

HP Cloud Service Automation

Customizing the Marketplace Portal



Release 4.00

Contents

- Custom Themes 2**
 - Pre-packaged Themes2
 - HP Simplified.....2
 - HP Playful3
 - HP Enterprise.....3
 - Custom Theme Option4
 - Theming Framework5
 - Glyph Icons5
- Custom Widgets 7**
 - Link Widget8
 - Featured Service Widget.....9
 - Mashup Widget10
 - Example of a generic custom widget: Clock.....10
 - Example of an HP CSA-specific custom widget: Most Expensive Subscription11
 - Example of an HP CSA-specific custom widget: Recurring Cost Breakdown12
 - Example of an HP CSA-specific custom widget: Recent 5 Subscriptions13
 - Limitations and Requirements.....16
 - Ways to Customize16

© Copyright 2014 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Restricted rights legend: Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation. AMD is a trademark of Advanced Micro Devices, Inc. Intel and Xeon are trademarks of Intel Corporation in the U.S. and other countries. Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Custom Themes

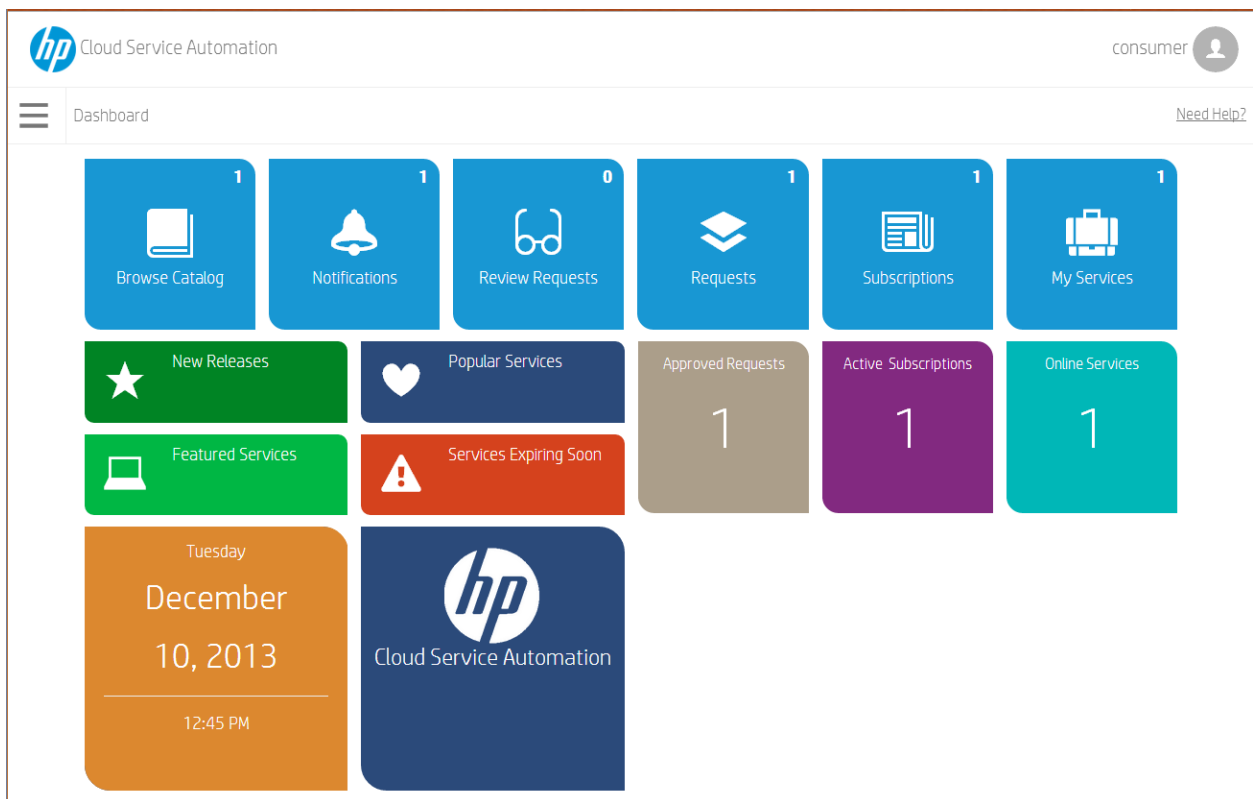
You can customize the display of the Marketplace Portal Dashboard to fulfill your organization's requirements.

