miLody

**Beheerdershandleiding**

Lysander Vogelzang <Lysander.vogelzang@meertens.knaw.nl>

Meertens Instituut

2012/10/16

Inhoudsopgave

1. Installatie 3

1.1. Over deze handleiding 3

1.2. Bestanden verkrijgen 3

1.3. Bronbestanden collectie klaarmaken 3

1.4. Eigen collectie aanmaken: de dataprovider 4

1.5. Configureren 6

1.5.1. Configureren 6

1.5.2. Apache 7

1.5.3. Het .htaccess-bestand 7

1.5.4. Logmechanisme 8

1.5.5. SMARTY 8

1.5.6. Configureren 8

1.5.7. Foutpagina’s 9

1.5.8. Uploaden van de code 9

2. Gebruik 9

3. Support 10

# Installatie

## Over deze handleiding

Deze handleiding dient als hulp voor het configureren van het miLody raamwerk en alignment server. Met deze software is het mogelijk om collecties van liederen te doorzoeken en te vergelijken op muzikale inhoud.

Ter verduidelijking van deze handleiding wordt gebruik gemaakt van speciale opmaak:

* Engelse, technische termen die niet zijn vertaald zijn weergeven in het lettertype Courier: Engelse computerterm
* Termen uit de code die worden gebruikt, zijn weergegeven in lettertype Courier en bovendien grijs gearceerd: miLody code

## Bestanden verkrijgen

Het installatiepakket bevat twee delen, namelijk de code voor het PHP raamwerk en de code en gecompileerde binaries voor de aligment-server. Deze projecten zijn ondergebracht op respectievelijk <https://sourceforge.net/p/milody/code/> en <https://sourceforge.net/p/milody-as/code/> , waar instructies staan hoe de code via GIT gedownload kan worden.

Het raamwerk in PHP handelt de verzoeken van de gebruiker af die via de webserver worden ontvangen, de alignment-server zorgt voor de vergelijking tussen de verschillende melodieën in de databank of melodieën en muzikale gebruikersinvoer.

Het PHP raamwerk is geschikt voor PHP > 5.2 en is uitvoerig getest in een opstelling met Apache en MySQL op een RedHat 6 distributie van het Linux besturingssysteem.

De bijgevoegde binaries is geschikt voor UNIX gebaseerde systemen. De bijgevoegde binaries zijn gecompileerd voor een x64 versie van RedHat 6. De broncode zou aangepast kunnen worden om ook onder Windows besturingssystemen te werken, een universele oplossing die voor beide besturingssystemen werkt is niet mogelijk.

## Bronbestanden collectie klaarmaken

De alignment-server accepteert JSON-bestanden in het formaat van Peter van Kranenburg als bronbestanden, het zogenaamde NLB-formaat. Deze bronbestanden kunnen uit bestanden met wce of krn notatie worden gegenereerd met behulp van de hulpmiddelen wce2krn en krn2json, beide bijgevoegd in de map /bin/ in het installatiepakket.

De JSON-bestanden dienen in een enkele directory geplaatst te worden, zonder submappen. De naamgeving van de bronbestanden moet uiteraard zorgen voor unieke bestandsnamen en zijn te herleiden naar de records in de (MySQL-)database, zodat er op enig moment een koppeling tussen bestanden en databaserecords gemaakt kan worden. Zie voor de werking hiervan de volgende paragraaf.

## Eigen collectie aanmaken: de dataprovider

Om dit softwarepakket te kunnen gebruiken voor een eigen collectie, moet voor elke collectie die moet worden aangesproken een nieuwe PHP klasse gemaakt worden: de dataprovider. Hierin wordt de koppeling tussen de database en het raamwerk beschreven en bronbestanden. Als voorbeeld is de dataprovider voor Speelmuziek[[1]](#footnote--1) bijgevoegd.

Er zijn verschillende constanten die moeten worden gedefinieerd, die hieronder in een tabel zijn samengevat:

