# Evaluate and Implement Dynamic Dashboard Layout Configuration with Forge Support for Charts

## Objective

Enable customizable dashboards in OutSystems with support for drag-and-drop tile arrangement, user-specific layout persistence, and integration of chart components. The layout must be dynamic, persistable, and reloadable with real-time configuration options.

## Step-by-Step Implementation Guide (Beginner-Friendly)

### Step 1: Install Required Forge Components

- Open Service Studio.
- Go to the Forge tab.
- Search and install these components:

  Drag and Drop

  Layout Grid

  OutSystems Charts or HighCharts Plugin

### Step 2: Create the Entity to Store Layout

- Create an entity named DashboardLayout with:

  Id, UserId, LayoutJson, Version, UpdatedAt

### Step 3: Define the Layout Model (JSON)

- Use structured JSON with tileId, title, position, visibility, chart type, and chart config.

### Step 4: Build the Dashboard Settings UI

- Use Layout Grid and Containers

- Add Chart and Text Widgets

- Enable Drag and Drop

- Add Save Button to persist layout

Step 5: Create Persistence Logic

- Create GetDashboardLayoutByUserId and SaveDashboardLayout actions

Step 6: Render the Dashboard Dynamically

- Use OnReady to load layout

- Iterate over tiles to show charts dynamically

AFS Example Tiles

- Loan Origination Volume (Bar)

- Delinquency Rate (Line)

- Customer Satisfaction (Gauge)

- Active Cases by Analyst (Bar)

- Task Completion Rate (Donut)

Deliverables Checklist

- UI, JSON Model, APIs, Renderer, Chart Integration, Documentation

Forge Plugin Evaluation

- Drag and Drop, Layout Grid, OutSystems Charts, HighCharts Plugin

Limitations

- No resizing, performance impact, chart format variations

Optional Enhancements

- Resizing, versioning, role-based dashboards, JSON export/import

Author: Carlos Baya | Date: 2025-06-27