Pre-packaged Themes

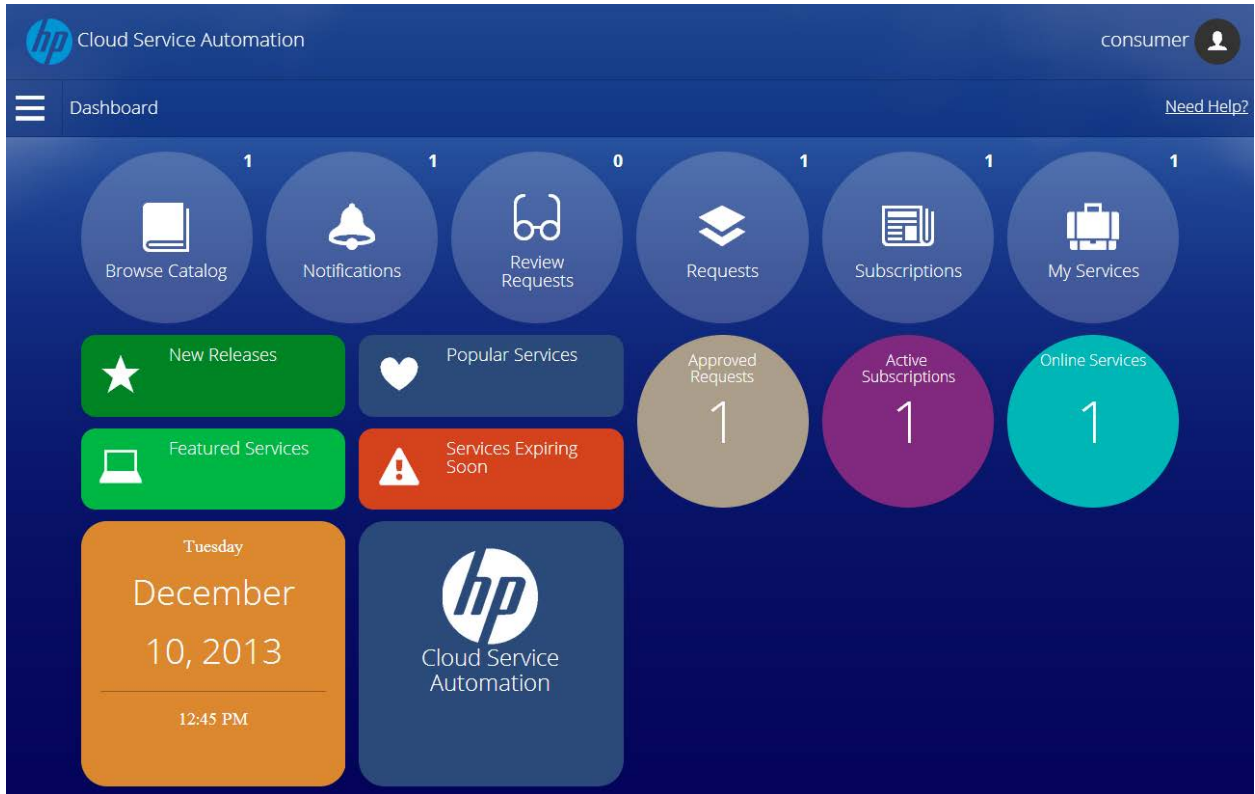
To help you easily customize your Marketplace Portal Dashboard, HP includes the following, pre-packaged themes:

