Kendo UI Builder 3.0

Modern Web UI for OpenEdge Business Apps

Shelley Chase
Software Fellow
October 2018
Agenda

- Modern Business Applications
- Steps to Modernize
- Kendo UI Builder
- Demo
Modern Business Applications
Modern Business Applications are dynamic, interactive systems that perform business critical tasks and increase productivity.
Key Features of a Modern Business Application

- **Seamless omnichannel user experience**
  - User access from any device
  - Compelling, inviting user experience, multi-lingual, multi-culture
  - Rapid technology evolution – staying ahead means constant updates and deployment
  - Multi-tasking support

- **Cloud deployment (public or private)**
  - Secure environment with authentication and authorization
  - High availability and scalability
  - Multi-tenant, SaaS licensing model
  - Evolving regulations, auditing, compliance and accessibility

- **Focus on your core competency**
  - Key differentiators for your market
  - Leverage existing ABL business logic
Reference Architecture for Cloud Deployment
Architecture

- Load-balanced Application Server
- Database
- REST API
- Message Queue
Data Management / Integration

- Replication DB (failover and/or reporting)
- Auditing
- DBA
Cloud / SaaS

- Firewalls
- DMZ
- Gateway/Router
- Load Balancer
Security and Compliance

- Authorization layer: SSO, Direct login
- External Auth Provider
- OE Auth Gateway
UI / UX
Steps to Modernize an Existing OpenEdge Business App
Recommendations are to:

• Start continuous modernization and transform the organization into an entity with an agile culture

• Convert monolithic applications to lightweight new architectures suitable for the cloud

• Create APIs to enable ecosystems and give the organization the ability to create new applications and functionality rapidly…"
Modernize Existing Applications

- Build App using a prescriptive application framework – OEAA / CCS
  
  **Client**
  - Kendo UI Components
  - Responsive UI
  - JSDO Data Source on Client

  **Server**
  - Business Entities = ABL for all business logic and data management
  - Service interface for Progress Data Objects

- Automate the modernization as much as possible

**Kendo UI Builder**
- Pre-defined UI templates using world-class Kendo UI components
- Data-driven Views (based on JSDO catalog)
- Free-form Views
Kendo UI Builder
Kendo UI Builder

- Visual designer for building business web apps
  - Modules, Views, Components
    - Pre-defined data-driven Views provide automated data binding
    - Build custom views using drag-and-drop designer
    - Extend by building custom templates
  - Data Providers, Data Sources
    - Progress Data Object Services, REST Services, OData Services
  - Meta-data based UI definition
    - UI templates separate framework details from UI definition
    - Multi-language support
  - Built-in authentication and session management
    - Authentication for data services
    - Role-based authorization
Kendo UI Builder Web Application

- **Modules**
  - Application
  - Custom Module
    - Data Views
    - Custom Views

- **Data Providers**
  - Data Sources
View Templates

- Data View Templates
  - Data Grid
  - Data Grid and Form
  - Data Grid Separate Form
  - Stacked Data Grids
  - Hierarchical Data Grids

- Custom View Templates
  - Blank
  - Your Custom Views

Blank View

- **Custom Layout** with Responsive Behavior
- **Drag-and-drop Components**
  - Layout
  - Data Management
  - Editors
  - Charts
  - Scheduling
  - Media
  - Navigation
  - Custom
    - Custom HTML
    - Your Custom Components
Data Providers

- **Data Provider**: Data Objects Service (Catalog)
  - Web App Service URI
  - PAS for OE or Classic AppServer
  - Contains one or more Resources
  - Resource = DataSet or Table

- **Data Source**: Table
  - Schema: Fields
    - Related tables
    - Label
    - Editor Type – defaulted from Semantic Type
    - ...
  - Operations: CRUD + Submit + Invoke + Count
Authentication

- Supported models
  - Anonymous
  - Basic
  - FORM
  - SSO (future)

- JSDOSession manages
  - Authentication token life cycle, automated token renewal
  - All http/s communication and error handling from multiple browsers
Client-side Dataset
  • Hierarchical tables
  • Foreign key relationships
  • Single records and multi-record transactional updates

Purposed for Progress Data Object services
  • Based on required catalog generated by Progress Developer Studio

Manages all communication to services
  • Supports authentication models
  • Manages session lifecycle
Demo
Design and Preview Web App
Kendo UIB Demo – New Web Application
Web App Design
Kendo UI Builder Web App Design
Runtime Architecture

- Model
- View
  - Kendo UI
  - HTML
  - CSS
- Controller
  - JavaScript
  - JSDO
- Kendo UI by Progress
- Data Object Service Catalog
- Progress OpenEdge
Built-in Extension Points for Customization

- App logo
- User Roles
- Properties in Designer
- Custom HTML Sections
- Event JavaScript functions
- Row Templates
- CSS and Themes
- Translations
- Modify existing Editors
- Modify existing View templates
- Build custom Editors
- Build custom View templates
- Custom App Framework (very ambitious)

Technical Knowledge Required

- Familiar with HTML, JS and CSS
- Complex JS, Angular/JS, EJS-templates

Point and Click
Components: Kendo UI and Custom

https://www.telerik.com/kendo-angular-ui/components/

https://www.progress.com/blogs/introduction-to-angular-custom-components-for-kendo-ui-builder
How Kendo UI Builder Compares to Other Options

- **Future-proof**: Meta-data and template driven
  - Other Web UI Design tools lock you into one UI technology
  - Templates can be customized and extended for your needs

- **Rapid UI development based on data schema**
  - Data-driven views with CRUD features accelerates development of maintenance screens
  - Award winning Kendo UI controls to present the data
  - Built-in extensibility options for customization

- **Purposed for OpenEdge business logic and data**
  - JSDO offers client-side dataset features like change tracking and relationships that map to ProDatasets
  - Invoke, Submit and Count operations provide much more flexibility than a generic REST service