|  |  |
| --- | --- |
| $COLLECTION\_NAME  | De naam van de collectie, waarmee deze is te benaderen in de URL. De URL van de collectie wordt: http://foo.bar/collections/$COLLECTION\_NAME/ |
| $COLLECTION\_DAEMONPIPE  | Het pad naar de PIPE waarop de alignment-server luistert. Dit moet overeenkomen met het tweede argument waarmee de alignment server is gestart. |
| $PATH\_TO\_PRIVATE\_FIFO  | Het pad naar de folder waarin PIPES aangemaakt mogen worden voor de communicatie tussen server en PHP-script. Deze map heeft speciale rechten nodig, zie de volgende paragraaf. |
|  |  |
| $PATH\_TO\_USERFILES  | Het pad waar door gebruikers ingevoerde melodieën worden opgeslagen. Let op de permissies. |
| $PATH\_TO\_NLBFILES  | Het pad naar de JSON-bestanden uit de collectie. Let op de permissies. |
| $PATH\_TO\_TEMPFILES  | Een pad om tijdelijke bestanden op te slaan. Let op de permissies. |
| $PATH\_TO\_TEMPLATES  | Het pad naar de SMARTY-templates |
| $RESULTS\_PER\_PAGE  | Het maximaal aantal resultaten dat per pagina wordt weergegeven. |
|   |  |
| $DB\_HOSTNAME  | Hostname voor de database |
| $DB\_USERNAME | Gebruikersnaam voor toegang tot de database |
| $DB\_PASSWORD  |  Wachtwoord voor toegang tot de database |
|  |  |
| $dataConstants  | Een array van extra waarden die nodig zijn voor modules aan het script, met name de usermethods. Voor de opgeleverde code is de constante c\_waon vereist voor de muzikale gebruikersinvoer van neuriën (hum).  |
|  |  |
| $infoPages  | Een lijst van ondersteunende pagina’s die geen deel uitmaken van de kern van het raamwerk, zoals een help- of colofonpagina. Dit is een associatieve array met als key de naam van het template en als waarde de bestandsnaam van het template. Voor de key “help” is de URL [http://BASE\_URL/collections/$collectionName/info/help](http://BASE_URL/collections/%24collectionName/info/help) |
|  |  |
| $searchParameters  | De geldige waarden waarop gezocht mag worden. Een associatieve array waarbij de key de waarde van het zoekveld is en de waarde altijd leeg. Deze array moet later worden geïnterpreteerd in de functie getTunes(). |
| $sortParameters  | De parameters waarop gesorteerd mag worden. Een associatieve array waarbij de key de naam van de parameter is en de waarde ASC of DESC voor respectievelijke oplopend of aflopend sorteren. Deze waarde moet overeenkomen met de GET parameter “sort”. |
|   |  |
| $daemonAdvancedDefault  | De standaardwaarden voor de aligment instructies gapOpeningScore en gapExtensionsScore. Deze twee waarden moeten als key worden meegeven in de array met als waarde een getal tussen 0 en 100. |

Merk op er ook er voor de klassen van uit is gegaan dat er een verhouding is tussen liederen, strofen en stemmen: een lied kan meerdere strofen hebben, een strofe kan meerdere stemmen hebben. Een stem het kleinste element in de liedabstractie uit dit softwarepakket.

|  |  |
| --- | --- |
| getTuneById ($id) | Een functie waarbij de benodigde velden uit de database worden gehaald voor record met id $id. $id is de parameters die wordt meegegeven in de url in http://foo.bar/collections/$collectionname/tunes/$id .De velden die in de templates worden aangeroepen, moeten worden opgehaald. |
| filterCollection() | Functie waarin het zoeken en filteren wordt afgehandeld. Doel van de functie is benoemen van de velden die nodig zijn voor de resultatenlijst en het construeren van de WHERE parameters. Dit laatste gebeurt aan de hand van de eerder genoemde searchParameters-array. De code is uitgebreid gedocumenteerd.**Opmerking:** Door middel van de functie createWordGroups kan positie-onafhankelijk gezocht worden, zodat de query ‘woordeen<spatie>woordtwee’ hetzelfde is als de query ‘woordtwee<spatie>woordeen’.**Opmerking:** Let er op dat in de SQL-query het SQL\_CALC\_FOUND\_ROWS commando wordt meegegeven, zodat het totaal aantal resultaten zonder paginering bekend is. |
| mapToFilename() | Een functie die de mogelijkheid biedt om de id’s uit de database te herleiden naar de bestandsnamen van de JSON-bestanden in de collectie.**Opmerking:** In veel gevallen is het niet noodzakelijk om deze functie te gebruiken, omdat de bestandsnaam bijvoorbeeld al als veldnaam in de database aanwezig is, of door middel van een SQL concat() on the fly kan worden samengesteld. Deze methoden zijn bijna altijd sneller dan dit handmatig in PHP te coderen. |

## Configureren

## Configureren

Het PHP raamwerk is geschreven met in het achterhoofd zo minimaal mogelijke configuratie. De configuratie voor het gehele raamwerk vindt plaats in index.php.