- HP Simplified
- HP Playful
- HP Enterprise

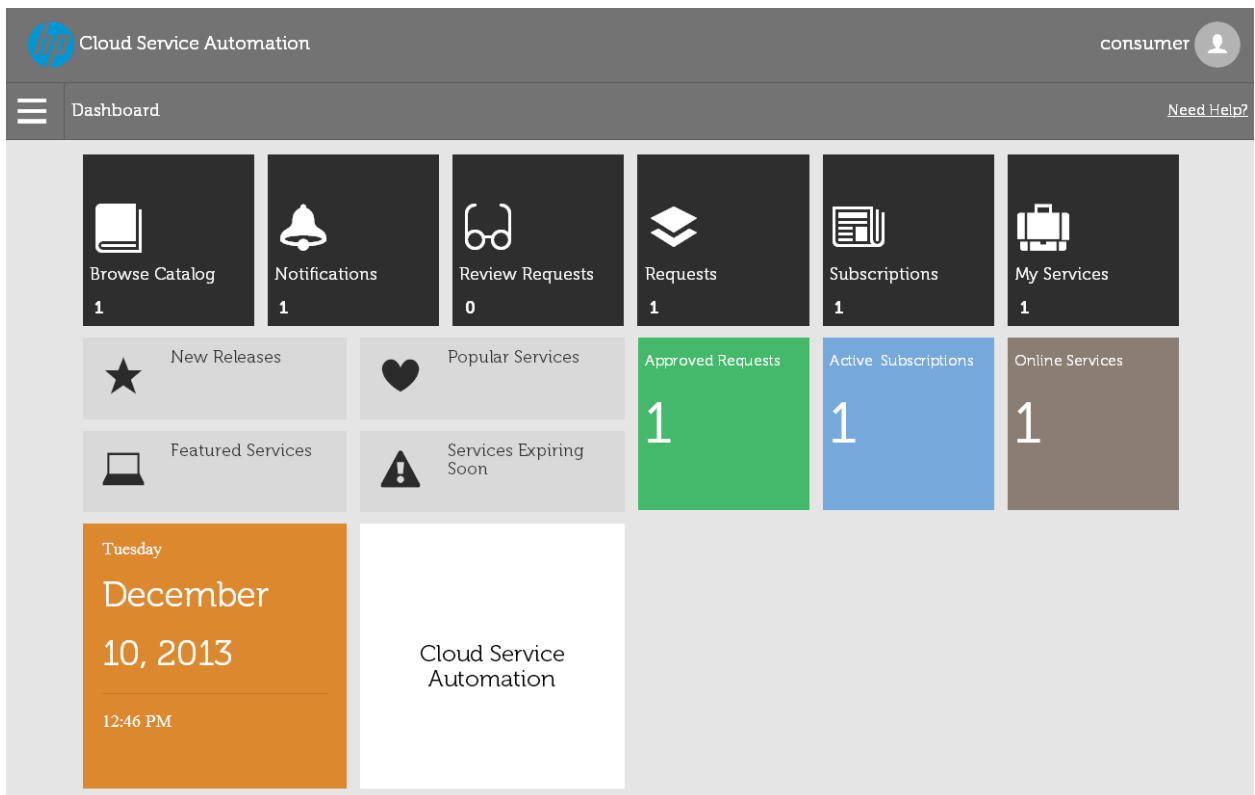
HP Simplified



HP Playful

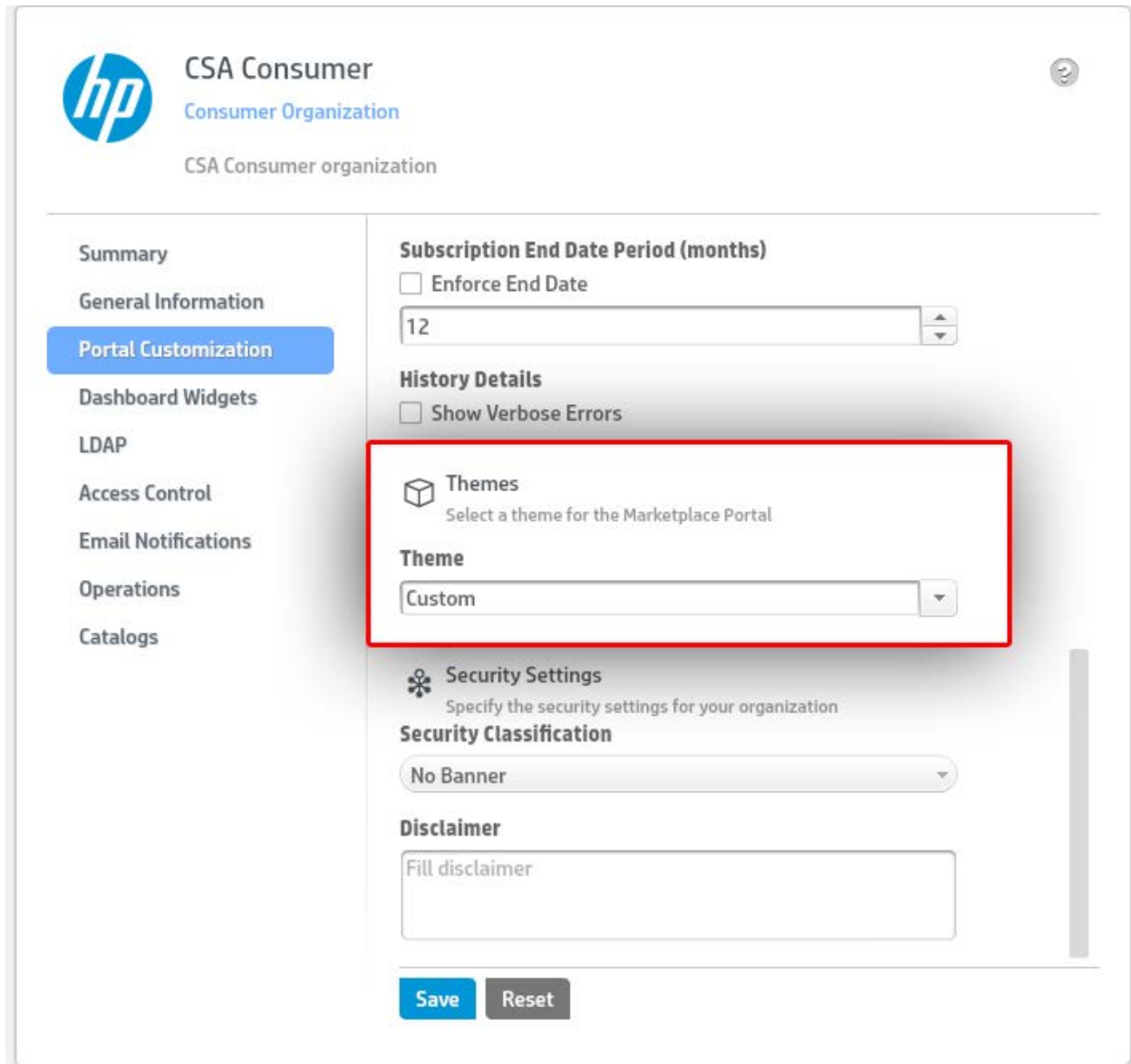


HP Enterprise



Custom Theme Option

HP CSA provides a Custom theme option, where you enter the name of your theme directory:



The screenshot shows the HP CSA Consumer Management Console interface. The left sidebar contains navigation links: Summary, General Information, Portal Customization (highlighted in blue), Dashboard Widgets, LDAP, Access Control, Email Notifications, Operations, and Catalogs. The main content area is titled 'CSA Consumer Consumer Organization' and 'CSA Consumer organization'. It features several configuration sections: 'Subscription End Date Period (months)' with an 'Enforce End Date' checkbox and a value of 12; 'History Details' with a 'Show Verbose Errors' checkbox; 'Themes' (highlighted with a red box) with a 'Select a theme for the Marketplace Portal' instruction and a dropdown menu set to 'Custom'; 'Security Settings' with a 'Security Classification' dropdown set to 'No Banner' and a 'Disclaimer' text input field. At the bottom are 'Save' and 'Reset' buttons.

The Marketplace Portal will check the Custom theme under two themes directories of the project:

```
CSA/portal/node_modules/mpp-ui/dist/themes/(theme directory)/styles/main.css
```

The Login page (IDM Service) will have its own theme structure, as follows:

```
CSA/jboss-as-7.1.1.Final/standalone/deployments/idm-service.war/themes/(theme directory)/styles/main.css
```

Example: In the CSA Management Console, if you enter **Custom** for the custom theme name, the Marketplace Portal will try to find the following file location:

```
CSA/portal/node_modules/mpp-ui/dist/themes/Custom/styles/main.css
```

The Login page (IDM service) will also try to find this location:

```
CSA/jboss-as-7.1.1.Final/standalone/deployments/idm-  
service.war/themes/Custom/styles/main.css
```

Best Practice: When you create a new theme directory, copy one of the existing theme directories and then modify the new theme directory name and the css file. To support customization, the css file must be named "main.css".

Note: Your custom directory must model the other theme directories. There's a main directory ("styles") that contains all styling components, such as .css, images, and fonts.

Pre-packaged themes can be minified to reduce or exclude whitespace. You will need an IDE/tool to help unminify the css file to make it more readable, such as <http://mrcoles.com/blog/css-unminify/>.

Theming Framework

The Marketplace Portal theming framework models the open-source project Bootstrap at <http://getbootstrap.com/>. Review the Bootstrap documentation to learn about how the Marketplace Portal's components are structured with class labels.

