



DHIS2 Application Platform

Austin McGee
austin@dhis2.org

29 October 2019

About Me

Austin McGee, Software Engineer

- **DHIS2 Core Team (1 year)** – based in Paris, France
 - Analytics Team, work closely with Apps team
 - Created d2 CLI & App Platform
- Previously
 - BigFix => IBM – enterprise network management
 - Executive Director, WellDone International – monitoring for rural water systems
 - Co-founder, Arch Systems – cost-effective modular sensors



Please sign in

 Username PasswordSign in



Line

UPDATE

File Options Download

< Interpretations

Search dimensions

Data

Period

Organisation Unit

Area

Commodities

Diagnosis

Donor

EPI/nutrition age

Facility Ownership

Facility Type

Funding Agency

Gender

HIV age

Implementing Partner

Location Fixed/Outreach

Location Rural/Urban

Main data element groups

Morbidity Age

Morbidity/Mortality

PMTCT

Pregnant/Non-pregnant

Project

Referrals Age

Rural and Urban

Target vs Result

Tracker-based data

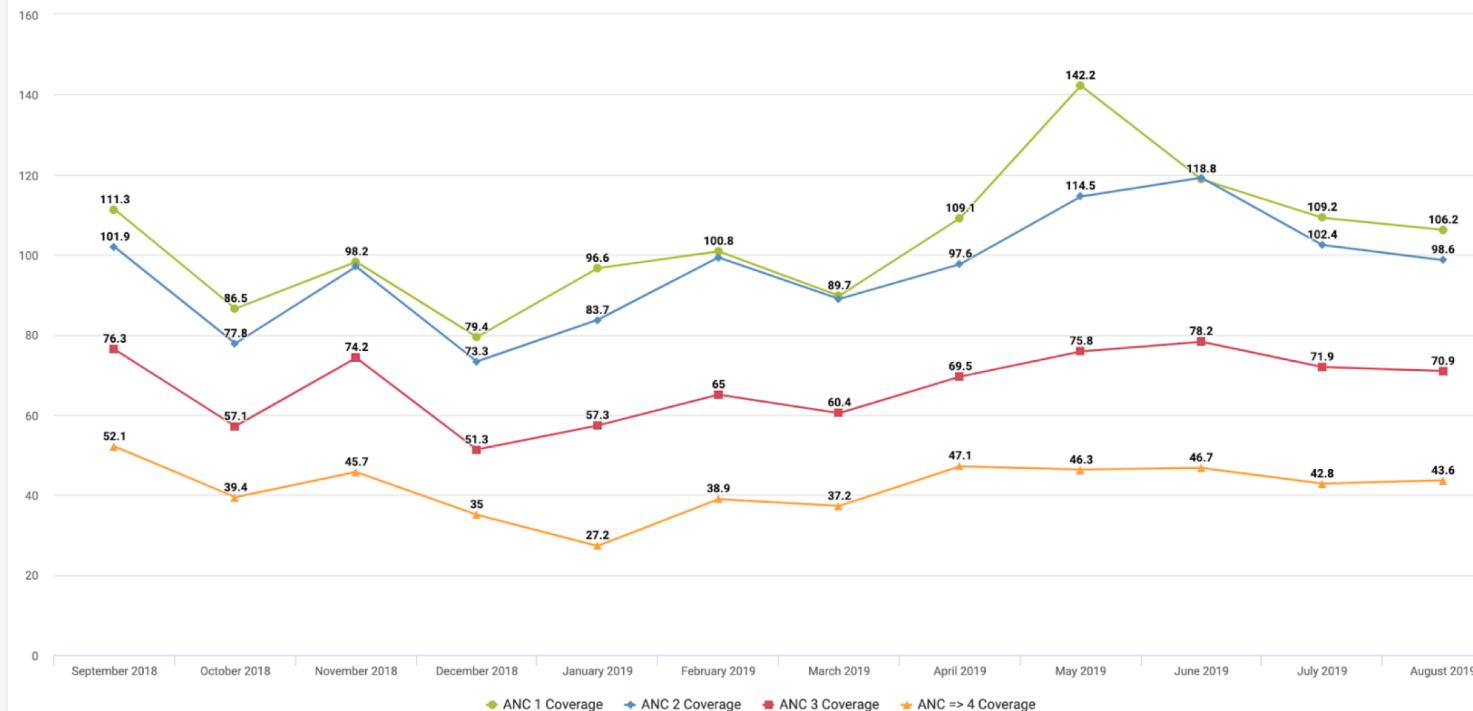
Series: Data: 4 selected ...

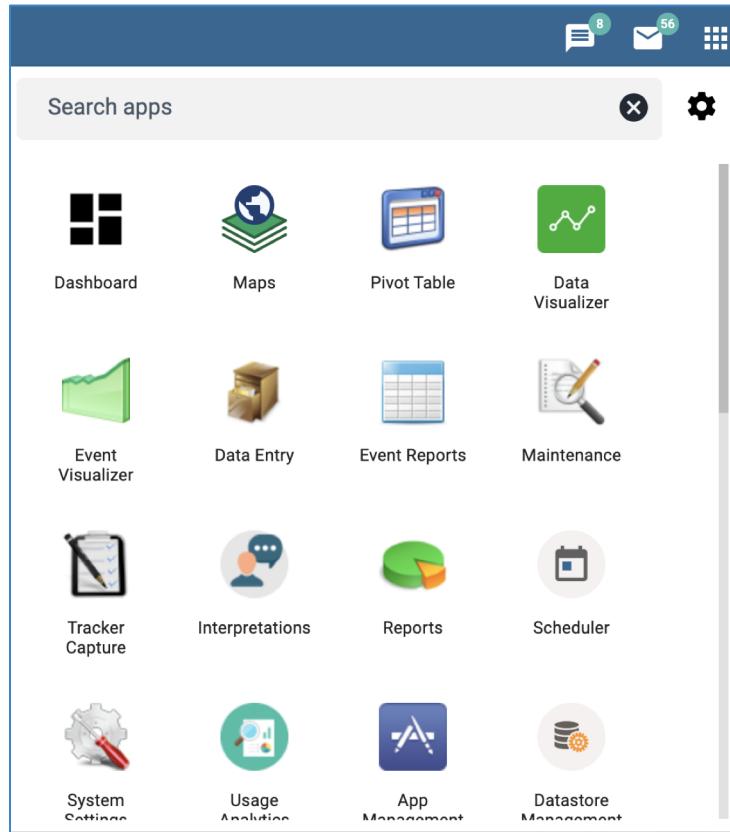
Filter: Organisation Unit: 1 selected ...

Category: Period: 1 selected ...

