DealWip Application

Prepared by Patrick Lee Dooley

CEO iTKunst corporation

on Behalf of

Iron Forge

May, 1, 2018

pdooley@itkunst.com

Dealwip has been developed with simplicity and scalability in mind. It is based on leading edge technoly such as Angular 5, Nodejs, Express, Hadoop and Material Design.

Over the course of development, an SPA framework from iTKunst corporation called kunstWerkNg (kWng) has evolved in order to support the unique requirements of DealWip.

kWng – a component based framework

KunstWerk is a third generation framework that originally appeared 7 years ago when there was very few frameworks available. It was the first truly “component” based architecture very similar in nature to what later became Reactjs.

kWng – completely configurable

Key and central to kunstWerk is the ability to define nearly all aspects of the application with declarations. KwNg at this moment requires 4 different json files. These include:

1. bootstrap.json. This file is used for application intialization. This file is used for defining services, authentication keys, bootstrap file apis and models, and most importantly debugging settings.
2. Apis.json. This file stores all the apis available to the application. It specifies settings for both debug and release. It specifies which service defined in the bootstrap.json file it will be using. It specifies which

token defined in the bootstrap file it will be using. Finally it specifies the signature, the port, the protocol and which actions are allowed.

3. Models.json This file stores the model definitions for all of the tables or documents that are used by the application. It is used by the application to verify data received, for transforming data in and out of the application and for controlling variable names.
4. View.json file. This file is the heart of the application. It defines the individual views or page creation. The html and styles are loaded dynamically on startup. Through Ids in the view.json file, various aspects of the view can be controlled by data or intialization state.

Each of these files are loaded at runtime. Modifications do not require rebuilding or redoploying of the application.

kWng – Infinitely Scalable

Kwng has been designed specifically to help developers build complex SPAs.

Its underlying architecture is publish/subscribe mechanism. All components are totally loosely coupled. Connectivies are generated through publishing and subscribing messages with data.

kWng – A Complex State Maintenance System

Kwng is totally state driven. Over 11 years of modelling the SPA problem space has allowed iTKunst to identify and created the objects and individual orthogonal states that exist in a typical SPA. These objects and states have been implemented via components and services in the Angualar world.