**Converting from simple card to APL**

**Configure your skill to support the Alexa.Presentation.APL interface**

1. If you are updating an Alexa-Hosted skill, remember to add Alexa Presentation Language in the Build > Interfaces tab, and rebuild the model.
2. If using lambda, update the manifest.apis.custom.interfaces array in your skill-package / skill.json code by adding interfaces. See Part 2

If you want to create your own Alexa skill, this is how you do it.

**Part 1. Alexa Hosted**

Create a new Alexa Hosted skill. Check the invocation. We’ll add a simple card, then update this to APL.

To use a simple card, add the following at the top of your code

from ask\_sdk\_model.ui import SimpleCard

Change the speak and return section in the launch code to:

speak\_output = "This displays a simple card, say hello to see the a.p.l."

card\_title = "Here's a simple card – this is the title"

card\_text = speak\_output

return (

handler\_input.response\_builder

.speak(speak\_output)

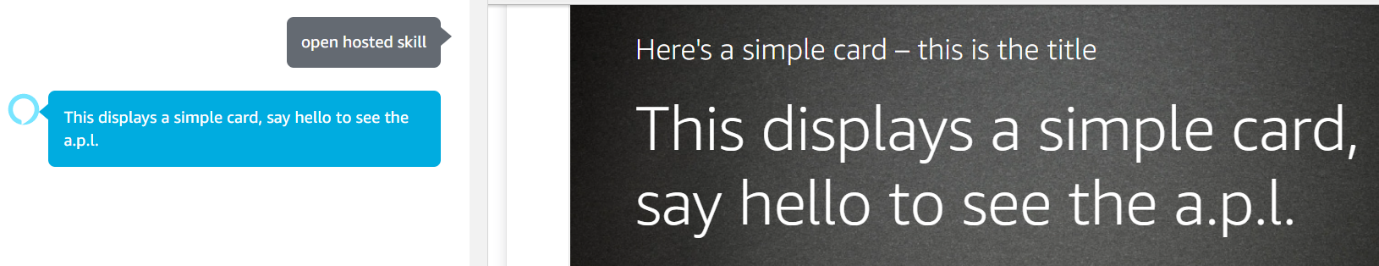
.ask(speak\_output)

.set\_card(SimpleCard(card\_title, card\_text))

.response

)

Deploy your code and check the simple card works.

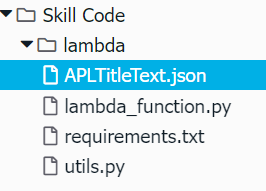


**If you haven’t done so, make sure you add APL to your model**. Build > Interfaces > Select APL. Save interfaces and rebuild the model

We’ll start with a simple APL json file that displays a title text and some main text.

You can create your APL using the Multimodal Authoring tool in the developer console, build tab. But we’ll just use the editor.

Create a new file called APLTitleText.json in the lambda folder.



This is the json code:

{

"type": "APL",

"version": "2022.1",

"settings": {},

"theme": "dark",

"mainTemplate": {

"parameters": [

"payload"

],

"items": [

{

"alignItems": "center",

"type": "Container",

"width": "100%",

"height": "100%",

"items": [

{

"alignItems": "center",

"type": "Container",

"width": "100%",

"height": "20%",

"items": [

{

"text": "Here's the title",

"fontSize": "32dp",

"fontStyle": "normal",

"type": "Text",

"width": "300dp",

"height": "64dp"

}

]

},

{

"alignItems": "center",

"type": "Container",

"width": "100%",

"height": "80%",

"items": [

{

"text": "Here's the main text in yellow",

"fontSize": "64dp",

"fontStyle": "normal",

"type": "Text",

"width": "300dp",

"height": "64dp",

"color": "#FFFF00"

}

]

}

]

}

]

}

}

This has one container the full height and width of the screen, containing two more containers – both are full width, but one at the top taking 20% of the screen and the other underneath it taking 80% of the screen. They both have a Text component. You can see the parameters of the Text components.

You can use Build > Multimodal Responses to create APL.

Now change the HelloWorld code to display the APL

Add the following at the top of your lambda\_function code

from ask\_sdk\_model.interfaces.alexa.presentation.apl import RenderDocumentDirective as APLRenderDocumentDirective

Add also this at the top:

import json

def \_load\_apl\_document(file\_path):

with open(file\_path) as f:

return json.load(f)

We’ll use this to load our APL json files.

