GEN3 DATA COMMONS

Gen3.2 - How to build a Gen3 data portal using the new frontend framework

Gen3 Community Forum
1 May 2024

THE UNIVERSITY OF CHICAGO CENTER FOR TRANSLATIONAL DATA SCIENCE

OHSU

OCC

Australian BioCommons

NeSI New Zealand eScience Infrastructure

The Agenda

- Introduction
- Gen3.2 Features and Capabilities - Craig Barnes, CTDS
- Migrating from Windmill to Gen3.2 - Matthew Peterkort, Oregon Health & Science University
- Q&A

Gen3.2 Features and Capabilities

Craig Barnes
Center for Translational Data Science
University of Chicago

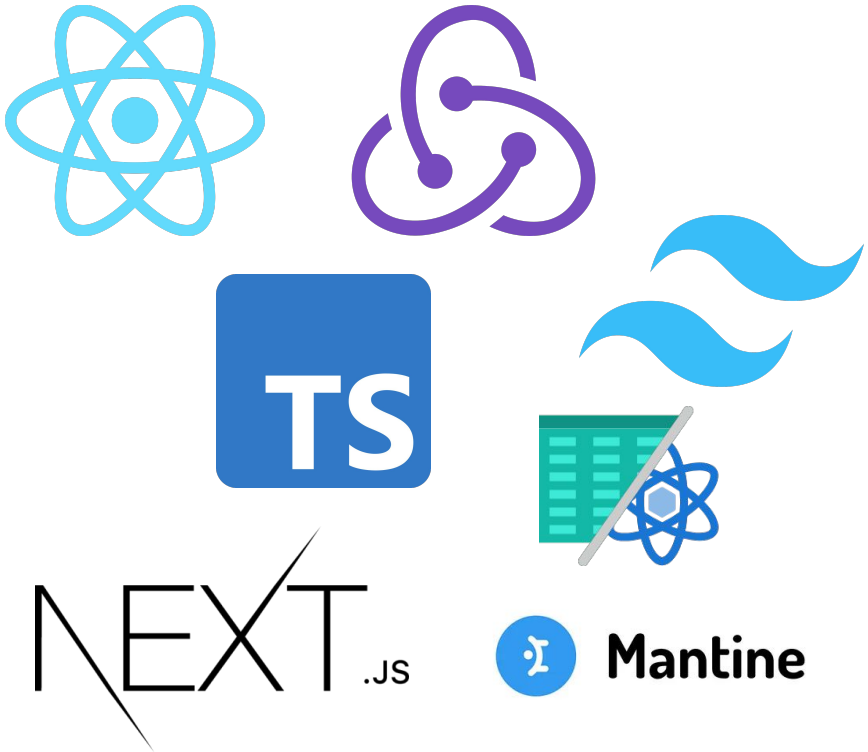
Gen3 Frontend Framework: Gen3.2

The Gen3 Frontend Framework provides:

- Integrated analysis tools
- Custom content
- Per commons codebase which enables a more flexible source code management and deployment
- Improved user, development, and administration experience
- Upgraded technology stack
- Extensibility and customization

Core Technologies

- Core:
  - React 18
  - Typescript
  - Redux-toolkit
- Application Framework: Next.js 14
- Styling: Tailwind CSS
- UI Components:
  - Mantine.dev
  - mantine react table
- Gen3 Components
- Content:
  - HTML and MDX based static pages
  - Next.js custom page



Gen3.2 Features

- Styling and theming
- Data renderer customization for tables, charts
- Global selection feature: My Data Library
- Analysis tools
- Commons specific pages
- Update designs and improved UX

Gen3.2 Applications

Existing:

- Explorer
- Discovery
- Workspaces
- Data Dictionary
- GraphQL query UI
- Profile
- Data Submission

New:

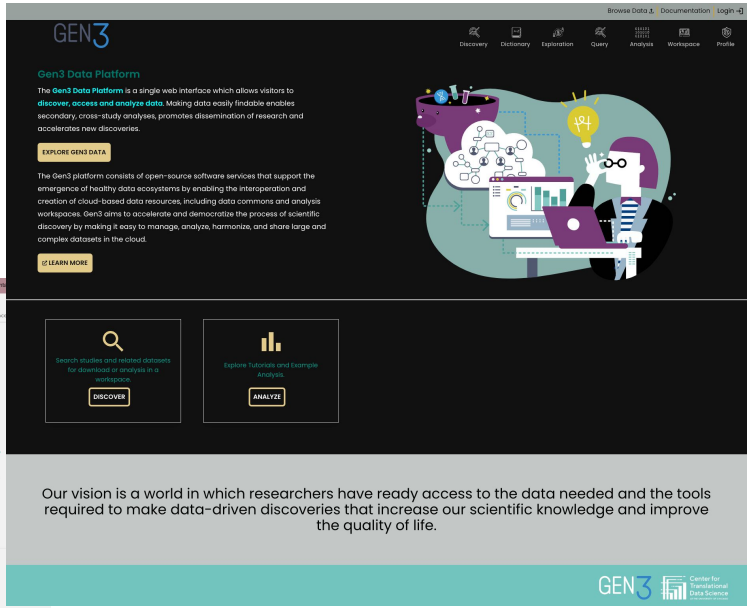
- Application Center
- My Data Library
- Administration UIs
- LLM Search
- Cohort Discovery
- 3rd party tools and applications

Styling and Theming

The portal's style can be customized by setting:

- Color theme
- Fonts
- Icons

Configuration tools
Style overrides



Colors

Gen3.2's Color Theme is based on USWDS theme color token: primary/secondary/accent following the proportional 60/30/10 relationship

Navigation

Configured like data-portal

Customizable icons

3 layouts:

Classic:

Horizontal:

Navigation: Vertical

GEN3 DATA COMMONS

GEN3

Discovery

Dictionary

Exploration

Query

Analysis

Workspace

Profile

Gen3 Data Platform

The **Gen3 Data Platform** is a single web interface which allows visitors to **discover, access and analyze data**. Making data easily findable enables secondary, cross-study analyses, promotes dissemination of research and accelerates new discoveries.

EXPLORE GEN3 DATA

The Gen3 platform consists of open-source software services that support the emergence of healthy data ecosystems by enabling the interoperation and creation of cloud-based data resources, including data commons and analysis workspaces. Gen3 aims to accelerate and democratize the process of scientific discovery by making it easy to manage, analyze, harmonize, and share large and complex datasets in the cloud.

LEARN MORE

Development support

- Striving to make developing commons as simple as possible.
- Initial set of documentation:
  - https://github.com/uc-cdis/gen3-frontend-framework/tree/**develop**/docs
- .env files for configuration
- Fine grain connection to remote Gen3 services
- Credentials based login
- Run Gen3.2 development outside of helm charts
- Local revproxy instruction
- Supported in **gen3-helm** charts
- Continuously refining development support

Environment Files

NextJS supports **.env** files for various deployments:

- .env.development
- .env.production

You can add more:

- .env.testing
- .env.staging

```
GEN3_COMMONS_NAME=gen3
NEXT_PUBLIC_GEN3_API=https://localhost:3010
```

Pending:

