA 11.
 - Right click on a payment project, select "Properties".
 - Select "Java Build Path".
 - Select "Libraries".
 - Select "JRE System Library" (If it says java 11 and NOT java 8, you are fine).

- If your workspace default JRE is java 11 check that (3rd radio button). Otherwise you will want to select "Alternate JRE" and select Java 11 from the dropdown. If Java 11 is not an option, click the "Installed JREs..." button and install it.

4 EngageOne Smart Bill API Shortcuts

The billing API consists of a number of shortcut URLs that can be called from AJAX, a script that interacts with a server or database to exchange information without refreshing the entire page. AJAX then returns a JavaScript Object Notation (JSON) object that contains the HTML or chart for the billing data. JSON is a format used to structure and organize data to transmit this information between the server and the web application. These API methods allow you to embed bill content within your existing HTML pages or applications. The shortcut definitions are defined in the use case - **billShortcuts.uc** - a subset of the Bill Library module.

The current implementation of the example application uses the Java methods that back (or support) these shortcuts to simplify the display of bills (without using AJAX). Many changes are possible by only modifying the FreeMarker templates provided, which is explained in the next section of this document.

The AJAX calls to these shortcuts are defined in the following AngularJS files:

1. **linked-root\js\bill.js**
2. **linked-root\js\zingchart.js.**

While **bill.js** includes an example AngularJS directive - **billtest** - for AJAX loading of the **Bill** and **Usage** views, **zingchart.js** includes the **getChart** shortcut for charting. Collectively, these files are a subset of the *brand.zip* file which is uploaded in the Agent portal **Configure Application** tab (and available for download by the customer).

A standard local development environment runs the application at:

<http://localhost:8080/personaWeb/user>. This is the base URL shown in the shortcut URL examples that follow.

4.1 getBillSummary

shortcut **getBillSummary** (returnBillSummaryResults)

[sBillingAccount, sBillDate, sPaymentGroup, sLanguageCode, sCountryCode]

The **getBillSummary** shortcut returns the FreeMarker-rendered HTML for the bill summary section.

- sBillingAccount – Internal billing account number ("C538876349C")
- sBillDate – Bill date in format YYYYMMDD ("20160601")
- sPaymentGroup – Payment group ("ogone_com")
- sLanguageCode – Language code ("en")
- sCountryCode – Country code ("US")

The standard local development environment that runs the **getBillSummary** shortcut is as follows:

http://localhost:8080/personaWeb/user/getBillSummary?sBillingAccount=C538876349C&sBillDate=20160601&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US

4.2 `getUsageCharges`

shortcut **`getUsageCharges`** (returnUsageSummaryResults)

[sBillingAccount, sBillDate, sPaymentGroup, sLanguageCode, sCountryCode]

The **`getUsageCharges`** shortcut returns the FreeMarker-rendered HTML for the Usage Summary Charges table.

- sBillingAccount – Internal billing account number (“C538876349C”)
- sBillDate – Bill date in format YYYYMMDD (“20160601”)
- sPaymentGroup – Payment group (“ogone_com”)
- sLanguageCode – Language code (“en”)
- sCountryCode – Country code (“US”)

The standard local development environment that runs the **`getUsageCharges`** shortcut is as follows:

http://localhost:8080/personaWeb/user/getUsageCharges?sBillingAccount=C538876349C&sBillDate=20160601&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US

4.3 `getUsageUsed`

shortcut **`getUsageUsed`** (returnUsageSummaryResults2)

[sBillingAccount, sBillDate, sPaymentGroup, sLanguageCode, sCountryCode]

The **`getUsageUsed`** shortcut returns the FreeMarker-rendered HTML for the Usage Summary Usage Used table.

- sBillingAccount – Internal billing account number (“C538876349C”)
- sBillDate – Bill date in format YYYYMMDD (“20160601”)
- sPaymentGroup – Payment group (“ogone_com”)
- sLanguageCode – Language code (“en”)
- sCountryCode – Country code (“US”)

The standard local development environment that runs the **`getUsageUsed`** shortcut is as follows:

http://localhost:8080/personaWeb/user/getUsageUsed?sBillingAccount=C538876349C&sBillDate=20160601&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US

4.4 `getUsageDetails`

shortcut **`getUsageDetails`** (returnUsageDetailsResults)

[sBillingAccount, sBillDate, sPaymentGroup, sLanguageCode, , sCountryCode, sAssetId, sAssetName, sUsageType]

The **getUsageDetails** shortcut returns the FreeMarker-rendered HTML for the Usage Details Popup table.

- sBillingAccount – Internal billing account number (“C538876349C”)
- sBillDate – Bill date in format YYYYMMDD (“20160601”)
- sPaymentGroup – Payment group (“ogone_com”)
- sLanguageCode – Language code (“en”)
- sCountryCode – Country code (“US”)
- sAssetId - Asset ID (phone number) of the phone (“264-479-7918”)
- sAssetName – Phone user name (“Sarah%20Frederick”)
- sUsageType – Type of details requested (“usagedata”, “usagetext”, or “usagetalk”)

The standard local development environment that runs the **getUsageDetails** shortcut is as follows:

http://localhost:8080/personaWeb/user/getUsageDetails?sBillingAccount=C538876349C&sBillDate=20160601&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US &sAssetId=264-479-7918&sAssetName=Sarah%20Frederick&sUsageType=usagedata

4.5 getSummaryChart

4.6 shortcut getSummaryChart (returnSummaryChartResults)

[sBillDate, sBillingAccount, sPaymentGroup, sLanguageCode, sCountryCode, sChartType, sIsMobile]

The **getSummaryChart** shortcut returns a ZingChart-rendered chart.

- sBillDate – Bill date in format YYYYMMDD (“20160601”), or numeric long version for drilldown by date charts (such as the bill summary drilldown). This value is passed through ZingChart and represents the time in milliseconds since the epoch of January 1, 1970, 00:00:00 GMT (such as “1468251814”).
- sBillingAccount – Internal billing account number (“C538876349C”)
- sPaymentGroup – Payment group (“ogone_com”)
- sLanguageCode – Language code (“en”)
- sCountryCode – Country code (“US”)
- sChartType – Chart type requested
 - “bill_summary”
 - “bill_summary_drilldown”
 - “usage_summary”
 - “usage_summary_drilldown_voice”
 - “usage_summary_drilldown_data”
 - “usage_summary_drilldown_text”
- sIsMobile – True if mobile device; otherwise, false (“false”)

The standard local development environment that runs the **getSummaryChart** shortcut is as follows:

http://localhost:8080/personaWeb/user/getChart?sBillDate=20160601&sBillingAccount=C538876349C&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US&sChartType=bill_summary&sIsMobile=false

http://localhost:8080/personaWeb/user/getChart?sBillDate=20160601&sBillingAccount=C538876349C&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US&sChartType=usage_summary_drilldown_voice&sIsMobile=false

4.7 getAssetChart

4.8 shortcut getAssetChart (returnAssetChartResults)

[sBillDate, sBillingAccount, sPaymentGroup, sLanguageCode, sCountryCode, sChartType, sIsMobile, sAssetId]

The **getChart** shortcut returns a ZingChart-rendered chart.

- sBillDate – Bill date in format YYYYMMDD (“20160601”), or numeric long version for drilldown by date charts (such as the bill summary drilldown). This value is passed through ZingChart and represents the time in milliseconds since the epoch of January 1, 1970, 00:00:00 GMT (such as “1468251814”).
- sBillingAccount – Internal billing account number (“C538876349C”)
- sPaymentGroup – Payment group (“ogone_com”)
- sLanguageCode – Language code (“en”)
- sCountryCode – Country code (“US”)
- sChartType – Chart type requested (none currently implemented)
- sIsMobile – True if mobile device; otherwise, false (“false”)
- sAssetId - Asset ID (phone number) of the phone (“264-479-7918”)

The standard local development environment that runs the **getAssetChart** shortcut is as follows:

http://localhost:8080/personaWeb/user/getChart?sBillDate=20160601&sBillingAccount=C538876349C&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US&sChartType=asset_summary&sIsMobile=false&sAssetId=264-479-7918

4.9 getUsageChart

4.10 shortcut getUsageChart (returnUsageChartResults)

[sBillDate, sBillingAccount, sPaymentGroup, sLanguageCode, sCountryCode, sChartType, sIsMobile, sAssetId, sUsageType]

The **getUsageChart** shortcut returns a ZingChart-rendered chart.

