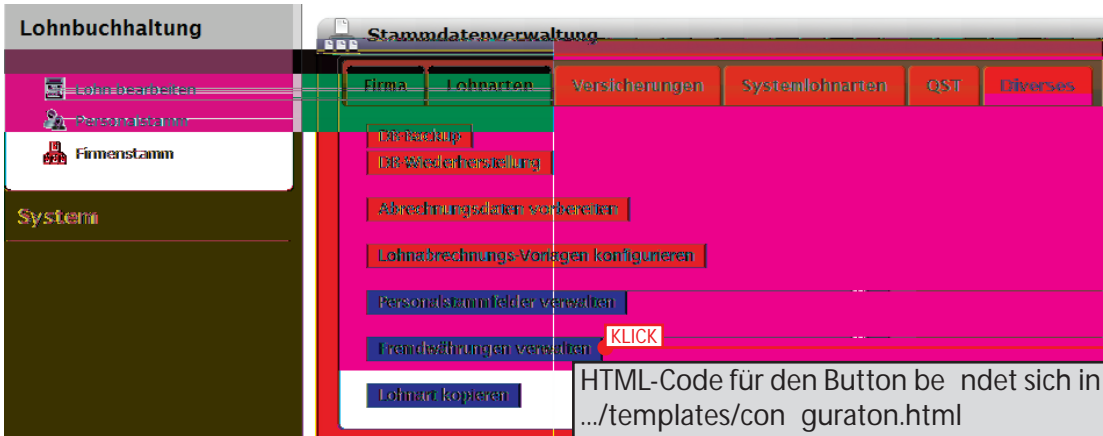


Client

Server



HTML-Code für den Button befindet sich in
.../templates/con guration.html
<button onclick="cb('payroll.currencyCon g');">Fremdwährungen verwalten</button>

WEBROOT/dispatcher.php wird aufgerufen (AJAX) und 'payroll.currencyCon g' als Argument übergeben.
Anhand des Funktions-Präfix 'payroll', wird die Anfrage durch dispatcher.php an WEBROOT/plugins/payroll_V00_01_00/code_ui/plugin.php zur Verarbeitung weitergeleitet.

WEBROOT/plugins/payroll_V00_01_00/code_ui/plugin.php

```
class payroll_UI {
public function sysListener($functionName, $functionParameters) {
    global $saafwCon g;

    switch($functionName) {
        ...
        case 'payroll.currencyCon g':
            if(!isset($functionParameters[0]["action"])) $functionParameters[0]["action"] = "edit";

            switch($functionParameters[0]["action"]) {
                case 'edit':
                    $tmpCncyAssigned = array();
                    $data["currency_table"] = array();
                    $res = blFunctionCall('payroll.getCurrencyList', array("type"=>"assigned"));
                    if($res["success"]) {
                        foreach($res["data"] as $row) {
                            if($row["default_currency"]==1) $data["default_currency"] = $row["currency_code"];
                            else $data["currency_table"][$row["currency_code"]] = $row;
                            $tmpCncyAssigned[] = $row["core_intl_currency_ID"];
                        }
                    }

                    $data["currency_list"] = array();
                    $res = blFunctionCall('payroll.getCurrencyList', array("type"=>"complete"));
                    if($res["success"]) {
                        // prlCurrencyEdit wird als HTML-ID verwendet.
                        // Damit kann die Dialogbox mittels JavaScript identifiziert werden.
                        $objWindow = new wgui_window("payroll" . "prlCurrencyEdit");
                        $objWindow->windowTitle("Währungen bearbeiten");
                        $objWindow->windowIcon("con g32.png");
                        $objWindow->windowWidth(300);
                        $objWindow->windowHeight(205);
                        $objWindow->dockable(false);
                        $objWindow->buttonMaximize(false);
                        $objWindow->resizable(false);
                        $objWindow->fullscreen(false);
                        $objWindow->modal(true);
                        $objWindow->loadContent("con guration",$data,"prlCurrencyEdit");
                        $objWindow->showWindow();

                        communication_interface::jsExecute("#prlCfncncySave").bind('click', function() {
                            communication_interface::jsExecute("#prlCfncncyCancel").bind('click', function() {
                                communication_interface::jsExecute("#prlCfncncy tr div").bind('click', function() {
                                    break;
                                }
                            )
                        });

                        case 'save':
                            ...
                            break;
                    }
                }
            }
        }
    }
}
```

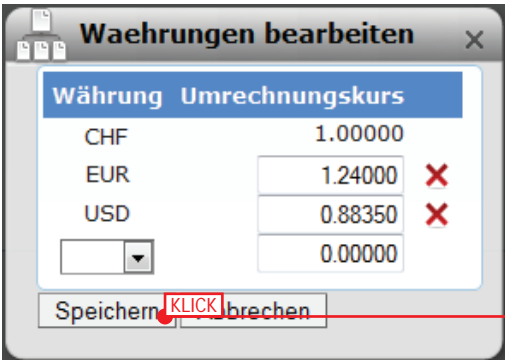
Hier werden die Daten (Währungscodes und Wechselkurse) in einem Assoc. Array (\$data) zusammengestellt.
Mit blFunctionCall wird die Funktion getCurrencyList in code_logic/plugin.php aufgerufen. So kann das UI-Plugin vom BL-Plugin Daten beziehen.

Hier werden die Dialogbox-Parameter gesetzt. Mittels loadContent wird das HTML-Template geladen.

loadContent-Argumente:
arg1: Verwende Template "con guration.html"
arg2: assoc. Array mit den Daten
arg3: Verarbeite Block "prlCurrencyEdit" innerhalb "con guration.html"

Zuletzt wird noch das clientseitige Event-Handling definiert. Das könnte auch in js/payroll.js erfolgen und von hier aus nur noch mittels einem einzigen JS-Funktionsaufruf ausgeführt werden.

Clientseitige Funktionen werden üblicherweise in der JS-Datei WEBROOT/plugins/...payroll_V00_01_00/code_ui/...js/payroll.js gespeichert.
In diesem Beispiel waren aber nur 3 JS-Codezeilen nötig und die habe ich in code_ui/plugin.php untergebracht (siehe rechts, die Zeilen mit communication_interface::jsExecute



Beim Speichern wird wieder das oben beschriebene Procedere abgearbeitet, aber dieses Mal wird beim AJAX-Call 'cb' neben 'payroll.currencyCon g' auch noch 'save' als Zusatzargument mitgegeben. So weiss plugin.php, dass jetzt nicht die Dialogbox angezeigt werden muss, sondern Daten gespeichert werden. Die Formulardaten werden clientseitig in einem JSON-Objekt aufbereitet und ebenfalls beim cb('payroll.currencyCon g'... mitgeschickt. Mit Firebug solltest Du in der Lage sein, Dir den genauen AJAX-Call anzeigen zu lassen. Einmal in code_ui/plugin.php angekommen, werden die Daten geringfügig aufbereitet und mittels blFunctionCall('payroll.saveCurrencyList'... an code_bl/plugin.php zum Speichern übergeben.