Remaining Modularity Work for Rice

Original Rice 2.0 Modularity Design and Analysis

To Dos

1. Finish work on Rice IMPL
    a. KCB and KEN stuff should be easy to work with by moved into a KEN IMPL Module
    b. KNS package should be move up to KNS module
    c. Module service base should be move up from KRAD to the KRAD APP or Framework
        i. allows for core service and location out of rice impl
    d. Analyze KEW to see how it needs to be refactored to it's own module
    e. Analyze KIM as well
    f. Analyze the KRAD pieces and decide which needs to go where by package and classes
2. Run latex tool against our code
3. Get rid of groovy stuff
4. Look at Web Content modularity remaining work
    a. Focus on KRAD and what makes most sense for it being our Web Content delivery method going forward
5. Figure out how to fix the EBO issue and replace it

Rice impl work Discussion Notes KCW2012

- Rice impl needs modularized
    - There are some things that just don't go into an implementation module,
    - Some need to go up to our framework as well. That will be the challenge since it's difficult to do.
    - The route context in the KEW section will be a bigger issue
    - Most of the KEW stuff in there was pretty jumbled, how do we do this in a way that's not majorly impactful
    - We'll probably have to create new versions and keep but deprecate the old methods.
        - This should affect version compatibility for client-server, but couldn't do this in a vacuum, would have to talk to the roadmap committees.
        - Should do this one step at a time instead of a larger roadmap version
    - Definitely want to get all the KRAD stuff pulled out as a first step.
    - Would want to keep the package names stay the same, we
    - Could run some packaging changes by the TRC
    - There's a bunch of stuff in KIM as well, the UI mostly, that shouldn't break our API (maintenance, trans, business objects)
    - KNS doesn't have a lot of stuff though
    - KIM and KEN do, though KIM should be cleaned up our business objects don't link up with our transfer objects
- Run latex tool against our code
- Get rid of groovy stuff
- Another idea would be to push this out to the 3.0 release, KRAD and JPA and such
- Need to do an analysis of the current module from the KRAD/KNS viewpoint and then another group look at the non-KRAD/KNS pieces
    - Do we still need and API and an IMPL for krad modules?
        - In anything that we expect the public to extend, we should have an API.
        - We need to look at what we can do to making extendable items safe and well documented
        - Would be nice to forgo the IMPL side if we can, but it may not always be the case
    - Want to see KRAD be well packaged and modularized
- Need to know if we can overlay multiple war packages in one jar file
- KFS uses EBOs extensively so we need to ensure we don't mess things up while/when we do this with our new design
    - If this is an instance of an externalized business object...