- sBillDate – Bill date in format YYYYMMDD (“20160601”), or numeric long version for drilldown by date charts (such as the bill summary drilldown). This value is passed through ZingChart and

represents the time in milliseconds since the epoch of January 1, 1970, 00:00:00 GMT (such as "1468251814").

- sBillingAccount – Internal billing account number ("C538876349C")
- sPaymentGroup – Payment group ("ogone_com")
- sLanguageCode – Language code ("en")
- sCountryCode – Country code ("US")
- sChartType – Chart type requested (none currently implemented)
- sIsMobile – True if mobile device; otherwise, false ("false")
- sAssetId - Asset ID (phone number) of the phone ("264-479-7918")
- sUsageType – Type of details requested ("usagedata", "usagetext", or "usagetalk")

The standard local development environment that runs the **getUsageChart** shortcut is as follows:

http://localhost:8080/personaWeb/user/getChart?sBillDate=20160601&sBillingAccount=C538876349C&sPaymentGroup=ogone_com&sLanguageCode=en&sCountryCode=US&sChartType=usage&sIsMobile=false&sAssetId=264-479-7918&sUsageType=usagedata

5 EngageOne Smart Bill FreeMarker Templates

FreeMarker templates are used by EngageOne Smart Bill to transform the XML bill data into HTML. This HTML is returned in the shortcut calls defined previously and displayed on the page at the appropriate location (according to the use case definition).

FreeMarker provides robust documentation for templates. For more information, please visit the FreeMarker site at: <http://freemarker.org/docs/index.html>.

The bill FreeMarker templates can be found in the **\linked-root\ftl\bill** directory. These templates are a subset of the *brand.zip* file, which is uploaded in the Agent portal **Configure Application** tab (and available for download by the customer). The template files have the following naming convention:

<type>-<billstream>-<billversion>.ftl

For example:

bill_summary-Smartbill-1-.ftl

As shown above, the template file names must match the bill stream and bill version for your application. This requirement allows EngageOne Smart Bill to have versioned templates (according to the bill stream and bill version).

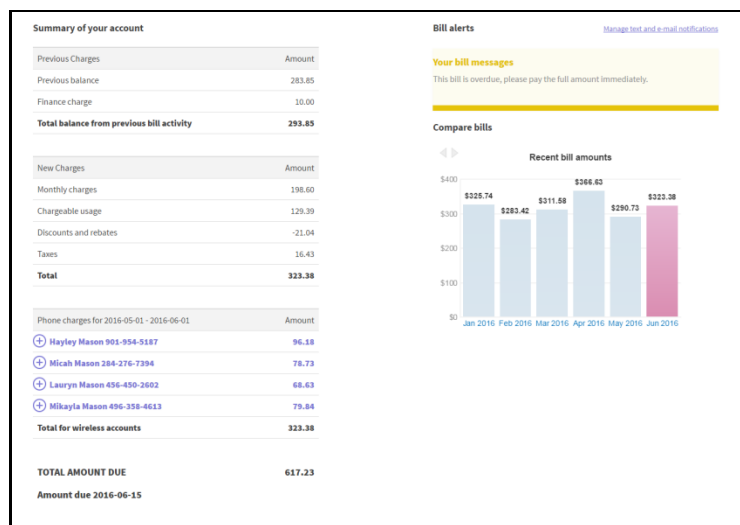
The following templates are defined for the current product:

- bill_container-Smartbill-1.ftl
- bill_summary-Smartbill-1.ftl
- usage_container-Smartbill-1.ftl
- usage_charges-Smartbill-1.ftl

- usage_used-Smartbill-1.ftl
- usage_details_container-Smartbill-1.ftl
- usage_details-Smartbill-1.ftl
- usage_details_csv-Smartbill-1.ftl

6 Container Templates (Java Injection)

To simplify the application and provide a more relaxed learning curve, the example application injects the bill contents directly into the use case screen using Java calls with **container** templates. This action allows you to experiment with FreeMarker templates without having to worry about JavaScript, AngularJS, AJAX, and the order of operations. These Java methods are the same methods that back (or support) shortcut API calls. It is easier to add JavaScript or AJAX calls within the templates for your existing content (such as marketing material) using this implementation. This action becomes problematic when you use the shortcut API to load a template instead (since you now have an AJAX call loading the template as well as additional AJAX calls within). Binding or instantiating the JavaScript to chain the calls can be challenging.



bill_container-Smartbill-1.ftl

The Bill Container template encapsulates the Bill Summary screen, and it injects the other templates through the use of the FreeMarker `#include` statement. This template does not have a corresponding AJAX shortcut call. The example application injects this container template directly into the HTML through Java. In the example below, it is extremely important to note the use of `#include` to inject the **bill_summary-SmartBill-1.ftl** template, the ZingChart element with the **chartCtrl** controller, and the ZingChart inner div element.

```
<div class="row " id="billSummary_billDetails" viewbill=""
account="${info.internalAccountNumber}" billdate="${info.billDate?date?string}"
paymentgroup="${info.paymentGroup}" language="en">
  <div class="col-md-5 " id="billSummary_col_left">
    <h4 class=" row st-padding-bottom" id="billSummary_summaryHeader">
```

```

        <span class=" " id="billSummary_sSummaryHeader">Summary of your
account</span>
    </h4>
    <div class="row " id="billSummary_row">
        <div class="col-md-12 " id="billSummary_col">
            <!-- Bill summary content - angular controller viewbillsummary -->
            <#include "bill_summary-SmartBill-1.ftl">
        </div>
    </div>
    <div class="row st-padding-bottom " id="billSummary_viewPaperBill">
        <#if "false" == "true">
            <a class=" btn btn-link" href="#"
id="billSummary_viewPaperBillLink" ng-click="downloadPdf($event)" tabindex="10">View paper
bill</a>
        </#if>
    </div>
</div>
<div class="col-md-2 " id="billSummary_col_center">
</div>
<div class="col-md-5 " id="billSummary_col_right">
    <div class="row " id="billSummary_topRow">
        <h4 class=" row col-md-6" id="billSummary_billAlertHeader">
alerts</span>
            <span class=" " id="billSummary_sBillAlertHeader">Bill
    </h4>
    <div class="col-md-6 st-padding-bottom st-padding-top "
id="billSummary_manageNotifications">
        <a class=" btn btn-link pull-right" href="gotoNotifications"
id="billSummary_manageNotificationsLink" tabindex="11">Manage text and e-mail notifications</a>
    </div>
</div>
<div class="row " id="billSummary_billAlerts">
    <div class="row col-md-12 alert alert-warning st-padding-bottom "
id="billSummary_msgAlert">
        <h4 class="">
            <span class=" " id="billSummary_msgBillAlert.sTitle">Your
bill messages</span>
        </h4>
        <p class="">
            <span class=" " id="billSummary_msgBillAlert.sBody">
                <!-- Bill notifications -->
                <#if !info.isAccountClosed && !info.isBillOverdue &&
!info.isFinalBill && !info.isPaymentDisabled>
                    No bill messages are available for this
                    billing period of ${info.billDate}.
                </#if>
                <#else>
                    <#if info.isAccountClosed>
                        This account has been closed. All
                        bills have been paid in full.<br/><br/>
                    </#if>
                    <#if info.isBillOverdue>
                        This bill is overdue, please pay the
                        full amount immediately.<br/><br/>
                    </#if>
                    <#if info.isFinalBill>
                        This is your final bill. You will not
                        receive any more bills for this account.<br/><br/>
                    </#if>
                    <#if info.isPaymentDisabled>
                        Payment is disabled for this
                        account.<br/><br/>
                    </#if>
                </#if>
            </span>
        </p>
    </div>
</div>
<div class="row st-padding-bottom " id="billSummary_bottomRow">
    <h4 class=" row st-padding-bottom" id="billSummary_billChartHeader">
EngageOne Smart Bill, Developers' Guide for Product Versions 4.30
Document 602-00007-006 – Published March 2021

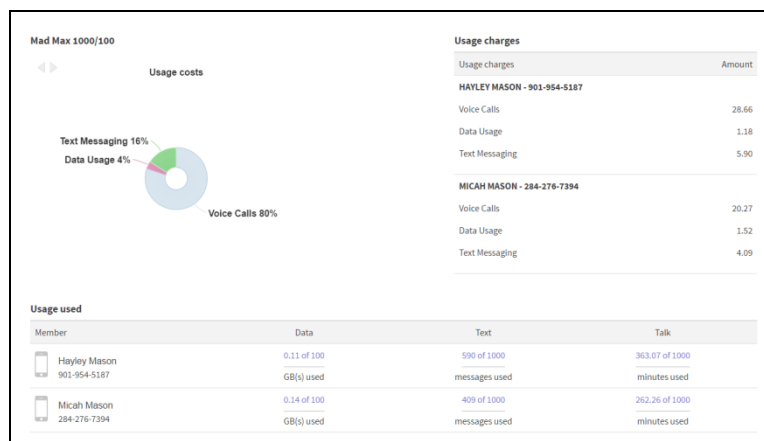
```

```

        <span class=" " id="billSummary_sCompareBills">Compare bills</span>
    </h4>
    <!-- Bill summary zingchart - angular controller chartCtrl -->
    <div class="ng-scope" id="billSummary_billChartController" ng-
controller="chartCtrl" ng-init="url =
'getSummaryChart?sBillDate=${dateHelper.getDateAsString('yyyy-MM-dd', 'yyyyMMdd',
info.billDate?date?string)}&sBillingAccount=${info.internalAccountNumber}&sPaymentGroup=${info.pa
ymentGroup}&sLanguageCode=${languageCode}&sCountryCode=${countryCode}&sChartType=bill_summary'">
        <div class="st-bills-chart ng-isolate-scope"
id="billSummary_billChart" zingchart="" zc-width="100%" zc-height="100%" zc-json="billChartJson"
zc-render="billChartRender">
        </div>
    </div>
</div>
</div>
</div>

