Introduction to the Gen3 Frontend Framework

Gen3 Community Forum
1 November 2023
The Agenda

- Introduction
- Presentation
- Q&A
- Topics for future Gen3 Community Events
Gen3 Frontend Framework
Windmill strengths and weaknesses

**Strengths**
- Highly configurable
- Can support a wide variety of different commons
- Tightly coupled with Gen3 services

**Weaknesses**
- Monolithic architecture and deployment
- Difficult to extend and configure
- Project/Business logic embedded in the code
- Inadequate support for data movement between pages
- Complex development environment
Introducing the Gen3 Frontend Framework

- Replacement for Gen3 data-portal
- Address frontend limitations
- Improved development experience and user experience
- Updated technology stack
- Enhancing:
  - Custom content and applications development
  - Performance
  - Deployment and maintenance
Overview

- Provide functions to access Gen3 services
- Introduce concept of Context
- Analysis Tool Framework
- Standardize Components
- Upgraded tech. stack
- Simpler custom custom pages

Gen3 Frontend Framework

- Gen3 Application
- Analysis Tools
- Gen3 Components
- Content management
- Analysis Tool Framework
- Context
- Data Access Layer
- Gen3 Services
- External Services
- Data
The concept of the user’s context is a powerful feature of the Gen3 Frontend Framework.

Context provides pages and tools with information about:
- Active cohort
- Selected studies
- Active analysis tool
- AuthN/AuthZ

Context is available to all pages and analysis tools.
Framework Technology Stack

- NextJS for full-stack web applications
- React as the UI framework
- Redux-toolkit and SWR
- Written in Typescript (with JavaScript compatibility)
- Theming and styling with Tailwind CSS
- Mantine.dev for component library
Framework Modules

- Framework is divided into 2 npm packages and a template
  - Core module `@gen3/framework`
  - Frontend module `@gen3/frontend`
  - Gen3 data commons web application template

- Goals:
  - Reduce code complexity
  - Abstract UI interactions
  - Support customization
  - Simplify deployment
Core Module: @gen3/framework

- Provides a data interface to Gen3 services independent of components
- Manages user's context: cohort, selected studies, files, and analysis tools
- Allows future features: user's command history
- Access to Gen3 services via "hooks"
- Abstracted API calls for simplified data retrieval
- Designed to isolate frontend components/web service from Gen3 API changes
Analysis Tool Framework (ATF)

- Supports the development of custom analysis tools
- Connect to:
  - Gen3 services
  - 3rd-party APIs
  - Other data sources
- Uses context to filter tools that can be applied to current data selection
- Frontend:
  - Support to register apps as plugins
  - Analysis tool page
Frontend Module: @gen3/frontend

- Standard Gen3 pages:
  - Discovery
  - Exploration
  - Data Dictionary
  - ...
- Theming and configuration functions
- Gen3 components build using mantine components
- Uses Gen3 services/data via @gen3/framework hooks
Commons Specific Code

- Commons Specific code is implemented as a functions that:
  - Retrieve
  - Transform
  - Render
- Gen3 Feature accept a function *hook* to override default behavior
- Default hooks are implemented to provide standard behavior
- Tables/charts can be customized by registering custom Cell and Chart components
- Planned provide templates and Developer documentation
Discovery Page Data Hook Example

Data-portal:
- Discovery page reads data from the Gen3 Metadata Service
- Pulls data from MDS and/or Aggregate MDS
- The complexity/variety of metadata makes configuration complex
- All metadata is loaded into the client to support search

Gen3 Frontend Framework:
- Developers pass a function (e.x. hook) to discovery page to load data/transform data
- Enables data transformation logic to be commons specific
Discovery Page Data Hook Example

Discovery

- Advanced Search
- Search bar
- Key/Value Search

Table

- Summary/Aggregations
- Pagination

Data for aggregations

Data Hook

Gen3 Metadata Service
export const LinkedStudyCell = {
  value: cellValue,
};

const CellRenderFunctionProps<
  boolean
> = Element => {
  const value: boolean = cellValue as boolean;
  return value ? (
    <Badge
      variant="outline"
      leftSection={<CheckCircleOutlined />}
      color="green"
    >
      Linked
    </Badge>
  ) : (  
    <Badge leftSection={<MinusCircleOutlined />} color="primary">
      Not Linked
    </Badge>
  );
};

const CommonsCellRenderers: {boolean: {...}, string: {...}} = {
  string: {
    default: WrappedStringCell,
  },
  boolean: {
    LinkedStudyCell,
  },
};

export const registerDiscoveryCustomCellRenderers = () : void => {
  DiscoveryCellRendererFactory.registerCellRendererCatalog(CommonsCellRenderers);
};
New Features

- Common location for storing selection: My Data Library
- Analysis tool plugins
- Static content
- Custom pages
- Custom content via markdown
- Context
- Updated site navigation and new UI design
- 508 compliant
New Design/UX
New Design/UX
NextJS Application Framework

- NextJS application server for Gen3 commons
- Frontend module implements standard Gen3 Features
- Supports access-controlled pages and page sections
- Analysis tool registration with the ATF
- Enables complex site navigation and middleware support
• Built into @gen3/frontend
• Current support: file-based configurations
• Future: database-stored configurations/introspection
• Separate configuration file for each page/application
• Backward compatibility with data-portal but with new capabilities
• Planned: Web-based configuration and validation in the pipeline
• Planned: hybrid data-portal/Gen3FF
Data-commons Web Application Framework

- Data common is a git repository:
  - configuration
  - icons
  - custom code and pages
  - content
- Addition pages can be added as NextJS pages, slugs, or Markdown
- Registered Analysis Tools
Custom Content and Analysis Tools

- Enable simple way to add commons-specific content
- Addition of custom pages via React JSX or TSX
- Markdown support
- More complex pages:
  - Custom components
  - Analysis tool built using React and Gen3 modules
Color Theming

Color tokens based on USWDS:
- Primary
- Secondary
- Accent
- ...

Command line tool to create:
- 10 shades per color token
- 508 compliant contrasting color

Color tokens are used exclusively in Gen3 front end components/pages
Development Environment

- Local development using Gen3 Helm charts for Gen3 Services
- Commons site runs as a NextJS development server
- Support development features:
  - hot reloading,
  - debugging w/breakpoints
- Plan to support debug commons
- Provide sample helm configurations and tools to run local development environment
## Status and Roadmap

### Current:
- Discovery Page
- Query Page
- Login
- Profile
- Basic theming
- Configuration
Conclusion

- Gen3 frontend framework overcomes current portal limitations
- Enriched user experience
- Easier development and deployment
- Many options for customization
- Facilitates complex data analysis workflows
Acknowledgements

● Speakers
  ○ Robert Grossman - Center for Translational Data Science, University of Chicago
  ○ Craig Barnes - Center for Translational Data Science, University of Chicago

● 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
Questions