



Diplomarbeit

Höhere Technische Bundeslehranstalt Leonding
Abteilung für Informatik

HomeDS

Eingereicht von: **Andrej Sakal, 5CHIF**

Felix Hofman, 5CHIF

Datum: **April 4, 2018**

Betreuer: **Thomas Stütz**

Inhaltsverzeichnis

1	Einleitung	2
1.1	Ausgangssituation	2
1.2	Ziele	2
1.3	Problemstellung	2
2	XIBO-Server	3
2.1	Beschreibung	3
2.2	API-Schnittstelle	3
2.3	Authentifizierung	3
2.4	Request-Helper	4
3	XIBO-Grundlagen	6
3.1	Digital Signage	6
3.2	Was ist XIBO?	6
3.3	Weboberfläche des XIBO	6
3.4	Designen mit XIBO	7
4	Summary	9
A	Additional Information	13
B	Individual Goals	14

Kapitel 1

Einleitung

1.1 Ausgangssituation

Die HTL-Leonding besitzt schon einige Multimedia Systeme verstreut im ganzen Schulgebäude um Projekte, aktuelle News und Änderungen im Unterrichtsablauf anzuzeigen. Doch ein großer Schwachpunkt dieser Multimedia Systeme ist, dass der Prozess vom erstellen der Anzeige bis zum zuordnen welcher Bildschirm, welche Information anzeigen soll sehr kompliziert, und mühselig ist. Sodass oftmals neue Informationen erst Verspätet oder gar nicht angezeigt wird.

1.2 Ziele

Ziel ist es, dass es der Schulverwaltung möglich ist Informationen, Warnungen oder Ankündigungen möglichst schnell überall in der Schule anzuzeigen. Die verschiedenen Multimediasysteme sollen einheitlich gesteuert und verwaltet werden können um schnell alle Anzeigen beliebig zu verändern. So ist es auch ein Teilziel festzustellen ob es möglich die derzeitig verwendeten Anzeigesysteme durch den XIBO Server zu ersetzen.

1.3 Problemstellung

Momentan wird um eine Anzeige zu ändern sehr viel Aufwand betrieben, zum Beispiel wird eine neue Präsentation erstellt in Form von Folien oder ein Video zusammengeschnitten Beispiel dafür ist die Anzeige im Eingangsbereich der Schule. Diese Vorgehensweise ist zeitaufwendig und werden Änderungen vorgenommen, kann man die alte Präsentation oder das Video meistens verwerfen.

Kapitel 2

XIBO-Server

2.1 Beschreibung

Als zentrale Steuereinheit wird ein XIBO-Server verwendet. Um diesen verwenden zu können, war es notwendig sich in die Dokumentation einzulesen und die API-Schnittstelle auszuprobieren. Die Website des Servers diente vorerst als Übungsumgebung dadurch wurde es leicht auch die einzelnen Funktionen, inklusive der Vorgangsweise, des Servers zu verstehen.

2.2 API-Schnittstelle

Die API-Schnittstelle des XIBO-Servers ist mittels Swagger Dokumentiert, diese Dokumentation deckt die Grundfunktionalitäten und die Form der Anfragen ab. Da die Schnittstelle des Servers später als wesentliches Verbindungsstück zwischen der eigens entwickelten Steuerungssoftware und dem Server dient, war es nötig diese gründlich zu Testen und diese auch zu verstehen. Anfangs wurde dafür mit Postman gearbeitet. Um mit Postman die Requests testen zu können mussten festgestellt werden welche Codierung für den Request verwendet wird, im Falle des XIBO-Servers wird "application/x-www-form-urlencoded" als Codierung verwendet.

2.3 Authentifizierung

Es stellte sich heraus, dass die Authentifizierung mittels OAuth2 sehr speziell war was zu Beginn zu einigen Schwierigkeiten führte, da es einige Anläufe brauchte um herauszufinden wie die Parameter übergeben werden müssen und in welcher Reihenfolge. Dazu

wurde eine Java-Klasse entwickelt welche die Authentifizierung automatisch übernimmt.
 VERWEIS!!!!!!!!!!!!!!!!!!!!!! <https://oauth.net/2/>

Der Server benötigt zur Authentifizierung mit einem Client eine Client_ID, diese wird vom Server für jeden Client mitgegeben.