```

FreeMarker Example



usage_container-Smartbill-1.ftl

The Usage Container template encapsulates the Usage Summary screen, and it injects the other templates through the use of the FreeMarker #include statement. This template does not have a corresponding AJAX shortcut call. The example application injects this container template directly into the HTML through Java. In the example below, it is extremely important to note the use of #include to inject the **usage_charges-SmartBill-1.ftl** and **usage_used-SmartBill-1.ftl** templates, the ZingChart element with the **chartCtrl** controller, and the ZingChart inner div element.

```

<div class="row" id="billUsage_usageDetails">
    <div class="col-md-5" id="billUsage_col_left">
        <h4 class="row st-padding-bottom" id="billUsage_sPlanNameHeader">
            <span class=" "
id="billFilterChild_sPlanName">${summary.summary.plan.@name}</span>
        </h4>
        <!-- Usage summary zingchart - angular controller chartCtrl -->
        <div class="ng-scope" id="billUsage_billChartController" ng-controller="chartCtrl"
ng-init="url = 'getSummaryChart?sBillDate=${dateHelper.getDateAsString('yyyy-MM-dd', 'yyyyMMdd',
info.billDate?date?string)}&sBillingAccount=${info.internalAccountNumber}&sPaymentGroup=${info.pa
ymentGroup}&sLanguageCode=${languageCode}&sCountryCode=${countryCode}&sChartType=usage_summary'">
            <div class="st-bills-chart ng-isolate-scope" id="billUsage_billChart"
zingchart="" zc-width="100%" zc-height="100%" zc-json="billChartJson" zc-
render="billChartRender">
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
    <div class="col-md-2 " id="billUsage_col_middle">
    </div>
    <div class="col-md-5 " id="billUsage_col_right">
        <div class=" " id="billUsage_topRow">
            <h4 class=" " id="billUsage_usageHeader1">
                <span class=" " id="billUsage_sUsageChargesHeader">Usage
charges</span>
            </h4>
            <div class="row " id="billUsage_row">
                <div class="col-md-12 " id="billUsage_col">
                    <!-- Usage summary charges content - angular controller
viewusagecharges -->
                    <#include "usage_charges-SmartBill-1.ftl">
                </div>
            </div>
        </div>
    </div>
    <div class="row st-padding-bottom " id="billUsage_lastRow">
        <div class="col-md-12 " id="billUsage_lastCol">
            <h4 class=" " id="billUsage_usageHeader2">
                <span class=" " id="billUsage_sUsageUsedHeader">Usage used</span>
            </h4>
            </div>
            <div class="row " id="billUsage_row">
                <div class="col-md-12 ng-scope" id="billUsage_col" ng-
controller="DefaultCtrl">
                    <!-- Usage summary used content - angular controller viewusageused
-->
                    <#include "usage_used-SmartBill-1.ftl">
                </div>
            </div>
        </div>
    </div>
</div>

```

FreeMarker Example

Individual usage details for Sarah Frederick | 264-479-7918

Billing period: 06/01/2016 View details by: Data [PRINT](#) [PDF](#) [CSV](#)

Show 25 entries

Date/Time	Units used	Amount (\$)
05-01-2016 12:12:51 PM	4868 Mb(s)	\$0.05
05-01-2016 12:16:52 PM	463 Kb(s)	\$0.00
05-01-2016 04:34:30 PM	2771 Mb(s)	\$0.03
05-01-2016 05:53:04 PM	844 Kb(s)	\$0.01
05-01-2016 08:32:26 PM	989 Kb(s)	\$0.01
05-02-2016 02:18:41 AM	571 Kb(s)	\$0.01
05-02-2016 04:12:40 AM	747 Kb(s)	\$0.01
05-02-2016 09:57:10 AM	162 Kb(s)	\$0.00
05-02-2016 03:43:33 PM	2034 Mb(s)	\$0.02
05-03-2016 12:01:42 AM	913 Kb(s)	\$0.01
05-03-2016 07:54:01 AM	2761 Mb(s)	\$0.03
05-03-2016 09:13:59 AM	117 Kb(s)	\$0.00
05-03-2016 09:24:23 AM	113 Kb(s)	\$0.00
05-03-2016 12:03:52 PM	529 Kb(s)	\$0.01
05-03-2016 09:08:50 PM	100 Kb(s)	\$0.00
05-04-2016 04:27:46 PM	386 Kb(s)	\$0.00
05-04-2016 07:09:15 PM	3091 Mb(s)	\$0.03
05-04-2016 11:09:41 PM	90 Kb(s)	\$0.00

usage_details_container-Smartbill-1.ftl

The Usage Details Container template encapsulates the Usage Details popup, and it injects the Details template through the use of the FreeMarker #include statement. This template does not have a corresponding AJAX shortcut call. The example application injects this container template directly into the HTML through Java. In the example below, it is extremely important to note the use of #include to inject the **usage_details-SmartBill-1.ftl** template.

```
<div class="row " id="billUsage_row">
  <div class="col-md-12 " id="billUsage_col">
    <!-- Usage details content - angular controller viewusagedetails -->
    <#include "usage_details-SmartBill-1.ftl">
  </div>
</div>
```

FreeMarker Example

7 Shortcut Templates (AJAX Injection)

For more complex applications (or to inject bill content into existing HTML pages or web applications), you can use the AJAX shortcut API methods. Instead of using a container template that includes the sub-templates, you can inject the sub-templates - themselves - into the screen by calling the shortcuts. This method is more flexible, at the cost of being more complicated. The complexity increases when trying to inject a FreeMarker template via AJAX which - itself - contains subsequent AJAX calls for marketing content or other functionalities.