Set env variables/config in helm `values.yaml`

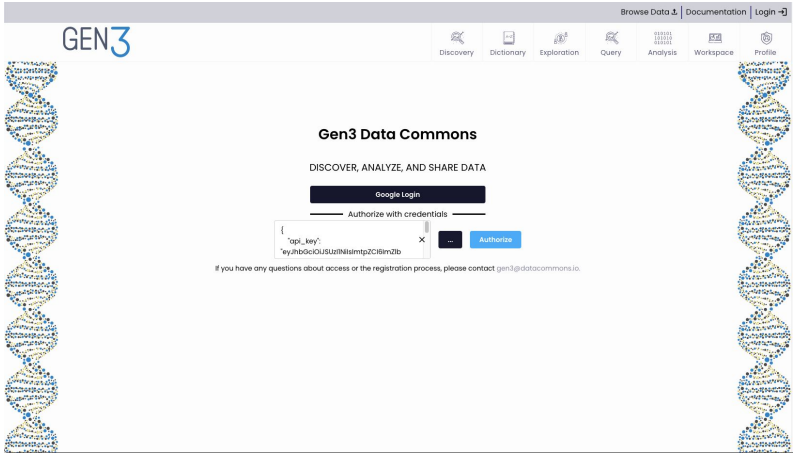
Environment Files

Override any Gen3 endpoint

```
NEXT_PUBLIC_GEN3_FENCE_API=https://localhost:3010
NEXT_PUBLIC_GEN3_DOMAIN=https://localhost:3010
NEXT_PUBLIC_GEN3_MDS_API=https://localhost:3010/mds
NEXT_PUBLIC_GEN3_AI_SEARCH_API=https://localhost:3010/ai
NEXT_PUBLIC_GEN3_GUPPY_API=https://localhost:3010/guppy
NEXT_PUBLIC_GEN3_FENCE_API=https://localhost:3010
NEXT_PUBLIC_GEN3_AUTHZ_API=https://localhost:3010/authz
NEXT_PUBLIC_GEN3_WORKSPACE_STATUS_API=https://localhost:3010/lw-workspace
NEXT_PUBLIC_GEN3_SUBMISSION_API=https://localhost:3010/api/v0/submission
```

Credentials based login

Mostly for development at CTDS:
Credentials based login



- Login with no fence redirect
- Connect FE to remote commons
- Only in development environment
- Scope restrictions on token

Explorer/Cohort Builder

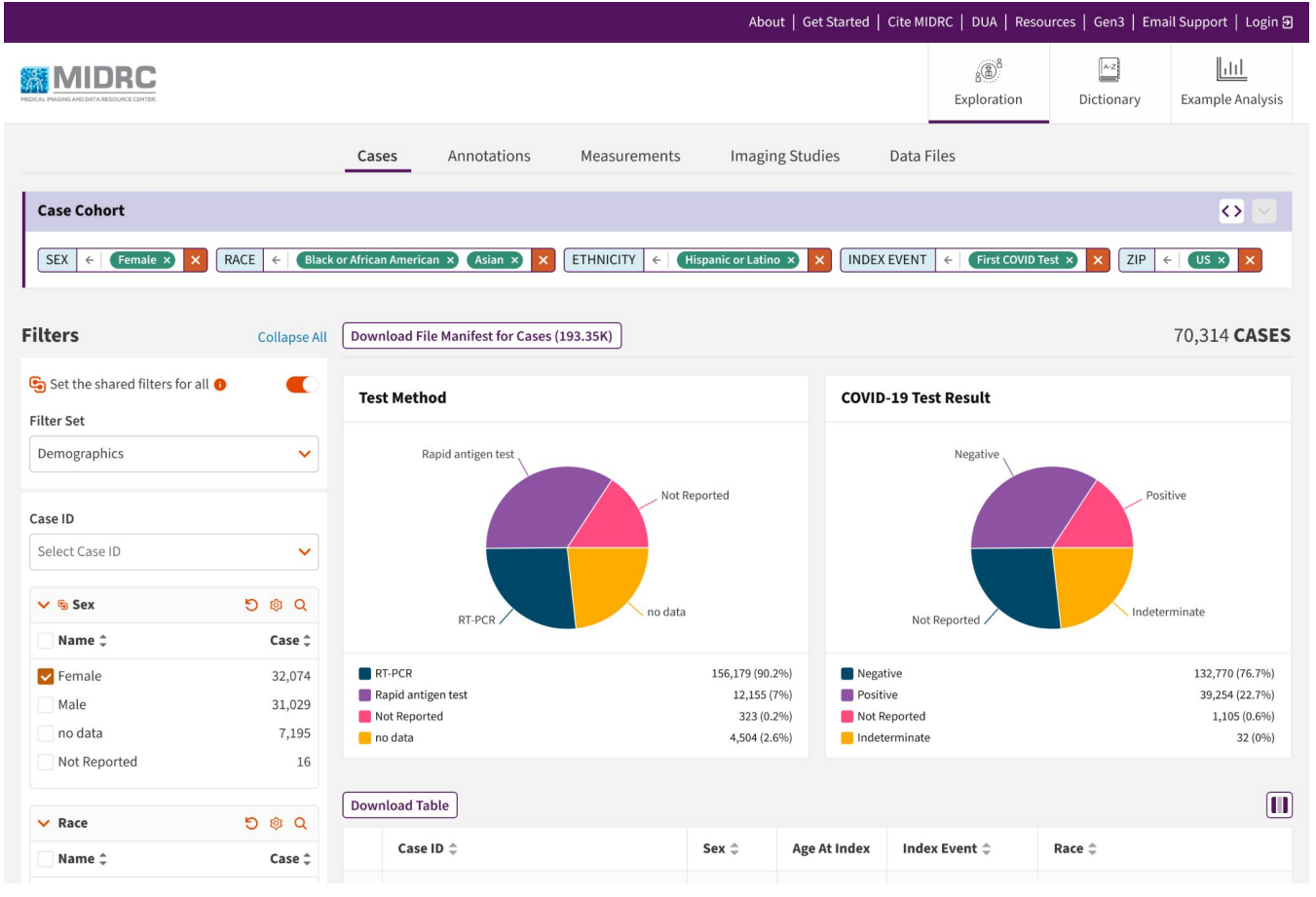
Extensible to custom data renderer for table cells and charts

Gen3.2 implementation of Guppy UI components

Pending work:

- Elimination of filter tabs
- Support for sharing selection with the same facet on other indexes
- Subtable in main table.
- Row details
- Authz based filtering

New Explorer UI



Explorer in action: imaging-hub

Discovery

Discovery Page for metadata browsing and searching
Support for custom renderers for table cells and rows
Once registered the cell renders can be added to the configuration
Define a data hook for retrieving and processing metadata from the Gen3 MDS
Basic functionality is working for all components
Todo:

- Advanced search
- Details page
- Selection
- Actions: export to workspace

Data Renderers

Gen3.2 support extensibility using data renderers

Enables data to have visual representation in Tables and Charts

Overview of the process:

1. Write a function that takes a value and returns a ReactElement
2. Register function:

```
DiscoveryCellRendererFactory.registerCellRendererCatalog({...});
```

