Development-WO Component-Binding Synchronization

Overview

When you create a subclass of WOComponent, the default configuration is to have all of your bindings automatically synchronized. Binding synchronization is the process of getting and setting parent and chldren .wod file bindings during the request-response loop. While binding synchronization is very convenient, the one downside is that your binding get and set methods can be called multiple times (as many as six) during the request-response loop. As a result, you should be careful about potential performance problems inside of get and set methods. If you would rather avoid the potential performance issues, it is possible to disable automatic binding synchronization in your component by simply overriding the synchronizesVariablesWithBindings method to return false as in the below example:

```
public boolean synchronizesVariablesWithBindings() {
  return false;
}
```

If synchronizesVariablesWithBindings is false, you will use the valueForBinding(bindingName) and setValueForBinding(value, bindingName) methods to get and set the binding value manually.

WOComponentContent

Because of the way WOComponentContent works, binding is often "backwards" from your intuition (often it ends up that your wrapper pushes its title value into your page instead of the other way around). One way to make your template wrapper component behave the way you expect is to turn off automatic binding synchronization:

```
public boolean synchronizesVariablesWithBindings() {
  return false;
}

public String pageTitle() {
  return (String)valueForBinding("pageTitle");
}
```

Kieran Kelleher

Have a look at WOComponent API valueForBinding and setValueForBinding methods. In your subcomponent, you can define the API (jigsaw icon in WOBuilder. Use lazy initialization to pull bindings on demand. Subcomponents in general pull from parent and push into parent through bindings. Here is an example of a simple manual pull and push of a 'customer' binding:

```
public CTCustomer customer() {
  if (customer == null) {
    customer = (CTCustomer)valueForBinding("customer");
  }
  return customer;
}

public void setCustomer(CTCustomer newCustomer) {
  customer = newCustomer;
  setValueForBinding(customer,"customer");
}
```

Also look at synchronizeVariablesWithBindings to turn on manual binding synchronization (recommended to synchronize on demand rather than 6 times or so through the R-R loop)

wojingo

If you use lazy initialization in your binding accessor methods as described above, you should also make the component stateless. If you do not make the component stateless you will find that your component may display incorrect values when used in a WORepetition. This situation occurs when the WORepetition's list is changed via a component action and null is returned. The existing nested component will be reused and the values in the original list will be displayed.

Make the component stateless and override its reset method, setting all instance variables to null. This will make the component behave as expected.