Summary of your account	
Previous Charges	Amount
Previous balance	189.12
Payment received - Thank you	-201.40
Total balance from previous bill activity	-12.28
New Charges	Amount
Monthly charges	101.75
Chargeable usage	132.55
Discounts and rebates	-8.21
Taxes	11.74
Total	237.83
Phone charges for 2016-05-01 - 2016-06-01	Amount
+ Sarah Frederick 264-479-7918	119.21
+ Lilah Frederick 472-772-1931	118.62
Total for wireless accounts	237.83
TOTAL AMOUNT DUE	225.55
Amount due 2016-06-15	

bill_summary-Smartbill-1.ftl

The Bill Summary template outputs the HTML for the bill summary data section. The following data is available for use inside of the template:

- **summaryDoc** - The bill summary UBF XML
- **info** – The bill info map which contains all content in the <info> section of the UBF XML
 - **address1** through **address5** – Address lines 1 through 5
 - **amountDue** – Total amount due for this bill and (past) overdue bills
 - **billAmount** – Amount of the bill
 - **billDate** – Date of the bill
 - **billDueDate** – Due date of the bill
 - **billStream** – Bill stream for the bill
 - **externalAccountNumber** – Display-friendly account number
 - **externalDocId** – Document ID
 - **flex1** through **flex20** – Flex fields 1 through 20 to store user-defined data
 - **id** – ID of the bill
 - **internalAccountNumber** – Internal account number (may not be display-friendly)
 - **invoiceNumber** – Invoice number
 - **location** – Storage location of the bill
 - **minimumDue** – Minimum amount due
 - **paymentGroup** – Payment group for the bill
 - **selfReg1** through **selfReg5** – Self-registration fields 1 through 5
 - **version** – Bill version
 - **isAccountClosed** – **true** if account is closed; otherwise, **false**
 - **isBillOverdue** – **true** if amount is overdue; otherwise, **false**
 - **isFinalBill** – **true** if this is the customer’s final bill; otherwise, **false**
 - **isPaymentDisabled** – **true** if payment is disabled; otherwise, **false**
- **dateHelper** – Java date helper that has a method, `getDateAsNumber` (String format, String date), that returns a date as a numeric long value representing the time in milliseconds since the epoch of January 1, 1970, 00:00:00 GMT (such as "1468251814"). This allows the table columns to sort on a numeric value rather than text. This object also has a method, `getDateAsString` (String szInputFormat, String szOutputFormat, String szDate), which takes in a String date (such as "2016-01-20") with an input format (such as "yyyy-MM-dd"), and reformats it with an output format (such as "yyyyMMdd").
- **languageCode** – Language code of the application (**en**)
- **countryCode** – Country code of the application (**US**)
- **i18n** – Internationalization object for multi-language support
- **serviceLocator** – Service locator object needed for i18n calls
- **userData** – User data object needed for i18n calls

```
<ubf:summary-xml layout="telco.demo" version="1">
  <summary>
    <invoice>C327148423C20160601</invoice>
    <iaccount>C327148423C</iaccount>
    <account>327148423</account>
    <plan name="Average Joe 300/20">
      <voice>300 Minutes</voice>
```

EngageOne Smart Bill, Developers’ Guide for Product Versions 4.30
Document 602-00007-006 – Published March 2021

```

        <data>20 Gb</data>
    </plan>
    <date>2016-06-01</date>
    <dateRange>
        <start>2016-05-01</start>
        <end>2016-06-01</end>
    </dateRange>
    <dueDate>2016-06-15</dueDate>
    <amountDue>255.77</amountDue>
    <billAmount>255.77</billAmount>
    <minDue>10.00</minDue>
    <overdue>true</overdue>
</summary>
<owed total="255.77">
    <overdue>168.66</overdue>
    <current>87.11</current>
</owed>
<previous total="168.66">
    <balance label="Previous balance">158.66</balance>
    <financeCharge label="Finance charge">10.00</financeCharge>
</previous>
<charges total="87.11">
    <monthly label="Monthly charges">52.24</monthly>
    <usage label="Chargeable usage">36.64</usage>
    <discounts label="Discounts and rebates">-6.23</discounts>
    <taxes label="Taxes">4.46</taxes>
</charges>
<phoneUsage>
    <phone target="5409958211.phone">
        <voice minutes="433.89" cost="33.37">5409958211_calls</voice>
        <data usage="0.10" cost="1.09">5409958211_data</data>
        <text count="218" cost="2.18">5409958211_text</text>
    </phone>
</phoneUsage>
<history>
    <bill date="2016-01-01" total="86.13">
        <phone name="Caitlyn Stephens">86.13</phone>
    </bill>
    <bill date="2016-02-01" total="78.94">
        <phone name="Caitlyn Stephens">78.94</phone>
    </bill>
    <bill date="2016-03-01" total="101.34">
        <phone name="Caitlyn Stephens">101.34</phone>
    </bill>
    <bill date="2016-04-01" total="80.78">
        <phone name="Caitlyn Stephens">80.78</phone>
    </bill>
    <bill date="2016-05-01" total="70.29">
        <phone name="Caitlyn Stephens">70.29</phone>
    </bill>
    <bill date="2016-06-01" total="87.11">
        <phone name="Caitlyn Stephens">87.11</phone>
    </bill>
</history>
<phones>
    <phone target="5409958211.phone">
        <summary>
            <name>Caitlyn Stephens</name>
            <number>540-995-8211</number>
            <plan>Unlimited local calls</plan>
            <amount>87.11</amount>
            <monthly>52.24</monthly>
            <usage>36.64</usage>
            <rebates>-6.23</rebates>
            <taxes>-6.23</taxes>
            <total>87.11</total>
        </summary>
        <monthly total="52.24">

```

```

        <planUnlimitedLocal label="Unlimited local
calls">41.99</planUnlimitedLocal>
        <unlimitedText label="Unlimited texting">1.37</unlimitedText>
        <callerId label="CallerId">1.92</callerId>
        <callConf label="Call conferencing">6.64</callConf>
        <extraVmStorage label="Additional voicemail
storage">0.32</extraVmStorage>
    </monthly>
    <usageSummary total="36.64">
        <voiceCalls label="Voice Calls">33.37</voiceCalls>
        <data label="Data Usage">1.09</data>
        <text label="Text Messaging">2.18</text>
    </usageSummary>
    <rebate total="-6.23">
        <internationalRebate label="International calls rebate">-
3.56</internationalRebate>
        <stateRebate label="State calls rebate">-2.67</stateRebate>
    </rebate>
    <taxes total="4.46">
        <monthly>2.62</monthly>
        <usage>1.84</usage>
    </taxes>
</phone>
</phones>
</ubf:summary-xml>

```

UBF XML Example

```

<!-- Previous bill -->
<#if summary.previous?has_content>
    <div class='st-div-table'>
        <div class='row st-div-table-header'>
            <div class='col-md-9'>Previous Charges</div>
            <div class='col-md-3 amount'>Amount</div>
        </div>
        <#if summary.previous.balance?has_content>
            <div class='row st-div-table-body'>
                <div class='col-md-9'>${summary.previous.balance.@label}</div>
                <div class='col-md-3 amount'>${summary.previous.balance}</div>
            </div>
        </#if>
        ... more code here ...
    </div>
</#if>

```

FreeMarker Example

This example shows the extraction of the **Previous Charges** section of the summary. The `<previous>` section is referenced through `summary.previous`. To extract the previous balance (`<previous>/<balance>`), the syntax is `summary.previous.balance` and `summary.previous.balance.@label` to get the text label attribute of the `<balance>` element.

Usage charges	
Usage charges	Amount
LILAH FREDERICK - 4727721931	
Voice Calls	52.94
Data Usage	2.81
Text Messaging	8.68
SARAH FREDERICK - 2644797918	
Voice Calls	55.64
Data Usage	2.16
Text Messaging	10.32

usage_charges-INV_2015-1.ftl

The Usage Charges template outputs the HTML for the Usage Summary screen in the Usage Charges table module.