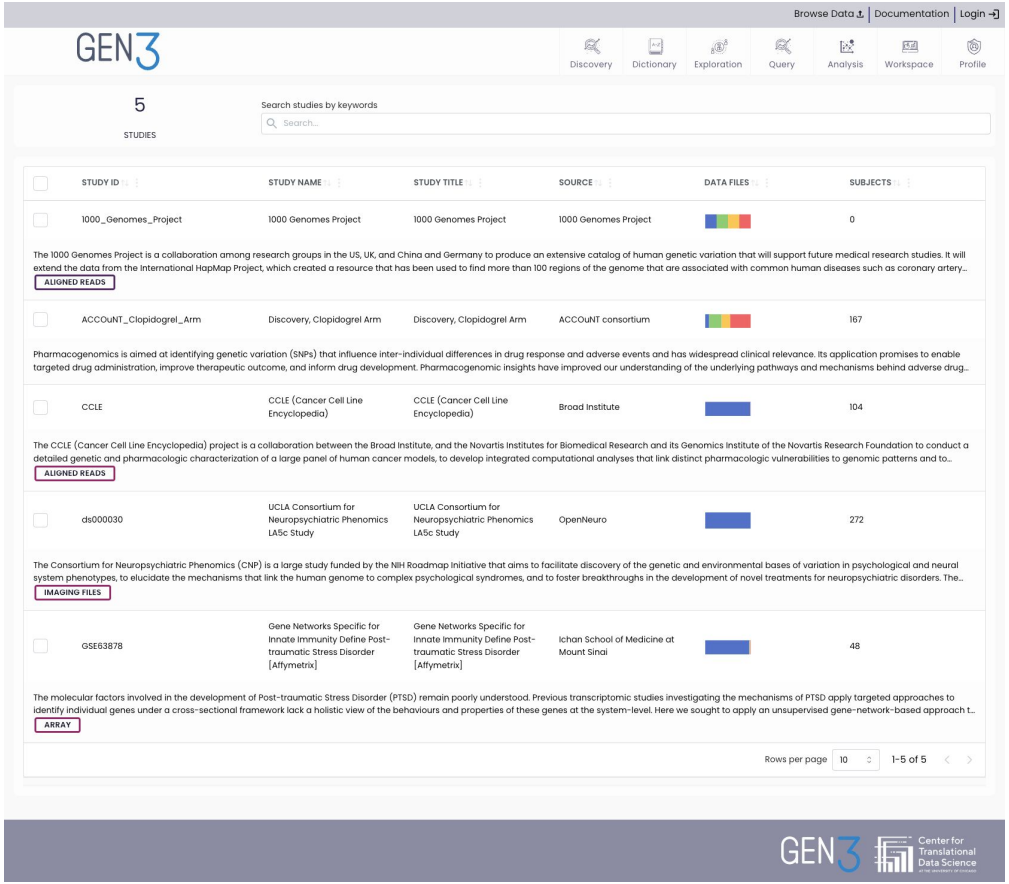
3. Add the function to the configuration:

```
"field": "__manifest",
"contentType": "manifest",
"cellRenderFunction": "inline",
```

Data Renderer Example

GEN3 DATA COMMONS

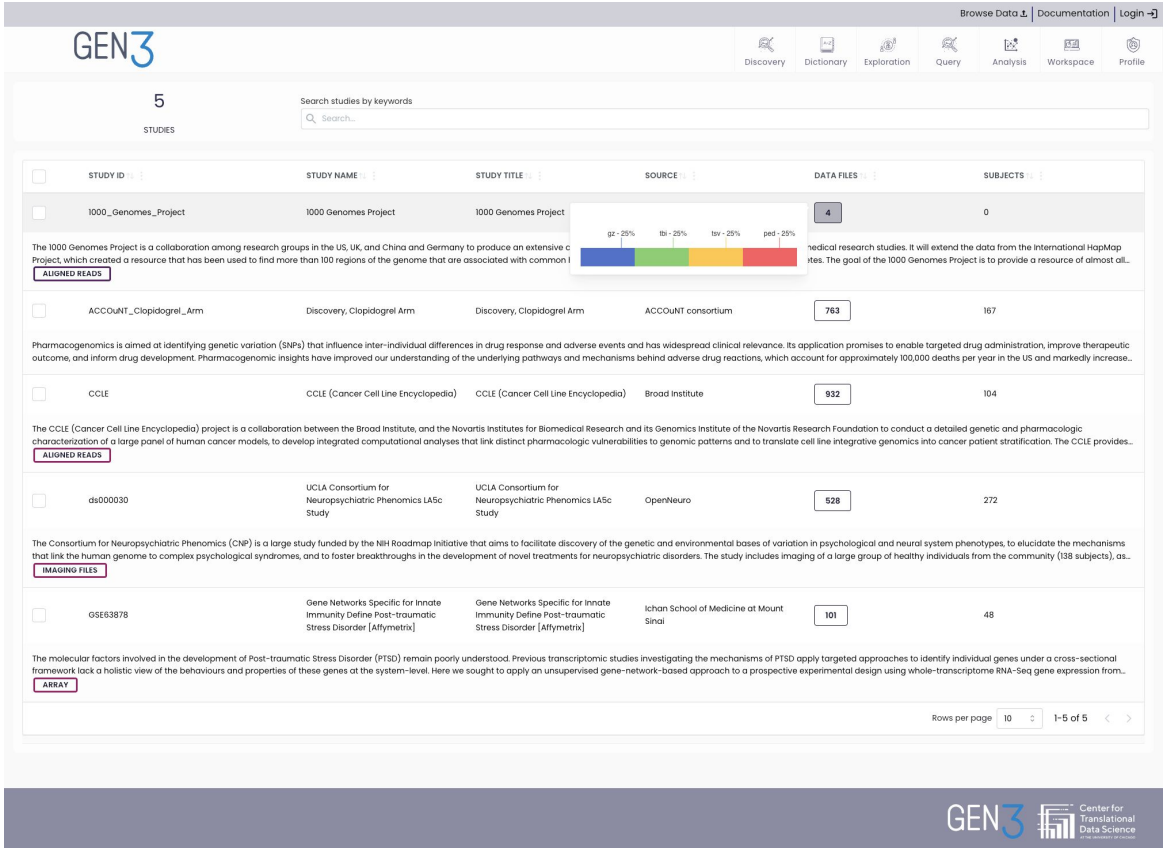
Inline chart renderer `__manifest` field Discovery Table



