# Using Dynamic Cards in your app

## Overview

In this guide, you will learn how to use dynamic cards in your apps. You will be making a simple news application that lists the top 20 news articles trending globally.

## Pre-requisites

Here are some things you should know before you can get started with this guide.

- Working with APIs and the Web Component
- Working with the JSON component
- Working with Dynamic Labels, Buttons, and Images

> **Caution**
>
> This is guide covers advanced topics like dynamic components and web APIs. For sake of keeping this guide at a readable length, some basic instructions will be glossed over with the assumption that you are aware of the workings of other components. If not, please take a look at more basic guides before getting started here.

## First steps

We will be using the News API to fetch the trending headlines. Head over to https://newsapi.org/> to get an API key.

> **Warning**
>
> You will not be able to monetize apps which use the News API. To place ads in apps that use this API, you will have to switch to their paid plan first. More information is available here

The API endpoint we'll be using in this guide is `https://newsapi.org/v2/top-headlines?language=en&apiKey={YOUR_API_KEY}`

This is what a sample response looks like:

```
{"status":"ok",
"totalResults":TOTAL_NUMBER_OF_ARTICLES,
"articles":[
{"source":
{"id":"SOURCE_ID","name":"SOURCE_NAME"},
"author":AUTHOR_NAME,
"title":"ARTICLE_TITLE",
"description":"ARTICLE_SUMMARY",
"urlToImage":"ARTICLE_IMAGE",
"publishedAt":"ARTICLE_PUBLISH_DATE",
"content":"ARTICLE_CONTENT"
}, {...}
]}
```

We will need the article `title`, `description`, and `urlToImage` to populate the cards. We'll be using the JSON component to extract data from the API.

## Designing the app

As usual, we'll start with a new project. Drag and drop the `Vertical Scroll Arrangement`, `Web`, `JSON`, `Dynamic Label`, `Dynamic Image`, and `Dynamic Card View` components.

Set the `Vertical Scroll Arrangement` 's height and width designer properties to `Fill Parent`, and its `Align Vertical` property to `Center`.

In the `Dynamic Card View` 's designer properties, set the `Corner Radius` to 5 and the `Elevation` to 0. Finally, enable its `Full Clickable` designer property.

Finally, add an `Image Utilities` component which we will use to load images asynchronously

This is what your designer should look like

The top 20 news articles will be displayed as `Dynamic Card View` s, which will be loaded inside the `Vertical Scroll Arrangement` . Each card will include the article's title, a short description, and a header image.

## Coding the blocks

We will need a variable to store the JSON response from the API. This can be a simple `Text` variable like so.

The API data is to be fetched when the screen initializes. Set the `Web` component's url to the API url, and perform the GET request.

When the `Web` component has fetched the data, set the response content to the global variable you created. In case there was an issue getting the content, we set the variable to custom error JSON. This is so that the JSON parsing, which has to be done later, doesn't throw errors.

Next, we need a procedure that populates the `Vertical Scroll Arrangement` with cards. This procedure will iterate 20 times if there are more than 20 news articles in the api response variable. If there are less than 20 articles, the procedure will only iterate through the number of articles in the api response.

We call this procedure inside the `Web` component's Got Text block.

Next, we create another procedure which takes the JSON object and the array index as parameters. This procedure will create the cards for us and add them to the `Vertical Scroll Arrangement` .

> **Tip**
>
> It is recommended you place blocks in procedures so that they are more readable and can be used several times.

For each news article, we first create a card view that holds all the content we wish to show. Use the `Create Card View` block inside the procedure to make a

dynamic card for each article. Each card will be 90% of the screen's width and will have automatic height so that it can resize according to the content inside.

Next, we create a dynamic image that contains the article's header image. We use the `Image Utilities` component to load the images asynchronously from the URL, so that the UI doesn't lag while the data is being fetched.

Using the `Dynamic Label` component, we create dynamic labels for each card which contain the article's title. Since this the article's title, we set the font size to 18.

> **Tip**
>
> It is recommended you set the `html` parameter to `False` when getting data from external sources. Unchecked HTML can be used to perform unauthorized actions in your app.

We use the `Dynamic Label` component one more time to create a label for the description. We set the id to a value that we are certain hasn't been used yet. If you are showing the top 200 news articles, then you might want to start your id's from 1000 (or a similarly appropriate number).

Finally, we place the procedure call block inside the `PopulateNewsArticles` procedure.

## Conclusion

The finished project

All the blocks we've used

Hooray! You've successfully made an app that uses dynamic card views!

Here is a summary of what you've learned in this guide.

- How to create dynamic card views.
- How to add other dynamic components inside dynamic card views.
- How to load images asynchronously inside dynamic images.

- How to decode JSON that is fetched from an API.

## Next steps

Want ideas for making this app better? We have some things for you to try.

- Open the full article in a web view when the card is clicked.
- Toggle the visibility of the card's descriptions when they are clicked on.
- Experiment with different font sizes and typefaces for the title and description.
- Add more information to each card, like the author's name and the date of publication.

## Downloads

Get the AIA file here.

---

Last update: January 24, 2020