

Stammdaten per RFC aus SAP holen

Voraussetzungen:

1. Es werden folgende Dateien benötigt: `php_sapnwrfc.dll`, `sapnwrfc.dll`
2. Es wird ein SAP RFC Baustein je Stammdaten Tabelle benötigt, welcher Stammdaten liefert
3. Es wird ein SAP Benutzer benötigt
4. Es werden die SAP Daten benötigt: Host, SysNr, Client, User, password

Vorbereitung:

1. SAP DLLs im Squeeze/php/ext ablegen
2. In der `php.ini` `extension=php_sapnwrfc.dll` hinzufügen
3. In der `Squeeze/repository/config/clients/<serverFQDN>.json` erweitern:

```
"sap": {  
    "host": "host",  
    "instance": "01",  
    "sysid": "C01",  
    "client": "940",  
    "user": "xyz",  
    "password": "geheim"  
}
```

1. Unter `Squeeze/repository/<ServerFQDN>/UserExits/MasterData` eine Datei `"SapRfcUpdateSettings.json"` anlegen und füllen mit:

```
{  
  "tables": [  
    {  
      "tableid": "1",  
      "query": "ZZDEX_MASTERDATA",  
      "variant": " "  
    }  
  ]  
}
```

```
}
```

1. Den folgenden UserExit anlegen:

```
<?php
use Dompdf\Exception;
use Squeeze\xTools;

// Disable xdebug to prevent CLI hanging
if(extension_loaded('xdebug')) {
    xdebug_disable();
}

// Alle Fehler ausser E_NOTICE melden
error_reporting(E_ALL ^ E_NOTICE);
ini_set('display_errors', true);
ini_set("log_errors", true);
ini_set("error_log", dirname(__DIR__) . "/logs/php-error.log");
//ini_set("memory_limit", "-1");
ini_set('max_execution_time', "1200");

require_once __DIR__ . ' ../../../../htdocs/bootstrap/app.php';

session_start();

if($argv[2] == '') $argv[2] = '80';

$_SESSION['SERVER_NAME'] = $argv[1];
$_SESSION['SERVER_PORT'] = $argv[2];

$logger = Logger::getLogger("main");

// =====
// start time
// =====
$time_start = xTools::microtime_float();

$logger->info("Refresh master data for client ". $argv[1]. " running on port " . $argv[2]);

// =====
```

```

// Get Root directory
// =====
$dirSep = DIRECTORY_SEPARATOR;
$config = new \Squeeze\SqueezeConfig();
$repoRoot = $config->get('repository.root');

// =====
// Get defined Table Settings
// =====
$settingsFile = $repoRoot . 'UserExits' . $dirSep . 'MasterData' . $dirSep .
'SapRfcUpdateSettings.json';
$logger->debug($settingsFile);
if( file_exists($settingsFile)){
    $settings = json_decode( file_get_contents($settingsFile));
    $logger->debug($settings);
    foreach($settings->tables as $table){
        $logger->debug($table);
        //$tableResult = refreshMasterDataTableViaRfc($logger, $table, $config,
$_SESSION['SERVER_NAME'], $_SESSION['SERVER_PORT']);
        $tableResult = MasterDataRefreshForSapRfc($table->tableid, $table->query,$table-
>variant);
    }
} else {
    throw(new \Exception('SapRfcUpdateSettings.json does not exist.', 400));
}

// end time
$time_end = xTools::microtime_float();
$time = $time_end - $time_start;
$logger->info("Refresh master data for client ". $argv[1]. " finished");

/**
 * @param Logger $logger
 * @param stdClass $table
 * @param \Squeeze\SqueezeConfig $config
 * @param string $host
 * @param string $port
 * @return \Squeeze\XReturnObject
 */
function refreshMasterDataTableViaRfc(Logger $logger, $table, \Squeeze\SqueezeConfig $config,

```

```

$host, $port){
    try {

        $curl = curl_init();
        $headers = array("Content-Type: multipart/form-data");
        $options = array(
            CURLOPT_URL => 'http://' . $host . ':' . $port .
"/api/MasterDataRefreshViaSapRfc",
            CURLOPT_HEADER => true,
            CURLOPT_POST => 1,
            CURLOPT_HTTPHEADER => $headers,
            CURLOPT_RETURNTRANSFER => true
        ); // cURL options
        curl_setopt_array($curl, $options);
        $args['tableid'] = $table->tableid;
        $args['sapRfcQuery'] = $table->query;
        $args['sapRfcVariant'] = $table->variant;
        curl_setopt($curl, CURLOPT_POSTFIELDS, $args);
        $logger->debug($curl);
        curl_exec($curl);
        if(!curl_errno($curl))
        {
            $info = curl_getinfo($curl);
            if ($info['http_code'] == 200) {
                $logger->debug('MasterData Refresh successful');
            } else {
                $msg = curl_error($curl);
                $logger->error($msg);
            }
        }
        else
        {
            $msg = curl_error($curl);
            $logger->error($msg);
        }
        curl_close($curl);

    } catch (Exception $e) {
        $logger = Logger::getLogger("main");
        $logger->error($e->getMessage());
    }
}