Data Renderer Example

Popup chart renderer

`__manifest` field Discovery Table



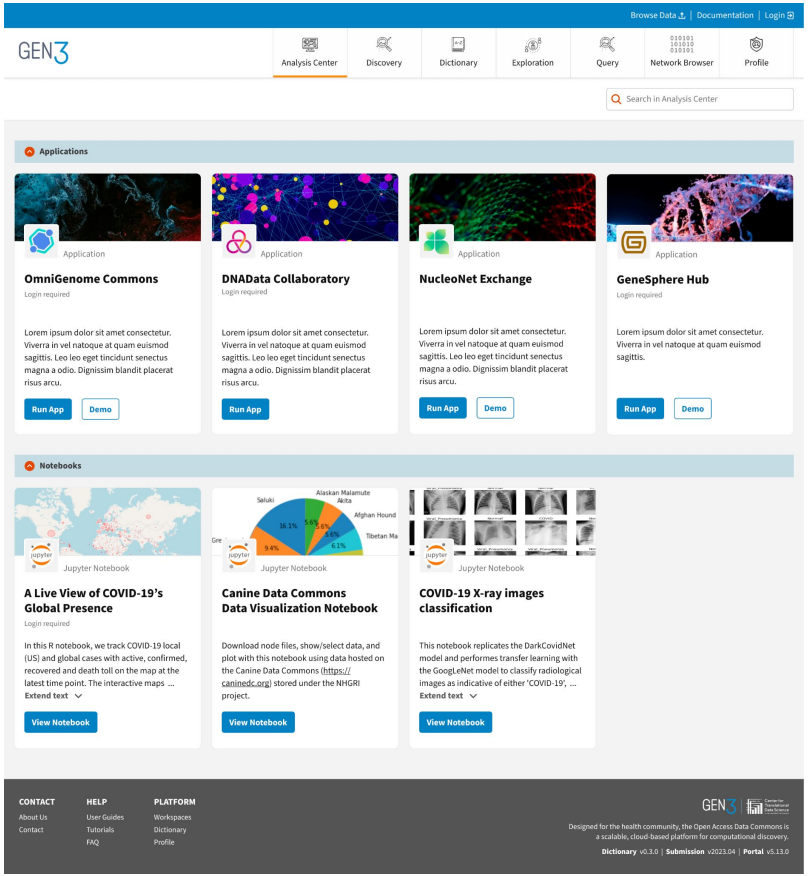
Custom Pages

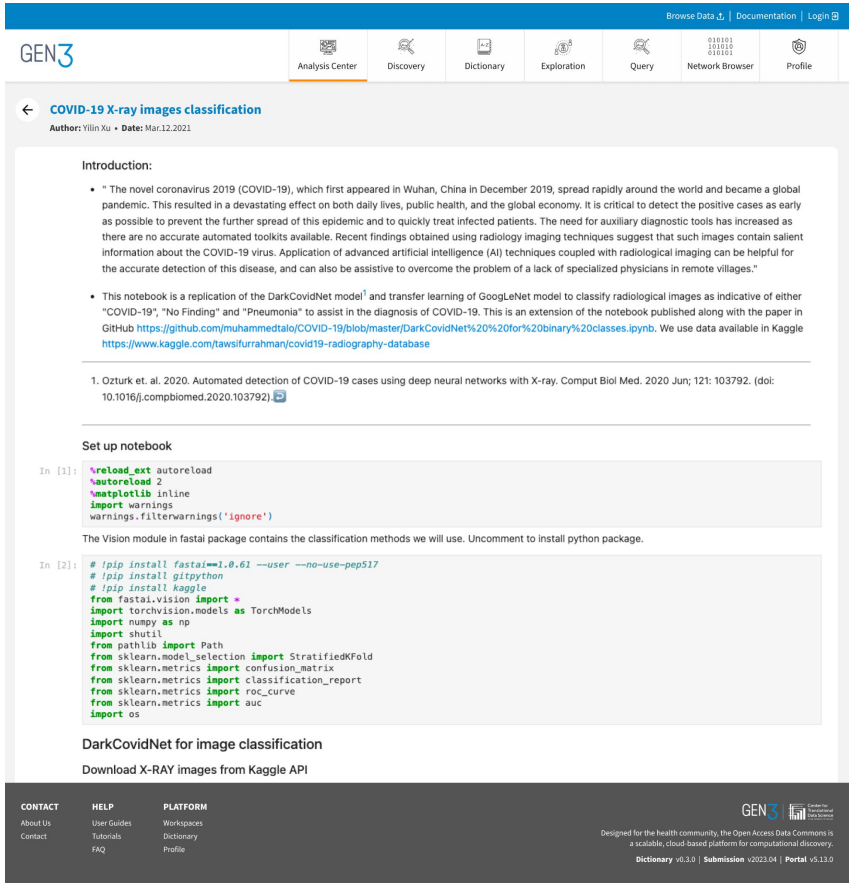
The application framework of Gen3.2 is NextJS (vers 14.1)
Adding new pages is a matter of adding a tsk file to `src/pages`:
There is a template in `pages/SamplePage.tsx`

```
const SamplePage = ({ headerProps, footerProps }:  NavPageLayoutProps ) => {
 return (
   <NavPageLayout {...{ headerProps, footerProps }}>
     <div className="w-full m-10">
       <Center>
       <Paper shadow="md" p="xl" withBorder>
         <Text>This is a example custom page in Gen3</ Text>
         <Text>
           You can add your own content here, and add a link to this page in
           the navigation bar by editing the config file in
           navigation.json
         </ Text>
       </ Paper>
       </ Center>
     </ div>
   </ NavPageLayout >
 );
};
```

Analysis Tool Center

GEN3 DATA COMMONS





Running with data-portal

Possible to run data-portal simultaneously with Gen3.2

In helm charts (currently on a gen3-helm branch)

- Add configuration for frontend-framework and data-portal in value.yaml
- Add frontendRoot: gen3ff

Gen3.2 will be on / and data-portal will be on `/portal`

frontendRoot: portal

data-portal will be on / Gen3.2 will be on `/ff`

Roadmap

May 2024

- Workspaces
- Update explorer UI
- Discovery page improvements

June 2024:

- Analysis Center
- My Data Library
- Data Dictionary

July 2024:

- New application
- Homepage layouts
- Data Submission

Remaining:

- Eliminate issues and flaws
- Testing framework
- Accessibility compliance
- Documentation
- Improved deployment configuration
- Modals
- Nextjs app router
- SSR components
- Simplify configuration:
  - admin UI
  - config validation
- …

Migrating from Windmill to Gen3.2

Matthew Peterkort
Oregon Health & Science University

Local Development

https://github.com/uc-cdis/gen3-frontend-framework/blob/develop/docs/Local%20Development/QuickStart.md

Gen3.2 Frontend Quickstart

This guide will get you up to speed on configuring a Gen3 Frontend Applications. There are other tutorial on running it locally (for example: (Local Development)[]).

Getting Started

Gen3 Data Commons using the Gen3 Frontend Framework is a matter of the following:

- create a clone of Gen3 Data Commons Application
- Configure the commons by editing the configuration files in the `config` directory.
- Add your pages and content
- Deploy via helm charts or Docker.

Changes to the Gen3 Data Commons Application can be pulled from the Common Frontend Repository. You need configure git to pull from the Common Frontend Repository.

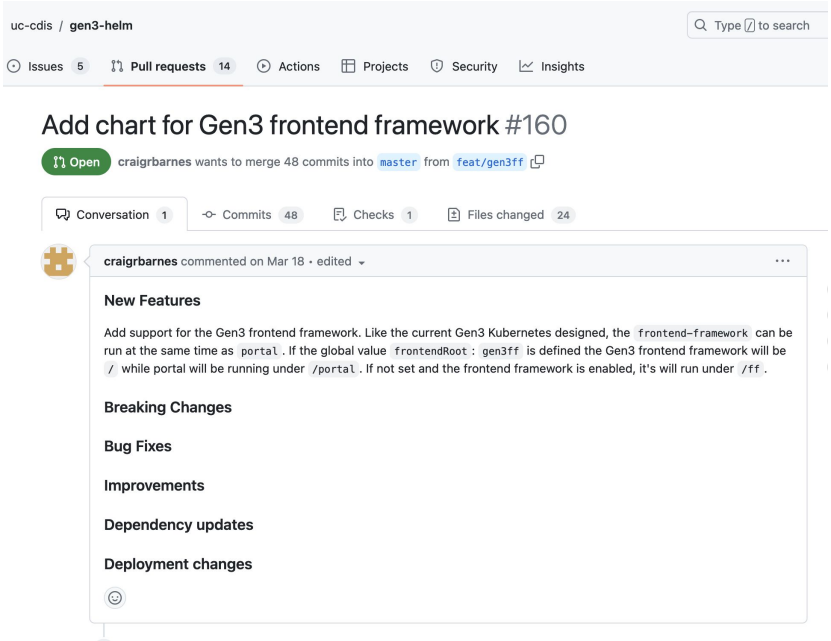
```
git remote add upstream https://github.com/uc-cdis/commons-frontend-app.git
```

Local Development Setup

- NVM is a painless way of managing Node versions
- Gen3 helm kubernetes instance required
- It is important to understand that the local development setup passes portal requests to your local helm nginx setup
- If strictly following the docs using example revproxy config make sure you change your values.yaml hostname to be localhost
- If you wish to customize and add to gen3-frontend-framework fork from: https://github.com/uc-cdis/commons-frontend-app
- If you wish to develop and make contributions to frontend framework fork from: https://github.com/uc-cdis/gen3-frontend-framework/tree/develop
- See gen3-frontend/framework/docs/Local Development/Using Helm Charts/ Local Development with Helm Charts on the develop branch for setup docs.

