

# HLM-NG | Backend User Manual

Oussama Zgheb

29. Mai 2015



Datum	Version	Änderung	Autor
14.04.15	1.0	Initial	Oussama Zgheb
16.04.15	1.1	Weitere Settings erfasst	Oussama Zgheb
29.05.15	1.2	Facebook & Twitter	Oussama Zgheb

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Voraussetzungen</b>	<b>3</b>
2.1	Google Cloud Messaging . . . . .	3
2.2	Facebook . . . . .	3
2.2.1	App erstellen . . . . .	3
2.2.2	Konfiguration . . . . .	3
2.3	Twitter . . . . .	3
2.3.1	App erstellen . . . . .	3
2.3.2	Konfiguration . . . . .	4
2.4	Tomcat . . . . .	4
<b>3</b>	<b>Konfiguration</b>	<b>4</b>
3.1	Ordner erstellen . . . . .	4
3.2	HLMNGSettings . . . . .	4
3.2.1	Pfade . . . . .	4
3.2.2	Login Daten . . . . .	5
3.2.3	Facebook . . . . .	5
3.2.4	Weiteres . . . . .	5
3.3	MySQL Verbindungsdaten . . . . .	6
<b>4</b>	<b>Installation</b>	<b>6</b>
<b>5</b>	<b>Fussnoten Index</b>	<b>7</b>

# 1 Einleitung

Dieses Dokument soll einer mit der Materie vertrauten Person eine Hilfestellung zur Konfiguration und Installation des HLMNG Backend geben.

## 2 Voraussetzungen

Die Software wurde auf unter Ubuntu 12.04 LTS, Ubuntu 14.10 sowie Debian 7 entwickelt und getestet, weitere Betriebssysteme sind höchst wahrscheinlich auch möglich einzusetzen.

Folgende Software muss auf dem Zielgerät installiert sein:

- Tomcat 7
- MySQL 5.6
- Eclipse EE (Kepler oder neuer)

### 2.1 Google Cloud Messaging

Um den Google Cloud Messaging Dienst zu benutzen, braucht man ein Google Konto sowie einen ApiKey. Informationen zur Eröffnung eines Accounts finden sich hier <https://developer.android.com/google/gcm/gs.html>.

Sobald dies erledigt ist kann man unter <https://console.developers.google.com/project> ein neues Project erstellen. Nun muss man unter »APIs & Auth« den Dienst »Google Cloud Messaging for Android« aktivieren. Unter »APIs & Auth -> Credentials« kann ein »Key for server applications« (API Key) erstellt werden.

### 2.2 Facebook

#### 2.2.1 App erstellen

Um auf eine Facebook Seite zu Posten wird eine App benötigt, diese kann unter <https://developers.facebook.com/> erstellt werden. Zuerst muss jedoch ein Facebook Konto bestehen sowie auf Facebook angemeldet sein. Wichtig: Der App Typ muss dabei »Website« sein und die Site URL richtig gesetzt sein, z.B. »<https://fix.confoxy.com/hlmng/frontend/>«. Nun findet man im Dashboard der App folgende Daten: »App ID« und »App Secret«.

#### 2.2.2 Konfiguration

Nun müssen die Parameter der Facebook4J Library <sup>1</sup> hinterlegt werden. Dies geschieht in der Datei facebook4j.properties im Rootverzeichnis »src«. Die Datei sollte also wie folgt aussehen:

```
oauth.appId=*****
oauth.appSecret=*****
oauth.permissions=publish_actions, manage_pages, publish_pages
```

Diese Tokens laufen nie ab, es Bedarf also keiner nachträglichen Änderungen mehr. <sup>2</sup>. Nach Bedarf kann auch »debug=true« hinzugefügt werden, diese liefert wertvolle Informationen auf Sysout.

### 2.3 Twitter

#### 2.3.1 App erstellen

Um die Twitter API zu benutzen muss man unter <https://apps.twitter.com/> ein App erstellen. Dafür wird ein Twitter Account benötigt, bei Bedarf sollte dieser zuerst erstellt und angemeldet sein. Wichtig ist, dass beim erstellen der App das Feld »Website« mit der richtigen URL ausgefüllt wird, also z.B. »<https://fix.confoxy.com/hlmng/frontend/>«. Nach dem erstellen der App sollte man nun unter »Keys and Access Tokens« die benötigten Daten finden.

<sup>1</sup><http://facebook4j.org/>

<sup>2</sup><https://developers.facebook.com/docs/facebook-login/access-tokens>

Folgende Parameter benötigt, »Consumer Key«, »Consumer Secret«, »Access Token«, »Access Token Secret«.

### 2.3.2 Konfiguration

Nun müssen die Parameter der Twitter4J Library <sup>3</sup> hinterlegt werden. Dies geschieht in der Datei twitter4j.properties im Rootverzeichnis »src«. Die Datei sollte also wie folgt aussehen:

```
oauth.consumerKey=*****
oauth.consumerSecret=*****
oauth.accessToken=*****
oauth.accessTokenSecret=*****
```

Diese Tokens laufen nie ab, es Bedarf also keiner nachträglichen Änderungen mehr, ausser das App wird durch den Twitter Account entfernt. <sup>4</sup>. Nach Bedarf kann auch »debug=true« hinzugefügt werden, diese liefert wertvolle Informationen auf Sysout.

## 2.4 Tomcat

Um sicherzustellen, dass genügend Ressourcen vorhanden sind, wird empfohlen den Java Heap Space etwas grösser als die Default Einstellung festzulegen. Dies geschieht im File »/etc/default/tomcat7«. Dort sollte die Zeile »JAVA\_OPTS« wie folgt aussehen:

```
JAVA_OPTS="-Djava.awt.headless=true -Xmx768m -Xms384m -XX:+UseConcMarkSweepGC"
```

Xmx ist dabei die maximale Heap Size, Xms die initiale Grösse. Bei grosser Benutzeranzahl muss die maximale Heap Size eventuell erhöht werden.