All CSS rules used in Bootstrap are in each provided theme, in the main.css file for each theme.

A CSS extension language that is used to create the framework is Syntactically Awesome Stylesheets (Sass). This extension language includes variables, nested rules, reusable mixins, and functions.

Glyph Icons

The Marketplace Portal styling framework provides a collection of glyph icons that can be used throughout the application and in the custom widgets.

Best Practice: Use the HTML class names shown in the following chart.

Customizing the Marketplace Portal

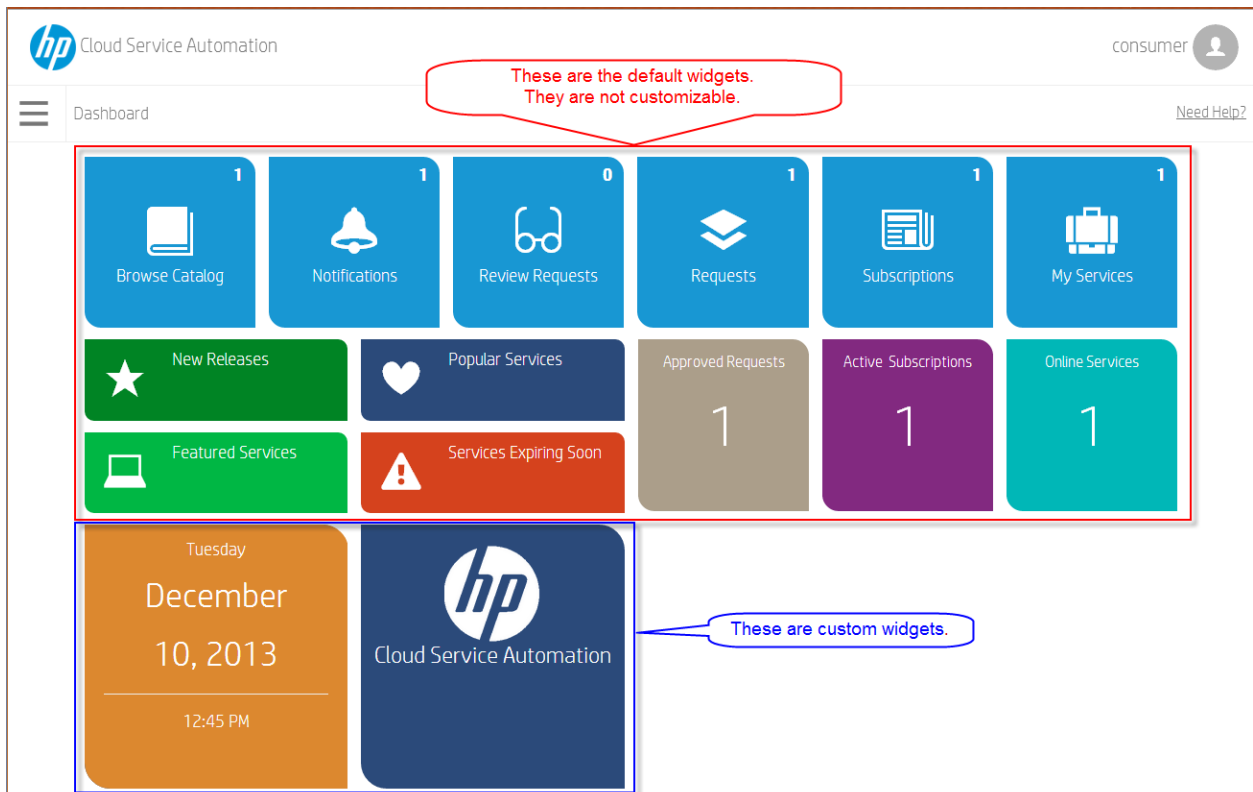
icon-zoom-out	icon-zoom-in	icon-youtube	icon-wifi
icon-web	icon-view	icon-video	icon-validation-denied
icon-validation-approve	icon-validation-alert	icon-user	icon-user-delete
icon-user-add	icon-unlock	icon-wallet	icon-iv
icon-trophy	icon-travel	icon-trash	icon-tools
icon-toolbox	icon-ticket	icon-target	icon-tag
icon-tablet	icon-surround	icon-subscription	icon-stop
icon-status	icon-star	icon-stamp	icon-spell-check
icon-speaker-mute	icon-speaker-medium	icon-speaker-max	icon-speaker-low
icon-shuffle	icon-shield	icon-settings	icon-settings-verticle
icon-settings-horizontal	icon-services	icon-server	icon-search
icon-sensors	icon-ris	icon-rocket	icon-rewind
icon-repeat	icon-relationship	icon-refresh	icon-record
icon-radar	icon-quote-start	icon-quote-end	icon-prohibited
icon-process	icon-print	icon-power	icon-play
icon-iphone	icon-pending	icon-payment	icon-pause
icon-page	icon-page-new	icon-page-multiple	icon-owner
icon-options	icon-notification	icon-network	icon-navigation
icon-mystuff	icon-mug	icon-mouse	icon-monitor
icon-microphone	icon-microphone-mute	icon-menu-expand	icon-menu-collapse
icon-mail	icon-loop	icon-layout	icon-login
icon-lock	icon-location	icon-locate	icon-loading
icon-list	icon-link	icon-likes	icon-lightbulb
icon-library	icon-legal	icon-leaf	icon-laptop
icon-labs	icon-keyboard	icon-key	icon-information
icon-inbox	icon-image	icon-hybrid	icon-hip
icon-home	icon-heart	icon-hd	icon-hd-raids
icon-hammer	icon-grid	icon-globe	icon-gift
icon-gift	icon-fox	icon-folder	icon-flame
icon-flag	icon-first-aid	icon-finder	icon-filter
icon-fast-forward	icon-facebook	icon-export	icon-export-option
icon-expand	icon-emoji-con-surprised	icon-emoji-con-sad	icon-emoji-con-happy
icon-ellipse	icon-edit	icon-drop	icon-download
icon-download-option	icon-dot	icon-donut	icon-documents
icon-document	icon-disk	icon-disconnect	icon-desktop
icon-delivery	icon-delete-circle	icon-date	icon-dashboard
icon-cube-solid	icon-cube-outline	icon-copyright	icon-contract
icon-contacts	icon-connections	icon-connection	icon-code
icon-cloud	icon-cloud-upload	icon-cloud-download	icon-close
icon-digboard	icon-dip	icon-child	icon-checkout
icon-chat	icon-chat-session	icon-chart-pie	icon-chart-line
icon-chart-bar	icon-catalog	icon-cart	icon-carousel
icon-calendar	icon-calculation	icon-business	icon-bulb
icon-bug	icon-bookmark	icon-blueprint	icon-barcode
icon-barcode-option	icon-attachment	icon-asterisk	icon-assistance
icon-assistance-option	icon-arrow-up	icon-arrow-span	icon-arrow-pin
icon-arrow-forward	icon-arrow-forward-option	icon-arrow-down	icon-arrow-back
icon-arrow-back-option	icon-archive	icon-applications	icon-alert
icon-alert-option	icon-address-book	icon-add	icon-add-document
icon-add-circle	icon-access	icon-id	icon-arrow-down-option
icon-arrow-up-option	icon-checkmark	icon-home-2	icon-stop-option
icon-play-option	icon-restart	icon-coat	icon-approvals-option
icon-review-option	icon-arrow-dash-forward	icon-arrow-dash-back	icon-arrow-line-forward
icon-arrow-line-back	icon-black-beard	icon-hat	icon-printer
icon-rocket-option	icon-traffic	icon-traffic-option	icon-plug
icon-plug-option	icon-magnet	icon-organization-option	icon-magic-wand
icon-layout	icon-bug-option	icon-sort-low	icon-sort-high
icon-sort	icon-brick	icon-link-option	icon-service-modify-fail
icon-infinite-option	icon-infinite	icon-windows	icon-apple
icon-linux	icon-date-option	icon-calendar-option	icon-checkout-complete
icon-linux	icon-date-option	icon-calendar-option	icon-checkout-complete
icon-checkout-next	icon-cart-option	icon-cart-next	icon-cart-complete
icon-measure	icon-measure-option	icon-weigh	icon-rule
icon-rule-option	icon-thermometer	icon-thermometer-option	icon-bookmark-option
icon-peace	icon-review	icon-approvals	icon-arrow-line-up
icon-arrow-line-down	icon-emoji-con-happy-2	icon-emoji-con-smile	icon-emoji-con-angry
icon-emoji-con-poo	icon-cpu	icon-subscription-add	icon-pause-option
icon-refresh-option	icon-saved	icon-square	icon-square-tip
icon-application-off	icon-applications-on	icon-application-on-option	icon-application-off-option
icon-star-half	icon-wallet	icon-basket-option	icon-basket
icon-umbrella	icon-radioactive	icon-expand-option	icon-contract-option
icon-chat-session-2	icon-conversation	icon-sun	icon-settings-option
icon-linkedin	icon-droptbox	icon-internet-explorer	icon-chrome
icon-fratix	icon-trash-option	icon-organization	icon-robot
icon-ghost	icon-diamond	icon-club	icon-saple
icon-archive-option	icon-moon	icon-phase-complete	icon-phase-start
icon-phase-crescent	icon-phase-gibson	icon-share	icon-watch
icon-pin	icon-generic-user	icon-battery-full	icon-battery-half
icon-battery-empty	icon-in-progress	icon-new	icon-urgent
icon-edit-field	icon-active	icon-active-option	icon-inactive-option
icon-inactive	icon-dog	icon-hall-option	icon-service-offline
icon-service-reserved	icon-service-transitioning	icon-service-deploy	icon-edit-option
icon-service-modify	icon-service-fail	icon-service-active	