Deployment Setup

- Gen3 Helm supports frontend framework deployments with minimal changes to existing helm gen3 deployments: https://github.com/uc-cdis/gen3-helm/pull/160
- Make sure that “npm run build” and “npm run start” work as expected before building a custom image.
- Separate production and development environment variables are defined in the sample Commons. Make sure that the production hostname matches environment defined in helm and in the frontend-framework.



Sample Commons Directory Layout

- src - Top level source code pages imported from @gen3/frontend
- public - The actual custom content files
- Config - Highly customizable settings files that control the look and feel of the website

```
src
├── lib
│   ├── CohortBuilder
│   │   └── CustomCellRenderers.tsx
│   ├── Discovery
│   │   ├── CustomCellRenderers.tsx
│   │   └── CustomRowRenderers.tsx
│   └── plugins
│       └── index.js
├── pages
│   ├── 404.tsx
│   ├── AISearch.tsx
│   ├── Colors.tsx
│   ├── DEMOGRIP.tsx
│   ├── DataDictionary.tsx
│   ├── Discovery.tsx
│   ├── Explorer.tsx
│   ├── GRIPREF.tsx
│   ├── Login.tsx
│   ├── Profile.tsx
│   ├── Query.tsx
│   ├── SamplePage.tsx
│   ├── Workspaces.tsx
│   ├── _app.tsx
│   ├── _document.tsx
│   ├── api
│   │   └── auth
│   │       ├── credentialsLogin.ts
│   │       ├── credentialsLogout.ts
│   │       ├── sessionLogout.ts
│   │       └── sessionToken.ts
│   └── index.tsx
└── styles
    └── globals.css
```

```
public
├── fonts
│   └── FranklinGothic.ttf
├── icons
│   ├── ACED_PARTNERS.jpg
│   ├── createdby.png
│   ├── gen3-dark.png
│   ├── gen3-dark.svg
│   ├── gen3.png
│   ├── logo.png
│   ├── logo_small_2.png
│   ├── midrc-logo.png
│   └── rolesPages
│       ├── analyze-local.png
│       ├── analyze-workspace.png
│       ├── behavior.png
│       ├── brain.png
│       ├── browse.png
│       ├── chronic.png
│       ├── database.png
│       ├── diagnose.png
│       ├── download.png
│       ├── effectiveness.png
│       ├── group.png
│       ├── intervention.png
│       ├── medication.png
│       ├── mental.png
│       ├── model.png
│       ├── new.png
│       ├── pain.png
│       ├── repository.png
│       ├── share.png
│       ├── state.png
│       └── therapy.png
└── images
    ├── brh
    │   ├── gene_bgy.svg
    │   └── logo.png
    ├── gene.svg
    └── sideTexture.png
```

```
config
├── brh
│   ├── colors.json
│   ├── discovery.json
│   ├── footer.json
│   ├── landingPage.json
│   ├── login.json
│   ├── modals.json
│   ├── navigation.json
│   ├── profile.json
│   ├── query.json
│   └── workspaces.json
├── drsHostnames.json
├── gen3
│   ├── colors.json
│   ├── discovery.json
│   ├── explorer.json
│   ├── footer.json
│   ├── landingPage.json
│   ├── login.json
│   ├── navigation.json
│   ├── profile.json
│   └── query.json
├── icons
│   ├── aiSearch.svg
│   ├── analysis.svg
│   ├── bar-chart.svg
│   ├── chevron-down.svg
│   ├── datafile.svg
│   ├── dictionary.svg
│   ├── download.svg
│   ├── exploration.svg
│   ├── gen3-dark.svg
│   ├── gen3.json
│   ├── gen3.svg
│   ├── loginCircle.svg
│   ├── profile.svg
│   ├── query.svg
│   ├── upload.svg
│   ├── user.svg
│   └── workspace.svg
├── modals.json
├── session.json
├── siteConfig.json
├── themeColors.json
└── themeFonts.json
```

Config Directory – Global Configurations

GEN3 DATA COMMONS

```
config
├── brh
│   ├── colors.json
│   ├── discovery.json
│   ├── footer.json
│   ├── landingPage.json
│   ├── login.json
│   ├── modals.json
│   ├── navigation.json
│   ├── profile.json
│   ├── query.json
│   └── workspaces.json
├── drsHostnames.json
├── gen3
│   ├── colors.json
│   ├── discovery.json
│   ├── explorer.json
│   ├── footer.json
│   ├── landingPage.json
│   ├── login.json
│   ├── navigation.json
│   ├── profile.json
│   └── query.json
├── icons
│   ├── aiSearch.svg
│   ├── analysis.svg
│   ├── bar-chart.svg
│   ├── chevron-down.svg
│   ├── datafile.svg
│   ├── dictionary.svg
│   ├── download.svg
│   ├── exploration.svg
│   ├── gen3-dark.svg
│   ├── gen3.json
│   ├── gen3.svg
│   ├── loginCircle.svg
│   ├── profile.svg
│   ├── query.svg
│   ├── upload.svg
│   ├── user.svg
│   └── workspace.svg
├── modals.json
├── session.json
├── siteConfig.json
├── themeColors.json
└── themeFonts.json
```

packages > sampleCommons > config > {} session.json > ...

```json
{
  "sessionConfig" : {
    "updateSessionTime": 5,
    "inactiveTimeLimit": 20,
    "logoutInactiveUsers": false
  }
}
```

packages > sampleCommons > config > {} siteConfig.json > ...

```json
{
  "commons": "gen3"
}
```

packages > sampleCommons > config > {} themeFonts.json > ...

```json
{
  "heading": ["FranklinGothic", "Montserrat", "sans-serif"],
  "content": ["FranklinGothic", "Noto Sans", "sans-serif"],
  "fontFamily": "FranklinGothic"
}
```

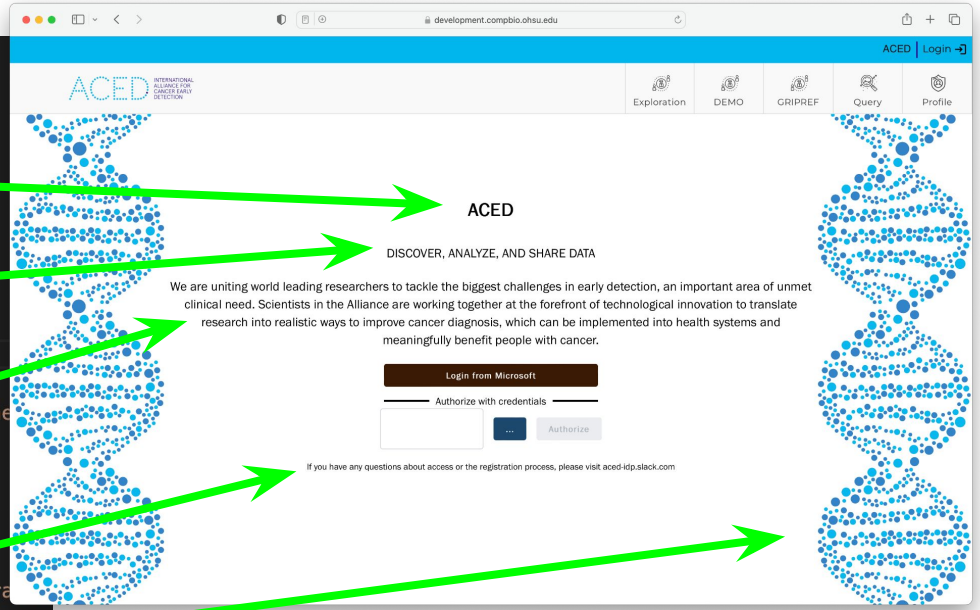
packages > sampleCommons > config > g

```json
{
  "primary": "#00b6ed",
  "secondary": "#BF7E06",
  "accent": "#2e008b",
  "base": "#CCCCCC",
  "accentWarm": "#594B04",
  "accentCool": "#908719",
  "chart": "#0d95A1"
}
```

