Protokollcleaner

Documentation

targetingsnake

31. Oktober 2017

Inhaltsverzeichnis

Teil I.

Nutzungsanleitung

1. Verwendung

1.1. Beschreibung

Mithilfe dieses Scripts werden Protokolle aus dem internen Teil des StuRa-Teils des Gremienwikis in den öffentlichen Teil des Gremienwikis kopiert. Damit sind sie Hochschulöffentlich für Studierende. Daher muss vor bzw. während des Kopiervorgangs der interne Teil des Protokolls entfernt werden, dies wird durch dieses Script realisiert.

2. Konfiguration

2.1. Grundlagen

Für eine Initiale Konfiguration müssen lediglich 16 Variablen auf entsprechende Werte gesetzt werden. Hierzu werden die entsprechenden Daten in die Variablen Definiton am Anfang der 'Main'-Klasse geschrieben. Diese werden für alle Funktionen und anderen Klassen verwendet.

2.2. Eingabepfad

$inputpath Der hier zwischen den Anführungszeichen angegebene Pfad, ist der Pfad des Ordners, der alle Protokolle die zu kopieren sind beinhaltet. Die enthaltenen Dateien sollten Text-Dateien im Dokuwikiformat sein. Dabei ist der Titel nach dem Datum des Protokolls zu benennen. Dies geschieht nach dem Schema yyyy-mm-dd. Hierbei ist yyyy die vierstellige Jahreszahl, mm die zweistellige Monatsnummer und dd die zweistellige Tageszahl. Am Ende des Schemas muss noch die korrekte Dateieindung für Textdateien ".txt"befinden. Der Ordner und dessen Inhalt muss für das Script lesbar öffenbar sein.

2.3. Ausgabepfad

$outputpath Der hier zwischen den Anführungszeichen angegebene Pfad, ist der Pfad des Ordners, in dem die verarbeiteten Protokolle, die zu kopieren waren, abgelegt werden. Die enthaltenen Dateien werden als Text-Dateien im Dokuwikiformat erzeugt. Dabei wird der Titel nach dem Datum des Protokolls benannt. Dies geschieht nach dem Schema yyyy-mm-dd. Hierbei ist yyyy die vierstellige Jahreszahl, mm die zweistellige Monatsnummer und dd die zweistellige Tageszahl Am Ende des Schemas wird noch die korrekte Dateieindung für Textdateien ".txtängehangen. All dies geschieht automatisch beim Ausführen des Scripts. Der Ordner und dessen Inhalt muss für das Script schreibbar öffenbar sein.

2.4. Beschlussliste

$decissionList Der hier zwischen den Anführungszeichen angegebene Pfad verweist direkt auf die Datei, in der alle Beschlüsse in einer Tabelle im Dokuwikiformat angegeben sind. Aus ihr werden die bereits beschlossenen Protokolle und alle beschlossenen Finanzanträge ausgelesen. Diese Datei wird nur lesend geöffnet.

2.5. Hilfsdatei

$helperFilePath Der hier zwischen den Anführungszeichen angegebene Pfad verweist direkt auf die Datei, in der das Script bereits erfolgte Ausgaben mitzeichnet, um bei Datenweitergabe an andere Websiten zu verhindern, dass diese Daten doppelt erhalten. Diese Datei muss les- und schreibar zu öffnen sein.

2.6. Starttag

$starttag Der hier zwischen den Anführungszeichen angegebene Wert definiert den Inhalt des Dokuwikitags, welches den Beginn eines nicht zu veröffentlichen Teiles kennzeichnet. Hierzu wird nur der Tagname zwischen die Anführungszeichen geschrieben.

2.7. Endtag

$endtag Der hier zwischen den Anführungszeichen angegebene Wert definiert den Inhalt des Dokuwikitags, welches das Ende eines nicht zu veröffentlichen Teiles kennzeichnet. Hierzu wird nur der Tagname zwischen die Anführungszeichen geschrieben.

2.8. Debug

$debug Wenn hier *false* eingetragen wird, so werden keine Debugausgaben mehr angezeigt. Hierdurch werden alle Ausgaben über die Finanzbeschlüsse, die noch nicht dem Tool bekannt sind, unterdrückt. Jedoch findet bei aktivierter weitergabe diese Trotzdem statt. Falls hier *true* steht, so werden alle Ausgaben über die Finanzbeschlüsse, die das

Tool noch nicht kennt erfolgen.
Bereits dem Tool bekannte Finanzbeschlüsse werden nie angezeigt.

