

Protokollcleaner

Documentation

targetingsnake

31. Oktober 2017

Inhaltsverzeichnis

I. Nutzungsanleitung	3
1. Verwendung	3
2. Konfiguration	3
2.1. Eingabepfad	3
2.2. Ausgabepfad	3
2.3. Beschlussliste	4
2.4. Hilfsdatei	4
2.5. Starttag	4
2.6. Endtag	4
2.7. Debug	5
2.8. Nur neue Finanzbeschlüsse ausgeben	5
2.9. Daten an andere Website weitergeben	5
2.10. Webadresse zur Weitergabe der Daten per HTTP-Post	5
2.11. Einstellen des ersten zu verarbeiteten Protokolls	5
3. Konfigurationsdatei erstellen	6
II. Code Dokumentation	6
4. Klassen	6
4.1. Main	6
4.1.1. Variablen	6
4.1.2. Funktionen	8

Teil I.

Nutzungsanleitung

1. Verwendung

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

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

2.1. 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 Dokuwiki-Format 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 offenbar sein.

2.2. 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 Dokuwiki-Format erzeugt. Da-

bei 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 ".txt" angehängen. All dies geschieht automatisch beim Ausführen des Scripts. Der Ordner und dessen Inhalt muss für das Script schreibbar offenbar sein.

2.3. Beschlussliste

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

2.4. 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 Websites zu verhindern, dass diese Daten doppelt erhalten. Diese Datei muss les- und schreibbar zu öffnen sein.

2.5. 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.

2.6. 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.

2.7. 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.

2.8. 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.

2.9. Daten an andere Website weitergeben

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

2.10. 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.

2.11. 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.

3. 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 Webseiten deaktiviert haben, so benötigen sie keinen Pfad zur Weitergabe.

Teil II. Code Dokumentation

4. Klassen

4.1. Main

4.1.1. 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.

\$decisionList 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 festgelegt, ob alle Testausgaben angezeigt werden. Die Variable ist als Wahrheitswert zu interpretieren.

\$onlyNew In dieser öffentlich statischen Variable wird festgelegt, 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 festgelegt, 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 festgelegt, an welche URL die Daten 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 festgelegt, ab welchem Jahr die Protokolle bearbeitet werden sollen. Die Variable ist als Ganzzahl zu interpretieren.

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

\$startday In dieser privaten Variable wird festgelegt, ab welchem Tag, des in *\$startMonth* angegeben Monats, des in *\$startYear* angegebenen 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 Ordner, 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.

4.1.2. Funktionen

__construct()