Page Level Configurations

packages > sampleCommons > config > gen3 > {} login.json > [ ] topContent

```
{
  "topContent": [
    {
      "text": "ACED",
      "className": "text-center text-3xl font-bold"
    },
    {
      "text": "DISCOVER, ANALYZE, AND SHARE DATA",
      "className": "text-center text-xl font-medium"
    },
    {
      "text": "We are uniting world leading researchers to tackle the
      "className": "text-center text-xl font-medium"
    }
  ],
  "bottomContent": [
    {
      "text": "If you have any questions about access or the registra
      "className": "text-center text-sm"
    }
  ],
  "image": "images/gene.svg",
  "showCredentialsLogin" : true
}
```



Page Level Configurations

```
{
  "body": [
    {
      "title": {
        "text": "ACED",
        "level": 2
      }
    },
    {
      "splitarea": {
        "left": [
          {
            "text": "We are <strong>uniting</strong> world leading re
          },
          {
            "link": {
              "href": "/Explorer",
              "linkType": "portal",
              "text": "Explore Gen3 Data"
            }
          },
          {
            "text": "Lorem ipsum dolor sit amet, consectetur adipisci
          },
          {
            "link": {
              "href": "https://gen3.datacommons.io",
              "text": "Learn More"
            }
          }
```

ACED

We are **uniting** world leading researchers to tackle the biggest challenges in early detection, an important area of unmet clinical need. Scientists in the Alliance are working together at the forefront of technological innovation to translate research into realistic ways to improve cancer diagnosis, which can be implemented into health systems and meaningfully benefit people with cancer.

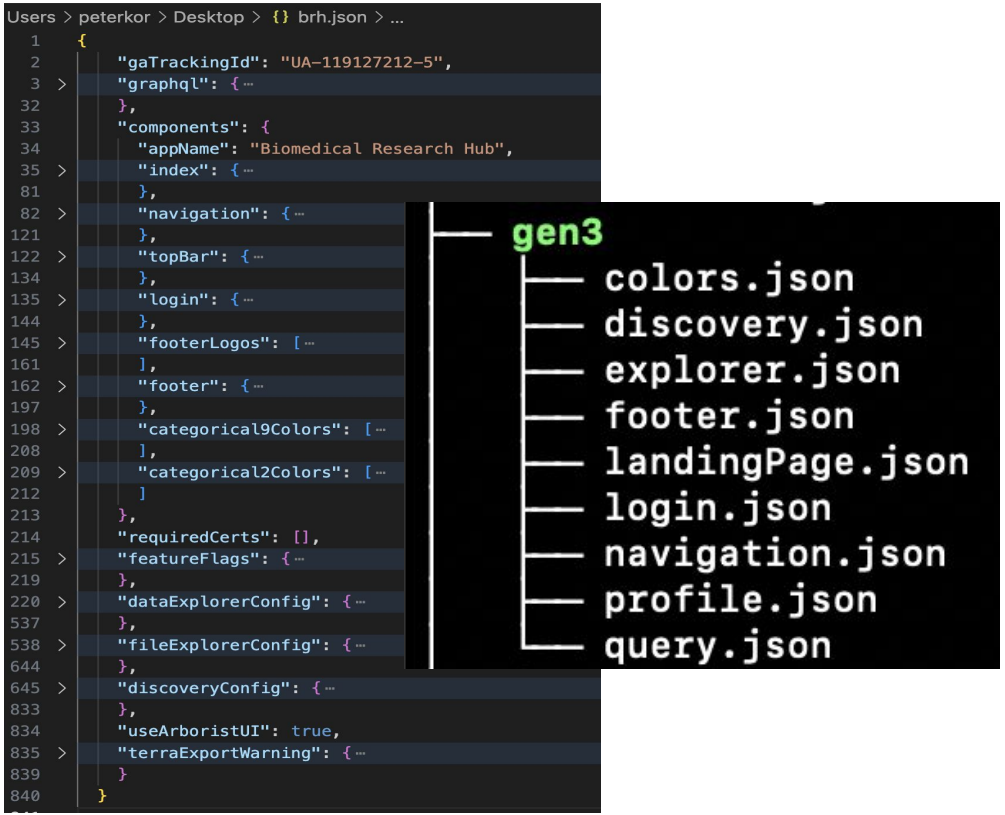
EXPLORE GEN3 DATA

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam elit dui, commodo a tortor ut, vestibulum consectetur tellus. Aliquam erat volutpat. Aenean sodales lacus quis venenatis fermentum. Phasellus varius tempor odio a elementum. Aliquam sed diam molestie, ornare neque in, porttitor mi. Aliquam hendrerit libero volutpat, bibendum ante at, auctor velit. Nulla eget augue a turpis facilisis dictum. Nulla vulputate accumsan aliquam. Fusce imperdiet arcu nibh, nec vehicula nisi placerat at. Phasellus eget magna elementum, tincidunt urna at, venenatis nulla. Cras ullamcorper ante sed ex porttitor, at elementum ligula dapibus. Sed malesuada mi turpis, eu venenatis lorem convallis in.

LEARN MORE

Gitops Migration

- Backwards compatible config files
- Migration from gitops to FF is as simple as a couple of copy and pastes
- Expanded configurations for landing page, colors, profile and others.
- Many configs map straight across to to the expected json file in sampleCommons/config/gen3



Adding a New Font

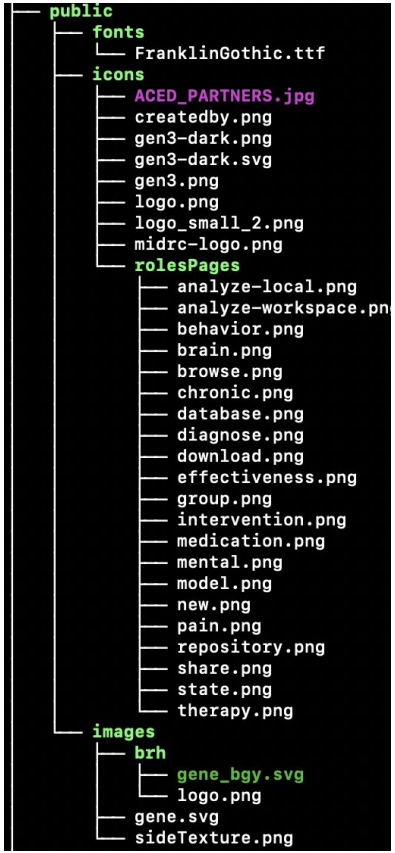
- Add font file to public/fonts
- Add path to font file to globals.css
- Add font name to config/themeFonts.json for corresponding text types

packages > sampleCommons > src > styles > # globals.css > ...