Zuerst wird ein Request-Body erstellt dieser hat folgende Parameter in der Form: "client_id=<CLIENT_ID>&client_secret=<CLIENT_SECRET>&grant_type=client_credentials", die im Body mitgegeben werden und als Format 'application/x-www-form-urlencoded' haben. Anschließend werden dem Header noch der content-type mit dem Wert "application/x-www-form-urlencoded" und der Parameter "cache-control" mit dem Wert "no-cache" hinzugefügt. Als ergebniss der Anfrage bekommt der Client einen "access_token", dieser ist nun bei jeder Anfrage von Nöten um sich beim Server zu authentifizieren und es dem Client zu ermöglichen Daten abzurufen beziehungsweise weiterzugeben.

2.4 Request-Helper

Um in weiterer folge die Anfragen an den Digital Signage Server einfach und einheitlich durchzuführen gibt es die Klasse RequestHelper. In dieser Klasse gibt es neben den beiden Parametern responseBody und responseCode, welche zur Fehlerausgabe und zum Erhalt der Daten aus der Anfrage vorhanden sind, auch noch die Methode executeRequest, diese übernimmt die Hauptaufgabe der Klasse und führt die Anfragen an das Signage System durch.

Die Parameter dieser Methode lauten wie folgt:

- *RequestTypeEnum*: Der Parameter vom Typ Enum wird genutzt um herauszufinden welche Http Anfrage vorliegt. Mögliche Werte sind hierbei GET, POST, PUT und DELETE.
- *Params*: Hier liegt eine HashMap vor, die als Key-Value Paare alle benötigten Parameter für den RequestBody beinhaltet. Beispielsweise: LayoutID:"78", hierbei ist LayoutID der Key und "78" das Value.
- *Url*: Beinhaltet die URL unter der die Anfrage erreichbar ist.
- *Token*: Ist jener Parameter der den "access_token", der benötigt wird um sich beim Server zu authentifizieren. Der Erhalt dieses Parameters, funktioniert wie bereits im vorigen Unterpunkt Authentifizierung beschrieben.

Zu beginn der Methode wird anhand des Parameters RequestTypeEnum unterschieden, um welche Http Anfrage es sich handelt. Wird GET oder DELETE geliefert wird durch die HashMap iteriert und die einzelnen Key-Value Paare als QueryParameter in der URL eingefügt. Beispielsweise: <URL>/layout?layoutID=78token=ajdlfjÄÄwkflkd6545. Handelt es sich um eine POST oder PUT Anfrage so werden die Key-Value Paare im Body mitgegeben und im Format application/x-www-form-urlencoded codiert. Anschließend

wird noch die URL mittels `HttpUrl.Builder` erstellt und ausgegeben. Anschließend wird per Switch-Case dem Request die richtige Art der Anfrage zugewiesen und danach wird auch noch die URL übergeben.

Um die Anfragen noch fertig zu stellen wird noch das Interface `Callback` implementiert. Mit den beiden Methoden `onFailure` und `onResponse` wird dem Interface zugewiesen was passiert wenn der Request fehlschlägt oder funktioniert.

Sollte der Request fehlschlagen so wird im Log-Fenster der Responsecode und die Fehlermeldung/Exception ausgegeben. Wird der Request ohne Fehler durchgeführt so wird im Log-Fenster ebenfalls der Responsecode und der Responsebody ausgegeben.

Der letzte Schritt ist es dem `OkHttpClient` mitzuteilen, dass er einen neuen Call ausführen soll, als Parameter wird der zusammengestellte Request mitgegeben. Über `.enqueue` wird dem Client gesagt er soll auf einen Response warten. Parameter für diese Methode ist das erstellte Interface `Callback`.

Als Ergebnis der Anfrage, mit den Parametern, in der Form `String` fragen ob bsp für request usw einbauen

Kapitel 3

XIBO-Grundlagen

3.1 Digital Signage

Digital Signage Systeme haben die Aufgabe viele Bildschirme mit Inhalten zu füllen und eventuell auch diese Inhalte zu designen. Damit soll das zeit- oder interaktions-gesteuerte Ändern von Inhalten auf den Bildschirmen einfach und unübersichtlich gehalten werden. Digital Signage bietet ein breites Spektrum an Anwendungsbereichen. https://de.wikipedia.org/wiki/Digital_Signage Anwendungsbeispiele :2 017.