Unsaved chart

Sierra Leone





Thirty-Two Roads Diverge in a Yellow Wood

-  Dashboard - DHIS 2 ×
-  Data Visualizer ×
-  DHIS 2 Pivot Tables ×
-  2 DHIS 2 Demo - Sierra Leone ×
-  Interactive Assignment ×



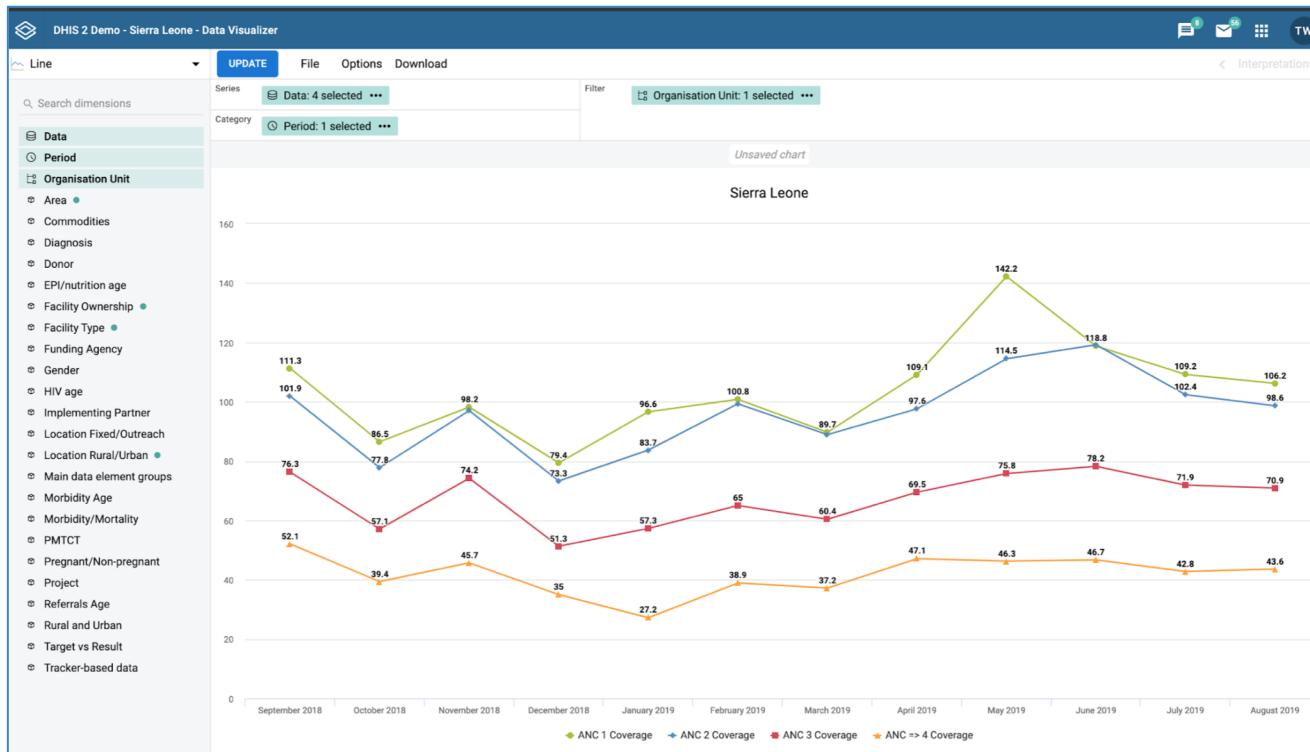
Thirty-Two Roads Diverge in a Yellow Wood

Dashboard - DHIS 2	×
DHIS 2 Pivot Tables	×
DHIS 2 Event Charts	×
DHIS 2 GIS	×
WHO Metadata App	×
Data Visualizer	×
DHIS 2 Demo - Sierra Leone	×
Cache Cleaner	×
DHIS2 Maps	×
DHIS 2 Maintenance	×
DHIS 2 Event Reports	×
Translations App	×
Interactive Assignment	×
Import/Export DHIS2	×
Usage Analytics App	×
DHIS2 Capture App	×
App Management	×
DHIS 2 Demo - Sierra Leone	×
Settings	×
Data Administration	×
Scorecard	×
Tracker Capture	×
DHIS2 Scheduler	×
Data Quality	×
DHIS2 Datastore Manager	×
DHIS 2 Demo - Sierra Leone	×
Interpretations App	×
WHO Data Quality Tool	×
DHIS 2 User Management	×
Reports App	×
Menu Management	×
Services Management	×

How many front-end developers?

On the DHIS2 core team?

13



DHIS 2 Demo - Sierra Leone

Report Reports

Standard Report

Data Set Report

Reporting Rate Summary

Resource

Organisation Unit Report

Data Approval

 **Standard Report**
View and add reports based on the JasperReports library. These can be based on report tables and can be designed in iReport.

 **Reporting Rate Summary**
Browse the reporting rates of data sets by organisation unit and period based on various criteria for submission.

 **Organisation Unit Distribution Report**
Browse the organisation unit distribution report based on the organisation unit group sets and its groups.

 **Data Set Report**
View data set reports. These reports are based on data entry screens and will produce a report with aggregated data.

 **Resource**
View and add resources. These resources can be uploaded documents or URLs on the web.

 **Data Approval**
View data and manage data approval by approving or unapproving, accepting or unaccepting data.

DHIS 2 Pivot Tables

Indicators

Select indicator group

Available > >> << < Selected

Creating a pivot table

- Select Items from any of the dimensions in the left menu
- Click Layout to arrange your dimensions on table rows and columns
- Click Update to create your table

Working with a pivot table

- Click Options to hide sub-totals or empty rows, adjust font size and more
- Click Favorites to save your table for later use
- Click Download to save table data to your computer

Your most viewed favorites

- HIV: Care data indicators at district
- ANC: ANC by area last 12 months
- Immunization: Data last 4 quarters for NGO, mission and public facilities
- Reporting rates: Major forms by orgunit last year
- ANC: ANC visits 1-3 last 12 months
- ANC: Indicators by districts last 4 quarters
- IDSR: Disease Week 1-12 Sierra Leone
- HIV: HIV counselling at facility level (hide empty columns)
- ANC: ANC visits by facility.type
- Child health: Immunization doses given (tracker)

DHIS 2 Demo - Sierra Leone

League Table

Search Apps  

 [League Table](#) [+ Create](#)

Welcome to League Table App!

Create a League Table instantly, over tea break

1. Click create new, Write your League Table a title and description
2. Write group header name for indicators you will create in next step
3. Select indicator group and click indicator to add to the group
 - o Edit title to change title to use on heading
 - o Enter effective gap to show increase/decrease from last period
 - o Set minimum and maximum range for each color codeSelect indicator group and click indicator to add to the group
4. Click Add another to add another indicator in next column in the group
5. Select the second indicator to add indicator in the same group
6. Click Add another in Group row to add next group

[New League Table](#)

« »

**“Easy things should be easy,
hard things should be possible.”**

Larry Wall, creator of Perl

Instance UI	Headerbar	Application Navigation	Error Handling	File Menu	Dialogs
Common UI	In-app Routing	Alerts	i18n Runtime & Lang Selection	Loading State Indicators	Common UI Components
Application	App State Management	App Interface	Data Visualization		
Data Access	API Fetch Read & Write	Metadata Schemas	Server Discovery	Data Caching & Offline Support	Authentication & Permissions
Build Tooling	i18n extract & generate	Manifest Generate	Tests Infrastructure	Compilation Browser Targets	Publishing, Release, Distribution

What's in an App?



34 Core applications (at least)

+ ~ 6 Libraries



34 Core applications (at least)

+ ~ 6 Libraries

* 4 supported versions

= **~160 maintained codebases**

```
{
  "name": "maps-app",
  "scripts": {
    "prestart": "yarn localize && yarn manifest",
    "start": "webpack-dev-server",
    "test": "jest",
    "prebuild": "yarn test && yarn localize && yarn manifest && rm -rf build && mkdir build && cp -r public/* ./build/",
    "build": "NODE_ENV=production webpack --progress",
    "extract-pot": "d2-i18n-extract -p src/ -o i18n/",
    "localize": "yarn extract-pot && d2-i18n-generate -n NAMESPACE -p ./i18n/ -o ./src/locales/",
  },
  "dependencies": {
    "@dhis2/d2-i18n": "^1.0.4",
    "@dhis2/d2-ui-analytics": "^0.0.3",
    "@dhis2/d2-ui-core": "5.2.10",
    "@dhis2/d2-ui-file-menu": "5.2.10",
    "@dhis2/d2-ui-interpretations": "5.2.10",
    "@dhis2/d2-ui-org-unit-dialog": "5.2.10",
    "@dhis2/d2-ui-org-unit-tree": "5.2.10",
    "@dhis2/ui": "1.0.0-beta.15",
    "@material-ui/core": "3.9.3",
    "@material-ui/icons": "3.0.2",
    "d2": "31.2.2",
    "d2-utilizr": "0.2.15",
    "prop-types": "15.6.2",
    "react": "16.8.6",
    "react-dom": "16.8.6",
  },
}
```

