DocFile prototype

Die DocFile Klasse wurde um einige spezifische Funktionen erweitert.

```
/** Fügt einen Kommentar zum Historienfeld "Comment" hinzu.
 * Bei einer Split-Mappe wird der Kommentar auch zur Haupt-Mappe hinzugefügt.
 * @param {string} comment Kommentar-Angabe
 * @returns {string} Leerer String oder Fehlermeldung
**/
DocFile.addComment( comment )

/** Prüft ob für den Benutzer eine Wiedervorlage zur Mappe definiert wurde.
 * @param {string} optUserLogin Optional kann ein Benutzer-Login mitgegeben werden.
Andernfalls wird der aktuell sperrende Benutzer geprüft.
 * @returns {boolean} true (Wiedervorlage ist definiert) / false
 * @since 1.1.015
**/
DocFile.checkResubmission( optUserLogin )

/** Prüft ob der aktuelle Workflow-Schritt für den aktuellen Benutzer automatisch im
Bearbeitungs-Modus startet.
 * @param {SystemUser} systemUser SystemUser Objekt
 * @returns {boolean} true (StartEditMode) / false
**/
DocFile.checkStartEditMode( systemUser )

/** Prüft alle Kopf-Pflichtfelder der Mappe, wenn der Parameter
"CheckMandatoryHeadFieldsOnActionEnd" auf 'true' steht.
 * Setzt context.errorMessage
 * @returns {boolean} true (Pflichtfelder gesetzt) / false (Pflichtangaben fehlen)
**/
DocFile.checkMandatoryHeadFields()

/** Vergleicht einen Feldwert mit einem String-Wert.
 * Diese Funktion wird zum Beispiel bei der Auswertung der Workflow-Regeln oder bei der
Parameter-Ermittlung verwendet.
 * @param {string} techFieldName Technischer Feldname
```

```
 * @param {string} stringValue Vergleichs-Angabe
 * @param {string} optCompareType Optionale Angabe eines Vergleichstyps. Als Default-Wert
wird "=" verwendet.
 *                                Erlaubte Typen sind: '=', '<', '<=', '>', '>=', '..', '|~',
'!=', 'expr'
 * @returns {string} true (Vergleichswert passt zum Feldwert) / false (Vergleichswert passt
nicht zum Feldwert)
**/
DocFile.compareFieldValueWithString( techFieldName, stringValue, optCompareType )

/** Löscht alle Sub-Mappen zu einer Haupt-Mappe
 * @returns {string} Leerer String oder Fehlermeldung
**/
DocFile.deleteSubFiles()

/** Liefert die ID eines ausgehenden Kontrollflusses zum Abschluss einer Aktion passend zur
angegebenen Navigation.
 * Bei einer Weiterleitung über eine benutzerdefinierte Aktion muss die anschließende
Navigation
 * über die Skript-Rückgabe gesteuert werden!
 * @param {string} paramNavigation Navigation("keepfile", "next", "overview", "inbox",
"folder").
 * @returns {string} Leerer String oder Kontrollfluss ID
**/
DocFile.getActionEndControlFlowIdByNavigation( paramNavigation)

/** Ermittelt die in den Workflow-Aktionen konfigurierte Navigation nach Abschluss der Aktion.
 * @returns {string} Leerer String oder Navigation("keepfile", "next", "overview", "inbox",
"folder")
**/
DocFile.getActionEndNavigation()

/** Liefert die ID zu einem ausgehenden Workflow-Kontrollfluss-Namen.
 * @param {string} paramCfName Kontrollfluss-Name
 * @param {boolean} paramCfStartsWith Optionale Angabe, wenn der Kontrollfluss-Name nur mit
dem 'paramCfName' beginnt.
 * @returns {string} Leerer String oder Kontrollfluss ID
**/
DocFile.getControlFlowIdByName( paramCfName, paramCfStartsWith )
```

```
/** Wenn zu einem Beleg Split-Mappen erstellt werden, dann wartet die Haupt-Mappe auf die vollständige Abarbeitung
 * aller Split-Mappen.
 * Der Workflow-Verlauf der Split-Mappen ist nur im Monitor der einzelnen Split-Mappen nachzuvollziehen.
 * Die Split-Mappen werden allerdings nach der Verarbeitung gelöscht.
 * Über diese Funktion können Monitor-Einträge zur Hauptmappe hinzugefügt werden.
 * @param {string} comment Optionale Angabe für den Kommentar beim Monitor-Eintrag.
 * @returns {boolean} true / false
**/
DocFile.forwardMainFile( comment )

/** Wenn eine Mappe in einer technischen Aktion durch einen Job verarbeitet werden soll,
 * dann kann der Job die Mappe mit dieser Funktion weiterleiten.
 * Das Skript wechselt hierfür in den Kontext eines sperrenden Benutzers.
 * @param {string} controlFlow OKontrollfluss-Name ('TechActionEnd' oder Default: 'TechActionEndManually').
 * @param {string} comment Optionale Angabe für den Kommentar beim Monitor-Eintrag.
 * @returns {boolean} true / false
**/
DocFile.forwardTechActionJob( controlFlow, comment )

/** Ermittelt die Anzahl der Nachkommastellen aus der Gentable-Feldkonfiguration für die Gentable-Spalte.
 * @param {string} colName Technischer Gentable-Spalten-Name.
 * @param {number} expValue Standardwert für die Anzahl der Nachkommastellen.
 * @returns {number} Anzahl Dezimalstellen
 * @since 1.1.015
**/
DocFile.getGentableColumnDecimal( colName, expValue )

/** Ermittelt das Wiedervorlagedatum für einen Benutzer.
 * @param {string} userLogin SystemUser-Login
 * @returns {string} Wiedervorlagedatum als String
 * @since 1.1.015
**/
DocFile.getResubmissionDate( userLogin )

/** Gibt eine Liste aller sperrenden Benutzer und Gruppen im Workflow aus.
 * Die Funktion iteriert über alle sperrenden Workflow-Schritte.
```

```
 * @param {string} pListSeparator Trennzeichen zwischen den Angaben.
 * @returns {string} true / false
**/
DocFile.listLockingUser( pListSeparator )

/** Setzt die Eigenschaft "ReadOnly" bei allen Feldern der Mappe auf 'true'.
 * @returns {string} true (sync() erfolgreich) / false
**/
DocFile.setAllFieldsReadOnly()

/** Berechnet ein "DeadlineDate"-Datumsfeld anhand eines Ausgangs-Datums und den Angaben vom Parameter "DeadlineDate".
 * Ebenfalls werden die Felder "DeadlineDays" und "DeadlineSign" gesetzt.
 * @param {boolean} syncFile Über den Parameter kann optional direkt ein sync() ausgeführt werden.
 * @param {string} optUseDateField Optional kann ein alternatives Datumsfeld verwendet werden (Default: 'WorkflowStart').
 * @param {boolean} optSetEmptyDate Optional wird ein leeres Datumsfeld auf das aktuelle Tagesdatum gesetzt.
 * @returns {boolean} true / false
**/
DocFile.setDeadlineDate( syncFile, optUseDateField, optSetEmptyDate )

/** Liest die Feldkonfiguration und die Alternativen und setzt die Feldeigenschaften.
 * @returns {boolean} true / false
**/
DocFile.setFieldAttributesByConfig()

/** Wendet die unter "Feldkonfigurationen" -> "Suchfelder" eingestellte Konfiguration an.
 * Die Funktion holt sich die Einträge aus definierten Positions-Spalten und schreibt die Angaben in ein Kopf-Feld.
 * @returns {string} Leerstring oder Fehlermeldung
**/
DocFile.setGentableSearchFields()

/** Wechselt den ausführenden Benutzer an der aktuellen Workflow-Aktion.
 * @param {string} optLogin Optionale Angabe eines gültigen Benutzer-Logins (Default: aktueller Benutzer).
 * @returns {*} true / Fehlermeldung
**/
```

```
DocFile.takeOver( optLogin )

/** Entsperrt eine Mappe.
 * @returns {string} true / false
**/
DocFile.unlockDocFile()
```