2.9. Nur neue Finanzbeschlüsse ausgeben

$onlyNew Wenn hier *false* eingetragen wird, so werden alle Finanzbeschlüsse, die noch nicht dem Tool bekannt sind, durch das Tool bearbeitet. Falls hier *true* steht, so werden nur Finanzbeschlüsse die das Finanzentool kennt, dieses Tool hier jedoch nicht, bearbeitet.
Diesem Tool bereits bekannte Finanzbeschlüsse werden nie verarbeitet.

2.10. Daten an andere Website weitergeben

$postData Wenn hier *false* eingetragen wird, so werden keine Daten zu Finanzbeschlüsse, die noch nicht dem Tool bekannt sind, durch das Tool an eine angegebene andere Website weitergegeben. Falls hier *true* steht, so werden Daten zu Finanzbeschlüsse, die noch nicht dem Tool bekannt sind, durch das Tool an eine angegebene andere Website weitergegeben.

2.11. Webadresse zur Weitergabe der Daten per HTTP-Post

$PostUrl Hier wird eine HTTP- oder HTTPS-Url angegeben, zu welcher Daten mittels POST-Methode übermittelt werden sollen. Dies geschieht jedoch nur wenn *$postData* den Wert *true* hat.

2.12. Einstellen des ersten zu verarbeiteten Protokolls

Vorbemerkung Wenn ein Datum eingestellt werden soll, so müssen die einzelnen Bestandteile des Datums auf die folgenden Parameter aufgeteilt werden.

$startYear Dieser Parameter stellt die Jahreszahl, ab der Protokolle verarbeitet werden sollen als Integerwert dar. Zum anpassen muss lediglich die vierstellige Jahreszahl angepasst werden.

$startMonth Dieser Parameter stellt die Monatszahl, ab der Protokolle verarbeitet werden sollen als Integerwert dar. Zum anpassen muss lediglich die zweistellige Monatszahl angepasst werden.

$startday Dieser Parameter stellt die Tageszahl, ab der Protokolle verarbeitet werden sollen als Integerwert dar. Zum anpassen muss lediglich die zweistellige Tageszahl angepasst werden.

2.13. Konfigurationsdatei erstellen

Beispiel Für eine funktionsfähige Konfigurationsdatei kopieren sie bitte die *config_ example.php* in eine Datei mit dem Namen *config.php*. Dies können sie unter Linux mittels des Befehls *cp config_ example.php config.php* durchführen. Anschließend ändern sie die darin enthaltenen Werte auf ihre benötigten Werte. Passen sie bitte unbedingt alle Pfadvariablen an. Diese sind alle Notwendig. Falls sie das Weitergeben von Daten an andere Websiten deaktiviert haben, so benötigen sie keinen Pfad zur Weitergabe.

Teil II.

Code Dokumentation

3. Klassen

3.1. Main

3.1.1. Variablen

$inputpath In dieser öffentlich statischen Variable wird der Pfad zum Ordner mit den internen Protokollen als Zeichenkette gespeichert. Die Variable wird nun einmal im Konstruktor der 'Main'-Klasse gesetzt und im weiteren Programm nur gelesen.

$outputpath In dieser öffentlich statischen Variable wird der Pfad zum Ordner mit den hochschulöffentlichen Protokollen als Zeichenkette gespeichert. Die Variable wird nun einmal im Konstruktor der 'Main'-Klasse gesetzt und im weiteren Programm nur gelesen.

$decissionList In dieser öffentlich statischen Variable wird der Pfad zur Datei, welche alle Beschlüsse beinhaltet, als Zeichenkette gespeichert. Die Variable wird nun einmal im Konstruktor der 'Main'-Klasse gesetzt und im weiteren Programm nur gelesen.

$helperFilePath In dieser öffentlich statischen Variable wird der Pfad zur Hilfsdatei als Zeichenkette gespeichert. Die Variable wird nun einmal im Konstruktor der 'Main'-Klasse gesetzt und im weiteren Programm nur gelesen.

$starttag In dieser öffentlich statischen Variable wird der Name des Tages gespeichert, welcher den Anfang des nicht zu kopierenden Teils des Dokumentes beschreibt. Die Zeile wird mit dem Tag entfernt. Die Variable ist als Zeichenkette zu interpretieren.