Package.json dependencies

A real example Core app

(abbreviated for display purposes)



```
"devDependencies": {
  "@dhis2/d2-i18n-extract": "^1.0.7",
  "@dhis2/d2-i18n-generate": "^1.0.19",
  "babel-cli": "^6.26.0",
  "babel-core": "6.26.0",
  "babel-jest": "22.4.3",
  "babel-loader": "7.1.2",
  "babel-plugin-istanbul": "4.1.4",
  "babel-plugin-transform-es2015-arrow-functions": "6.22.0",
  "babel-plugin-transform-es2015-modules-commonjs": "6.26.0",
  "babel-plugin-transform-object-assign": "6.22.0",
  "babel-polyfill": "6.26.0",
  "babel-preset-es2015": "6.24.1",
  "babel-preset-react": "6.24.1",
  "babel-preset-stage-2": "6.24.1",
  "babel-regenerator-runtime": "6.5.0",
  "babel-runtime": "6.26.0",
  "css-loader": "0.28.10",
  "cypress": "3.1.5",
  "cypress-pipe": "1.3.2",
  "d2-manifest": "1.0.0",
  "enzyme": "3.3.0",
  "enzyme-adapter-react-16": "1.1.1",
  "enzyme-to-json": "3.3.4",
  "jest": "22.4.2",
  "webpack": "3.11.0",
  "webpack-dev-server": "2.11.1"
},
"manifest.webapp": {
  "name": "DHIS2 Maps",
  "icons": {
    "48": "icon.png"
  },
  "developer": {
    "url": "https://www.dhis2.org",
    "name": "DHIS2"
  },
  "activities": {
    "dhis": {
      "href": ".."
    }
  }
}
```

This is **complex, expensive, and error-prone.**

What about 3rd-party apps?

Instance UI	Headerbar	Application Navigation	Error Handling	File Menu	Dialogs
Common UI	In-app Routing	Alerts	i18n Runtime & Lang Selection	Loading State Indicators	Common UI Components
Application	App State Management	App Interface	Data Visualization		
Data Access	API Fetch Read & Write	Metadata Schemas	Server Discovery	Data Caching & Offline Support	Authentication & Permissions
Build Tooling	i18n extract & generate	Manifest Generate	Tests Infrastructure	Compilation Browser Targets	Publishing, Release, Distribution

Why

For Implementers, Users, & Funders

**Easy things easy
Hard things possible**

- More consistency across all applications
- Get the latest features faster
- Simplified custom application development
 - Get features without creating JIRA tickets!
- More high-quality Community Apps!

Why

For Developers

Create-react-app for DHIS2

- **Get things “out of the box”**
 - Modern UI components
 - Up-to-date security
 - Robust data engine
- **Develop apps which are:**
 - Decoupled from DHIS2 versions
 - Worry about “what” not “how”

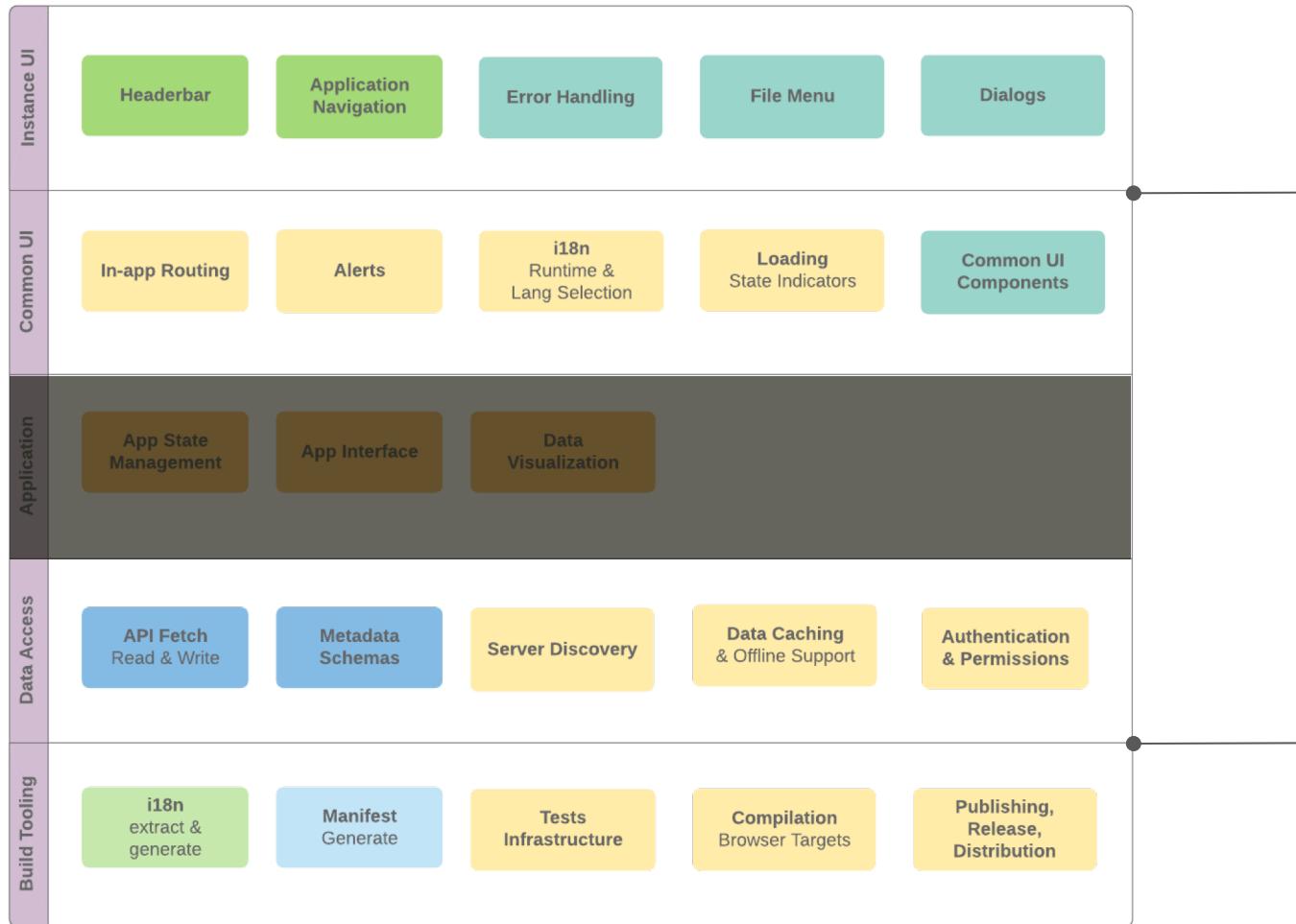
App Platform Principles

- **Inversion of control**
- **Unified build tooling** (@dhis2/cli-app-scripts)
- **Common runtime** (@dhis2/app-runtime)
- **Declarativity** (app configuration, data flow)

Instance UI	Headerbar	Application Navigation	Error Handling	File Menu	Dialogs
Common UI	In-app Routing	Alerts	i18n Runtime & Lang Selection	Loading State Indicators	Common UI Components
Application	App State Management	App Interface	Data Visualization		
Data Access	API Fetch Read & Write	Metadata Schemas	Server Discovery	Data Caching & Offline Support	Authentication & Permissions
Build Tooling	i18n extract & generate	Manifest Generate	Tests Infrastructure	Compilation Browser Targets	Publishing, Release, Distribution

What's in an App?





