

Getting Data to cf_sebForm

Posted At : November 21, 2007 8:00 AM | Posted By : Steve

Related Categories: sebtags

Before I can effectively cover anything else in cf_sebForm, I have to cover how to get data to the fields. Fortunately, this by itself will make cf_sebForm somewhat useful.

Setting Values Directly in Fields

The following is an example field:

```
<cf_sebField name="fName" label="First Name">
```

You might expect that you could use a "value" attribute to set the value for this field. In sebForm, however, no such attribute exists. The reason for this is the inherent ambiguity in such a field. The ambiguity doesn't exist in an HTML form, but would here (for reasons that will hopefully make sense shortly).

The most common method to set a value on a field directly would be to use the "defaultValue" attribute.

```
<cf_sebField name="fName" label="First Name" defaultValue="Bob">
```

This sets the value of the field if not other value is given either from a form field or from the form itself.

If you want to set a value on the field that will supercede any value given from the form (but not from a form submission), use the "setValue" attribute:

```
<cf_sebField name="fName" label="First Name" setValue="Bob">
```

Loading Data from a Query

You can also use a query to populate the values of several fields at once. The data from this query will supercede any values given in "defaultValue" attributes, but won't supersede any values given in "setValue" attributes.

When you pass a query to cf_sebForm, it assumes that you are passing a one record query. If you are passing a query with more than one record, subsequent records will be ignored.

Here is an example of a form using a query to populate data:

```
<cf_sebForm query="qEmployee">
  <cf_sebField name="fName" label="First Name">
  <cf_sebField name="lName" label="Last Name">
  <cf_sebField name="bdate" label="Birth Date" type="date">
  <cf_sebField type="submit" label="Save">
</cf_sebForm>
```

The qEmployee query will only be used if it exists, is a query, and has at least one record. Otherwise, it will be ignored. If it does exist, it will populate the value of each field that has a matching column name in the query with the value of that column in the query.

If the form has fields that aren't in the query, then they simply won't be populated with any value. If the query has columns that aren't in the form, then they will simply be ignored.

Getting Data from a CFC

You can also have cf_sebForm get data from a CFC. It is important to note, however, that this data must be in form of a query.

```
<cf_sebForm
  pkfield="employee_id"
  recordid="#URL.id#"
  CFC_Component="#Application.Employees#"
  CFC_GetMethod="getEmployee"
>

<cf_sebField fieldname="fName" label="First Name">
<cf_sebField fieldname="lName" label="Last Name">
<cf_sebField fieldname="bdate" label="Birth Date" type="date">
<cf_sebField type="submit" label="Save">

</cf_sebForm>
```

In this case, cf_sebForm will try to load a query from the getEmployee() method of Application.Employees. It will also pass in an argument with the same name as the "pkfield" attribute. The value that it will pass in is the same as the value passed in to cf_sebForm in the "recordid" attribute.

So, it will effectively use the following code to get a query:

```
Application.Employees.getEmployee(employee_id=URL.id)
```

If the code errors or doesn't return a query with at least one record, it will be ignored.

Note that the recordid attribute will default to URL.id if that variable exists and to zero if it does not. So, if you are using URL.id, you can leave out the attribute completely.

If you need to pass in more than one argument to the get method, cf_sebForm allows for that as well. Additional arguments will need to be put into a structure. That structure can then be passed to the CFC_GetArgs attribute of cf_sebForm.

So, if you have a structure of additional arguments named "sMoreArgs", you could use the following code:

```
<cf_sebForm
  pkfield="employee_id"
  recordid="#URL.id#"
  CFC_Component="#Application.Employees#"
  CFC_GetMethod="getEmployee"
  CFC_GetArgs="#sMoreArgs#"
>

<cf_sebField fieldname="fName" label="First Name">
<cf_sebField fieldname="lName" label="Last Name">
<cf_sebField fieldname="bdate" label="Birth Date" type="date">
<cf_sebField type="submit" label="Save">

</cf_sebForm>
```

I should also mention that cf_sebForm was originally built to load data directly from a database table (using a combination of "datasource" and "table" attributes). While that functionality still exists, I do not recommend its use and won't guarantee that it will be supported in future versions.

That covers loading data into cf_sebForm. This also lays the groundwork for adding and editing records with cf_sebForm, but that is a topic for another day.

The cf_sebForm custom tag is part of the [sebtags custom tag set](#), which is open source and free for any use.