Custom Widgets

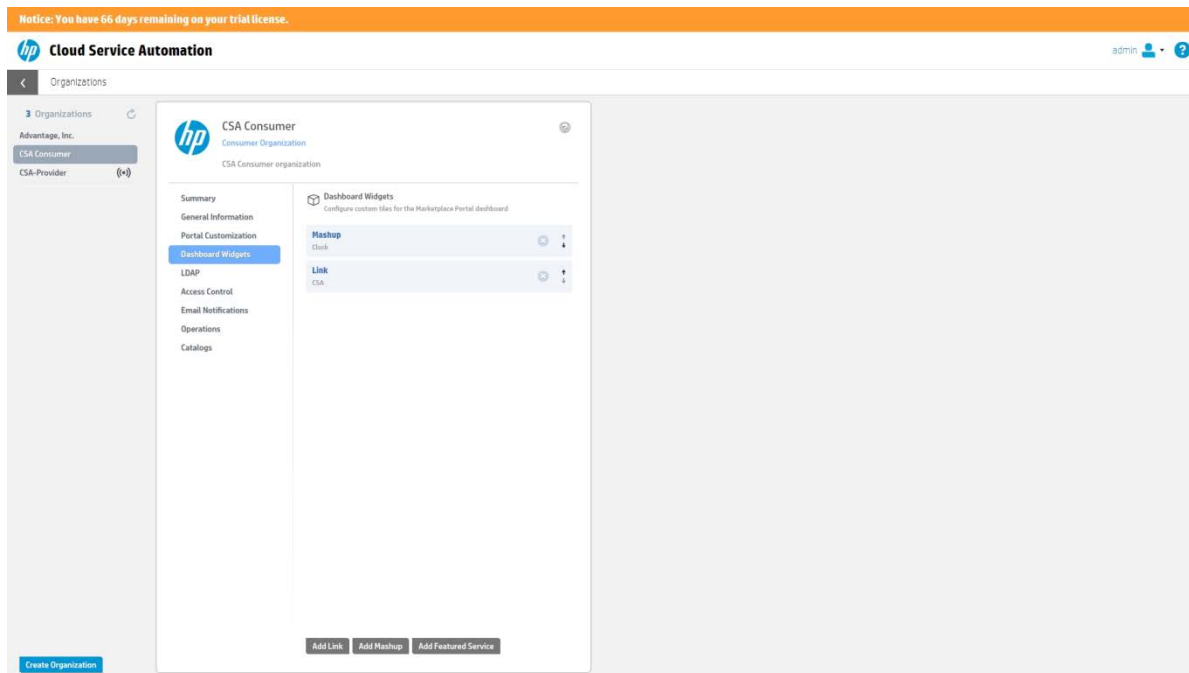
The Marketplace Portal Dashboard includes all default widgets and (optionally) custom widgets.