App-shell
Common “wrapper”

Redefine "App"

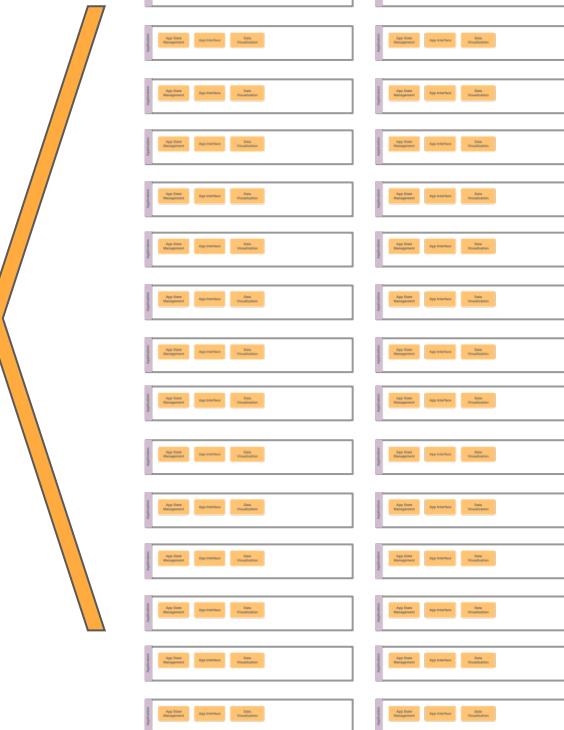
DHIS2 App != standalone web app (CRA)

We can control:

- The development ecosystem (Build process, libraries)
- The delivery mechanism (AppStore => DHIS2 Core)
- The data access layer (DHIS2 Core API)



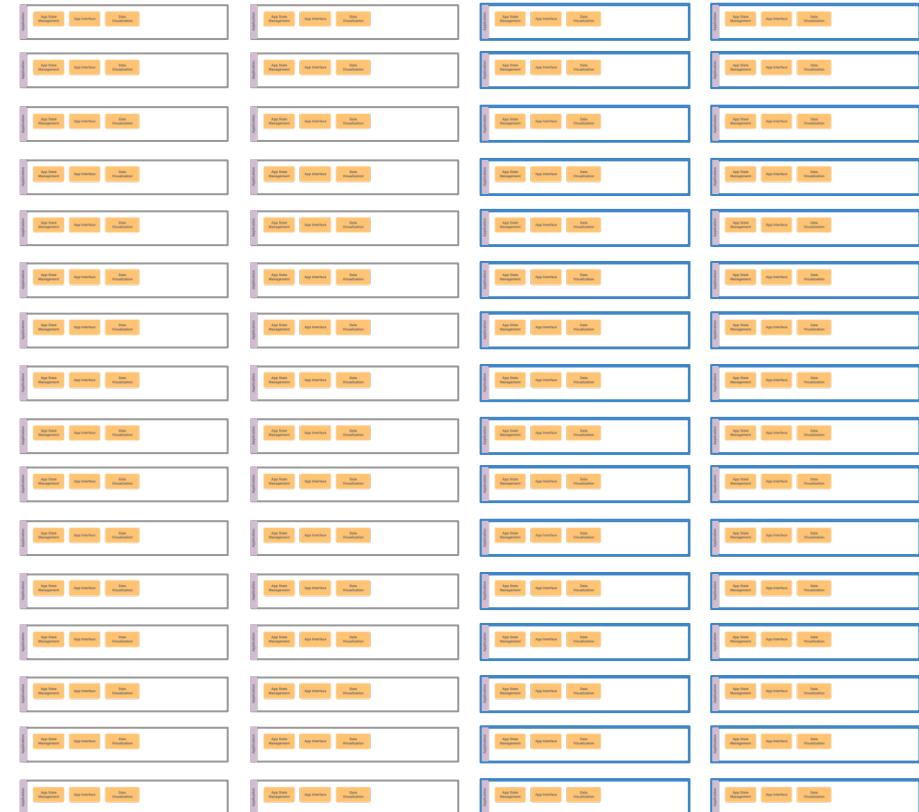
App-Shell



Core Apps



App-Shell

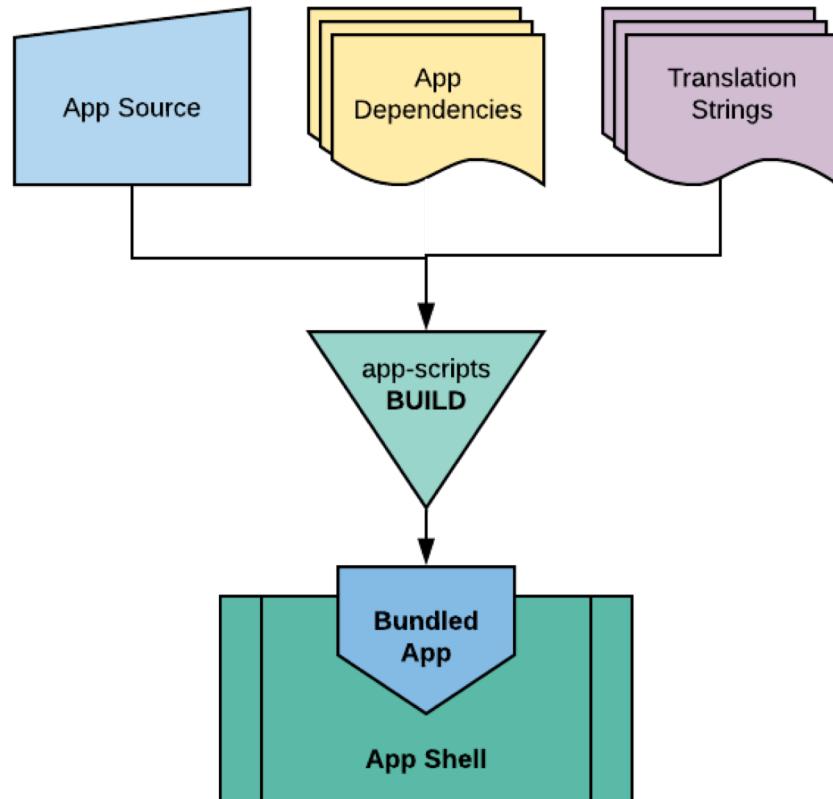


Core Apps

3rd-party Apps...

