ImprovedValidationProposal

NOTE: This information applies to an older version of Tapestry. For current information see http://tapestry.apache.org/input-validation.html.

Problem Description

Tapestry's validation subsystem is quite powerful, but has evolved over time in fits and starts. The Tapestry 3.0 implementation is powerful and flexible, but could be *simpler* and more *consistent*.

- Creating validation delegate objects for Form components is somewhat awkward
- Creating validator objects for ValidField components is much too verbose
- There is no easy way to *cancel* a Form (i.e., disable client- and server-side validation)
- Is it necessary to have both TextField and ValidField?
- It would be nice to integrate validation for other (perhaps all) form element components, especially TextArea, DatePicker, ProperySelection

Proposed Solutions

Validation Delegates

Forms shall, by default, have a validation delegate. The `org.apache.tapestry.validation-delegate-class` property will define the validation delegate class to instantiate, with a default of the `org.apache.tapestry.valid.ValidationDelegate` implementation. The property can be overriden by a page, a namespace, or globally.

Streamlined Validator Specification

The `validator` parameter of the ValidField component requires an instance of `IValidator`. This shall be changed to also allow a string to be specified. The string shall be an encoding of a type of validator and configured properties for that validator type, for example, `string,required,minLength=5` or `number,int,maxValue=20`. A HiveMind configuration point will allow new validator types to be defined (details forthcoming).

To be determined: perhaps an additional configuration property to control whether client side scripting is enabled or disabled by default.

Note: through the magic of OGNL, something like this can be done in Tapestry 3.0, i.e.:

```
<binding name="validator">
  bean.validatorFactory.get("string,required,minLength=5")
</binding>
```

The trick 💡 is to write the validatorFactory bean.

Form Cancellation

It would be nice to have the ability to cancel a form, such that client-side validation is skipped and the form submitted (in such a way that the Form component knows it should not rewind).

I would propose that a special query parameter, `cancel` be used for this purpose. A non-null value will be recognized by the Form component to indicate that the form has been canceled.

Possibly, the Form class (and IForm interface) should have a `canceled` property. Another possiblility is to add a `canceledListener` parameter.

A Cancel component may also be added, though it's not obvious that it is necessary, `<input type="submit" name="cancel" value="true">` would be sufficient.

Part of this placys into the need to re-organize how client-side scripting related to the Form is managed. There needs to be a differentiation of which Form submit handlers should execute under which scenarious (normal submit, cancel submit, other?). I have seen cases where a Form should be submitted so that it can update in a refreshed mode (for example, when there is a dependency between the selection of one drop down list and the options in another) that causes some headaches on the client side.

Unified Validation

Make `validator` parameter optional on all form element components and remove ValidField component entirely. Allow displayName on all form element components.

How would a validaiton delegate decorate a non-text field, such as a check box or drop down list? How can the validation delegate determine the "flavor" of component it is decorating (i.e., is it a traditional text field, a drop down list, a check box, a text area, etc.). This may involve some significant API changes. Possibly, IFormComponent would need an additional method for this purpose.