Zudem können Parameter-Angaben zum DocFile-Objekt ermittelt werden.

```
/** Ermittelt das Parameter-Objekt zum angegebenen Parameter.
 * @param {string} paramName Technischer Parameter-Name.
 * @param {string} optValue Optionale Angabe, ob direkt der 1., 2. oder 3. Wert zum Parameter zurückgegeben werden soll.
 * @returns {*} Gibt entweder ein Parameter-Objekt zurück oder den spezifischen 1., 2. oder 3. Wert oder null bei Fehlern
**/
DocFile.getParamObject( paramName, optValue )
/* Beispiel */
var docFile = context.file;
var value1String = docFile.getParamObject("MeinParameter", 1);

/** Liefert einen spezifischen Wert zu einem Parameter-Objekt (siehe DocFile.getParamObject()).
 * @param {string} paramObj Optionale Angabe eines gültigen Benutzer-Logins (Default: aktueller Benutzer).
 * @param {string} valueNo Angabe, ob der 1., 2. oder 3. Parameter-Wert zurückgegeben werden soll.
 * @param {*} standardValue Standardwert im passenden Datentypen.
 * @param {string} optParamType Optionale Angabe für den Datentyp des Parameters (Standard: 'string').
 *    Die angaben werden in der Datenbank immer als String gespeichert, werden aber in den Datentypen gewandelt.
 *    Gültige Angaben: 'boolean'/'bool', 'percent'/'numeric'/'decimal', 'select'/'string'
 * @returns {*} In passenden Datentypen umgewandelter Parameter-Wert / Standardwert
**/
DocFile.getParamValue( paramObj, valueNo, standardValue, optParamType)
/* Beispiel */
var docFile = context.file;
var paramObj = docFile.getParamObject("MeinParameter");
var value1Bool = docFile.getParamValue( paramObj, 1, true, "boolean" );
```

```
var value2String = docFile.getParamValue( paramObj, 2, "Test", "string" );
var value3Numeric = docFile.getParamValue( paramObj, 3, 1.5, "numeric" );
```

Revision #13
Created 11 September 2020 10:19:13 by Markus Meisner
Updated 5 August 2022 10:03:56 by Markus Meisner