App Platform Principles

- **Inversion of control**
- **Unified build tooling** (@dhis2/cli-app-scripts)
- **Common runtime** (@dhis2/app-runtime)
- **Declarativity** (app configuration, data flow)





```
{  
  "name": "maps-app",  
  "scripts": {  
    "prestart": "yarn localize && yarn manifest",  
    "start": "webpack-dev-server",  
    "test": "jest",  
    "prebuild": "yarn test && yarn localize && yarn manifest && rm -rf build && mkdir build && cp -r public/* ./build/",  
    "build": "NODE_ENV=production webpack --progress",  
    "extract-pot": "d2-i18n-extract -p src/ -o i18n/",  
    "localize": "yarn extract-pot && d2-i18n-generate -n NAMESPACE -p ./i18n/ -o ./src/locales/",  
  },  
  "dependencies": {  
    "@dhis2/d2-i18n": "^1.0.4",  
    "@dhis2/d2-ui-analytics": "^0.0.3",  
    "@dhis2/d2-ui-core": "5.2.10",  
    "@dhis2/d2-ui-file-menu": "5.2.10",  
    "@dhis2/d2-ui-interpretations": "5.2.10",  
    "@dhis2/d2-ui-org-unit-dialog": "5.2.10",  
    "@dhis2/d2-ui-org-unit-tree": "5.2.10",  
    "@dhis2/ui": "1.0.0-beta.15",  
    "@material-ui/core": "3.9.3",  
    "@material-ui/icons": "^3.0.2",  
    "d2": "31.2.2",  
    "d2-utilizr": "^0.2.15",  
    "prop-types": "15.6.2",  
    "react": "16.8.6",  
    "react-dom": "16.8.6",  
  },  
},
```

```
"devDependencies": {  
  "@dhis2/d2-i18n-extract": "^1.0.7",  
  "@dhis2/d2-i18n-generate": "^1.0.19",  
  "babel-cli": "^6.26.0",  
  "babel-core": "6.26.0",  
  "babel-jest": "22.4.3",  
  "babel-loader": "7.1.2",  
  "babel-plugin-istanbul": "4.1.4",  
  "babel-plugin-transform-es2015-arrow-functions": "6.22.0",  
  "babel-plugin-transform-es2015-modules-commonjs": "6.26.0",  
  "babel-plugin-transform-object-assign": "6.22.0",  
  "babel-polyfill": "6.26.0",  
  "babel-preset-es2015": "6.24.1",  
  "babel-preset-react": "6.24.1",  
  "babel-preset-stage-2": "6.24.1",  
  "babel-regenerator-runtime": "6.5.0",  
  "babel-runtime": "6.26.0",  
  "css-loader": "0.28.10",  
  "cypress": "3.1.5",  
  "cypress-pipe": "1.3.2",  
  "d2-manifest": "1.0.0",  
  "enzyme": "3.3.0",  
  "enzyme-adapter-react-16": "1.1.1",  
  "enzyme-to-json": "3.3.4",  
  "jest": "22.4.2",  
  "webpack": "3.11.0",  
  "webpack-dev-server": "2.11.1"  
},  
"manifest.webapp": {  
  "name": "DHIS2 Maps",  
  "icons": {  
    "48": "icon.png"  
  },  
  "developer": {  
    "url": "https://www.dhis2.org",  
    "name": "DHIS2"  
  },  
  "activities": {  
    "dhis": {  
      "href": ".."  
    }  
  }  
}
```

Before

(abbreviated for display purposes)



```
{  
  "name": "simple-app",  
  "version": "1.0.0",  
  "devDependencies": {  
    "@dhis2/cli-app-scripts": "^1.4.2"  
  },  
  "scripts": {  
    "start": "d2-app-scripts start",  
    "build": "d2-app-scripts build",  
    "test": "d2-app-scripts test"  
  },  
  "peerDependencies": {  
    "@dhis2/app-runtime": "*",  
    "@dhis2/d2-i18n": "*",  
    "moment": "*",  
    "react": "^16.8",  
    "react-dom": "^16.8"  
  }  
}
```

package.json

(after app-shell)

```
{  
  "name": "simple-app",  
  "version": "1.0.0",  
  "devDependencies": {  
    "@dhis2/cli-app-scripts": "^1.4.2"  
  },  
  "scripts": {  
    "start": "d2-app-scripts start",  
    "build": "d2-app-scripts build",  
    "test": "d2-app-scripts test"  
  },  
  "peerDependencies": {  
    "@dhis2/app-runtime": "*",  
    "@dhis2/d2-i18n": "*",  
    "moment": "*",  
    "react": "^16.8",  
    "react-dom": "^16.8"  
  }  
}
```



- I18n extract/generate
- Manifest generation
- Dependency injection
- Tests infrastructure
- Transpilation & Bundling
- App-shell encapsulation
- Publication & Release



```
const config = {  
    name: 'simple-app',  
    title: 'Simple Example App',  
    description: 'This is a simple example application',  
  
    entryPoints: {  
        app: './src/App',  
    },  
}  
  
module.exports = config  
...
```

D2.config.js

DHIS2-specific app configuration



```
1 import React from 'react'
2 import i18n from './locales'
3 import { useDataQuery } from '@dhis2/app-runtime'
4 import style from './App.style'
5
6 const query = {
7     me: {
8         resource: 'me'
9     }
10}
11
12 const Component = () => {
13     const { error, loading, data } = useDataQuery(query)
14     return (
15         <div style={style}>
16             {error && <span>ERROR</span>}
17             {loading && <span>...</span>}
18             {data && (
19                 <h1>
20                     |     {i18n.t('Hello {name}', { name: data.me.name })}
21                 </h1>
22             )}
23         </div>
24     )
25 }
26
27 export default Component
```

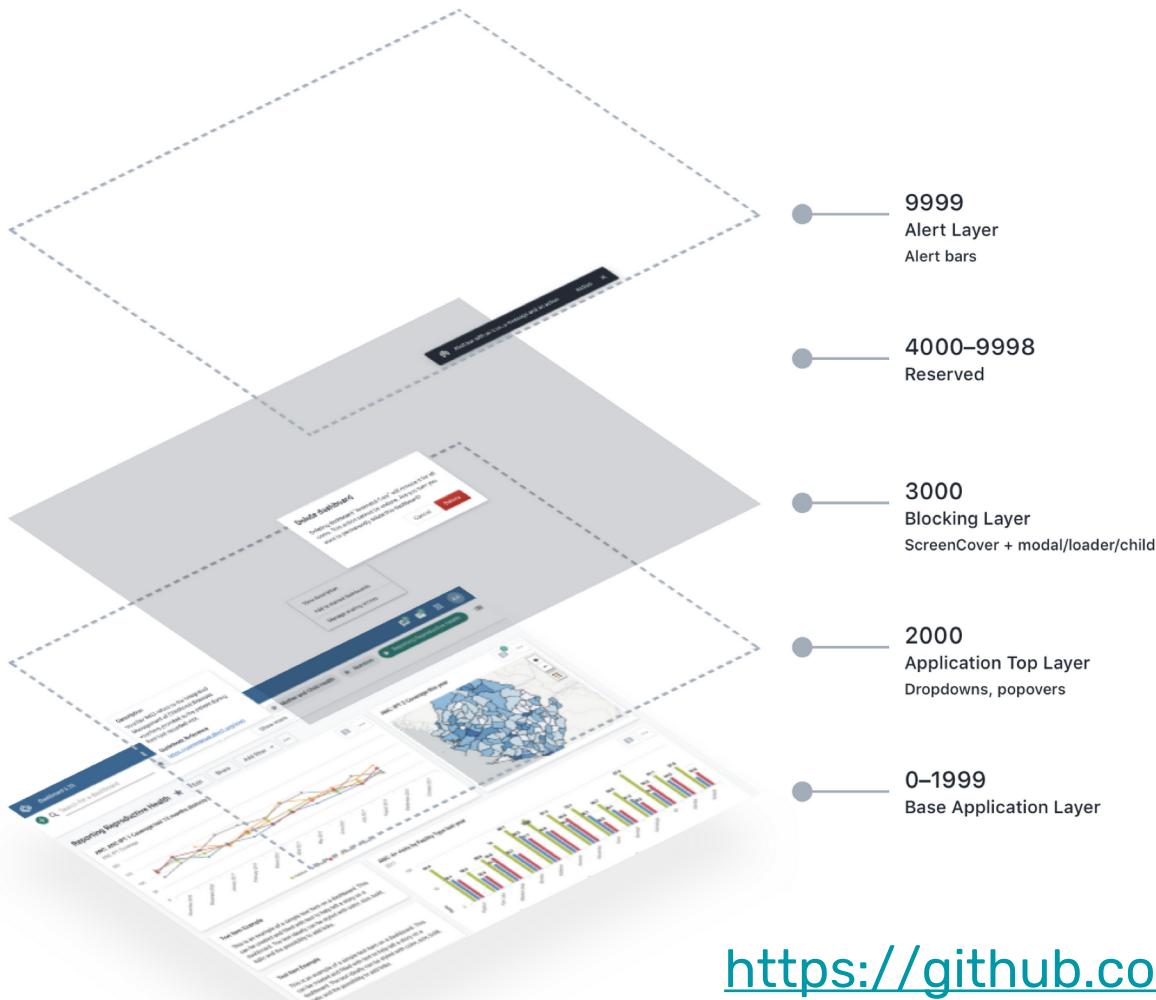
App Platform Principles

- **Inversion of control**
- **Unified build tooling** (@dhis2/cli-app-scripts)
- **Common runtime** (@dhis2/app-runtime)
- **Declarativity** (app configuration, data flow)

```
{  
  "name": "simple-app",  
  "version": "1.0.0",  
  "devDependencies": {  
    "@dhis2/cli-app-scripts": "^1.4.2"  
  },  
  "scripts": {  
    "start": "d2-app-scripts start",  
    "build": "d2-app-scripts build",  
    "test": "d2-app-scripts test"  
  },  
  "peerDependencies": {  
    "@dhis2/app-runtime": "*",  
    "@dhis2/d2-i18n": "*",  
    "moment": "*",  
    "react": "^16.8",  
    "react-dom": "^16.8"  
  }  
}
```



