

ServoyCommons - Feature #103

Show multiple errors at the time

07/25/2010 08:08 AM - Patrick Talbot

Status:	Feedback	Start date:	07/25/2010
Priority:	Normal	Due date:	
Assignee:	Patrick Talbot	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Browser (if web client):			

Description

The current implementation of errors is using a single String, which I think is insufficient. Although you can concatenate the error messages into one, it is not that practical to do so.

I propose a different approach, using an array of errors, which could be one error, either a field name, or a message, depending on what the users would prefer.

To better show what I'm proposing, please find a patch for the orderDialog form of the servoyCommonsExample solution. I am basically using this array (that I only use here in this example) to show all the fields in error in one go, coloring the labels in red using the solutionModel.

History

#1 - 07/27/2010 08:12 PM - Sean Devlin

- Status changed from New to Feedback

- Assignee changed from Sean Devlin to Patrick Talbot

I thought about this a bit, but I'm not sure where to draw the line between simplicity and flexibility. I like the idea of a single error message because it can directly bind to a data label, be shown in a dialog. I'd like to add statusMessage and warningMessage

```
errorMessage = 'xyz'; // can be instantly shown in an implementation form.
```

But I thought about having error objects (not just strings - they could hold info), status objects and warning objects (or maybe just messages, with different types/severities). These would provide info that could be shown in different ways on implementation forms.

i.e. in a base form validate method, fill an errors array

```
errors = [];  
errors.push({errorCode:MISSING_REQUIRED_FIELD, dataProviderID:'last_name'});  
errors.push({errorCode:PASSWORD_MISMATCH, dataProviderID:'confirmPassword'});
```

also statuses and warnings

```
statuses.push({statusCode:PASSWORD_CHANGED});  
warnings.push(warningCode:ORDER_NOT_CONFIRMED);
```

then in implementation form....

```
for(i in formState.errors){  
var e = formState.errors[i];  
if(e.errorCode == MISSING_REQUIRED_FIELD){  
for(j = 0; j < elements.length; j++){  
if(e.dataProviderID == elements[j].getDataProviderID()){  
e.setBorder('Line, 1, #FF0000');  
}  
}  
}  
}  
errorMessage = 'Missing required fields: ' + formState.errors.join(',');
```

#2 - 07/27/2010 08:17 PM - Sean Devlin

I thought about this a bit, but I'm not sure where to draw the line between simplicity and flexibility. I like the idea of a single error message because it can directly bind to a data label, be shown in a dialog. I'd like to add statusMessage and warningMessage

```
errorMessage = 'xyz'; // can be instantly shown in an implementation form.
```

But I thought about having error objects (not just strings - they could hold info), status objects and warning objects (or maybe just messages, with different types/severities). These would provide info that could be shown in different ways on implementation forms.

i.e. in a base form validate method, fill an errors array

```
errors = [];  
errors.push({errorCode:MISSING_REQUIRED_FIELD, dataProviderID:'last_name'});  
errors.push({errorCode:PASSWORD_MISMATCH, dataProviderID:'confirmPassword'});
```

also statuses and warnings

```
statuses.push({statusCode:PASSWORD_CHANGED});  
warnings.push(warningCode:ORDER_NOT_CONFIRMED);
```

then in implementation form....

```
for(i in formState.errors){  
  var e = formState.errors[i];  
  if(e.errorCode == MISSING_REQUIRED_FIELD){  
    for(j = 0; j < elements.length; j++){  
      if(e.dataProviderID == elements[j].getDataProviderID()){  
        e.setBorder('Line, 1, #FF0000');  
      }  
    }  
  }  
}  
errorMessage = 'Missing required fields: ' + formState.errors.join(',');
```

Files

servoyCommonsExample.patch	2.54 KB	07/25/2010	Patrick Talbot
----------------------------	---------	------------	----------------