In dit bestand moet een enkele constante worden gedefinieerd voor de WWW-root van de huidige opstelling. Deze constante wordt gedefinieerd als BASE\_URL. Een voorbeeld is <http://www.liederenbank.nl/temp/>.

Ook is het verplicht om de collecties toe te voegen aan het raamwerk. Dit bestaat uit twee delen. Ten eerste moet de dataprovider per include, require of require\_once toevoegd worden. De aangewezen plek om dit te doen is aan het begin van index.php. Ten tweede moet de klasse worden geregistreerd. Dit moet gebeuren binnen het aanwezig try…catch blok en gebeurt met de volgende opdracht:

 $mRestTron->registerCollection([naam],[klasse]);

Hierbij is [naam] de naam zoals deze collectie wordt aangeroepen in het webadres. Bij de naam ‘speelmuziek’ en BASE\_URL ‘http://www.example.com’ wordt de URL dus: <http://www.example.com/collections/speelmuziek/> . De [klasse] parameters is de naam van de klasse die bij deze collectie hoort.

**Optioneel:** Het raamwerk voorziet in een modulaire opbouw van zogenaamde ‘speciale acties’. Dit zijn acties waarbij de gebruiker op een of andere manier zelf een melodie aanlevert, door bijvoorbeeld een bestand te uploaden, een deuntje te neuriën of zelf iets in te spelen. Voor meer informatie over de vervaardiging van deze speciale klassen, zie de technische handleiding. Speciale acties kunnen worden geregistreerd door middel van de opdracht:

 $mRestTron->registerSpecialAction([actie],[klasse]);

Waarbij [actie] de afkorting is van de actie die wordt meegegeven in het POST verzoek (zie de technische handleiding) en [klasse] de naam van de klasse.

## Apache

De enige vereiste die aan de Apache omgeving wordt gesteld, is dat REWRITE\_MOD is ingeschakeld. Standaard is deze optie ook ingeschakeld. Voor instructies hoe deze optie in te schakelen, consulteer de Apache handleiding voor de gebruikte versie.

## Het .htaccess-bestand

Het .htaccess bestand geeft de Apache webserver regels over het afhandelen van binnenkomende verzoeken. Dit bestand is nodig omdat het miLody raamwerk gebruik maakt van virtuele paden die niet corresponderen met echte bestanden op de harde schijf. Een van de nadelen van deze oplossing is dat sommige paden die wel verwijzen naar echte bestanden op de schijf, niet meer gevonden kunnen worden. Dit kan ook worden opgelost door andere regels. Voor het miLody raamwerk zijn deze regels nodig om geluid mogelijk te maken bij de bijgeleverde module voor de muzikale gebruikersinvoer voor het spelen van de virtuele piano.

Het .htaccess-bestand is een verborgen bestand dat is te vinden in de root van de code. Zorg er dus voor de verborgen bestanden zichtbaar zijn. Het bestand ziet er als volgt uit:

# Soundmanager: Absolute path for soundconfig

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule ^(.\*)/sound-config.xml ***/temp/audio/sound-config.xml*** [R,L]

#Soundmanager: Absolute path for MP3 files

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_URI} !^***/temp/***audio/

RewriteRule ^.\*/(.\*).mp3$ ***/temp/***audio/$1.mp3 [R,L]

# Get REST paths

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_URI} !^/errordocuments/

RewriteRule ^(.\*)$ index.php?apacheparam=$1&%{QUERY\_STRING} [L,QSA]

De vetgedrukte, cursieve paden moeten overeenkomen met de locatie van de geluidsbestanden op de server. Wanneer het raamwerk op een eigen domein is geïnstalleerd, bijvoorbeeld [www.milody.nl](http://www.milody.nl), dan is het pad **/audio/.** Wanneer het raamwerk niet op een eigen domein wordt geïnstalleerd maar in een directory **temp**, dan is het correcte pad **/temp/audio/**, zoals hierboven staat.

## Logmechanisme

Voor elke gegenereerde fout wordt een melding in het logboek weggeschreven. Het logboek wordt gedefinieerd in /frestune/clsLogger.php. Het pad naar het logboek kan worden veranderd , standaard is dit het bestand errors.log. Ook biedt het logboek de mogelijkheid om in het geval van kritieke fouten een e-mail naar de beheerder te sturen. In dat geval moet ook het emailadres in de constante mailFrom worden gewijzigd.