- **UI Primitives**
- **API Data fetching**
- **Translations**
- Routing
- Alerts
- **Configuration**
- **Server discovery**
- Loading & **Error handling**
- **Authentication**
- Data caching



<https://github.com/dhis2/design-system>

- [AlertBar](#)
- [AlertStack](#)
- [Button: Basic](#)
- [Button: Primary](#)
- [Button: Secondary](#)
- [Button: Destructive](#)
- [ButtonStrip](#)
- [Card](#)
- [Checkbox](#)
- [Chip](#)
- [CircularLoader](#)
- [ComponentCover](#)
- [CssReset](#)
- [DropdownButton: Basic](#)
- [DropdownButton: Primary](#)
- [DropdownButton: Secondary](#)
- [DropdownButton: Destructive](#)
- [Help](#)
- [InputField: Outlined](#)
- [InputField: Filled](#)
- [LinearLoader](#)
- [Logo](#)
- [Menu](#)
- [Modal](#)
- [Node](#)
- [Radio](#)

Button

Buttons are used for triggering actions. There are different types of buttons in the design system which are intended for different types of actions.

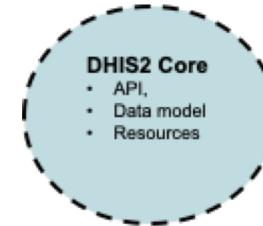
Usage

Type	View	Usage
Basic	Basic	The most often used button that will suit the majority of actions. Should be the default choice. Several basic buttons can be in the same area.
Primary	Primary	Used to highlight the most important/main action on a page. A 'Save' button for a form page should be primary, for example. Use sparingly, rarely should there be more than a single primary button per page.
Secondary	Secondary	Used for passive actions, often as an alternative to the primary action. If 'Save' is primary, 'Cancel' could be secondary. Not intended to draw user attention. Do not use for the only action on a page.
Destructive	Destructive	Used instead of a primary button when the main action is destructive in nature. Used to highlight to the user the seriousness of the action. Destructive buttons must only be used for destructive actions.

App Platform Principles

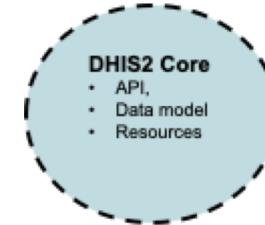
- **Inversion of control**
- **Unified build tooling** (@dhis2/cli-app-scripts)
- **Common runtime** (@dhis2/app-runtime)
- **Declarativity** (app configuration, data flow)

Accessing the API



- In a browser, using a log-in session
 1. Navigate to the URL (<https://<dhis2-url>/api>)
 2. Log in with your credentials
- With an “Authorization” header
 - *Basic <Base 64-encoded string>*
 - Encoded string: btoa(username:password)
 - Example for admin:district; Basic YWRtaW46ZGlzdHJpY3Q=

Navigating the API



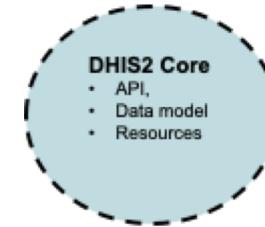
- Can be viewed in any web browser
 - With ordinary GET-requests
 - Returns *xml* by default, *json* with a .json suffix
- `/api/resources`
 - Contains a list of all **metadata** endpoints
- `/api/<resource[s]>`
 - List all metadata items of a certain type
 - Available parameters:



```
{  
  "resources": [  
    {  
      "displayName": "Data Set Notification Templates",  
      "singular": "dataSetNotificationTemplate",  
      "plural": "dataSetNotificationTemplates",  
      "href": "http://localhost:8080/api/dataSetNotificationTemplates"  
    },  
    {  
      "displayName": "Program Tracked Entity Attribute Groups",  
      "singular": "programTrackedEntityAttributeGroup",  
      "plural": "programTrackedEntityAttributeGroups",  
      "href": "http://localhost:8080/api/programTrackedEntityAttributeGroups"  
    }  
  ]  
}
```

Parameter	Explanation	Example
?paging=false	Disable paging	
?filter	Filter items on given constraint	?filter=id:eq:IpHINAT79UW
?fields	Show given fields	?fields=id,displayName
	Show all fields	?fields=:all
	Show properties of embedded object	?fields=id,programStages[id,displayName]

Exploring Resources with Schema



- `/api/schemas`
 - Show key attributes for all available resources
- `/api/schemas/<resource>`
 - Show *all* attributes for one specific resource



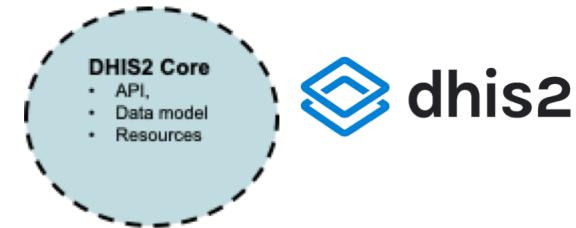
A screenshot of a JSON schema for the "Program" resource. The schema includes properties such as "relativeApiEndpoint", "displayName", and a "properties" array containing three objects. Each object has fields like "fieldName", "propertyType", "collection", and "required". The "collection" field is set to false for all properties.

```
{  
  "relativeApiEndpoint": "/programs",  
  "displayName": "Program",  
  "properties": [  
    {  
      "fieldName": "dataEntryForm",  
      "propertyType": "REFERENCE",  
      "collection": false,  
      "required": false  
    },  
    {  
      "fieldName": "publicAccess",  
      "propertyType": "TEXT",  
      "collection": false,  
      "required": false  
    },  
    {  
      "fieldName": "ignoreOverdueEvents",  
      "propertyType": "BOOLEAN",  
      "collection": false,  
      "required": false  
    }  
  ]  
}
```



A screenshot of a detailed JSON schema for the "Program" resource, showing all attributes and their values. The schema includes a "schemas" array with one item, which contains a "klass" field pointing to the "AttributeValue" class and various properties like "shareable", "metadata", and "plural". It also includes sections for "references", "authorities", and "properties" with their own detailed definitions.

```
{  
  "schemas": [  
    {  
      "klass": "org.hisp.dhis.attribute.AttributeValue",  
      "shareable": false,  
      "metadata": false,  
      "plural": "attributeValues",  
      "displayName": "Attribute Value",  
      "collectionName": "attributeValues",  
      "implicitPrivateAuthority": false,  
      "nameableObject": false,  
      "name": "attributeValue",  
      "namespace": "http://dhis2.org/schema/dxf/2.0",  
      "singular": "attributeValue",  
      "persisted": true,  
      "references": [  
        "org.hisp.dhis.attribute.Attribute"  
      ],  
      "authorities": [],  
      "properties": [  
        {  
          "fieldName": "lastUpdated",  
          "simple": true,  
          "required": false,  
          "writable": true,  
          "nameableObject": false,  
          "klass": "java.util.Date",  
          "propertyType": "DATE"  
        }  
      ]  
    }  
  ]  
}
```



- API supports the following other methods:
 - **POST**
 - Creates a new entry
 - See resource schema on `/api/schemas/<resource>` for required fields
 - **DELETE**
 - Delete an entry
 - Might have dependencies!
 - **PUT**
 - *Replace* the whole item
 - Requires app to download whole object
 - **PATCH**
 - Change specific attributes
 - Might not work on all endpoints – try!