The following data is available for use inside of the template:

- **summaryDoc** - The bill summary UBF XML
- **info** – The bill info map which contains all content in the <info> section of the UBF XML. (Please review the bill summary above for additional information.)
- **dateHelper** – Java date helper. (Please review the bill summary above for additional information.)
- **languageCode** – Language code of the application (**en**)
- **countryCode** – Country code of the application (**US**)
- **i18n** – Internationalization object for multi-language support
- **serviceLocator** – Service locator object needed for i18n calls
- **userData** – User data object needed for i18n calls

```
<ubf:summary-xml>
  <monthly total="52.24">
    <planUnlimitedLocal label="Unlimited local calls">41.99</planUnlimitedLocal>
    <unlimitedText label="Unlimited texting">1.37</unlimitedText>
    <callerId label="CallerId">1.92</callerId>
    <callConf label="Call conferencing">6.64</callConf>
    <extraVmStorage label="Additional voicemail storage">0.32</extraVmStorage>
  </monthly>
  <usageSummary total="36.64">
    <voiceCalls label="Voice Calls">33.37</voiceCalls>
    <data label="Data Usage">1.09</data>
    <text label="Text Messaging">2.18</text>
  </usageSummary>
  <rebate total="-6.23">
    <internationalRebate label="International calls rebate">-
3.56</internationalRebate>
    <stateRebate label="State calls rebate">-2.67</stateRebate>
  </rebate>
```

```

    <taxes total="4.46">
      <monthly>2.62</monthly>
      <usage>1.84</usage>
    </taxes>
  </ubf:summary-xml>

```

UBF XML Example

```

<#list summary.phones.phone as phone>
  <#if phone.usageSummary?has_content>
    <#if phone.summary?has_content>
      <div class='row st-div-table-body-header'>
        <div class='col-md-9'>${phone.summary.name} -
      </div>
      <div class='col-md-3 amount'></div>
    </div>
    </#if>
    <#if phone.usageSummary.voiceCalls?has_content>
      <div class='row st-div-table-body-noborder'>
        <div class='col-md-9'>
          ${phone.usageSummary.voiceCalls.@label}
          <#if phone.usageSummary.voiceCalls.@help?has_content>
            <@help
message='${phone.usageSummary.voiceCalls.@help}'/>
          </#if>
        </div>
        <div class='col-md-3 amount'>${phone.usageSummary.voiceCalls}</div>
      </div>
    </#if>
    ... more code here ...
  </#if>
</#list>

```

FreeMarker Example

This example shows iterating over a list, using the FreeMarker `#list` syntax. The **summary.phones** object passed to the context contains a list of **phone** objects. As such, `<#list summary.phones.phone as phone>` in the template iterates over these - assigning each item to the variable: **phone**. Thus, we can retrieve the individual **phone.summary.name**, **phone.summary.number**, and the **phone.usageSummary** to get the data needed for display.

Usage used		Add user		
Member	Data	Text	Talk	
 Sarah Frederick 264-479-7918	0.21 of 100	1032 of 1000	697.27 of 1000	
	GB(s) used	messages used	minutes used	
 Lilah Frederick 472-772-1931	0.27 of 100	868 of 1000	709.19 of 1000	
	GB(s) used	messages used	minutes used	

usage_used-INV_2015-1.ftl

The Usage Used template outputs the HTML for the Usage Summary page in the Usage Used Table module.

The following data is available for use inside of the template:

- **summaryDoc** - The bill summary UBF XML

- **info** – The bill info map which contains all content in the <info> section of the UBF XML. (Please review the bill summary above for additional information.)
- **dateHelper** – Java date helper. (Please review the bill summary above for additional information.)
- **languageCode** – Language code of the application (**en**)
- **countryCode** – Country code of the application (**US**)
- **i18n** – Internationalization object for multi-language support
- **serviceLocator** – Service locator object needed for i18n calls
- **userData** – User data object needed for i18n calls

```
<#if summary.phoneUsage?has_content && summary.phones?has_content>
  <#list summary.phoneUsage.phone as phoneUsage>
    <#list summary.phones.phone as phone>
      <#if phone.@target == phoneUsage.@target>
        <#assign assetName = phone.summary.name>
        <#assign assetNumber = phone.summary.number>
      </#if>
    </#list>

    <#if assetName?has_content && assetNumber?has_content>
      <div class='row st-div-table-body'>
        <div class='col-md-3 st-usage-member'><span class="st-usage-member-
name">${assetName}</span><br/><span>${assetNumber}</span></div>
        <div class='col-md-3 text-center'>
          <#if phoneUsage.data?has_content &&
phoneUsage.data.@usage?has_content>
            <a class="st-usage-used" href="#" id="usageDataLink"
ng-click="openPopIn($event,
{controller: 'DefaultCtrl',size: 'lg',templateUrl: 'openUsagePopup?sAssetId=${assetNumber}&sAssetNam
e=${assetName}&sUsageType=usagedata'})"><span>${phoneUsage.data.@usage} of
${summary.summary.plan.data?keep_before(" ")}</span></a><span>GB(s) used</span>
          </#if>
        </div>
        ... more code here ...
      </div>
    </#if>
  </#list>
</#if>
```

FreeMarker Example

This example (again) shows iterating over a list, using the FreeMarker #list syntax. This example also shows how to #assign values to variables for use later in the template. The inner #list loop <#list summary.phones.phone as phone> on Line 3 matches the phone object from two different summary XML blocks to get the display name and number of the phone via <#assign assetName = phone.summary.name> and <#assign assetNumber = phone.summary.number>.

Individual usage details for Sarah Frederick | 264-479-7918

Billing period: 06/01/2016 View details by: Data PRINT PDF CSV

Show 25 entries

Date/Time	Units used	Amount (\$)
05-01-2016 12:12:51 PM	4868 Mb(s)	\$0.05
05-01-2016 12:16:52 PM	463 Kb(s)	\$0.00
05-01-2016 04:34:30 PM	2771 Mb(s)	\$0.03
05-01-2016 05:53:04 PM	844 Kb(s)	\$0.01
05-01-2016 08:32:26 PM	989 Kb(s)	\$0.01
05-02-2016 02:18:41 AM	571 Kb(s)	\$0.01
05-02-2016 04:12:40 AM	747 Kb(s)	\$0.01
05-02-2016 09:57:10 AM	162 Kb(s)	\$0.00
05-02-2016 03:43:33 PM	2034 Mb(s)	\$0.02
05-03-2016 12:01:42 AM	913 Kb(s)	\$0.01
05-03-2016 07:54:01 AM	2761 Mb(s)	\$0.03
05-03-2016 09:13:59 AM	117 Kb(s)	\$0.00
05-03-2016 09:24:23 AM	113 Kb(s)	\$0.00
05-03-2016 12:03:52 PM	529 Kb(s)	\$0.01
05-03-2016 09:08:50 PM	100 Kb(s)	\$0.00
05-04-2016 04:27:46 PM	386 Kb(s)	\$0.00
05-04-2016 07:09:15 PM	3091 Mb(s)	\$0.03
05-04-2016 11:09:41 PM	90 Kb(s)	\$0.00

usage_details-Smartbill-1.ftl

The Usage Details template outputs the HTML for the Usage Details Popup table.

The following data is available for use inside of the template:

- **summaryDoc** - The bill summary UBF XML
- **info** – The bill info map which contains all content in the <info> section of the UBF XML. (Please review the bill summary above for additional information.)
- **dateHelper** – Java date helper. (Please review the bill summary above for additional information.)
- **languageCode** – Language code of the application (**en**)
- **countryCode** – Country code of the application (**US**)
- **i18n** – Internationalization object for multi-language support
- **serviceLocator** – Service locator object needed for i18n calls
- **userData** – User data object needed for i18n calls
- **dataGroupId** – Data group ID for the usage details requested (in the <assetId>_<usageType> format)
- **usageType** – The type of usage details requested: **usagetalk**, **usagetext**, or **usagedata**.
- **detailsHeaderDoc** – The usage details UBF XML header (if available)
- **detailsContentsDoc** – The usage details UBF XML contents
- **detailsFooterDoc** – The usage details UBF XML footer (if available)

```
<ubf:dataGroup id="2644797918_calls" linkId="2644797918_calls" layout="basic.call">
  <ubf:name type="id">Voice Calls</ubf:name>
  <ubf:display>
    <ubf:language>EN</ubf:language>
```

```

        <ubf:layout>telco</ubf:layout>
    </ubf:display>
    <ubf:currency>
        <ubf:code>USD</ubf:code>
        <ubf:symbol>$</ubf:symbol>
    </ubf:currency>
    <ubf:rawContents>
        <call called="962-849-9956" date="2016-05-05" time="03:10:37 AM" minutes="4.31"
cost="0.31"/>
        <call called="622-524-4178" date="2016-05-03" time="12:46:21 PM" minutes="2.06"
cost="0.01"/>
        <call called="793-347-1502" date="2016-05-23" time="12:46:50 AM" minutes="1.91"
cost="0.18"/>
        <call called="698-466-4085" date="2016-05-03" time="06:03:19 PM" minutes="3.01"
cost="0.72"/>
        <call called="861-217-0777" date="2016-05-30" time="05:24:12 PM" minutes="13.35"
cost="1.07"/>
        <call called="313-184-6338" date="2016-05-04" time="02:21:21 AM" minutes="1.90"
cost="0.24"/>
        <call called="347-488-6739" date="2016-05-16" time="10:21:03 AM" minutes="4.19"
cost="0.40"/>
        <call called="347-904-9351" date="2016-05-29" time="03:08:42 AM" minutes="0.07"
cost="0.01"/>
        <call called="435-228-7628" date="2016-05-21" time="05:54:06 PM" minutes="1.08"
cost="0.24"/>
        <call called="815-374-4551" date="2016-05-18" time="12:29:44 PM" minutes="1.35"
cost="0.04"/>
        ... more code here ...
    </ubf:rawContents>
    <ubf:header>
        <totals>
            <minutes>697.27</minutes>
            <cost>55.64</cost>
        </totals>
    </ubf:header>
    <ubf:footer>
    </ubf:footer>
</ubf:dataGroup>

