**Doku FrameWork-Kommunikation**

2014-04-03

Ich habe versucht, den Ablauf beim Aufruf einer Dialogbox einigermassen übersichtlich zu visualisieren und mit den nötigen Infos zu versehen (s. PDF). Zudem findest Du gleich nachstehend die Beschreibung zweier Methoden zum Befüllen von Formularfeldern in einer Dialogbox.  
  
  
Für das Befüllen einer Dialogbox (Formularfelder) gibt es zwei Möglichkeiten: Entweder rein serverseitig in den HTML-Templates (d.h. die Dialogbox wird fertig ausgefüllt an den Client geschickt) oder Du schickst die leeren Felder, bereitest serverseitig ein JSON-Objekt mit den Daten auf und schickst dieses an eine JS-Funktion, welche die Felder clientseitig befüllt. In der Regel habe ich es so gemacht, dass ich bei kleineren Formularen die Lösung mit den HTML-Templates gewählt habe und bei grösseren habe ich JSON+JS verwendet.  
  
Einfacher wäre für Dich vermutlich die Lösung mit JSON+JS, denn bei den HTML-Templates müsstest Du Dich zuerst mit den speziellen Templating-Tags auseinandersetzen. Das ist zwar nicht allzu komplex, aber sicher neu für Dich.  
  
Du kannst auswählen, welche Variante Dir lieber ist.  
  
Hier ein vereinfachtes Beispiel für JSON+JS (zusammengehörende Daten/Felder/Variabeln habe ich farblich gekennzeichnet):

* Auszug aus **code\_ui/templates/configuration.html** (Dummy-HTML-Code, nicht getestet... WICHTIG: Bei jeder Änderung der HTML-Datei muss die Datei *cache/de-configuration.php* gelöscht werden):  
  <wgui:block name="**meinTest**">  
  <input id="feld1" type="text" />  
  <input id="feld2" type="text" />  
  </wgui:block name="meinTest">
* Auszug aus **code\_ui/js/payroll.js** (Dummy-JS-Code, nicht getestet):  
  function initMeinTest(varData) {  
       $("#feld1").val(varData.feld1);  
       $("#feld2").val(varData.feld2);  
  }
* In der Datei **code\_ui/plugin.php** wären in etwa folgende Codezeilen nötig:  
  **$data** = array(); //leerer Array, weil wir der Template-Engine im Fall von "JSON+JS" keine Daten übergeben müssen  
  $objWindow->loadContent("**configuration**",**$data**,"**meinTest**"); //Aus der Template-Datei "**configuration.html**" wird der HTML-Code innerhalb des Blocks "**meinTest**" geladen.  
  $objWindow->showWindow();  
  communication\_interface::jsExecute("initMeinTest({'feld1':'eins', 'feld2':'zwei'});"); //Funktion initMeinTest in code\_ui/js/payroll.js aufrufen

Hier ein vereinfachtes Beispiel für die Methode mit der Template-Engine:

* Auszug aus **code\_ui/templates/configuration.html** (Dummy-HTML-Code, nicht getestet):  
  <wgui:block name="**meinTest**">  
  <input id="feld1" type="text" value="**<wgui:var name="meinFeld1"/>**" />  
  <input id="feld2" type="text" value="**<wgui:var name="meinFeld2"/>**" />  
  </wgui:block name="meinTest">
* In der Datei **code\_ui/plugin.php** wären in etwa folgende Codezeilen nötig:  
  **$data["meinFeld1"]** = "eins";  
  **$data["meinFeld2"]** = "zwei";  
  $objWindow->loadContent("**configuration**",**$data**,"**meinTest**"); //Aus der Template-Datei "**configuration.html**" wird der HTML-Code innerhalb des Blocks "**meinTest**" geladen. Die Werte im assoc. Array $data werden von der Template-Engine an der entsprechenden Stelle im Template eingefügt (siehe orange Markierung)  
  $objWindow->showWindow();

Ich hofffe, diese Infos sind hilfreich. Falls Du Fragen hast, stehe ich Dir gerne zur Verfügung.  
  
Gruss  
Daniel