Prototype with Postman, Curl or a similar tool!



Without app-runtime

```

import React, { useState, useEffect } from 'react'
const baseUrl = 'http://localhost:8080'

const MyComponent = ({ id }) => {
  const [ loading, setLoading ] = useState(true)
  const [ error, setError ] = useState(undefined)
  const [ data, setData ] = useState(undefined)

  useEffect(() => {
    setLoading(true)
    const controller = new AbortController()
    fetch(baseUrl + '/api/32/indicators/' + id, { signal: controller.signal })
      .catch(() => {
        if (!controller.aborted) {
          setError('An unknown network error occurred')
        }
      })
      .then(response => {
        if (controller.aborted) {
          return;
        }
        ...
      })
  })
}

```

```

      if (
        response.status === 401 ||
        response.status === 403 ||
        response.status === 409
      ) {
        setError('Access denied')
      } else if (response.status < 200 || response.status >= 400) {
        setError('An unknown error occurred')
      } else {
        setData(response.json())
      }
    setLoading(false)
  })

  return () => {
    controller.abort()
  }
}, [id])

return (
  <div>
    {loading && '...'}
    {error && 'ERROR'}
    {data && ('Indicator name: ' + data.displayName)}
  </div>
)
}

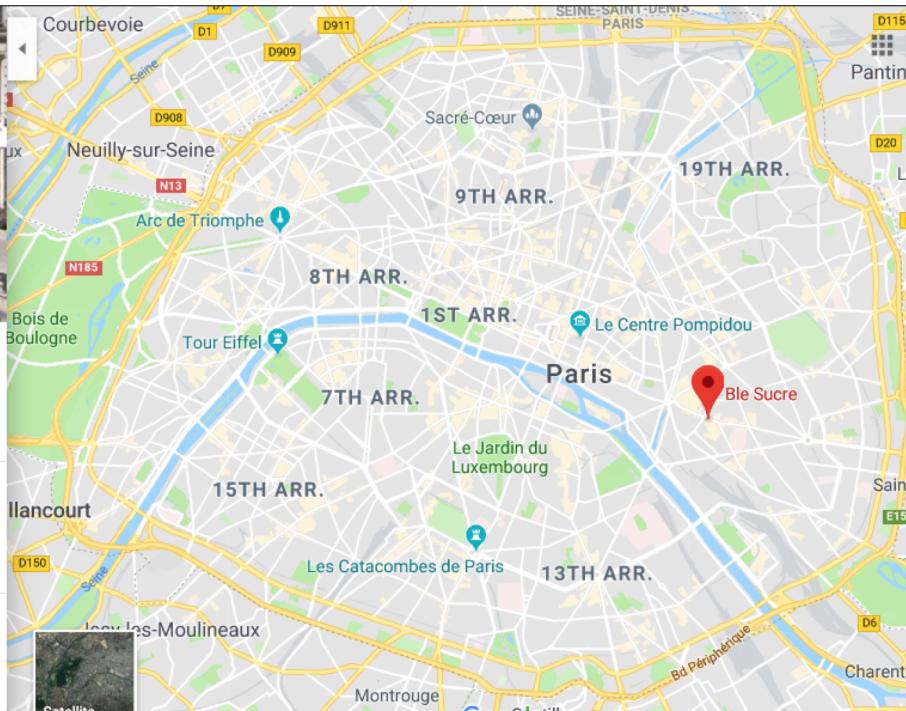
```



Ble Sucre
4.3 ★★★★☆ (3)
Bakery

 Directions  Save  Nearby  Send to your phone  Share

 7 Rue Antoine Vollon, 75012 Paris
 V92G+6M Paris



The map displays the Parisian arrondissements (1ST ARR., 7TH ARR., 8TH ARR., 9TH ARR., 13TH ARR., 15TH ARR., 19TH ARR.) and major landmarks (Arc de Triomphe, Tour Eiffel, Bois de Boulogne, Le Jardin du Luxembourg, Les Catacombes de Paris). The Seine River flows through the city. A red location pin marks the exact position of Ble Sucre on the map.

A destination is declarative



10 min (800 m)



via Passage Charles Dallery and Avenue Ledru-Rollin
Mostly flat

- ↑ Head southwest on Rue de la Roquette toward
Passage Charles Dallery

47 m

- ↶ Turn left onto Passage Charles Dallery

210 m

- ↷ Turn right onto Avenue Ledru-Rollin

400 m

- ↶ Turn left onto Rue du Faubourg Saint-Antoine

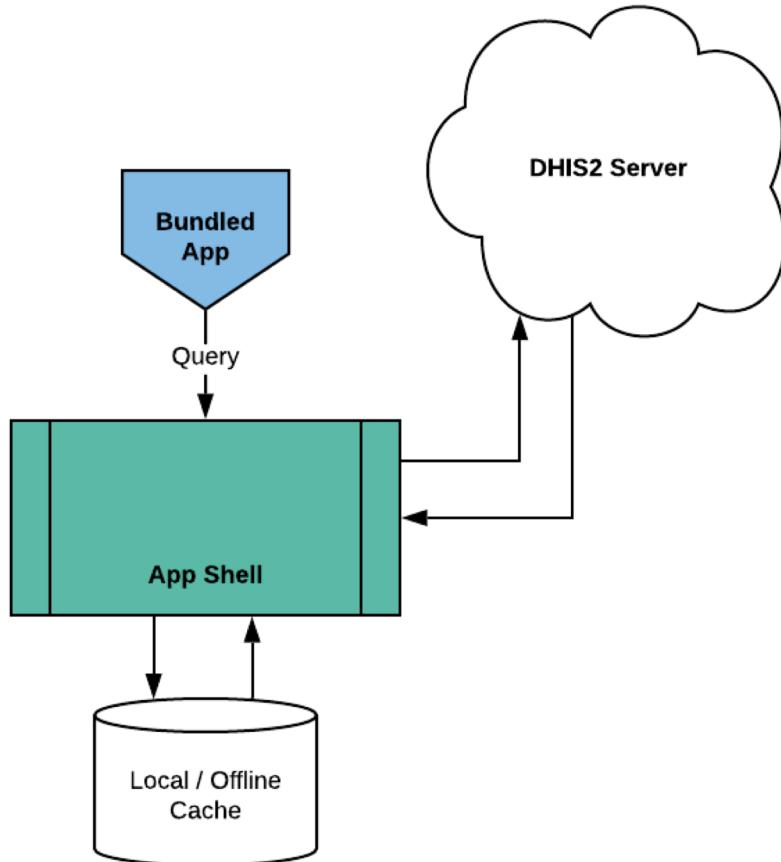
79 m

- ↷ Turn right onto Rue Antoine Vollen

ⓘ Destination will be on the right

59 m

Turn-by-turn directions are imperative



With app-runtime

```
1 import React from 'react'
2 import { useDataQuery } from '@dhis2/app-runtime'
3
4 const query = {
5   indicator: {
6     resource: 'indicators',
7     id: ({ id }) => id
8   }
9 }
10 const MyComponent = ({ id }) => {
11   const { loading, error, data } = useDataQuery(query, {
12     variables: {
13       id
14     }
15   })
16
17   return (
18     <div>
19       {loading && ...}
20       {error && 'ERROR'}
21       {data && ('Indicator name: ' + data.indicator.displayName)}
22     </div>
23   )
24 }
```



```
1 import React from 'react'
2 import { useDataQuery } from '@dhis2/app-runtime'
3
4 const query = {
5   indicator: {
6     resource: 'indicators',
7     id: ({ id }) => id
8   },
9   me: {
10     resource: 'me'
11 }
12 }
13 const MyComponent = ({ id }) => {
14   const { loading, error, data } = useDataQuery(query, {
15     variables: {
16       id
17     }
18   })
19
20   return (
21     <div>
22       {loading && '...'}
23       {error && 'ERROR'}
24       {data && (
25         <div>
26           <span>My name: {data.me.name}</span>
27           <span>Indicator name: {data.indicator.displayName}</span>
28         </div>
29       )}
30     </div>
31   )
32 }
```