```

UBF XML Example

```

<#if usageType == "usagetalk">
    <#assign totalItems = details.call?size>
    <#list details.call as call>
        <tr>
            <td class="" data-order="${dateHelper.getDateAsNumber('yyyy-MM-dd hh:mm:ss
a', call.@date + ' ' + call.@time)?replace(' ','')}">${call.@date?date?string("MM-dd-yyyy")} + '
' + call.@time?time}</td>
            <td class="">${call.@called}</td>
            <td class="">${call.@minutes}</td>
            <td class="amount" data-order="${call.@cost?number?c}">${call.@cost}</td>
        </tr>
    </#list>
<#elseif usageType == "usagetext">
    <#assign totalItems = details.text?size>
    <#list details.text as text>
        <tr>
            <td class="" data-order="${dateHelper.getDateAsNumber('yyyy-MM-dd hh:mm:ss
a', text.@date + ' ' + text.@time)?replace(' ','')}">${text.@date?date?string("MM-dd-yyyy")} + '
' + text.@time?time}</td>
            <td class="">${text.@number}</td>
            <td class="amount" data-order="${text.@cost?number?c}">${text.@cost}</td>
        </tr>
    </#list>
<#elseif usageType == "usedata">
    <#assign totalItems = details.data?size>
    <#list details.data as data>
        <#if data.@units == "Kb">
            <#assign multiplier = 1>

```

```

        <#elseif data.@units == "Mb">
            <#assign multiplier = 1024>
        <#elseif data.@units == "Gb">
            <#assign multiplier = 1024 * 1024>
        </#if>
        <tr>
            <td class="" data-order="${dateHelper.getDateAsNumber('yyyy-MM-dd hh:mm:ss
a', data.@date + ' ' + data.@time)?replace(',', ' ')}">${data.@date?date?string("MM-dd-yyyy") + '
' + data.@time?time}</td>
            <td class="" data-order="${data.@xfer?number * multiplier}">${data.@xfer +
' ' + data.@units}(s)</td>
            <td class="amount" data-order="${data.@cost?number?c}">$$${data.@cost}</td>
        </tr>
    </#list>
</#if>

```

FreeMarker Example

This example shows the use of the **usageType** value passed in to retrieve details related only to the requested usage type, using FreeMarker **#if** and **#elseif** blocks. The example also shows the use of table column sorting, using the **data-order** attribute of the table cell **<td>**. For instance, the date/time column needs to sort by a numeric value. As such, we find the long value of the timestamp through the helper class **dateHelper**. This helper provides the **getDateAsNumber** (String format, String date) function, where the format string is the **SimpleDateFormat** string needed to parse the second parameter: the string version of the timestamp.

usage_details_csv-Smartbill-1.ftl

The Usage Details CSV template outputs the same information as the previous template; however, it is formatted as comma separated values (CSV) instead of HTML (for downloading).

8 EngageOne Smart Bill FreeMarker Internationalization

There are two methods available to internationalize templates for multi-language support. FreeMarker supports named templates for languages; as such, the language code and country code with which the user is viewing the application will be sent to FreeMarker when looking up template files. Therefore, one way to internationalize templates for multi-language support is to make different template files for each language. For example, if the user is viewing the application in English (**en** language code and **US** country code), FreeMarker will look for the following templates:

- my_template.ftl
- my_template_en.ftl
- my_template_en_US.ftl

Another method by which to internationalize templates for multi-language support is to use the **i18n** object available in the FreeMarker templates. The **i18n** service looks up translations in the **\resources\internationalization** directory of the application. This directory contains properties files named using the **_en** language code pattern. If a key is defined in one of the properties files, it can be accessed in FreeMarker. This method of internationalization allows for a single FreeMarker template that can drive multiple languages (rather than multiple templates as in the previous example).

The **i18n** syntax in FreeMarker is as follows:

```
${i18n.translate(serviceLocator, userData, "billUsage_sPopinTitlePrefix", "{Individual usage details for}")}
```

This syntax will navigate the properties files looking for the key **billUsage_sPopinTitlePrefix**; it will return the defined value, based on the language with which the user is viewing the application. The second value, **{Individual usage details for}**, is the default value shown if the key is not found. The **serviceLocator** and **userData** are internal Java objects needed to call the service, and they are available in all templates (to include the **i18n** object itself).

9 EngageOne Smart Bill Charts

EngageOne Smart Bill uses the ZingChart library for rendering charts. ZingChart is a robust charting library that uses HTML5 features. The library supports many chart types, drilldowns into additional charts, and a well-documented JSON method of configuration. The chart JSONs are built using the FreeMarker templates, which can be found in the **\linked-root\ftl\charts** directory. In addition, there is a chart theme file (located at **\linked-root\ftl\charts\themes**), which defines the base styling for chart elements. These files are a subset of the *brand.zip* file, which is uploaded in the Agent portal **Configure Application** tab (and available for download by the customer).

ZingChart provides a wealth of documentation for the JSON configurations. For more information, please visit the site at: <https://www.zingchart.com/docs/>.

The following chart templates are defined for the current product:

- bill_summary_chart-SmartBill-1.ftl
- bill_summary_drilldown_chart-SmartBill-1.ftl
- usage_summary_chart-SmartBill-1.ftl
- usage_summary_drilldown_chart-SmartBill-1.ftl
- usage_summary_drilldown_voice_chart-SmartBill-1.ftl (voice wrapper for usage drilldown)
- usage_summary_drilldown_data_chart-SmartBill-1.ftl (data wrapper for usage drilldown)
- usage_summary_drilldown_text_chart-SmartBill-1.ftl (text wrapper for usage drilldown)

chart_theme.json

The chart theme JSON file provides the base styling of all charts in the application. Styling added to the base **graph** element will apply to all charts, and the **palette** element applies colors to chart types. The palette defines five colors for the vertical bar and pie charts. If you need more colors (for example, you have customers with more than five phones), simply add more colors to this theme palette.

The **graph** elements can be overridden per chart type as shown in the JSON, by using chart types such as **bar** then overriding the **plot** element (as shown below). This action would change (or, perhaps, allow) bar charts to use this more specific plot styling, rather than the base graph plot styling. Also, the theme settings can, themselves, be overridden in the actual chart JSON formed by the FreeMarker templates.

This configuration defines the initial state of the charts. You can modify this file to change fonts, the **scale-x** and **scale-y** scale colors and lines, the tooltip styling for mouse-over information, the **value-box** layout for showing plotted values, and other options.