```

```

        return new \Squeeze\xReturnObject(false, 500, $e->getMessage(), null);
    }
}

```

```

function MasterDataRefreshForSapRfc($tableId, $queryName, $queryVariant ) {
    try {
        $logger = Logger::getLogger("main");
        $logger->info("Start");

        ini_set("memory_limit", "-1");

        // =====
        // Check if tableid is defined
        // =====
        if ($tableId == "") {
            $msg = __FUNCTION__ . ' ' . "No Table ID defined!";
            $logger->error($msg);
            return false;
        }

        // =====
        // Check if sapRfcQuery is defined
        // =====
        if ($queryName == "") {
            $msg = __FUNCTION__ . ' ' . "No Query defined!";
            $logger->error($msg);
            return false;
        }

        // =====
        // Check if sapRfcVariant is defined
        // =====
        if ($queryVariant == "") {
            $msg = __FUNCTION__ . ' ' . "No Variant defined!";
            $logger->error($msg);
            return false;
        }

        $uuid = uniqid();
    }
}

```

```

$dirSep = DIRECTORY_SEPARATOR;

// =====
// Get Master Data Table object
// =====
$xMasterDataTable = new \Squeeze\xMasterDataTable();
if(strtoupper($queryName) == 'MD_VENDOR') {
    $xMasterDataTable->loadByName('creditors');
}
elseif(strtoupper($queryName) == 'ZZDEX_MASTERDATA') {
    $xMasterDataTable->loadByName('creditors');
}
elseif(strtoupper($queryName) == 'MD_PO') {
    $xMasterDataTable->loadByName('orders');
}

// =====
// Get Configuration value
// =====
$config = new \Squeeze\SqueezeConfig();
$sapConfig = [
    'ashost' => $config->get('sap.host'),
    'sysnr' => $config->get('sap.instance'),
'sysid' => $config->get('sap.sysid'),
'lang' => 'DE',
    'client' => $config->get('sap.client'),
    'user' => $config->get('sap.user'),
    'passwd' => $config->get('sap.password'),
'trace' => '1'
];

$logger->debug($sapConfig);

$c = new \SAPNWRFC\Connection($sapConfig);
$f = $c->getFunction('ZZDEX_MASTERDATA');
//$result = $f->invoke([
//    'I_WORKSPACE' => 'G',
//    'I_USERGROUP' => '/QUERY',
//    'I_QUERY' => $queryName,
//    'I_VARIANT' => $queryVariant

```

```

    //]);
$result = $f->invoke();

if($result['RETURN']['TYPE'] == 'E'){
    $msg = __FUNCTION__ . ' ' . $result['RETURN']['MESSAGE'];
    $code = 400;
    $logger = \Logger::getLogger("main");
    $logger->error(__CLASS__ . " " . __FUNCTION__ . " " . $msg);
    return false;
}

// =====
// Check if a userexit script for this table exists
// =====
//$root = \Squeeze\xTools::getRoot();
$repo = $config->get('repository.root');
$userExit = xTools::buildAbsolutePath($repo, 'UserExits' . $dirSep . 'MasterData',
true);
if (file_exists($userExit . $xMasterDataTable->getName() . '.php')) {
    //$logger->info("File for User Exit of table " . $xMasterDataTable->getName() . "
found.");
    include_once($userExit . $xMasterDataTable->getName() . '.php');
} else {
    //$logger->info("No UserExit for table " . $xMasterDataTable->getName() . "
defined.");
}

$resultTable = array();
$i = 0;

if(strtoupper($queryName) == 'MD_VENDOR') {
    foreach ($result['E_DATA'] as $line) {

        $cells = explode(';', $line['LINE']);

        $newLine['CreditorId'] = $cells[0];
        $newLine['CompanyId'] = $cells[1];
        $newLine['Name1'] = $cells[2];
        $newLine['Name2'] = $cells[3];
        $newLine['Name3'] = $cells[4];
    }
}

```

```

$newLine[' Phone' ] = $cells[5];
$newLine[' Street' ] = $cells[6];
$newLine[' CountryCode' ] = $cells[7];
$newLine[' Zip' ] = $cells[8];
$newLine[' City' ] = $cells[9];
$newLine[' EUTaxId' ] = $cells[10];
$newLine[' NationalTaxId' ] = $cells[11];
$newLine[' BankCountry' ] = $cells[12];
$newLine[' BankCode' ] = $cells[13];
$newLine[' BankAccount' ] = $cells[14];
$newLine[' IBAN' ] = $cells[15];

$logger->debug($newLine);
if (function_exists('Enhance_' . $xMasterDataTable->getName()))
{
    $newLine = call_user_func('Enhance_' . $xMasterDataTable->getName(),
$newLine);
}
$resultTable[] = $newLine;
}
}

elseif(strtoupper($queryName) == 'ZZDEX_MASTERDATA') {
    foreach ($result['ET_VND'] as $line) {
//0      1      2      3      4      5      6      7      8      9      10      -11 12      13
14      15      16      17      18      19
//Kreditor BuKr. Land Name1 Name2 ??? Name Ort Ortsteil Postfach PLZ-Postf. PLZ Region
Suchbegr. Straße USt-Id.Nr SteuerNr. IBAN Adresse ??? ??? ???

$logger->debug($line);
    $cells = explode(';', $line['']);
$logger->debug($cells);

    $newLine[' CreditorId' ] = ltrim( $cells[0], '0' );
    $newLine[' CompanyId' ] = $cells[1];
    $newLine[' Name1' ] = $cells[3];
    $newLine[' Name2' ] = $cells[4];
    //$newLine[' Name3' ] = $cells[4];
    //$newLine[' Phone' ] = $cells[5];
    $newLine[' Street' ] = $cells[14];
    $newLine[' CountryCode' ] = $cells[2];