3.2 Was ist XIBO?

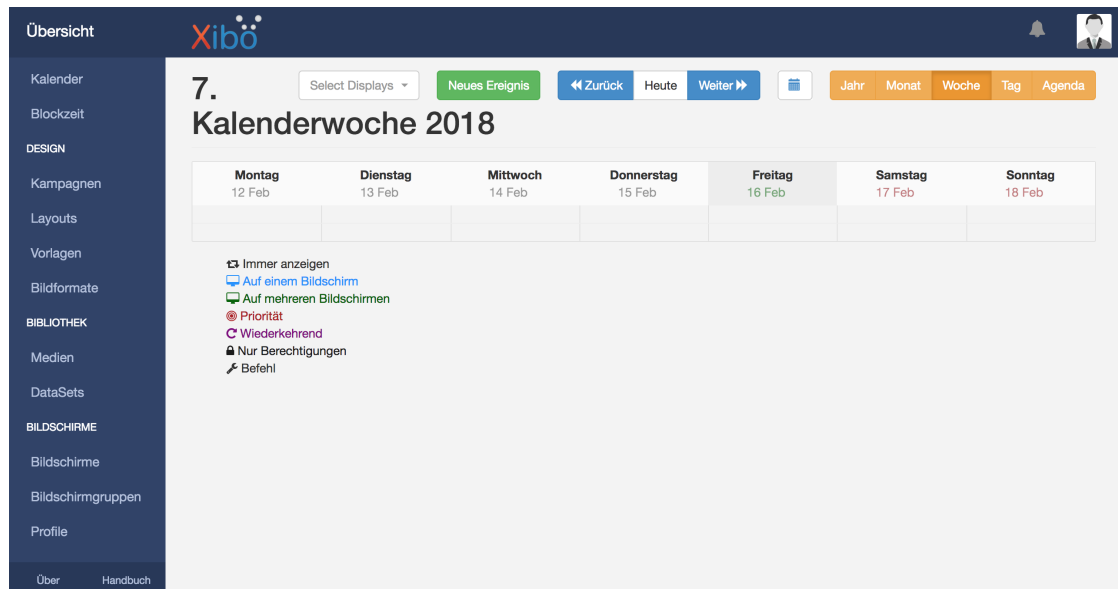
Das XIBO ist ein Open Source Digital Signage System entwickelt von der Spring Signage LTD. Das XIBO-System besteht aus vielen verschiedenen Komponenten. Das XIBO Paket besteht aus einem klassischen Server-Client Konstrukt. Der Server besteht aus 3 Komponenten Content Management System das mithilfe von ZeroMq bei Änderung der Inhalte diese aktualisieren soll, einer Datenbank und einer Weboberfläche die es dem Benutzer ermöglichen soll das System zu bedienen.

SYSTEM ARCH PLAN eventuell noch über zeromq schreiben

3.3 Weboberfläche des XIBO

Das Steuerungszentrum des ganzen Signage System ist die Weboberfläche die ganz einfach über einen Browser unter der Serveradresse aufgerufen werden kann. Auf der Willkommenseite sind die wichtigsten Funktionen dargestellt:

1. *Kalender*: Mit der Kalender Funktion kann eingetragen werden zu welchen Zeitpunkt welcher Inhalt auf welchem Bildschirm angezeigt werden soll. In dem Xibo-Kalender werden auch bereits eingetragene Aktivitäten angezeigt.