## SMARTY

De uitvoer naar HTML vindt plaats via SMARTY-templates. Omdat SMARTY vaak al beschikbaar is binnen een serveromgeving, is er voor gekozen om deze niet mee te leveren in het installatiepakket. SMARTY is echter simpel te configureren en kan gevonden worden op <http://www.smarty.net/> .

Let wel op het aanpassen van het pad naar SMARTY in index.php.

## Configureren

Het grootste deel van de configuratie is al behandeld bij het aanmaken van een dataprovider omdat in deze klasse de instellingen staan. Let er op dat de paden de correctie permissies hebben:

* /Warehouse/nlb Geen rechten nodig voor PHP, wel voor bin
* /Warehouse/temp Lees- en schrijfrechten
* /Bin Uitvoerrechten
* /Audio Leesrechten
* /templates\_c/ Lees- en schrijfrechten
* /pipes/ Lees- en schrijfrechten

De overige directories met PHP-code moeten uiteraard leesbaar zijn voor Apache, maar hebben geen speciale permissies nodig. Merk op dat bij het gebruik van seLinux ook deze permissies worden aangepast. In het bijzonder is het van belang dat de PIPE beschreven mag worden. Dit kan worden bereikt met de opdracht:

chcon -R -t httpd\_tmpfs\_t PAD\_NAAR\_PIPE

Dit geldt voor de waarde opgegeven in $COLLECTION\_DAEMONPIPE en voor de directory waarin de private pipes worden aangemaakt in $PATH\_TO\_PRIVATE\_FIFO.

Wat betreft het configureren van de database, wordt aangeraden om een aparte gebruiker aan te maken met enkel leesrechten voor de tabellen die worden opgevraagd en de inloggegevens voor deze gebruiker op te geven voor de connectie met de database.

## Foutpagina’s

Optioneel kan er voor gekozen worden voor aanpassing van de errordocuments, de webpagina’s die fouten weergeven. Deze zijn te vinden in de map /errordocuments in het softwarepakket. Wanneer er een kritieke fout optreedt, wordt het correcte errordocument – namelijk het errordocument dat correspondeert met de ondersteuden HTTP codes 400, 401, 500 – aangeroepen met de GET parameters [error] en [url], waarin respectievelijk zijn vermeld de foutmelding en de URL van de pagina waarop de foutmelding plaatsvond.

## Uploaden van de code

Het uploaden van de code en de binaries kan gebeuren met een standaard FTP-programma en is een simpele procedure. Let er op dat ook verborgen bestanden worden gestuurd. Het is van cruciaal belang dat het verborgen .htaccess-bestand in de root van het PHP raamwerk ook wordt gestuurd.

# Gebruik

Nadat de serveromgeving is geconfigureerd rest het starten van de alignment server. Deze is gecompileerd te vinden als /bin/colservd. Aangeraden wordt om de server niet te start als root, vanwege de beveiligingsrisico’s die dat met zich meebrengt. De parameters zijn als volgt:

 colservd *actie[start|stop] locatiePIPE [locatieCollectie]*

Waarin de volgende parameters dienen opgegeven te worden:

* Actie: start of stop.
	+ Start: start de server. LocatiePIPE en locatieCollectie zijn vereist.
	+ Stop: stopt de server. LocatiePIPE is vereist
* LocatiePIPE: de locatie van de PIPE, een bestand
* LocatieCollectie: de locatie van de collectie, een map

**Opmerking.** De actie **stop** is een alias voor het schrijven naar de pipe. Bij het geautomatiseerd aanroepen van deze commando’s kan ook gekozen worden om direct naar de PIPE te schrijven. Zie hiervoor de technische handleiding.

In veel gevallen is het wenselijk om de alignment server buiten een terminal te laten opereren. Hiervoor kan het UNIX commando nohup gebruikt worden:

nohup ./colservd start *locatiePIPE [locatieCollectie]* 2>&1 &

Alle output wordt opgeslagen in het bestand nohup.log in de /bin directory. Met het bovenstaande commando worden ook alle log-meldingen uit de tweede stream in het logboek weggeschreven.

**Opmerking:** Wanneer het commando nohup niet beschikbaar is, kan ook screen gebruikt worden.

# Support

Mochten er nog vragen zijn over het miLody-pakket, dan kunnen deze gericht worden aan Lysander.vogelzang@meertens.knaw.nl of aan lied@meertens.knaw.nl.

1. Live te zien op: http://www.liederenbank.nl/temp/collections/speelmuziek/ [↑](#footnote-ref--1)