Legacy - Android Interface Customization

The Alchemer Mobile Android SDK is built to automatically use your app's theme. When your app is using an AppCompat or a Material theme, we will inherit all of your app's styles. That means that you shouldn't need to do anything to modify our SDK to suit your app.

If your app is not using an AppCompat or Material design theme, then our SDK will retain the default styling used in all of our screenshots. However, it is still easy to override these styles if you like.

Using an AppCompat or Material Theme

If you are specifying an AppCompat or Material theme for your app, we will inherit all of your styles by default. The code necessary to do this is located in your `AndroidManifest.xml` , and will look something like this:

```
<application android:name=".YourApplication"
        android:label="Your App"
        android:icon="@drawable/icon"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
```

Specifying Your Theme Type

Alchemer Mobile needs to know one piece of information in order to provide the best user experience. We need you to tell us whether you are using a light or dark theme, with a light or dark ActionBar. To do so, follow one of these four examples:

Light Theme with Dark ActionBar

```
 <!-- If your app is using a Light Theme with a Dark ActionBar: -->
<style name="YourAppTheme" parent="Theme.AppCompat.Light.DarkActionBar"/>
<!-- No Action needed, as this is our default theme -->
```

Light Theme with Light ActionBar

```
 <!-- If your app is using a Light Theme with a Light ActionBar: -->
<style name="YourAppTheme" parent="Theme.AppCompat.Light"/>
<!-- Add this to make sure Apptentive is as well -->
<style name="ApptentiveTheme" parent="ApptentiveDefaultTheme.Light"/>
```

Dark Theme with Dark ActionBar

```
 <!-- If your app is using a Dark Theme with a Dark ActionBar: -->
<style name="YourAppTheme" parent="Theme.AppCompat"/>
<!-- Add this to make sure Apptentive is as well -->
<style name="ApptentiveTheme" parent="ApptentiveDefaultTheme.Dark"/>
```

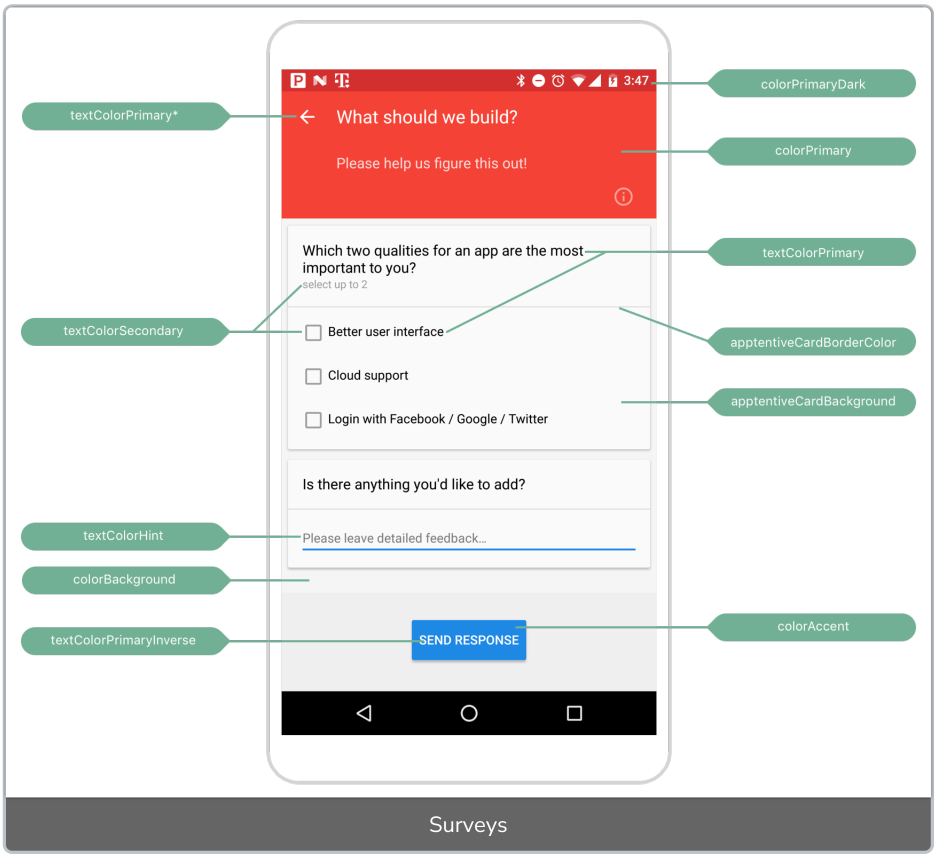
Dark Theme with Light ActionBar

```
<!-- If your app is using a Dark Theme with a Light ActionBar.
    There is no Material theme for this, but if you have set your colorPrimary
    to a dark color to make this happen: -->
<style name="YourAppTheme" parent="Theme.AppCompat">
  <item name="colorPrimary">#E6E6E6</item>
  <item name="colorPrimaryDark">#FFFFFF</item>
</style>
<!-- Add this to make sure Apptentive is as well -->
<style name="ApptentiveTheme" parent="ApptentiveDefaultTheme.Dark.LightActionBar"/>
```

