Hi I am giving brief understanding of “How @Config annotations are used in the framework with examples” :

* Nimbus FrameWork can be used to provide/customise different applications based on the client needs.

To build the application, we use framework configurations.

What is config

for any application first step is to design business entities(domain model) as per the requirements and standards,

we can define the configuration for the view, workflow and the rules. The view definition configs, the mapping to the domain model, the workflow(if any) and the corresponding view and core domain rules can be written.

ie., on a high level types of configurations are :

Configuration

Business Entity configuration

View configuration

Business Rule configuration

Workflow Configuration

Business Entity Configuration :

it is simple Entity(POJO) contains the below details (specifying with annotations).

@Domain – specifies the business entity and its attributeds like value and what are listeners to support different functionalities (ex : persistence, Update, etc.,).

@Repo – specifies that where to/from busine model’s data can be stored/fetched(ex : DB, cache).

View Configuration :

basically view is a java class where the flow starts from view root entity.

It contains Pages and its data interms of model.

i.e., View Root definition holds corresponding Pages.

@Page - specifies defaultPage attribute to tell it is default page are not. And layout is an attribute tells what kind of layout we are showing.

anda ViewPage is a static class and its having their own properties to view to user.

Propertie are :

@Model - it is again static nested classes which holds the data from different streams (ex: from another service or form or DB table etc.,).

@Config – represents where our services are located (through url attribute) and

@Configs can holds one or multiple services using @Configs . Which take array of config data.

For example :

@Configs({

 @Config(url="/vpOwners/vtOwners/vsSearchOwnerCriteria/vfSearchOwnerCriteria/\_update"),

@Config(url="/vpOwners/vtOwners/vsOwners/owners.m/\_process?fn=\_set&amp;url=/p/owner/\_search?fn=example")

        })

along with above details a Page can maintain Grid data also.

A Grid will hold list of entites and also can render the entities in page nation format.

For example :

@Config(url="/p/ownerlandingview/vpOwners/vtOwners/vsOwners/owners.m/\_process?fn=\_set&amp;url=/p/owner/\_search?fn=example")

@Grid(onLoad=true, isTransient = true, pageSize = "7")

private List<OwnerLineItem> owners;

all our @config data can be controlled from drl.