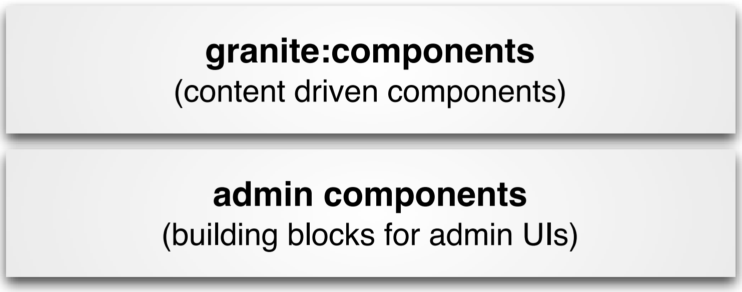
WARNING - this is a draft document!

Introduction to the Granite UI

The Granite UI merges the sling RESTful architecture with Coral’s UI elements to implement component libraries intended for building content-centric web applications. The libraries may be used within any Granite-based platform (e.g. CQ).

foundation components



granite:components
(content driven components)

admin components
(building blocks for admin UIs)

WARNING - this is a draft document!

Granite UI foundation components:

The Granite UI foundation components, as name implies, is a library of base components other libraries may use or extend from. All components follow the granite:component architecture, thereby directly benefiting from all features provided by the Granite platform such as drag&drop authoring or automatic UI/unit testing.

The library contains a Granite UI component for each Coral Element. A component is content driven, with its configuration residing in the repository. This makes it possible to compose a Granite UI application without writing HTML markup by hand.

purpose:
- component model for HTML Elements
- component composition
- automatic unit and functionality testing

implementation:
- use granite:component architecture (e.g. sling resolution, inheritance)
- repository based composition and configuration
- leveraging testing facilities provided by the Granite platform
- JSP templating

NEEDS some examples here

WARNING - this is a draft document!

Granite admin UI components:

The Granite admin UI components build upon the foundation to provide generic building blocks that any admin application, adhering to Granite UI's UX, should implement. Each component is designed to be easily customizable and/or overridable, facilitating both the rapid application development and customizability by a customer.
The ultimate goal is to have all admin applications used within the Granite to look and behave in one consistent way.

purpose:

- unified L&F for admin applications
- RAD for admin applications

implementation:

- pre-made components using the foundation components
- component customizability

NEEDS some examples here