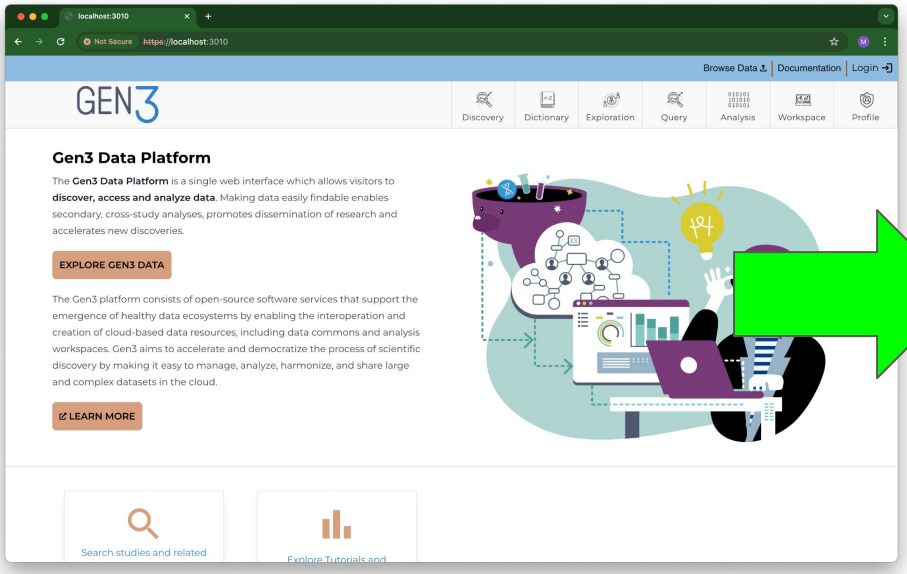
```
80
81  @font-face {
82    font-family: 'FranklinGothic';
83    src: url('/fonts/FranklinGothic.ttf')
84    format('truetype');
85  }
86
```

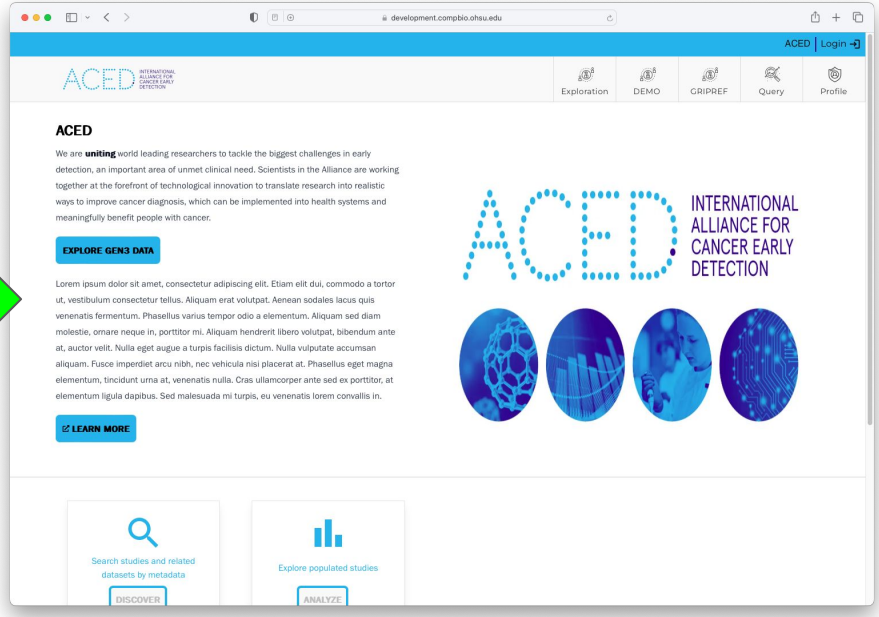
packages > sampleCommons > config > {} themeFonts.json > ...

```
1  {
2    "heading": ["FranklinGothic", "Montserrat", "sans-serif"],
3    "content": ["FranklinGothic", "Noto Sans", "sans-serif"],
4    "fontFamily": "FranklinGothic"
5  }
6
```



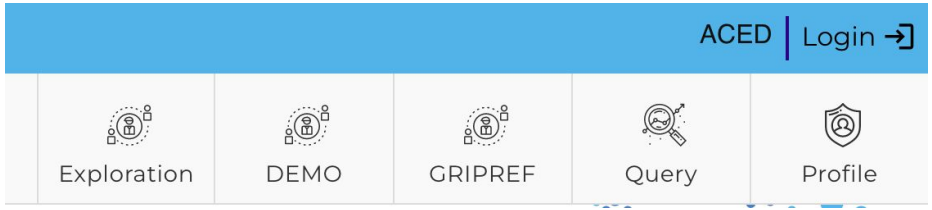
Before / After





Adding a New Page

- Add a new .tsx file to gen3-frontend-framework/packages/sampleCommons/src/pages.
- SamplePage.tsx is given as a template for adding new pages
- <NavPageLayout> component and props serve as boilerplate, ex:

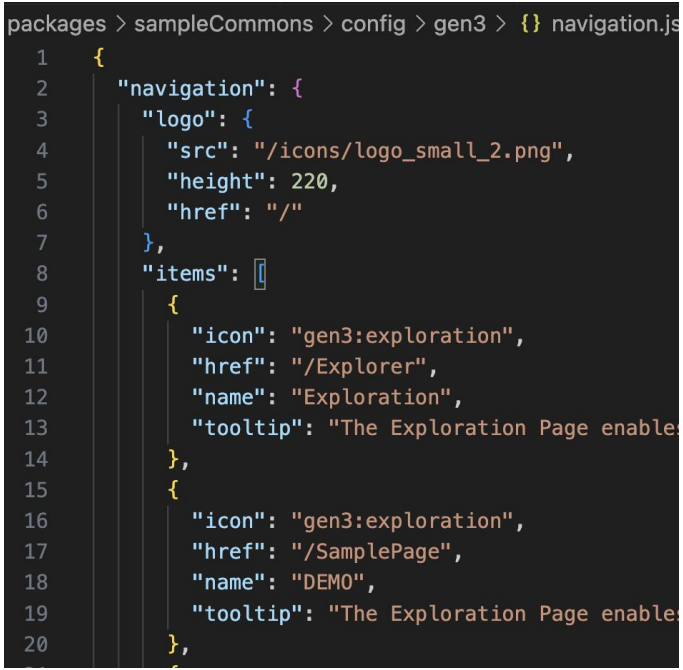


packages > sampleCommons > src > pages > SamplePage.tsx > SamplePage

```tsx
import React from 'react';
import { Text, Paper } from '@mantine/core';
import {
  NavPageLayout,
  NavPageLayoutProps,
  getNavPageLayoutPropsFromConfig,
} from '@gen3/frontend';
import { GetServerSideProps } from 'next';

const SamplePage = ({ headerProps, footerProps }: NavPageLayoutProps) => {
  return (
    <NavPageLayout {...{ headerProps, footerProps }}>
      <div className="w-96 m-10">
        <Paper shadow="md" p="xl" withBorder>
          <Text>This is a example custom page in Gen3</Text>
          <Text>
            You can add your own content here, and add a link to this page in
            the navigation bar by editing the config file in{' '}
            <em>COMMONSNAME</em>/navigation.json
          </Text>
        </Paper>
      </div>
    </NavPageLayout>
  );
};

export const getServerSideProps: GetServerSideProps<
  NavPageLayoutProps
> = async () => {
  return {
    props: {
      ...(await getNavPageLayoutPropsFromConfig()),
    },
  };
};

export default SamplePage;
```

Linking to Navigation Bar

- Add a link to SamplePage.tsx to sampleCommons/config/gen3/navigation.json, or anywhere else that allows linking pages
- To do this add a dict entry to “items” list as shown in right-hand image
- Supports full customization of icon, Name, and tooltip
- SamplePage is attached to the “DEMO” button in right-hand example



Guppy Api Fetch

- Guppy API fetching function part of the @gen3/core library
- More advanced fetching techniques are used in the cohort builder.
- Api endpoints fetching techniques are also supported in the core library

```
50  /**
51   * Similar to graphQLAPI except this
52   * guppyAPIFetch function is a branch of guppy core API defined below and
53   * graphQLAPI is a branch of gen3Services API.
54   */
55  export const guppyAPIFetch = async <T>(
56    query: guppyApiSliceRequest,
57  ): Promise<guppyApiResponse<T>> => {
58    const res = await fetch(`${GEN3_GUPPY_API}/graphql`, {
59      headers: {
60        Accept: 'application/json',
61        'Content-Type': 'application/json',
62        'Access-Control-Allow-Origin': '*',
63      },
64      method: 'POST',
65      body: JSON.stringify(query),
66    });
67
68    if (res.ok) return res.json();
69
70    throw await buildGuppyFetchError(res, query);
71  };
72
```