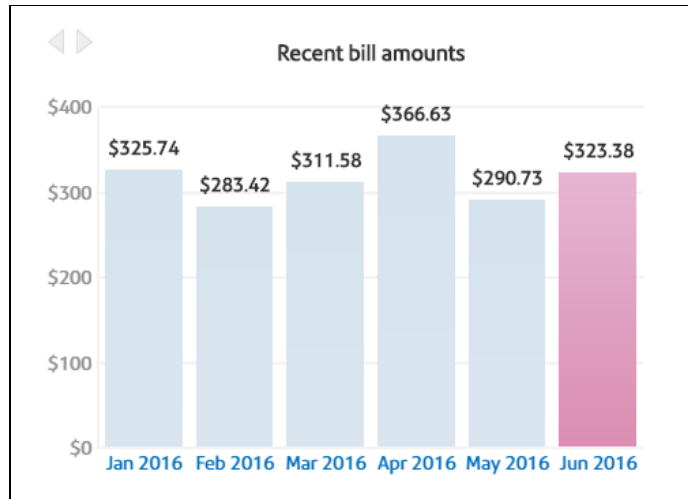
```
{
  "palette": {
    "vbar": [
      ["#B2CCDE", "#D5E3ED"],
      ["#E6B5D2", "#DA8DB1"],
      ["#DBF5DB", "#88D388"],
      ["#FCC6CA", "#F9A1A7"],
      ["#9793F4", "#5653A7"]
    ],
    "pie": [
      ["#B2CCDE", "#D5E3ED"],
      ["#E6B5D2", "#DA8DB1"],
      ["#DBF5DB", "#88D388"],
      ["#FCC6CA", "#F9A1A7"],
      ["#9793F4", "#5653A7"]
    ]
  },
  "graph": {
    "backgroundColor": "#FFFFFF",
    "title": {
      "background-color": "#FFFFFF",
      "height": "40px",
      "font-size": "16px",
      "font-color": "#333333"
    },
    "scale-x": {
      "line-color": "#F3F3F3",
      "line-width": "2px",
      "item": {
        "color": "#0170BB",
        "font-size": "14px"
      },
      "guide": {
        "visible": false
      },
      "tick": {
        "size": 0,
        "line-width": 0,
        "line-color": "#F3F3F3",
        "placement": "outer"
      }
    },
    "scale-y": {
      "line-color": "#F3F3F3",
      "line-width": "2px",
      "item": {
        "padding": "5px",
        "color": "#949494",
        "font-size": "14px"
      },
      "guide": {
        "line-color": "#F3F3F3",
        "line-style": "solid",
        "line-width": "2px"
      },
      "tick": {
        "size": 0,
        "line-width": 0,
        "line-color": "#F3F3F3",
        "placement": "outer"
      }
    },
    "scale-r": {
```

```

        "ref-angle": 270
    },
    "tooltip": {
        "background-color": "#fff",
        "border-color": "#333",
        "border-width": "1px",
        "border-radius": "8px",
        "font-color": "#333",
        "font-size": "14px",
        "thousands-separator": ",",
        "negation": "currency"
    },
    "plot": {
        "value-box": {
            "font-color": "#333333",
            "font-size": "14px"
        },
        "alpha": 1.0,
        "hover-state": {
            "border-color": "#333333",
            "border-width": 1
        },
        "detach": false
    }
},
"bar": {
    "plot": {
        "value-box": {
            "type": "all",
            "placement": "top",
            "angle": 0,
            "thousands-separator": ",",
            "negation": "currency"
        },
        "fill-angle": 90
    }
},
"pie": {
    "plot": {
        "value-box": {
            "placement": "out",
            "connected": true,
            "thousands-separator": ",",
            "negation": "currency"
        },
        "layout": "auto"
    }
},
"ring": {
    "plot": {
        "value-box": {
            "placement": "out",
            "connected": true
        },
        "layout": "auto"
    },
    "scale-r": {
        "ref-angle": 45
    }
},
"shadow": false,
"font-family": "\"Precision Sans Rg\", CenturyGothic, AppleGothic, sans-serif"
}

```

Configuration Theme Changes



Bill Summary Chart

The Bill Summary chart is shown on the Bill Summary screen as the landing chart for the page.

Note: This chart defines a **history** object as well as series **target** and **url** values, which are necessary to perform drilldowns to subsequent charts.

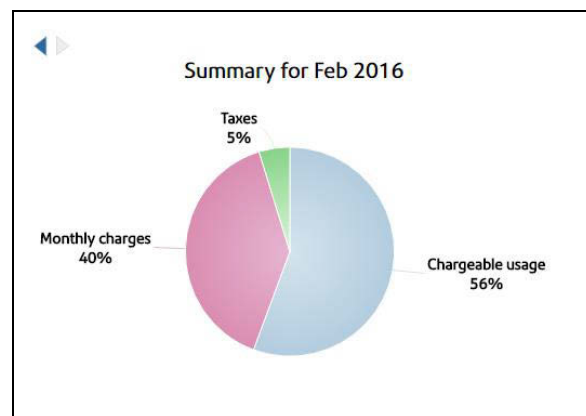
```
<#ftl ns_prefixes={"ubf":"com/sorrisotech/saas/ubf/v1/UBF"}>
<#assign summary = summarydoc["ubf:xml"]>
<!-- If viewing on mobile, only show up to the number of months defined with numMonthsMobile -->
<#assign startIndex = 0>
<#assign numMonthsMobile = 3>
<#assign totalNumMonths = summary.history.bill?size>
<#if isMobile == "true" && totalNumMonths > numMonthsMobile>
  <#assign startIndex = totalNumMonths - numMonthsMobile>
</#if>
{
  "history": {
    "y": "0px",
    "width": "40px",
    "height": "25px"
  },
  "graphset": [{
    "id": "zc",
    "type": "bar",
    "plot": {
      "value-box": {
        "text": "$%v"
      },
      "rules": [{
        "rule": "%i == %N - 1",
        "background-color-1": "#E6B5D2",
        "background-color-2": "#DA8DB1"
      }]
    },
    "tooltip": {
      "text": "$%v"
    },
    "scale-x": {
      "step": 2629743000,
      "transform": {
        "type": "date",
        "all": "%M %Y"
      },
      "values": [
        <!-- Extract bill dates as numbers for x-axis values -->
```

```

        <#list startIndex..totalNumMonths-1 as index>
            <#assign billhistory = summary.history.bill[index]>
            ${dateHelper.getDateAsNumber('yyyy-MM-dd',
billhistory.@date)?replace(',',' ')}<#if index < totalNumMonths-1>,</#if>
        </#list>
    ]
    },
    "scale-y": {
        "format": "$%v",
        "thousands-separator": ",",
        "negation": "currency",
        "multiplier": true
    },
    "title": {
        "text": "Recent bill amounts"
    },
    "series": [{
        "values": [
            <!-- Extract bill total amounts for series values -->
            <#list startIndex..totalNumMonths-1 as index>
                <#assign billhistory = summary.history.bill[index]>
                ${billhistory.@total?number?c}<#if index <
totalNumMonths-1>,</#if>
            </#list>
        ],
        <!-- Drilldown url -->
        "url": "getSummaryChart?sBillDate=%scale-key-
label&sBillingAccount=${accountNumber}&sPaymentGroup=${paymentGroup}&sLanguageCode=${languageCode
}&sChartType=bill_summary_drilldown&sIsMobile=${isMobile}",
        "target": "graph=zc"
    }]
}

```

bill_summary_chart-SmartBill-1.ftl



Bill Summary Drilldown Chart

The Bill Summary Drilldown chart is available on the Bill Summary screen; it is the drilldown chart visible when clicking a month bar in the Bill Summary bar chart.

```

<#ftl ns_prefixes={"ubf":"com/sorrisotech/saas/ubf/v1/UBF"}>
<#assign summary = summarydoc["ubf:xml"]>
{
    "graphset": [{
        "id": "zc",
        "type": "pie",
        "plot": {
            "value-box": {
                "text": "%t<br/>%npv%"
            }
        }
    }]
}

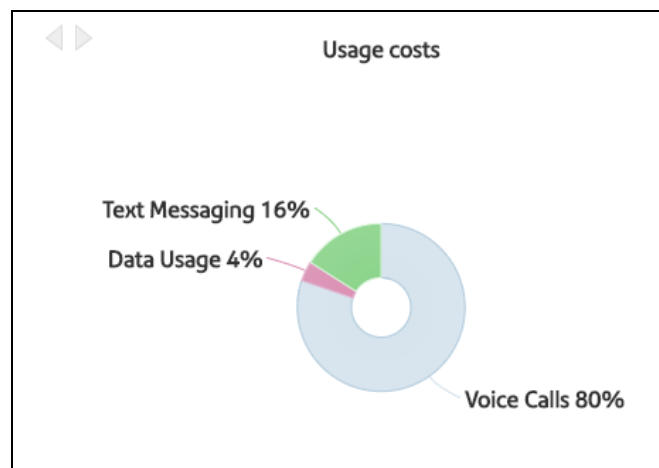
```

```

    },
    "tooltip": {
        "text": "%t<br/>${v}"
    },
    "title": {
        "text": "Summary for ${dateHelper.getDateAsString('yyyy-MM-dd', 'MMM
yyyy', summary.summary.date)}"
    },
    "series": [
        <#if summary.charges.onetime?has_content>
            {
                "values": [
                    ${summary.charges.onetime?number?c}
                ],
                "text": "${summary.charges.onetime.@label}"
            },
        </#if>
        <#if summary.charges.monthly?has_content>
            {
                "values": [
                    ${summary.charges.monthly?number?c}
                ],
                "text": "${summary.charges.monthly.@label}"
            },
        </#if>
        <#if summary.charges.usage?has_content>
            {
                "values": [
                    ${summary.charges.usage?number?c}
                ],
                "text": "${summary.charges.usage.@label}"
            },
        </#if>
        <#if summary.charges.taxes?has_content>
            {
                "values": [
                    ${summary.charges.taxes?number?c}
                ],
                "text": "${summary.charges.taxes.@label}"
            }
        </#if>
    ]
}

```

bill_summary_drilldown_chart-SmartBill-1.ftl



Usage Summary Chart

The Usage Summary chart is shown on the Usage Summary screen as the landing chart for the page.