```



```

$newLine[' Zip' ] = $cells[11];
$newLine[' City' ] = $cells[7];
$newLine[' EUTaxId' ] = $cells[15];
$newLine[' NationalTaxId' ] = $cells[16];
//$newLine[' BankCountry' ] = $cells[12];
//$newLine[' BankCode' ] = $cells[13];
//$newLine[' BankAccount' ] = $cells[14];
$newLine[' IBAN' ] = $cells[17];

//$logger->debug( $newLine);
if ( function_exists(' Enhance_' . $xMasterDataTable->getName()))
{
    $newLine = call_user_func(' Enhance_' . $xMasterDataTable->getName(),
$newLine);
}
$resultTable[] = $newLine;
}
}
elseif( strtoupper( $queryName) == ' MD_PO' ) {
    foreach ( $result[' E_DATA' ] as $line) {

        $cells = explode(';', $line[' LINE' ]);

        $newLine[' CompanyId' ] = $cells[0];
        $newLine[' CreditorId' ] = $cells[1];
        $newLine[' OrderNumber' ] = $cells[2];
        $newLine[' OrderItem' ] = $cells[3];
        $newLine[' Material' ] = $cells[4];
        $newLine[' Description' ] = $cells[5];
        $newLine[' Quantity' ] = $cells[6];
        //$newLine[' Quantity' ] = $cells[7];
        $newLine[' QuantityUnit' ] = $cells[8];
        $newLine[' PriceUnit' ] = $cells[9];
        //$newLine[' Client' ] = $cells[10];
        $newLine[' NetAmount' ] = $cells[11];
        $newLine[' Currency' ] = $cells[12];

        //$logger->debug( $newLine);
        if ( function_exists(' Enhance_' . $xMasterDataTable->getName()))
        {

```

```

$newLine = call_user_func('Enhance_' . $xMasterDataTable->getName(),
$newLine);

        }
        $resultTable[] = $newLine;
    }
}

$result = $xMasterDataTable->createNewTempTable($uuid);
//$logger->info("xMasterDataTable created");
if ($result->isSuccess() == false) {
    $logger->error($result->getMessage());
    return false;
}
//$logger->info("DB Created");

$result = $xMasterDataTable->insertData($uuid, $resultTable);
$resultTable = null;
if ($result->isSuccess() == false) {
    $logger->error($result->getMessage());
    return false;
}
//$logger->info("DB Insert done");

$result = $xMasterDataTable->dropTableAndReplace();
if ($result->isSuccess() == false) {
    $logger->error($result->getMessage());
    return false;
}
//$logger->info("Drop and replace done.");

// =====
// Check if a UserExit script saving the document exists
// =====
$repo = $config->get('repository.root');
$userExit = xTools::buildAbsolutePath($repo, 'UserExits/MasterData/', true);
if (file_exists($userExit . 'AfterMasterDataRefresh.php')) {
    // $logger->info("File for UserExit AfterMasterDataRefresh found.");
    include_once($userExit . 'AfterMasterDataRefresh.php');
    $userExitResult = call_user_func('AfterMasterDataRefresh',
$xMasterDataTable);

```

```

    } else {
        // $logger->info("No UserExit for AfterMasterDataRefresh defined.");
    }

    $logger->info(json_encode($xMasterDataTable));

} catch (\Exception $e) {
    xTools::handleException($e);
    $code = 500;
    $msg = $e->getMessage();
    $logger = \Logger::getLogger("main");
    $logger->error(__CLASS__ . " " . __FUNCTION__ . " " . $msg);
    return false;
}
}

```

Anpassen des Mappings im UserExit in Zeilen 219ff.

```
Aufruf per "D:\EASY\Squeeze\php\php.exe" RefreshMasterDataViaRfc.php ServerFQDN 80
```

Es wird im UserExit Ordner oder teilweise im Windows/System32 eine *.trc Datei erzeugt, welche die Verbindungsdaten wiedergibt.

<https://github.com/gkralik/php7-sapnwrfc/releases>

Revision #4

Created 25 November 2021 19:09:41 by Simon Pust

Updated 25 November 2021 19:33:34 by Simon Pust