Now change the Hello\_World def handle code to

def handle(self, handler\_input):

# type: (HandlerInput) -> Response

logger.info("In Hello world intent")

speak\_output = "Simple a.p.l container with title and main text"

return (

handler\_input.response\_builder

.speak(speak\_output)

.add\_directive(

APLRenderDocumentDirective(

token="pagerToken",

document=\_load\_apl\_document("APLTitleText.json"),

datasources={}

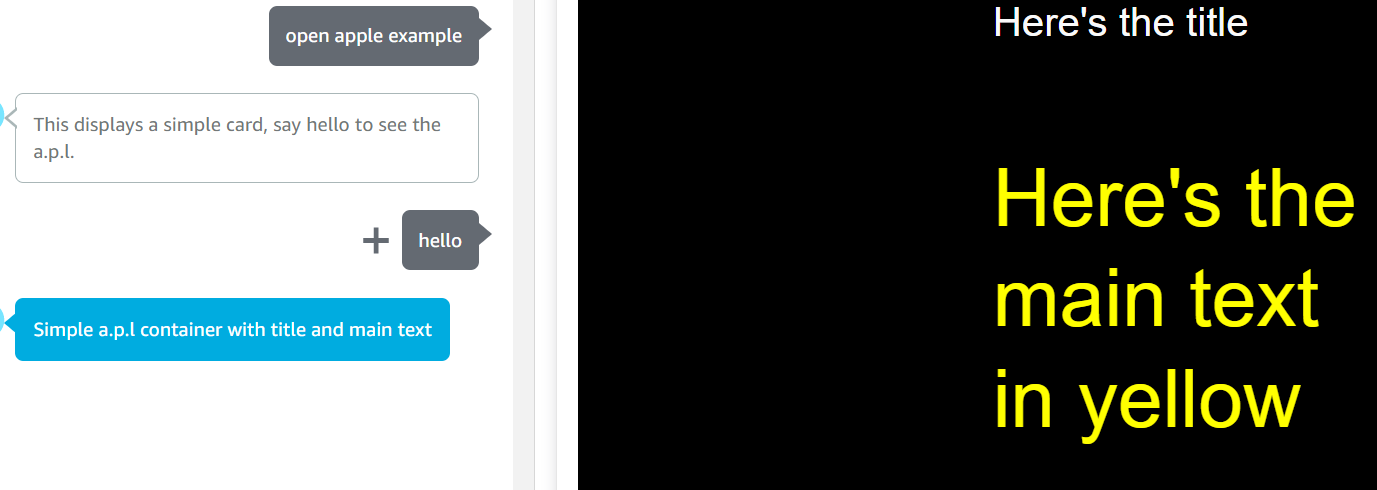
)

)

.response

)

Save, deploy and test. A simple card should be displayed on launch and this when you type/say hello:



If it doesn’t work, check the cloudwatch logs. (Click code > Cloudwatch logs, and choose your locale)

That should let you change your simple card to APL. You can change font size and type using

"fontStyle": "normal",

"fontFamily": "arial",

Alexa’s fontFamily example is "times new roman, times, georgia, serif". fontStyle is either normal (the default) or italic.

See <https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-text.html>

Now let’s add an Alexa Headline

**Alexa Headline Template**

This displays short text and cand include header footer and background.

Alexa Headline is part of the Alexa layouts package, currently version 1.6.0.

<https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/apl-alexa-headline-layout.html>

To use this, you import the Alexa layouts package to your APL. Here’s the basic APL:

{

"type": "APL",

"version": "2022.1",

"license": "Copyright 2021 Amazon.com",

"theme": "dark",

"import": [

{

"name": "alexa-layouts",

"version": "1.6.0"

}

],

"mainTemplate": {

"parameters": [

"payload"

],

"item": [

{

"type": "AlexaHeadline",

"id": "PlantHeadline",

"primaryText": "primaryText",

"secondaryText": "secondaryText",

"headerBackButton": false,

"headerAttributionPrimacy": true,

"footerHintText": "footerHintText",

"backgroundColor": "@colorRed800",

"backgroundColorOverlay": false,

"headerTitle": "headerTitle",

"headerSubtitle": "subtitle"