> We do not support Dark Mode at this time. If you are using a Dark Mode theme, it could cause issues with how Alchemer Mobile is displayed. To solve these issues, please see the Overriding Our Styles section below to modify how Alchemer Mobile looks when used in Dark Mode.

What Styles We Use

When you use the standard Android or AppCompat theming styles, such as `colorPrimary`, and `android:textColorPrimary`, our UI will also use those in the appropriate places. Here are some screenshots showing you which styles we use in our UI.



What should we build?

Please help us figure this out!

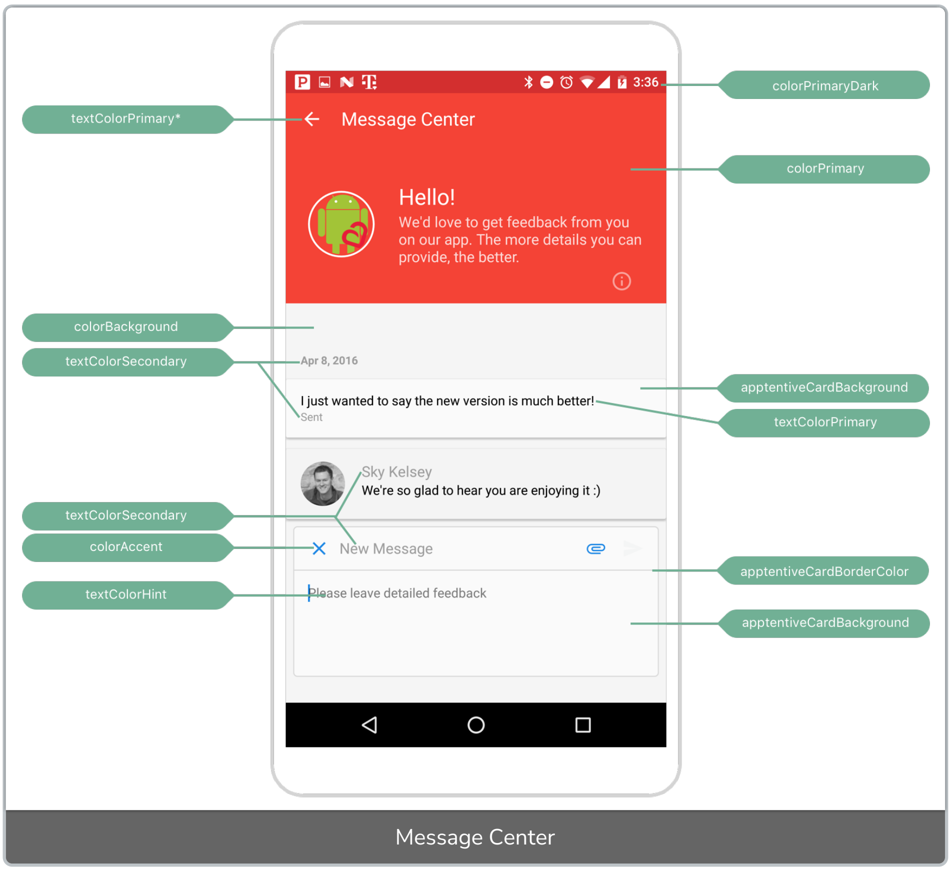
Which two qualities for an app are the most important to you?
select up to 2