## 3 Konfiguration

### 3.1 Ordner erstellen

Das Backend muss Daten auf dem Filesystem ablegen können. Empfohlen ist, die folgenden Ordner unter »/var/lib/hlmng« abzulegen. Es werden die Unterordner »logs«, »media«, und »qr« benötigt. Wichtig ist, dass der User tomcat7 owner ist.

```
chown tomcat7 /var/lib/hlmng/*
```

### 3.2 HLMNGSettings

Unter »src/settings/HLMNGSettings.java« können die Parameter für das Backend vorgenommen werden.

#### 3.2.1 Pfade

- jdbcPath (String) = Der JDBC Pfad zur Datenbank  
z.B. »jdbc:mysql://127.0.0.1/hlmng«, wobei hlmng der Datenbank Schema Name ist
- pubURL = Der Pfad unter welchem der Public Rest API Teil sein soll  
z.B. »/pub«
- admURL = Der Pfad unter welchem der Admin Rest API Teil sein soll  
z.B. »/adm«

<sup>3</sup><http://twitter4j.org/>

<sup>4</sup><https://dev.twitter.com/oauth/overview/faq>

- `qrFileRootDir (String)` = Der Pfad unter welchem alle die gerenderten Qr Codes abgelegt werden  
z.B. »`/var/lib/hlmng/qr/`«
- `mediaFileRootDir (String)` = Der Pfad unter welchem alle hochgeladenen Dateien abgelegt werden  
z.B. »`/var/lib/hlmng/media`«
- `logFileRootDir (String)` = Der Pfad unter welchem die Logfiles abgelegt werden (falls aktiv)  
z.B. »`/var/lib/hlmng/logs`«
- `restAppPath (String)` = Der Pfad unter welchem die Schnittstelle erreichbar sein soll  
z.B. »`https://fix.confosy.com/hlmng/rest`«
- `gcmURL (String)` = Die URL der Google Cloud Messaging API um Nachrichten zu senden  
z.B. »`https://android.googleapis.com/gcm/send`«
- Den Subpfad zur Rest API wird im File »`WebContent/web.xml`« festgelegt:

```
<servlet-mapping>
  <servlet-name>
    Jersey REST Service
  </servlet-name>
  <url-pattern>
    /rest/*      <---- */
  </url-pattern>
</servlet-mapping>
```

### 3.2.2 Login Daten

- `jdbcUser (String)` = Der JDBC User für die Datenbank
- `jdbcPassword (String)` = Das JDBC Passwort für den User
- `apiKey (String)` = Der vom Google erhaltene Key für ihre API's

### 3.2.3 Facebook

- `facebookAppId (String)` = Die Facebook App ID
- `facebookPageId (String)` = Die Facebook Page ID auf welche gepostet werden soll

### 3.2.4 Weiteres

- `cacheTime (int)` = Cache Max-Age in Sekunden
- `qrCodeWidth , qrCodeHeight (int)` = Die Dimension der gerenderten Qr Codes in Pixel
- `maxMediaImageSizeMB (Double)` = Die maximale Dateigröße für die Datei uploads in Megabyte
- `logSysErr (boolean)` = Falls Wahr wird jeglicher Log Output auf `System.Error` umgeleitet anstatt direkt in das Logfile geschrieben zu werden
- `selectLimit (int)` = Die Anzahl von Elementen die bei limitierten Queries geladen werden sollen (z.B. bei `/newest calls`)
- `maxActionsAllowed (int)` = Die Anzahl Aktionen (siehe API Dokumentation -> Spam Schutz) welche in »`actionGraceTime`« durchgeführt werden dürfen
- `actionGraceTime (int)` = Die Aktionen Zeitfenster (in ms)
- `Angestrenztheit (int)` = Die Länge der QR Code Payload in Bit
- `mediaUploadThumbnailPixel (int)` = Die maximale Höhe oder Länge eines Thumbnails

### 3.3 MySQL Verbindungsdaten

Unter »hlmng/WebContent/META-INF/context.xml« werden die Informationen für die MySQL Verbindung festgelegt, siehe »url« sowie »username« und »password«.

```
<Resource name="jdbc/hlmng" auth="Container" type="javax.sql.DataSource"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/hlmng?autoReconnect=true"
username="user" password="password" />
```

## 4 Installation

Die neueste Version kann per GitHub <sup>5</sup> bezogen werden.

Unter Eclipse kann man das Projekt exportieren:

Project Explorer -> Rechtsklick -> Export -> War File

Oder per Commandline

```
cd /path/to/project
jar cvf hlmng.war *
```

Nun kann man das WAR File in den »WebApps« Ordner kopieren und die Anwendung sollte nun gestartet werden.

---

<sup>5</sup><https://github.com/ozzi-/hlmng-app>

## 5 Fussnoten Index

1	<a href="http://facebook4j.org/">http://facebook4j.org/</a> . . . . .	3
2	<a href="https://developers.facebook.com/docs/facebook-login/access-tokens">https://developers.facebook.com/docs/facebook-login/access-tokens</a> . . . . .	3
3	<a href="http://twitter4j.org/">http://twitter4j.org/</a> . . . . .	4
4	<a href="https://dev.twitter.com/oauth/overview/faq">https://dev.twitter.com/oauth/overview/faq</a> . . . . .	4
5	<a href="https://github.com/ozzi-/hlmng-app">https://github.com/ozzi-/hlmng-app</a> . . . . .	6