Default widgets: These widgets display in the first two rows of the Dashboard. You cannot customize them.

Custom widgets: You can add custom widgets to the Dashboard, such as the calendar widget and the HP link widget. You cannot add widgets to other pages or components in the Marketplace Portal, such as in the Sidebar Menu. The size of a widget is fixed. You cannot change the size of a custom widget.



Your administrator can create custom widgets and make them available to the consumer. Widgets are defined per organization and will be available for all users of that organization, as shown below.



The Marketplace Portal supports the following types of widgets:

- Link Widget
- Featured Service Widget
- Mashup Widget

Link Widget

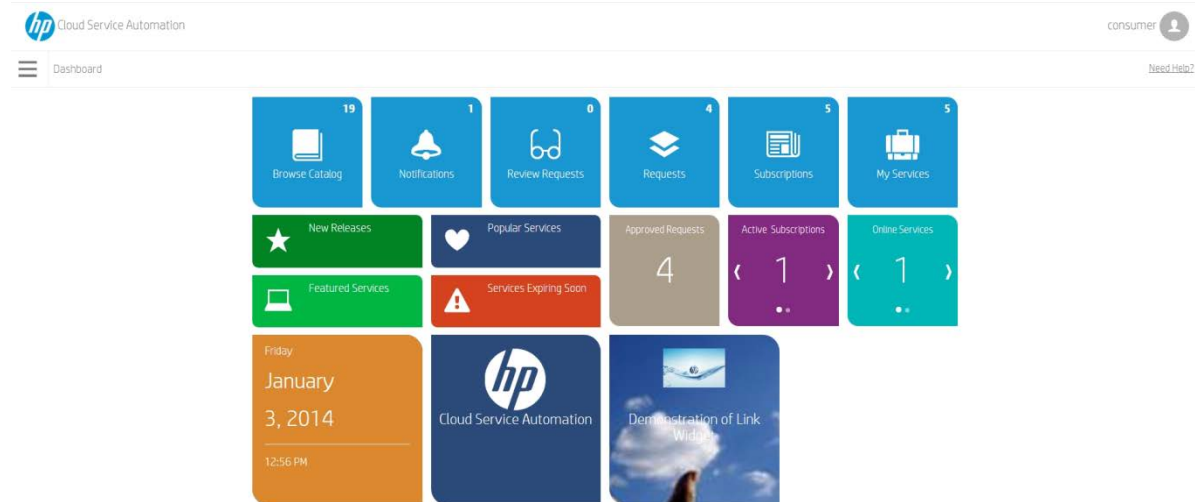
A link widget is a quick-and-easy shortcut to access a URL from the Marketplace Portal Dashboard.

To configure a link widget, review the following images—the bottom row of the Dashboard and the Edit Link Widget dialog. You can define the background image, the icon image, and the target URL.

Best Practice: Administrators can create link widgets to support custom, organization-specific links.

Link widgets display in the bottom row of the Dashboard.

Customizing the Marketplace Portal



Use the Edit Link Widget dialog to create a custom widget.

Edit Link Widget

Update the widget properties in the fields below

Name *

Title *

URL *

Icon URL

Background Image URL

Target

Featured Service Widget

Features Service widgets are helpful when there are several services available for provisioning. Services that are part of the featured service categories can be added to the Marketplace Portal Dashboard for quick-and-easy access. The Featured Service widget selects a random service from the featured services list and displays it in the Dashboard. See the Edit Featured Service Widget dialog below.

Customizing the Marketplace Portal

Edit Featured Service Widget

Update the widget properties in the fields below

Each featured service widget will result in a dashboard tile appearing in the Marketplace Portal that contains a random service offering in the featured category configured for this organization.

Name *

Note: In the current implementation, the Marketplace Portal does not support pinning a service. For the same consumer, this widget may show different services upon refresh.

The screenshot shows the HP Cloud Service Automation dashboard. At the top, there is a navigation bar with the HP logo, 'Cloud Service Automation', a user profile icon labeled 'consumer', and a 'Need Help?' link. Below the navigation bar is a grid of widgets. The first row includes 'Browse Catalog' (19), 'Notifications' (1), 'Review Requests' (0), 'Requests' (4), 'Subscriptions' (5), and 'My Services' (5). The second row includes 'New Releases', 'Popular Services', 'Approved Requests' (4), 'Active Subscriptions' (1), and 'Online Services' (1). The third row includes 'Featured Services', 'Services Expiring Soon', and a large number '4'. The bottom row includes a calendar for Friday, January 3, 2014, at 1:12 PM; the HP logo; a 'Demonstration of Link Widget'; and a widget for 'Advantage - Corporate Email Server' which is highlighted with a red box.

Mashup Widget

Mashup widgets are used to display a calendar, a clock, and other helpful, HP CSA-specific information in the Dashboard.