□ Better user interface
□ Cloud support
□ Login with Facebook / Google / Twitter

Is there anything you'd like to add?

Please leave detailed feedback...

Surveys



Message Center

Hello!

We'd love to get feedback from you on our app. The more details you can provide, the better.

Apr 8, 2016

I just wanted to say the new version is much better!

Sent

Sky Kelsey

We're so glad to hear you are enjoying it :)

New Message

Please leave detailed feedback

Message Center

Styles specified in the “extended ActionBar/Toolbar” part of our UI will use styles from the `ThemeOverlay` that is used for the regular ActionBar. For example, if you are using a Light theme with Dark ActionBar, the `textColorPrimary` in the ActionBar will be light, when `textColorPrimary` in the rest of the UI will be dark, just like in your app.

Overriding Our Styles

If you have applied the correct base style to suit your app’s theme, and still don’t feel our SDK’s UI looks good, you can override specific styles within our SDK by specifying an `ApptentiveThemeOverride` theme. Our override theme is applied to our UI after all other styles, so placing styles in the override theme guarantees our UI will use them.

Here is an example override theme that changes a few of the styles our SDK uses.

```
<style name="ApptentiveThemeOverride">
  <item name="colorPrimary">#FFEB3B</item>
  <item name="colorPrimaryDark">#FDD835</item>
  <item name="colorBackground">#FFFFFF</item>
  <item name="apptentiveCardBorderColor">#757575</item>
</style>
```

Tip: If you aren’t using an AppCompat or Material theme, then specifying the

`ApptentiveThemeOverride` style will also give you an opportunity to customize our UI.

Overriding the Toolbar

The Toolbar in Message Center can be customized by overriding and adding to the appropriate toolbar style for your theme, and then assigning it to the `apptentiveToolbarTheme` custom attribute in your override theme. Choose from:

- Dark Toolbar: `ThemeOverlay.AppCompat.Dark.ActionBar`
- Light Toolbar: `ThemeOverlay.AppCompat.Light`

Here is how you would change the color of the control buttons:

```
<style name="ApptentiveThemeOverride">
  <item name="apptentiveToolbarTheme">@style/ApptentiveToolbarOverride</item>
</style>

<style name="ApptentiveToolbarOverride" parent="ThemeOverlay.AppCompat.ActionBar">
  <item name="colorControlNormal">#FF00FF</item>
</style>
```

Troubleshooting

Below are several common scenarios and suggested fixes.

- If you are using Material Design components and they are not working well with an AppCompat theme, we suggest you to override styles for those selected UI-components. For example, to override the look and feel of a button:

```
<style name="ApptentiveButtonStyle" parent="Apptentive.Widget.Button.Borderless">
    <item name="android:textColor">@color/colorPrimary</item>
</style>
<style name="ApptentiveThemeOverride">
    ...
    <item name="apptentiveInteractionButtonStyle">@style/ApptentiveButtonStyle</item>
</style>
```

- If you're using a global property and it's impacting Alchemer Mobile's UI, we suggest you override the property inside our ApptentiveThemeOverride style. For example:

```
<style name="ApptentiveThemeOverride">
    ...
    <item name="colorPrimary">#FFFFFF</item>
</style>
```

Other Styling Needs

Our Android SDK is 100% open source, and as such, you can modify the UI and underlying code at any time. Keep in mind though, that it may be more difficult to upgrade to newer versions of our SDK if you modify it significantly.

Related Articles