Note: This chart defines a **history** object as well as series **target** and **url** values, which are necessary to perform drilldowns to subsequent charts.

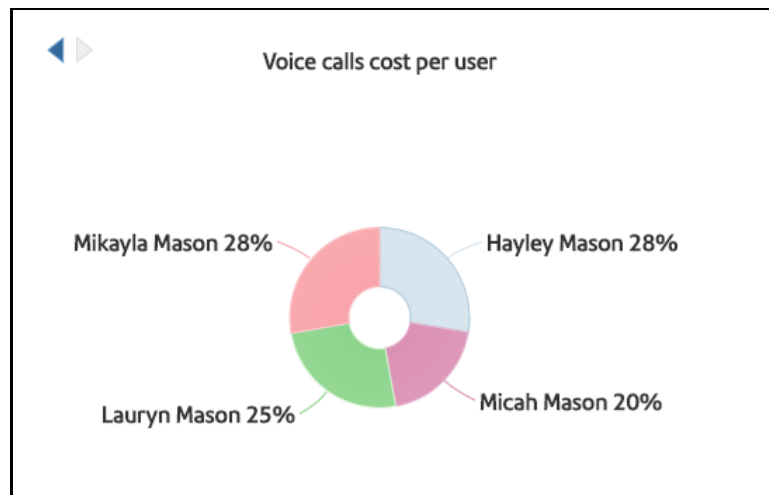
```
<#ftl ns_prefixes={"ubf":"com/sorrisotech/saas/ubf/v1/UBF"}>
<#assign summary = summarydoc["ubf:xml"]>
{
    "history": {
        "y": "0px",
        "width": "40px",
        "height": "25px"
    },
    "graphset": [{
        "id": "zc",
        "type": "ring",
        "plot": {
            "value-box": {
                "text": "%t %npv%"
            }
        },
        "tooltip": {
            "text": "%t<br/>$$v"
        },
        "title": {
            "text": "Usage costs"
        },
        "series": [
            <#assign hasVoice = false>
            <#assign hasData = false>
            <#assign hasText = false>
            <#assign voiceCost = 0.00>
            <#assign dataCost = 0.00>
            <#assign textCost = 0.00>
            <#list summary.phoneUsage.phone as phone>
                <#if phone.voice?has_content>
                    <#assign hasVoice = true>
                    <#assign voiceCost += phone.voice.@cost?number>
                </#if>
                <#if phone.data?has_content>
                    <#assign hasData = true>
                    <#assign dataCost += phone.data.@cost?number>
                </#if>
                <#if phone.text?has_content>
                    <#assign hasText = true>
                    <#assign textCost += phone.text.@cost?number>
                </#if>
            </#list>
            <#if hasVoice == true>
                {
                    "values": [
                        ${voiceCost?c}
                    ],
                    "text": "Voice Calls",
                    "url":
"getSummaryChart?sBillDate=${dateHelper.getDateAsString('yyyy-MM-dd', 'yyyyMMdd',
summary.summary.date)}&sBillingAccount=${accountNumber}&sPaymentGroup=${paymentGroup}&sLanguageCo
de=${languageCode}&sChartType=usage_summary_drilldown_voice&sIsMobile=${isMobile}",
                    "target": "graph=zc"
                },
            </#if>
            <#if hasData == true>
                {
                    "values": [
                        ${dataCost?c}
                    ],
                    "text": "Data Usage",
```

```

        "url":
"getSummaryChart?sBillDate=${dateHelper.getDateAsString('yyyy-MM-dd', 'yyyyMMdd',
summary.summary.date)}&sBillingAccount=${accountNumber}&sPaymentGroup=${paymentGroup}&sLanguageCo
de=${languageCode}&sChartType=usage_summary_drilldown_data&sIsMobile=${isMobile}",
        "target": "graph=zc"
    },
</#if>
<#if hasText == true>
    {
        "values": [
            ${textCost?c}
        ],
        "text": "Text Messaging",
        "url":
"getSummaryChart?sBillDate=${dateHelper.getDateAsString('yyyy-MM-dd', 'yyyyMMdd',
summary.summary.date)}&sBillingAccount=${accountNumber}&sPaymentGroup=${paymentGroup}&sLanguageCo
de=${languageCode}&sChartType=usage_summary_drilldown_text&sIsMobile=${isMobile}",
        "target": "graph=zc"
    }
</#if>
    ]
}
}

```

usage_summary_chart-SmartBill-1.ftl



Usage Summary Drilldown Chart

The Usage Summary Drilldown chart is available on the Usage Summary screen; it is the drilldown chart visible when clicking a usage type slice in the Usage Summary donut chart. This chart differs from the others, as it uses multiple templates with FreeMarker `#include` statements. Upon clicking a chart slice in the previous chart, the system will load the appropriate wrapper template for voice, data, or text. This template sets the `usageType` context for the main template.

```

<#ftl ns_prefixes={"ubf":"com/sorrisotech/saas/ubf/v1/UBF"}>
<#assign summary = summarydoc["ubf:xml"]>
<#assign usageType = "voice">
<#include "usage_summary_drilldown_chart-SmartBill-1-en.ftl">

```

usage_summary_drilldown_voice_chart-SmartBill-1.ftl (wrapper for voice drilldown)

```

<#ftl ns_prefixes={"ubf":"com/sorrisotech/saas/ubf/v1/UBF"}>
<#assign summary = summarydoc["ubf:xml"]>

```

```
<#assign usageType = "data">
<#include "usage_summary_drilldown_chart-SmartBill-1-en.ftl">
```

usage_summary_drilldown_data_chart-SmartBill-1.ftl (wrapper for data drilldown)

```
<#ftl ns_prefixes={"ubf":"com/sorrisotech/saas/ubf/v1/UBF"}>
<#assign summary = summarydoc["ubf:xml"]>
<#assign usageType = "text">
<#include "usage_summary_drilldown_chart-SmartBill-1-en.ftl">
```

usage_summary_drilldown_text_chart-SmartBill-1.ftl (wrapper for text drilldown)

```
{
  "graphset": [{
    "id": "zc",
    "type": "ring",
    "plot": {
      "value-box": {
        "text": "%t %npv%"
      }
    },
    "tooltip": {
      "text": "%t<br/>${%v}"
    },
    "title": {
      <#if usageType == "voice">
        "text": "Voice calls cost per user"
      <#elseif usageType == "data">
        "text": "Data usage cost per user"
      <#elseif usageType == "text">
        "text": "Text messaging cost per user"
      </#if>
    },
    "series": [
      <#list summary.phoneUsage.phone as phoneUsage>
        <#list summary.phones.phone as phone>
          <#if phone.@target == phoneUsage.@target>
            <#assign assetName = phone.summary.name>
          </#if>
          </#list>
          <#if phoneUsage.voice?has_content && usageType == "voice">
            <#assign cost = phoneUsage.voice.@cost?number>
          </#if>
          <#if phoneUsage.data?has_content && usageType == "data">
            <#assign cost = phoneUsage.data.@cost?number>
          </#if>
          <#if phoneUsage.text?has_content && usageType == "text">
            <#assign cost = phoneUsage.text.@cost?number>
          </#if>
          {
            "values": [
              ${cost?c}
            ],
            "text": "${assetName}"
          }<#if phoneUsage?index < summary.phoneUsage.phone?size-1>,</#if>
        </#list>
      ]
    }
  ]
}
```

usage_summary_drilldown_chart-SmartBill-1.ftl
(main template included from above templates)