There are two categories of mashup widgets:

- **Generic custom widgets:** These widgets show generic information. This information excludes HP CSA-specific data.
- **HP CSA-specific custom widgets:** These widgets retrieve data from the HP CSA server the subscriber is using and then displays that data,

Example of a generic custom widget: Clock

```
<div style="background-color:#DC882F;background-image: url('');  
color: #FFFFFF;font-family: HPSimplified;  
font-size: 18px;background-repeat: no-repeat;  
padding-left: 20px;padding-top: 20px;padding-right: 20px;padding-bottom: 20px;line-height:
```

Customizing the Marketplace Portal

```
.7; ">
<p id="weekday" style="font-size: 18px;"></p>
<h1 id="month" style="font-size: 36px;"></h1>
<h1 style="font-size: 36px;"><span id="day"></span>, <span id="year"></span></h1>
<hr style="color: #FFF;" />
<p style="font-size: 18px;"><span id="hmm"></span> <span id="ampm"></span></p>
</div>

<script type="text/javascript">
function startClock() {
    // inspired by http://www.tristanwaddington.com/2010/08/javascript-clock/
    var months = [ "January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December" ];
    var weekdays = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

    function updateClock() {
        var now = new Date();
        var hours = now.getHours();
        var minutes = now.getMinutes();
        var ampm = "AM";
        // Convert hours to AM/PM format - see http://en.wikipedia.org/wiki/12-hour_clock
        if (Math.floor(hours/12)==1) {
            ampm = "PM";
            hours = hours - 12;
        }
        if (hours == 0) {
            hours = 12;
        }
        var ms = "" + minutes;
        if (ms.length == 1) {
            ms = "0" + ms;
        }
        document.getElementById("weekday").textContent = weekdays[now.getDay()];
        document.getElementById("month").textContent = months[now.getMonth()];
        document.getElementById("day").textContent = now.getDate();
        document.getElementById("year").textContent = now.getFullYear();
        document.getElementById("hmm").textContent = hours + ":" + ms;
        document.getElementById("ampm").textContent = ampm;
    }
    updateClock();
    setInterval(updateClock, 1000);
}
startClock();
</script>
```

Example of an HP CSA-specific custom widget: Most Expensive Subscription

```
<div style="background-color:#F79032;padding:10px;">
<h4>Most Expensive Subscription</h4>
<hr />
<p id="subs"></p>
<p id="init"></p>
<p id="recurring"></p>
</div>

<script type="text/javascript">
$.ajax({
```

Customizing the Marketplace Portal

```
url: "api/subscription?count=1&offset=0&sort=expensive&status=Active",
success: function(data){
debugger;

    document.getElementById("subs").textContent = data[0].name;
    document.getElementById("init").textContent = data[0].initPrice.price + " "
+data[0].initPrice.currency;
    document.getElementById("recurring").textContent = data[0].recurringPrice.price +
" " +data[0].recurringPrice.currency + " " +data[0].recurringPrice.basedOn;
    }
});
</script>
```

Example of an HP CSA-specific custom widget: Recurring Cost Breakdown

```
<div style="background-color:#46B525; padding:10px;">
<p id="subs"></p>
<hr/>
<h4>Recurring Cost Breakdown</h4>
<p id="yearly"></p>
<p id="monthly"></p>
<p id="hourly"></p>
<hr style="color: #FFF;"/>
</div>
<script type="text/javascript">
var hourlyRecurring =0;
var monthlyRecurring =0;
var yearlyRecurring = 0;
var count;
$.ajax({
url: "api/subscription?count=500&offset=0&sort=newest&status=Active",
success: function(data){
count = data.length;
data.forEach(function (item) {
if(item.recurringPrice.basedOn === 'Yearly' && item.recurringPrice.currency === 'USD')
yearlyRecurring += item.recurringPrice.price;
if(item.recurringPrice.basedOn === 'Monthly' && item.recurringPrice.currency === 'USD')
monthlyRecurring += item.recurringPrice.price;
if(item.recurringPrice.basedOn === 'Hourly' && item.recurringPrice.currency === 'USD')
hourlyRecurring += item.recurringPrice.price;
});
document.getElementById("subs").textContent = count+ " Subscriptions";
```

Customizing the Marketplace Portal

```
document.getElementById("yearly").textContent = "Yearly Recurring Costs: " +yearlyRecurring
+"USD";

document.getElementById("monthly").textContent = "Monthly Recurring Costs: "
+monthlyRecurring +"USD";

document.getElementById("hourly").textContent = "Hourly Recurring Costs: " +hourlyRecurring
+"USD";
}
})
</script>
```

Example of an HP CSA-specific custom widget: Recent 5 Subscriptions

```
<div id="recentSubs">
    <h4>Recent Subscriptions</h4>
    <ul id="list_recent_subscriptions">
    </ul>
</div>
<script type="text/javascript">
/// here you can put all the code that you want to run after page load
var $list = $('#list_recent_subscriptions');

function getSubscriptions(options) {
$.ajax({
    type: "GET",
    url: "/api/subscription?count=5&offset=0&sort=newest&status=Active",
    success: function(data, textStatus, jqXHR) {
        if (options.success) {
            options.success(data);
        }
    },
    error: function(jqXHR, textStatus, errorThrown) {
        window.alert(textStatus);
    }
});
}

function render() {
$list.empty();
getSubscriptions({
success: function(data) {
    if (data && data.length > 0) {
```