Multiple Queries (in parallel)

Pagination

(using dynamic queries & refetch)

```

1  ✓ import React from 'react'
2    import { Button } from '@dhis2/ui-core'
3    import { useDataQuery } from '@dhis2/app-runtime'
4
5  ✓ const query = {
6    indicators: {
7      resource: 'indicators',
8      params: ({ page = 1 }) => ({
9        order: 'displayName:asc',
10       page,
11       pageSize: 10
12     })
13   }
14 }

```



```

16  const IndicatorList = () => {
17    const { loading, error, data, refetch } = useDataQuery(query)
18
19    if (loading) return <span>...</span>
20    if (error) return <span>` ERROR: ${error.message}` </span>
21
22    const { pager, indicators } = data.indicators;
23    const { page, pageCount } = pager
24
25    return (<>
26      <h3>Indicators</h3>
27      <ul>
28        {indicators.map(ind => (
29          <li>{ind.displayName}</li>
30        )))
31      </ul>
32      <div style={{ fontSize: '0.8rem' }}>
33        Page {page} of {pageCount}
34      </div>
35      <Button
36        disabled={page === 1}
37        onClick={() => refetch({ page: page - 1 })}
38      >
39        &lt;- Previous
40      </Button>
41      <Button
42        disabled={page === pageCount}
43        onClick={() => refetch({ page: page + 1 })}
44      >
45        Next -&gt;
46      </Button>
47    </>)

```

Parameter	Explanation	Example
?paging=false	Disable paging	
?filter	Filter items on given constraint	?filter=id:eq:IpHINAT79UW
?fields	Show given fields	?fields=id,displayName
	Show all fields	?fields=:all
	Show properties of embedded object	?fields=id,programStages[id,displayName]

```
1 import React from 'react'
2 import { Button } from '@dhis2/ui-core'
3 import { useDataMutation } from '@dhis2/app-runtime'
4
5 const mutation = {
6   resource: 'indicators',
7   id: ({ id }) => id,
8   type: 'delete',
9 }
10
11 export const DeleteButton = ({ indicatorId, onDelete }) => {
12   const [mutate] = useDataMutation(mutation, {
13     onComplete: onDelete,
14     variables: {
15       id: indicatorId,
16     },
17   })
18
19   return <Button onClick={mutate}>Delete</Button>
20 }
```

Mutation

(create / update / delete)

Un-Versioning

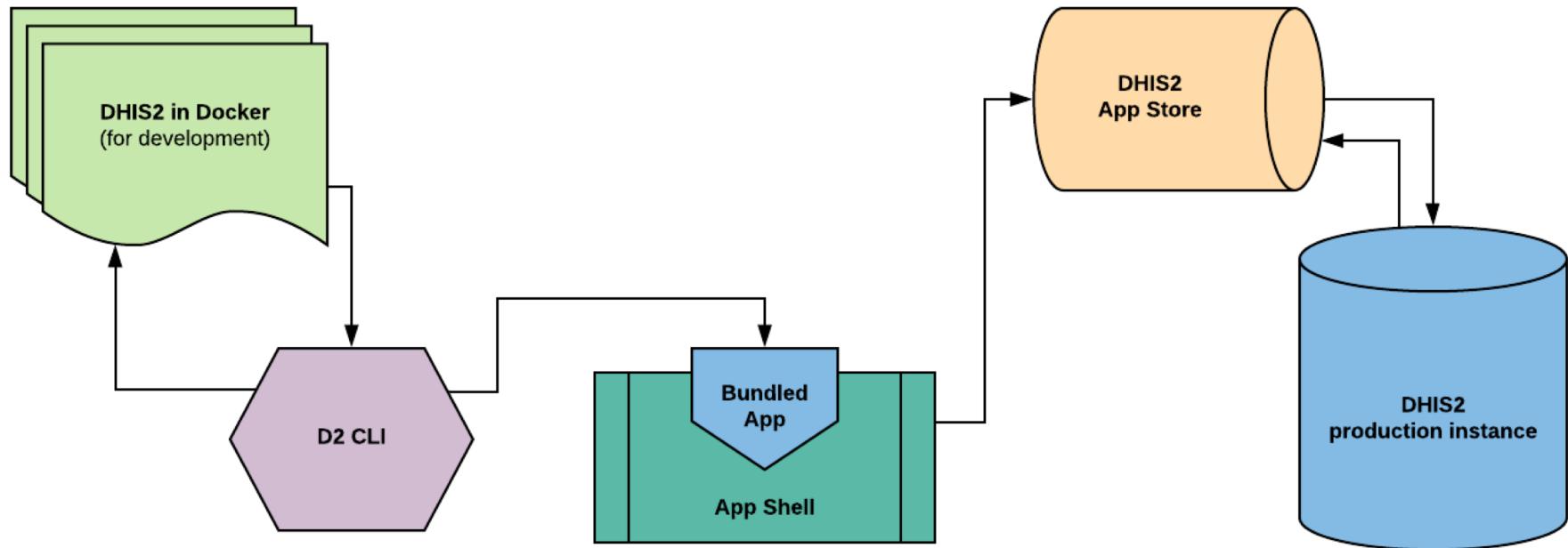
Goal:

Decouple the application version from the Core version

Use **Feature Toggling** to support multiple Core versions in a single application

- Check the version of the server
- Disable unsupported features
- Translate data requests and responses in the AppShell

Application Development Ecosystem





To spin up a local DHIS2 instance (version 2.32.0):

1. Install yarn (<https://yarnpkg.com/>)

2. Install **d2 CLI**

```
> yarn global add @dhis2/cli
```

3. Install Docker (<https://www.docker.com/products/docker-desktop>)

4. Run **d2 cluster up**

```
> d2 cluster up 2.32.0 --db-version 2.32 --seed
```

5. Run **d2 cluster logs** and wait until you see **Server startup in xxxxx ms**

```
> d2 cluster logs 2.32.0
```

6. Open <http://localhost:8080>, login as **admin / district**

7. Run **Analytics Table => Export** from the **Data Administration App**

Resources

- Developer Portal - dhis2.nu
- App Platform
 - platform.dhis2.nu
 - runtime.dhis2.nu
- d2 CLI - github.com/dhis2/cli
- UI & Design
 - github.com/dhis2/design-system
 - ui-core.dhis2.nu
 - ui-widgets.dhis2.nu
- General DHIS2 - docs.dhis2.org (v2.32)
 - User Manual
 - Developer Manual => Web API
- Examples
 - github.com/dhis2/usage-analytics-app
 - github.com/dhis2/app-platform
=> /examples
 - github.com/dhis2/app-runtime
=> /examples

Questions?

Aussi en français

También en español

Austin McGee

austin@dhis2.org

25 September 2019