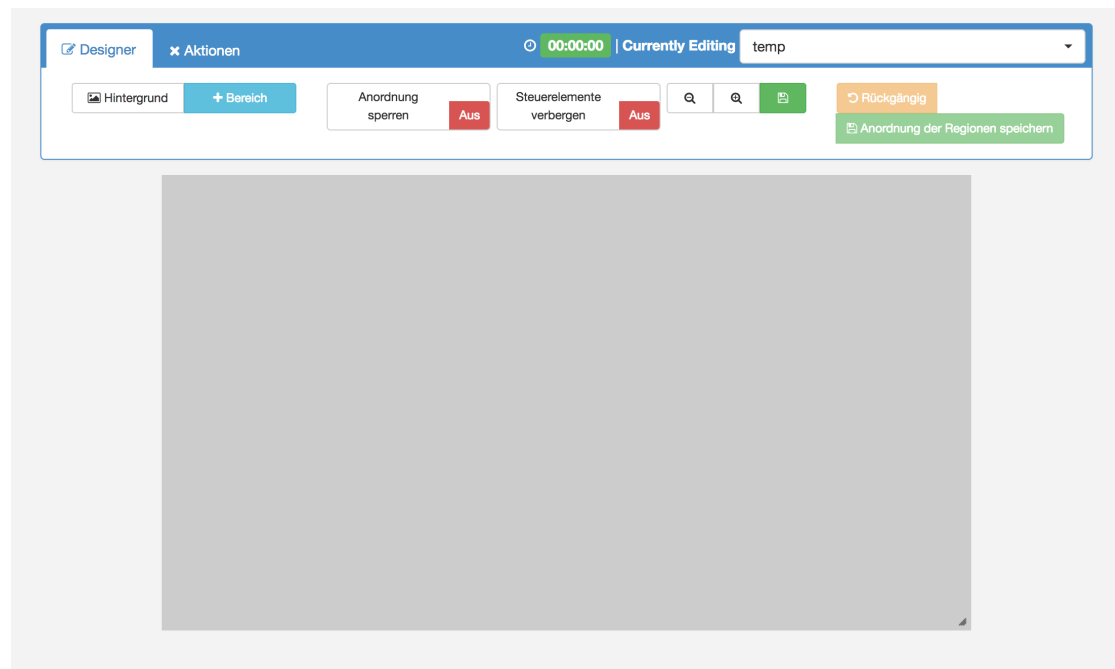
2. *Layouts*: Die Layout Funktion ist einer der wichtigsten Komponenten des Signage Systems es beschäftigt sich mit dem designen der Inhalte. Auf diese Funktion kommen wir noch einmal zurück
3. *Bibliothek*: Die Bibliothek Funktion ist zuständig für das Verwalten der Medien. Hier können Sie verschiedene Dateien hochladen. Diese Medien können dann in Layouts eingebunden und angezeigt werden.
4. *Benutzer*: Im Menüpunkt Benutzer können neue Benutzer angelegt werden und bereits bestehende bearbeitet oder gelöscht werden. Dabei gibt es auch ein Rechtssystem. Es können auch Datenmengen Begrenzungen pro Benutzer eingestellt werden.
5. *Einstellungen*: Der Menüpunkt Einstellung gibt dem Nutzer die Möglichkeit verschiedene Optionen einzustellen. So sind zum Beispiel die richtige Zeitzone, E-Mail Benachrichtigungen wichtige Einstellungen die für ein Einwandfreies funktionieren des Xibo-Servers zuständig wichtig sind.

3.4 Designen mit XIBO

Beim Designen von einem neuen Layout im XIBO muss zuerst die Bildschirm auflösung ausgewählt werden. Und dem Layout ein passender Name zugewiesen werden sowie op-

tional auch eine Beschreibung.

Layout Maske



Dem Layout kann nun eine Region oder auch mehrere hinzugefügt werden eine Region kann wiederum mehrere Widgets enthalten. Mit einem Doppelklick auf die Region kann ein Widget hinzugefügt werden. Es gibt viele verschiedene Arten von Widgets:

Bibliothek: Mit diesem Widget können Dateien aus der Medienbibliothek in die Region angezeigt werden.

Uhr: Dieser Widgettyp bindet eine Uhr in die aus

Bibliothek: Die Bibliothek Funktion ist zuständig für das Verwalten der Medien. Hier können Sie verschiedene Dateien hochladen. Diese Medien können dann in Layouts eingebunden und angezeigt werden.

Benutzer: Im Menüpunkt Benutzer können neue Benutzer angelegt werden und bereits bestehende bearbeitet oder gelöscht werden. Dabei gibt es auch ein Rechte-System. Es können auch Datenmengen Begrenzungen pro Benutzer eingestellt werden.

Einstellungen: Der Menüpunkt Einstellung gibt dem Nutzer die Möglichkeit verschiedene Optionen einzustellen. So sind zum Beispiel die richtige Zeitzone, E-Mail Benachrichtigungen wichtige Einstellungen die für ein Einwandfreies funktionieren des Xibo-Servers zuständig wichtig sind.

Kapitel 4

Summary

Here you give a summary of your results and experiences. You can add also some design alternatives you considered, but kicked out later. Furthermore you might have some ideas how to drive the work you accomplished in further directions.

Abbildungsverzeichnis

Tabellenverzeichnis

Project Log Book

Date	Participants	Todos	Due
------	--------------	-------	-----

Anhang A

Additional Information

If needed the appendix is the place where additional information concerning your thesis goes. Examples could be:

- Source Code
- Test Protocols
- Project Proposal
- Project Plan
- Individual Goals
- ...

Again this has to be aligned with the supervisor.

Anhang B

Individual Goals

This is just another example to show what content could go into the appendix.