Customizing the Marketplace Portal

```
    for (var i = 0; i < data.length; i++) {
        var name = data[i].name;
        var id = data[i].id;
        var url = "/subscription/" + id;
        if(i < data.length-1){
            $list.append("<li><a target='_parent' href='" + url + "'" + data[i].name +
"</a></li>");
        }
        else{
            $list.append("<li><a target='_parent' href='" + url + "'" + data[i].name +
"</a></li>");
        }
    }
} else {
    $list.append("<li>No Recent Subscriptions Available</li>");
}
}
});
}
render();
</script>
<style type="text/css">
#recentSubs {
    font-family:HPSimplified,Helvetica,Arial,sans-serif;
    background-color: #7e7e7e;
    padding: 10px;
    color: white;
    font-weight: 500;
}
#recentSubs > h4 {
    margin: 0px;
    padding-bottom: 10px;
    border-bottom: 1px solid white;
    font-weight: 500;
}
#recentSubs >ul {
    padding-left: 0px;
    margin: 0px;
}
#recentSubs > ul > li {
    display: block;
```

Customizing the Marketplace Portal

```
padding-top: 10px;
padding-bottom: 10px;
border-bottom: 1px solid rgba(255,255,255,0.5);
}
#recentSubs > ul > li:last-child {
border-bottom: none;
}
#recentSubs > a {
color: white;
text-decoration: none;
}
#recentSubs > a:visited {
color: white;
}
#recentSubs > a:hover, #recentSubs > a:focus {
color: #1897d3;
}
</style>
```

Note: The key function in the above HP CSA-specific widgets is the url: "/api/subscription?count=5&offset=0&sort=newest&status=Active".

The options to "/api/subscription?" can also be seen on the URL when the Subscriptions page is visited using the Market place Portal, as shown below.


https://csademo4.cdllocal:8080/api/subscription?sort=cheap&category=ACCESSORY&status=ACTIVE

Cloud Service Automation

Subscriptions (1)

Search

Accessory Active Least Expensive

 **Erick Medical subscription**

Active Subscription

Subscribe to Medical Response

consumer

December 18, 2013 - June 18, 2014

\$0.00 | \$0.00 /y

Similarly, you can use following to get additional widgets:

Requests: "/api/request?"

Catalog entries: "/api/service?"

Notification: "/api/notification?"

Approvals: "/api/approval?"

My Services: "/api/myservice?"

Limitations and Requirements

From a performance perspective, more widgets will take more time to get the data and update. This will result in slow performance of rendering the Marketplace Portal Dashboard. This will impact following actions:

- Login time.
- Refreshing the Dashboard view.
- Returning back to the Dashboard.

When you create custom widgets, be aware of the following limitations and requirements:

- Variables are shared between the widgets. If you are using the same variable name, such as 'list', in multiple widgets, this might cause unexpected behavior because the values might not be accurate.
- Widgets cannot be shared between organizations. The widget must be created for each organization.
- If widgets are deleted from the organization, all information will be lost. As a best practice, keep a copy of the widget and use the 'disable' functions.

Ways to Customize

There are several ways to customize widgets in the Marketplace Portal. As long as you adhere to the security restrictions around running external scripts on HTTPS you can implement widgets anyway you want. However, there are ways that are simpler than others.

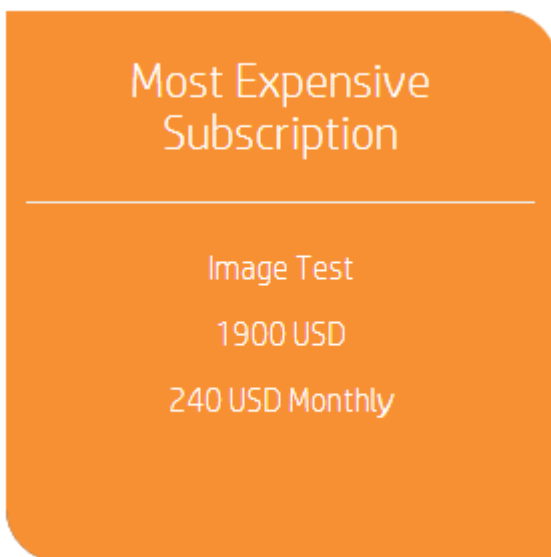
For convenience, all custom widgets have access to jQuery and SugarJS. When you use the `jquery.ajax()` function to make calls to the Marketplace Portal APIs, the user session authorization token will be automatically included. You do not need to extract the user data from the session. A sample script that would return data from CSA could be as simple as:

```
<div id="expensiveSubsDiv" style="background-color:#F79032;padding:10px;">
<h4>Most Expensive Subscription</h4>
<hr/>
<p id="sub"></p>
<p id="init"></p>
<p id="recur"></p>
</div>
<script type="text/javascript">
$.ajax({
  url: "api/subscription?count=1&offset=0&sort=expensive",
  success: function(data){
    data.forEach(function (item){ //data for this API returns an Array & .forEach is a helper
function provided by SugarJS, in this case we only request one item (count=1) so it will only
execute once.
      document.getElementById("sub").textContent = item.name;
      document.getElementById("init").textContent = item.initPrice.price + " "
```


Customizing the Marketplace Portal

```
+item.initPrice.currency;  
        document.getElementById("recur").textContent = item.recurringPrice.price + " "  
+item.recurringPrice.currency + " " +item.recurringPrice.basedOn;  
    });  
}  
})  
</script>
```

Note: The sample code above creates a widget similar to this:



Optionally, you can extend this code to include an inline `<style>` tag for more complex CSS cases:

```
</script>  
<style type="text/css">  
    #expensiveSubsDiv > p {  
        color: black  
    }  
</style>
```

When you append a style tag, wrap your widget in a top level div, such as `<div id="expensiveSubsDiv">`. This will ensure that your styling does not contaminate elements outside of your widget.

Using jQuery, combined with SugarJS, should make developing widgets relatively simple, even when you are trying to use data from HP Public APIs, as shown in the example above. However, there are limitations when running external scripts in an HTTPS environment. Do not expect to be able to copy and paste an iframe you found on a widget-generating website to work when running in an HTTPS environment. This is not supported at the browser level.

Note: For more information about Marketplace Portal API documentation, see <http://rndwiki.atlanta.hp.com/confluence/display/HPUI/MPP+API>.

