

AfterExtraction

Der UserExit AfterExtraction wird immer nach der Extraktion eines Dokumentes ausgeführt und kann genutzt werden um das Ergebnis zu beeinflussen, bevor das Dokument in die Validierung übergeben wird.

- [Eine Position je Bestellnummer](#)
- [Übergabe jeder Bestellnummer Alternative als kommaseparierte Liste](#)

Eine Position je Bestellnummer

Gerade bei SAP Projekten kann es erforderlich sein, je Bestellnummer eine Position erzeugen zu müssen. In diesem Fall kann der folgende UserExit genutzt werden, um nach der Extraktion des Beleges eine Position je gefundener Bestellnummer zu erzeugen.

Voraussetzungen dieses UserExits:

- Das Kopffeld für Bestellnummern heißt "**OrderNumber**".
- Es existiert ein Feld in der Positionstabelle mit dem Namen "**PosOrderNumber**".
- Das Feld für die Positionen heißt "**LineItems**".

```
<?php

use Squeeze\xDoc;
use Squeeze\xDocumentField;

function AfterExtraction( xDoc $xDoc)
{

    $logger = Logger::getLogger("main");
    createLineItemsByHeadField($xDoc, $logger, "OrderNumber", "PosOrderNumber",
"LineItems");

    return true;
}

/**
 * @param xDoc $xDoc
 * @param Logger $logger
 * @param string $originHeadField
 * @param string $targetLineField
 * @param string $lineItemName
 */
```

```

function createLineItemsByHeadField(
    xDoc $xDoc,
    Logger $logger,
    string $originHeadField,
    string $targetLineField,
    string $lineItemName
): void {
    $head = $xDoc->getDocumentFieldByName($originHeadField);
    $lines = $xDoc->getDocumentFieldByName($lineItemName);

    $head->alternatives = getUniqueAlternatives($head, $logger);
    if (count($head->alternatives) > 1) {
        $head->getValue()->value = '';
        $head->getValue()->text = '';
        foreach ($head->alternatives as $alternative) {
            $line = new Squeeze\xDocValue(0, "", "", "string", 100, 0, 99999, 99999, 0,
0);

            $line->subfields[$targetLineField] = $alternative;
            $line->subfields[$targetLineField]->subFieldName = $targetLineField;
            $lines->alternatives[] = $line;
        }
    }
}

/**
 * @param xDocumentField $field
 * @param Logger $logger
 * @return xDocumentField[]
 */
function getUniqueAlternatives(xDocumentField $field, Logger $logger): array
{
    // =====
    // Only keep unique alternatives
    // =====
    $tmpStore = [];
    $uniqueAlternatives = [];
    $logger->debug('The field ' . $field->name . ' has ' . count($field->alternatives) . '
alternatives');
    foreach ($field->alternatives as $alternative) {
        $logger->debug('Alternative = ' . $alternative->value);
    }
}

```

```
        if (!in_array($alternative->value, $tmpStore)) {
            $logger->debug(' Alternative will be kept ' . $alternative->value);
            $tmpStore[] = $alternative->value;
            $uniqueAlternatives[] = $alternative;
        }
    }

    if (count($uniqueAlternatives) > 0) {
        $field->alternatives = $uniqueAlternatives;
    }

    return $field->alternatives;
}
```

Übergabe jeder Bestellnummer Alternative als kommaseparierte Liste

Um jede Bestellnummer Alternative in einer kommaseparierten Liste als Rechnungskopffeld zu übergeben folgende Erweiterungen hinzufügen:

```
function allOrderNumbersInHeadField(xDoc $xDoc, Logger $logger, string $originHeadField): void
{
    $head = $xDoc->getDocumentFieldByName($originHeadField);

    $head->alternatives = getUniqueAlternatives($head, $logger);
    if (count($head->alternatives) > 1) {
        $head->getValue()->value = '';
        $head->getValue()->text = '';
        $strOrdernr = '';
        foreach ($head->alternatives as $alternative) {
            $strOrdernr .= ',' . $alternative->value;
        }
        $strOrdernr = ltrim($strOrdernr, ",");
        $head->getValue()->value = $strOrdernr;
        $head->getValue()->text = $strOrdernr;
    }
}

/**
 * @param xDocumentField $field
 * @param Logger $logger
 * @return xDocumentField[]
 */
function getUniqueAlternatives(xDocumentField $field, Logger e$logger): array
{
    // =====
```

```

// Only keep unique alternatives
// =====
$tmpStore = [];
$uniqueAlternatives = [];
$logger->debug('The field ' . $field->name . ' has ' . count($field->alternatives) . '
alternatives');
foreach ($head->alternatives as $alternative) {
    $logger->debug('Alternative = ' . $alternative->value);
    if (!in_array($alternative->value, $tmpStore)) {
        $logger->debug('Alternative will be kept ' . $alternative->value);
        $tmpStore[] = $alternative->value;
        $uniqueAlternatives[] = $alternative;
    }
}

if (count($uniqueAlternatives) > 0) {
    $field->alternatives = $uniqueAlternatives;
}

return $field->alternatives;
}

```

Aufruf der Funktion per:

```
allOrderNumbersInHeadField($xDoc, $logger, "OrderNumber");
```