Key Dates Project

Introduction

Key Dates Project is a collection of four applications and associated documentation.

The application is designed to work on Domino 9.0.1 with XPages Extension Library v12 or higher. For all except the basic XPages application, you will also need OpenNTF Domino API v2.0.0 or higher installed on the server. For the CrossWorlds application you will need the CrossWorlds feature installed on a Websphere Liberty Server.

For simplest configuration, install the basic XPages application at **xpagessamples/KeyDates.nsf**. The other releases will point to that location for their data. Amend the ACL as appropriate to allow Editor access to data. Mickey Mouse/Disney will need at least Editor access to the basic XPages application and the server it is on.

Applications

Basic XPages Application

This application, best installed at **xpagessamples/KeyDates.nsf**, holds the data for all applications. This uses basic drag and drop, simple SSJS, but some minor optimisations. All XPages developers should be able to understand this application.

Advanced XPages Application

This application demonstrates a few techniques:

- Separating XPages application from database, the kind of approach required for Bluemix development but equally valid on premises.
- Using an extension to Tim Tripcony's SmartDocumentModel approach (see part one, part two, part three and part four), access to the document is via a wrapper.
- The document wrapper is handled as a managed bean and accessed via XPages and Java accordingly.
- A "page controller" is used for the document-level page. This is just a Java class instantiated via a dataObject (from Extension Library).
- Neither page controller or document wrapper are intended as best practice approaches. They are intended to demystify controllers or beans. Whether they are created via the faces-config, via the *create* method of a dataObject or using the *new* keyword in Java code, they are all just an instance of a Java class.

It is not designed as a best practice, fully Java application. It is designed to help lead intermediate XPages developers into the world of Java and MVC.

The application can be installed anywhere on the same Domino server as the basic XPages application.

Vaadin OSGi Application

Although in terms of learning, this is the third application, in actuality it was the last.

Developing OSGi applications has its challenges and quirks. Developing a standard Vaadin web application on CrossWorlds is actually much easier.

I suspect XPages classes (dominoDocument, dominoView etc) would be available for an OSGi application running on Domino. However, in this case the document-level wrapper used for the advanced XPages application has been supplemented with wrappers for the views.

The open source Vaadin framework is used for the UI. This is not based on any value proposition of the merits of Vaadin vs XPages. It is intended to demonstrate alternate approaches, designed to encourage greater flexibility for developers and show how to access Domino data from a different Java framework. It is this aspect which requires some “re-inventing” of wrapper components that XPages provides out-of-the-box.

The application uses OsgiWorlds to wrap servlet requests and allow OpenNTF Domino API to be used to interact with the Domino server.

The update site can be uploaded to an Update Site database on the same server as the basic XPages application. Then it will be accessible at http://your.server.com/keyDates/.

Vaadin CrossWorlds Application

This application is a standard war (**W**eb **AR**chive) file that can run on a Websphere Liberty Profile server that is configured for CrossWorlds. It depends on the j2eenabler feature which pulls together all the various jar files that constitute CrossWorlds. It expects the Websphere Liberty Profile server to be using the same Domino server that the basic XPages application is on.

Beyond that requirement the application is virtually identical to the OSGi application.