Development-WOForm Best Practices

Attaching EO's to a Form

Pierre Bernard

I do create the EO and insert it into an editing context dedicated to the task of creating and eventually saving that EO. That editing context is peer to the default editing context. Actually all my edits happen in dedicated peer contexts. The session default is used for reads only. This eliminates the risk of having a pending change hang around.

Doing so I get the default values set in awakeFromInsertion() in my initail form.

I however do not bind the EO directly to the form. I transform the EO into a dictionary of strings. Using key-value coding I first extract a set of values I want in the form. Then again using key-value coding as well as formatters I derive a string representation for each value. The form is bound to a *clone* that dictionary of values.

After the form is submitted I compare the modified clone to the original dictionary. Only for the changed values I do a reverse transformation from String to object or EO value. This is done by first parsing the string using a formatter and the possibly calling a fetch specification.

This has the following advantages:

- I can map to-one relationships to a plain form
- I only set actually modifed values on the EO
- I do not lose precision on a value which internally had a higher precision than what the form shows
- I handle parsing exceptions in code. For one this allows keeping the unparsable value in the form

Ondra Cada

Depends on the application.

But most often, I am eating my cake and still having it by creating the EO first, but saving it into EC only when properly set up. That way, it can be wired up directly, and still, if the user abandons it, it just vanishes when the GC decides so 🙂

Geoff Hopson

I find it much cleaner to have a formValues NSMutableDictionary and have the key names the same as my EO attribute names. In my WOD file I can bind text field values etc. to 'formValues.myAttribute. It also allows me to preset some defaults, hide any hidden fields I may need, stuff like that. I can then easily blat the dictionary into an inserted EO once I am pretty happy with the data that has been collected.

Mike Schrag

Do most of you create the EO up-front and attach the form fields directly to the attributes of the EO, or do you clone the fields into the WOComponent and attach to those, then move them into a new EO in the "save" action? The problem with the first route is that if someone navigates away, you've got an empty EO inserted into the EC and the user doesn't realize that it's still sitting in their session, but it's obviously nicer because you can just wire up directly to the EO attributes and you don't have to keep the Component and the EO fields in-sync

Jean-François Veillette

I use a new EO, in a 'peer' editingContext used for insertion/edition. That way, If changes are made to the eo, but not saved, and user change page, or even use the browser back button, I do not care, he doesn't screw-up the 'main' editingContext.

I forgot to mention another option that I used in some special situation :
Instead of binding to an EO, bind to a NSMutableDictionary, then when you're ready to save, you can create your eo and apply the dictionary values to it (there is a method for that).