}

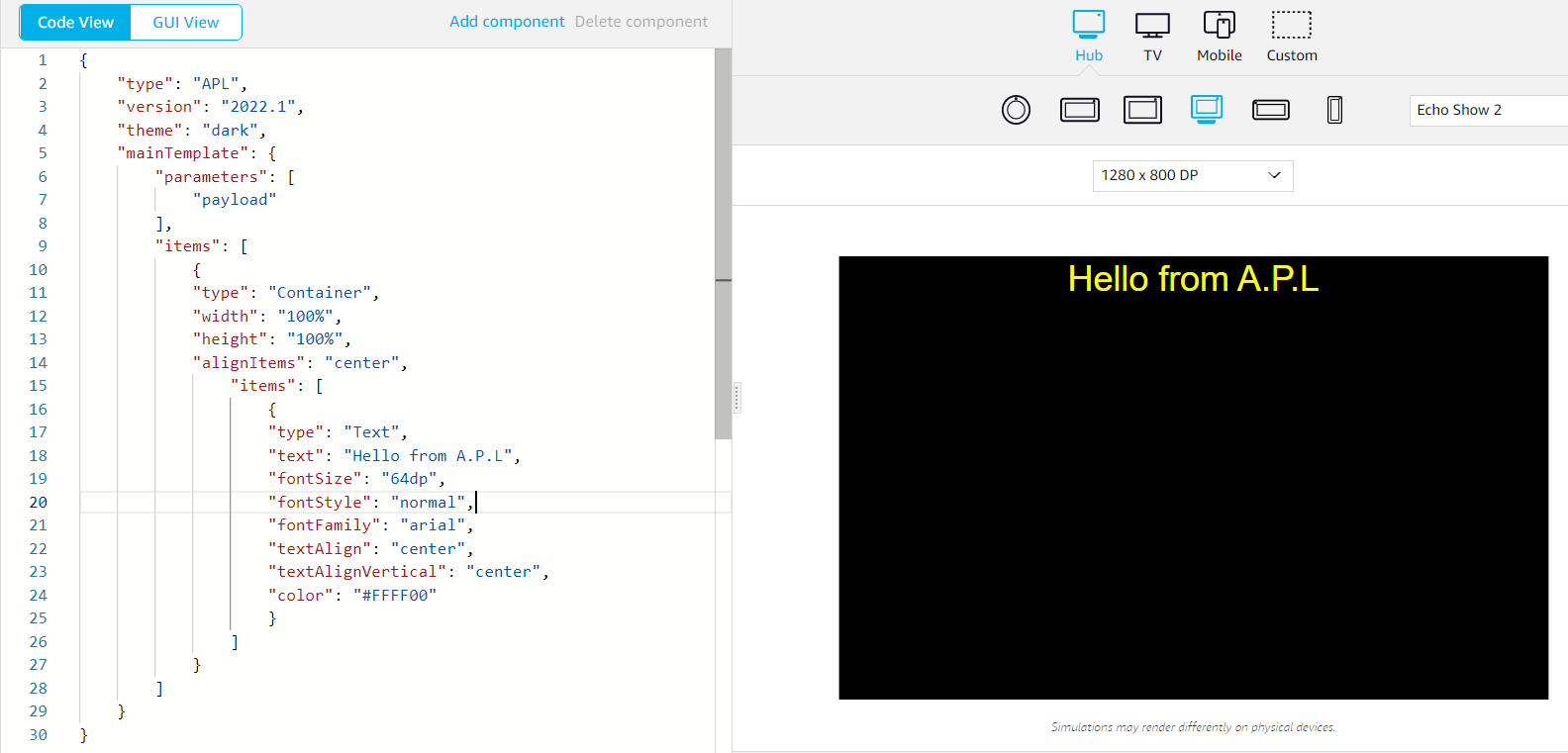
]

}

}

Note that you can only change the background colour not the primary or secondary text colour.

You can see your APL displayed if you paste the code into the developer console Multimodal Authoring tool:



**Using datasources**

Let’s remove the text to a datasource.

1. Create a json file that is going to contain your datasource (I’ve called it myData.json). In there place the items you want, say the primaryText and secondaryText for the Headline, so that it’s like this:

myData.json:

{

"HeadlineData": {

"text": {

"primaryText": "Primary text from datasource",

"secondaryText": "Secondary text from datasource"

},

"image": {

"someImageURL": "WeCouldHaveAnImageURLHereIfWeUsedIt.jpg"

}

}

}

Save this at the same level as your lambda\_function.

We’ll refer to the primaryText using **"${payload.HeadlineData.text.primaryText}"**

1. Change the RenderDocumentDirective code in lambda\_function.py to read the datasource from the file:

.add\_directive(

APLRenderDocumentDirective(

token="pagerToken",

document=\_load\_apl\_document("APLdatasourceExample.json"),

datasources=\_load\_apl\_document("myData.json")

)

)

1. Change the items in APL json to refer to the datasource:

{

"type": "APL",

"version": "2022.1",

"license": "Copyright 2021 Amazon.com",

"theme": "dark",

"import": [

{

"name": "alexa-layouts",

"version": "1.6.0"

}

],

"mainTemplate": {

"parameters": [

"payload"

],

"item": [

{

"type": "AlexaHeadline",

"id": "PlantHeadline",

"primaryText": **"${payload.HeadlineData.text.primaryText}",**

"secondaryText": **"${payload.HeadlineData.text.secondaryText}",**

"headerBackButton": false,

"headerAttributionPrimacy": true,

"footerHintText": "footerHintText",

"backgroundColor": "@colorRed800",

"backgroundColorOverlay": false,

"headerTitle": "headerTitle",

"headerSubtitle": "subtitle"