$endtag In dieser öffentlich statischen Variable wird der Name des Tages gespeichert, welcher das Ende des nicht zu kopierenden Teils des Dokumentes beschreibt. Die Zeile

wird mit dem Tag entfernt. Die Variable ist als Zeichenkette zu interpretieren.

$debug In dieser öffentlich statischen Variable wird fesgelegt, ob alle Testausgaben angezeigt werden. Die Variable ist als Wahrheitswert zu interpretieren.

$onlyNew In dieser öffentlich statischen Variable wird fesgelegt, ob nur durch das Finanztool mit Links versehene Finanzanträge ausgelesen werden sollen. Die Variable ist als Wahrheitswert zu interpretieren.

$postData In dieser öffentlich statischen Variable wird fesgelegt, ob neu hinzugekommene Finanzbeschlüsse durch das Tool mittels Http-POST-Methode an eine andere Website weitergegeben werden sollen. Die Variable ist als Wahrheitswert zu interpretieren.

$PostUrl In dieser öffentlich statischen Variable wird fesgelegt, an welche URL die Datem am Ende durch das Tool mittels Http-POST-Methode weitergegeben werden sollen. Die Variable ist als Zeichenkette zu interpretieren.

$startYear In dieser privaten Variable wird fesgelegt, ab welchem Jahr die Protokolle bearbeitet werden sollen. Die Variable ist als Ganzzahl zu interpretieren.

$startMonth In dieser privaten Variable wird fesgelegt, ab welchem Monat, des in *$startYear* angegbenen Jahres, die Protokolle bearbeitet werden sollen. Die Variable ist als Ganzzahl zu interpretieren.

$startday In dieser privaten Variable wird fesgelegt, ab welchem Tag, des in $startMonth angegeben Monats, des in *$startYear* angegbenen Jahres, die Protokolle bearbeitet werden sollen. Die Variable ist als Ganzzahl zu interpretieren.

$financialResolution In diesem öffentlich statischem Array werden alle weiterzugebenden Finanzbeschlüsse gesammelt. Die Initialisierung erfolgt bei der ersten Verwendung der Variable.6

$files In dieser privaten Variable werden alle Namen der Dateien, die sich im Ordnerm, welcher in der Variablen *$inputpath* als Pfad angegeben ist, als Array gespeichert.

$knownDecissions In dieser privaten Variable werden alle in der Hilfsdatei gespeicherten, bereits verarbeiteten Finanzbeschlüsse gespeichert. Die Variable ist als Zeichenkette zu interpretieren.

3.1.2. Funktionen

__construct()

Dies ist der Konstruktor. Hier wird immoment lediglich die Konfiguration einbezogen. Der Konstruktor ist öffentlich, nicht statisch.

getAllFiles()

Die Funktion ist öffentlich, nicht statisch.

Vorbereitung Dies ist die Hauptfunktion der Klasse. Die Funktion stellt die Hauptschleife des Scriptes dar. Es werden die Variablen *$knownDecissions* und *$files* als Array instanziiert. Anschließend werden alle Dateinamen des Ordners, welcher unter *$inputpath* angegeben ist, als Array gespeichert. Anschließend wird dieses Array mithilfe einer 'Für-alle-Schleife' angearbeitet.

Hauptschleife In der Schleife der Funktion wird zunächst überprüft, ob das Protokoll ein Datum nach dem Datum, ab welchem das Script Protokolle verarbeiten soll, hat. Anschließend wird für diese Datei eine Instanz der 'File'-Klasse mittels deren Konstruktor angelegt. Nun wird der Dateiname des bearbeiteten Protokolls festgelegt. Dieser ergibt sich aus dem statischen Pfad zum Ausgabeordner, welcher in der Variablen *$outputpath* angegeben ist. Anschließend wird geprüft, ob ein Protokoll bereits kontrolliert und abgestimmt wurde, falls dies geschehen ist, so wir es als Endgültig veröffentlicht, andernfalls wird es als Entwurf veröffentlicht. Entsprechend der Überprüfung wird nun eine entsprechende Ausgabe erzeugt. Nun wird das Protokoll mithilfe der *copy*-Methode kopiert. abschließend wird Dateiname in das *$file*-Array geschrieben.