Fetching Data from Guppy with FF

- Uses a basic useEffect hook to fetch data and use data in visualization
- Guppy expects query format

```
export interface guppyApiSliceRequest {
  readonly query: string;
  readonly variables?: Record<string, unknown>;
}

export interface guppyApiResponse<H = JSONObject> {
  readonly data: H;
  readonly errors: Record<string, string>;
}
```

```
55  const query =
56    `query($filter: JSON){
57      file(filter: $filter first: 10000){
58        subject
59      }
60     }`;
61
62  const variables = {
63    filter: {
64      AND: [{ IN: { project_id: ['synthea-test'] } }],
65    },
66  };
67
    const SamplePage = ({ headerProps, footerProps }: SamplePageProps) => {
      const [items, setItems] = useState<Item[]>([]);
      const [isLoading, setLoading] = useState<boolean>(true);

      useEffect(() => {
        const fetchData = async () => {
          try {
            const result: GuppyAPIFetchResult = await guppyAPIFetch({
              query: query,
              variables: variables,
            });
            setItems(result.data.file);
            setLoading(false);
          } catch (error) {
            console.error('Error fetching data:', error);
          }
        };
85
86      fetchData();
87    }, []);
88
```

Guppy Output Processing

- Data Component used for processing and mapping data into a grid of “reference” entries.
- Essentially a grid of guppy results where each result is a “reference” to a patient

```
interface Item {
  id: string;
  subject: string;
}

interface DataComponentProps {
  data: Item[];
}
const DataComponent = ({ data }: DataComponentProps) => {
  return (
    <Grid>
      {data.map((item) => (
        <Grid.Col key={item.id} span={4} style={{ marginBottom: 5, marginTop: 5 }}>
          <Paper style={{ padding: 'lg', boxShadow: 'xs' }}>
            <div style={{ display: 'flex', alignItems: 'center' }}>
              <Text style={{ margin: 10 }}>  ✅  </Text>
              <div>
                <div>Subject: {item.subject}</div>
              </div>
            </div>
          </Paper>
        </Grid.Col>
      ))}
    </Grid>
  );
};
```

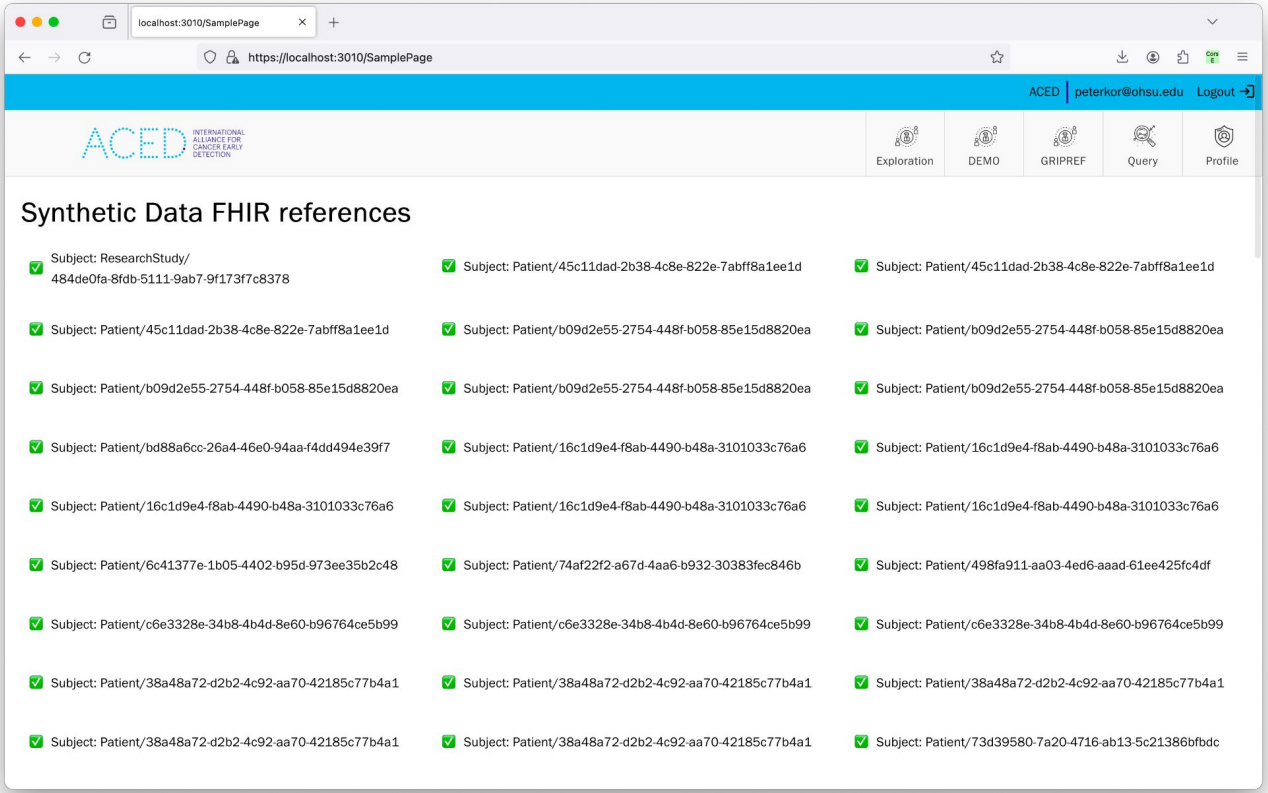
Putting it all together

- Code demonstrates using basic JS hooks, fetching functions, and Mantine components to fetch data from Guppy
- Guppy data is displayed as a grid of entries
- Header and footer props are maintained to keep look and feel consistent with the rest of the site

packages > sampleCommons > src > pages > SamplePage.tsx > DataComponentProps

```
69  const SamplePage = ({ headerProps, footerProps }: SamplePageProps) => {
70    const [items, setItems] = useState<Item[]>([]);
71    const [isLoading, setLoading] = useState<boolean>(true);
72
73    useEffect(() => {
74      const fetchData = async () => {
75        try {
76          const result: GuppyAPIFetchResult = await guppyAPIFetch({
77            query: query,
78            variables: variables,
79          });
80          setItems(result.data.file);
81          setLoading(false);
82        } catch (error) {
83          console.error('Error fetching data:', error);
84        }
85      };
86
87      fetchData();
88    }, []);
89
90    return (
91      <NavPageLayout headerProps={headerProps} footerProps={footerProps}>
92        {isLoading ? (
93          <p>Loading data...</p>
94        ) : (
95          <Group style={{ margin: 20 }}>
96          <Title> Synthetic Data FHIR references </Title>
97          <DataComponent data={items} />
98          </Group>
99        )}
100     </NavPageLayout>
101   );
102 };
```

Putting it all together



Acknowledgements

- Speakers
  - Craig Barnes - Center for Translational Data Science, University of Chicago
  - Matthew Peterkort - Oregon Health & Science University
- Gen3 Forum Steering Committee
  - Robert Grossman - Center for Translational Data Science, University of Chicago
  - Steven Manos - Australian BioCommons
  - Claire Rye - New Zealand eScience Infrastructure
  - Plamen Martinov - Open Commons Consortium
  - Michael Fitzsimons - Center for Translational Data Science, University of Chicago