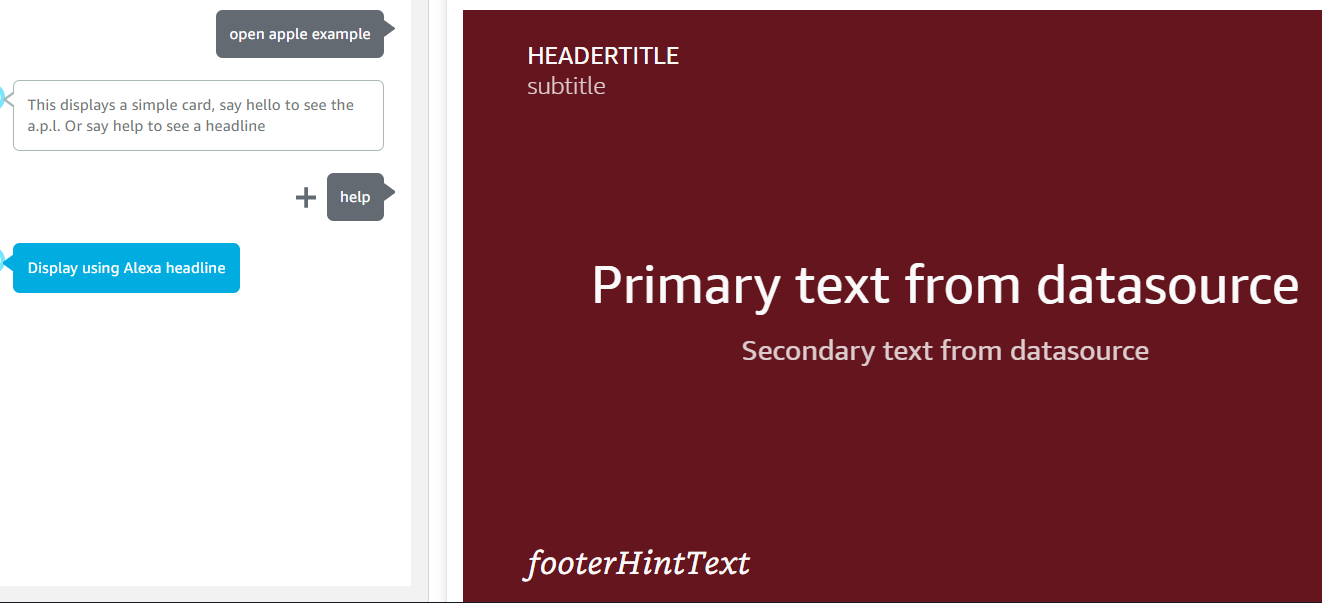
}

]

}

}

Save and deploy – you should get this:



**Part 2. Using lambda skill**

If you have a lambda skill, say on your PC, then update the the manifest.apis.custom.interfaces array in your skill-package / skill.json code as follows.

The rest is the same

      "custom": {

        "endpoint": {

          "uri": "arn:aws:lambda:us-east-1:559144307262:function:ask-alexaapl-default-default-166….5"

        },

        "interfaces": [

          {

            "supportedViewports": [

              {

                "maxHeight": 540,

                "maxWidth": 960,

                "minHeight": 540,

                "minWidth": 960,

                "mode": "TV",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 599,

                "maxWidth": 599,

                "minHeight": 100,

                "minWidth": 100,

                "mode": "HUB",

                "shape": "ROUND"

              },

              {

                "maxHeight": 959,

                "maxWidth": 1279,

                "minHeight": 600,

                "minWidth": 960,

                "mode": "HUB",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 1279,

                "maxWidth": 1920,

                "minHeight": 600,

                "minWidth": 1280,

                "mode": "HUB",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 599,

                "maxWidth": 1279,

                "minHeight": 100,

                "minWidth": 960,

                "mode": "HUB",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 1279,

                "maxWidth": 2560,

                "minHeight": 960,

                "minWidth": 1920,

                "mode": "HUB",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 2560,

                "maxWidth": 1279,

                "minHeight": 1920,

                "minWidth": 960,

                "mode": "HUB",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 1920,

                "maxWidth": 959,

                "minHeight": 320,

                "minWidth": 600,

                "mode": "MOBILE",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 1920,

                "maxWidth": 1279,

                "minHeight": 320,

                "minWidth": 960,

                "mode": "MOBILE",

                "shape": "RECTANGLE"

              },

              {

                "maxHeight": 1920,

                "maxWidth": 1920,

                "minHeight": 320,

                "minWidth": 1280,

                "mode": "MOBILE",

                "shape": "RECTANGLE"

              }

            ],

            "type": "ALEXA\_PRESENTATION\_APL"

          }

        ]

I hope that helps