Nachbereitung Im Nachgang an die Hauptschleife der Funktion werden alle bekannten und durch das Tool bereits verarbeiteten Finanzbeschlüsse aus der Hilfsdatei eingelesen. Anschließend wird die Beschlussliste nach neuen Finanzbeschlüssen durchsucht, dies geschieht mittels *exportFinancial*-Methode. Sollte nun die *$postData*-Variable auf true

gesetzt sein, so wird der Datenaustausch mittels der *sendData*-Funktion ausgeführt. Danach wird die Hilfsdatei mit den neuen Finanzbeschlüssen neu geschrieben.

copy($fileName, $fn, $check)

Eingabevariablen

$fileName Hier wird der komplette Pfad des zu kopierenden Protokolls angegeben. Die Variable ist als Zeichenkette zu interpretieren.

$fn Hier wird der komplette Pfad zur zu schreibenden Datei angegeben. Die Variable ist als Zeichenkette zu interpretieren.

$check Die Variable gibt an, ob ein Protokoll bereits beschlossen wurde. Falls sie den Wert *true* hat, so wurde das Protokoll bereits beschlossen. Die Variable ist als Wahrheitswert zu interpretieren.

Funktionsinterne Variablen

$OffRec Diese Variable wird nur Funktionsintern benutzt. Wenn sie den Wert *true* hat, so werden die danach kommenden Zeilen übersprungen. Die Variable ist als Wahrheitswert zu interpretieren.

$lines Diese Variable wird nur Funktionsintern benutzt. In dieser Variablen werden alle Zeichenketten gespeichert, die in das öffentliche Protokoll geschrieben werden sollen. Die Variable ist als Zeichenketten-Array zu interpretieren.

Funktionsablauf

Hauptschleife Nach dem Öffnen der Eingabedatei wird die Datei Zeile für Zeile durchgegangen. Sobald hier auf das Start- oder Endtag eines internen Blockes getroffen wird, so werden, bei einem Starttag, die Zeile selbst und alle nachfolgenden mit der Zeile des Endtags entfernt. Das Endtag wird nur nach einem Starttag erkannt.

Nachschleife Vor der zweiten Schleife wird die zu schreibende Datei, welche unter dem in der *$fn*-Variablen angegebenen Pfad befindet, geöffnet. Anschließend wird in die entsprechende Datei alle Zeilen mittels "Für-alle-Schleife"geschrieben.

Funktionsabschluss Am Ende wird noch der Dateiname ohne Pfad, sowie ein HTML-Seitenumbruch und eine EOL ausgegeben.

getDateFromFileName($Filename)

Eingabevariablen

$Filename Hier wird nicht der komplette Pfad einer Protokolldatei angegeben, sondern nur der Dateiname mit Dateiendung. Die Variable ist als Zeichenkette zu interpretieren.

Funktionsinterne Variablen

$Name Diese Variable beinhaltet den in *$Filename* angegebe Zeichenkette ohne die entsprechende Dateiendung. Die Variable ist als Zeichenkette interpretiert und wird nur funktionsintern genutzt.

$d Diese Variable stellt die Tageskomponente eines Datums dar. Sie ist als Zeichenkette zu interpretieren und wird nur funktionsintern genutzt.

$m Diese Variable stellt die Monatskomponente eines Datums dar. Sie ist als Zeichenkette zu interpretieren und wird nur funktionsintern genutzt.

$y Diese Variable stellt die Jahresskomponente eines Datums dar. Sie ist als Zeichenkette zu interpretieren und wird nur funktionsintern genutzt.

$Date Diese Variable ist vom Typ *Date*, welche ein vollständiges Datum repräsentiert. Sie wird nur Funktionsintern und als Rückgabewert der Funktion genutzt.

Funktionsablauf Zuerst wird die Länge des Dateinamens bestimmt. Anschließend wird mithilde der Länge des Dateinamens die Dateiendung aus der unter *$Filename*-Zeichenkette entfernt. Aus dieser gekürzten Zeichenkette wird nun mithilfe der Unterzeichenkettenfunktion die einzelnen Datumsbestandteile herausgetrennt. Abschließend wird eine neue *Date*-Klasse mithilfe des entsprechenden Konstruktors und der *$d*-, *$m*- und *$y*-Variablen erzeugt. Diese erzeugte Klasseninstanz wird nun von der Funktion zurück an den Aufrufer geliefert.

checkApproved($germanDate)

exportFinancial()

formatLine($line, $withToken)

sendData()

readAlreadyKnownFinancialDecissions()

checkAlreadyPostedData($DecissionKey):bool

writeHelperFile()

3.2. Date

3.2.1. Variablen

3.2.2. Funktionen

3.3. File

3.3.1. Variablen

3.3.2. Funktionen