

# Barrierefreiheitsdatenbank

---

Christian Schramm

*06. Mai 2014*



BERUFSAKADEMIE SACHSEN Staatliche Studienakademie  
Dresden

Belegarbeit Webprogrammierung

## **Barrierefreiheitsdatenbank**

Christian Schramm

*Gutachter*

**Maksim Gudow**

BERUFSAKADEMIE SACHSEN Staatliche Studienakade-  
mie Dresden

06. Mai 2014

**Christian Schramm**

*Barrierefreiheitsdatenbank*

Belegarbeit Webprogrammierung, 06. Mai 2014

Gutachter: Maksim Gudow

**BERUFSAKADEMIE SACHSEN Staatliche Studienakademie Dresden**

Hans-Grundig-Strasse 25

01307 Dresden

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielstellung und Abgrenzung der Arbeit . . . . .	2
1.3	Untersuchungsgegenstand . . . . .	2
1.3.1	Forschungsfrage . . . . .	2
1.3.2	Hypothesen . . . . .	3
<b>2</b>	<b>Theorie/Begriffsbestimmung</b>	<b>4</b>
2.1	Native Apps . . . . .	4
2.1.1	Vorteile . . . . .	5
2.1.2	Nachteile . . . . .	5
2.2	Web Apps . . . . .	5
2.2.1	Vorteile . . . . .	6
2.2.2	Nachteile . . . . .	6
2.3	Hybridapps . . . . .	6
2.3.1	Vorteile . . . . .	7
2.3.2	Nachteile . . . . .	7
2.4	Wahl der Appvariante . . . . .	7
<b>3</b>	<b>Synchronisation</b>	<b>11</b>
3.1	Datenbanksynchronisation am Beispiel “WebSqlSync” . . . . .	11
3.2	Dateisynchronisation am Beispiel “ownCloud” . . . . .	13
<b>4</b>	<b>Fazit</b>	<b>17</b>
4.1	Auswertung . . . . .	17
4.2	Ausblick . . . . .	18
<b>5</b>	<b>Abkürzungsverzeichnis</b>	<b>21</b>

# Einleitung

” *There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.*

— C.A.R. Hoare

(British computer scientist,  
winner of the 1980 Turing Award)

Das Thema Barrierefreiheit betrifft einen jährlich zunehmend größer werdenden Teil der Bevölkerung. Im Jahr 2011 gab es mit knapp 7,3 Millionen schwerbehinderten Menschen rund 2,6 Prozent mehr als noch im Jahr 2009 [@Sta11]. Dabei leiden zweidrittel der schwerbehinderten Menschen unter körperlichen Behinderungen, die je nach Art und Grad der Behinderung zu sehr starken Einschränkungen im Alltag führen können.

Im Bereich des öffentlichen Lebens wird sehr viel für die Aufhebung der Einschränkungen, um z.B. die Zugänge zu öffentlichen Einrichtungen wie Museen, Ämtern oder öffentlichen Verkehrsmitteln zu erleichtern. In der Verordnung DIN 18040 [@Hyp10] Barrierefreies Bauen sind grundlegende Anforderungen an öffentlich zugängliche Gebäude beschrieben. Die Einführung dieser Verordnung bzw. der einzelnen Punkte in die Technischen Baubestimmungen obliegt jedoch den einzelnen Bundesländern.

Um zu erfassen, welche Einrichtungen die behindertengerechten Anforderungen erfüllen, wurde von der Thüringer Tourismus GmbH in Zusammenarbeit mit dem Dresdner Unternehmen webit! Gesellschaft für neue Medien mbH das Konzept der Barrierefreiheitsdatenbank entwickelt. Sie ermöglicht es alle Daten über eine Einrichtung zu erfassen. Dazu zählen z.B. Zugänge, Treppen, Liftanlagen und vieles mehr. Alle Daten, die von Mitarbeitern des Unternehmens erfasst werden, kommen in einer Datenbank zusammen und sollen behinderten Menschen bei der Planung ihres Urlaubs oder Ausflügen helfen.

## 1.1 Motivation

Die Idee dieser Arbeit entwickelte sich bei der Erfassung der ersten öffentlich begehbaren Einrichtungen. Um Daten erfassen zu können, wird eine Internetverbindung benötigt. Diese kann aber bei vielen Einrichtungen, wie z.B. Kellergewölben, Museen oder Schlössern nicht gewährleistet werden. Die baulichen Gegebenheiten verhindern häufig eine stabile Funkverbindung zwischen dem mobilen Endgerät und dem Server. Daher entstand der

Wunsch die Daten unabhängig von einer Internetverbindung einpflegen zu können. Die aufgenommenen Bilder und Informationen zu den baulichen Gegebenheiten sollen dann mit den vorhandenen Daten auf dem Server abgeglichen und sowohl auf dem Client, als auch auf dem Server ergänzt werden.

## 1.2 Zielstellung und Abgrenzung der Arbeit

Smartphones und Tablets gehören mittlerweile zum Standard der mobilen Kommunikation. Mit diesen kleinen “Alleskönnern” kann man Fotos aufnehmen, im Internet surfen, Dokumente erfassen und vieles mehr, was noch vor ein paar Jahren nur von Desktoprechnern aus möglich war. Das macht die heutige Datenerfassung sehr viel flexibler als noch vor ein paar Jahren.

Ziel der Arbeit ist es eine Möglichkeit zu finden, die mit mobilen Endgeräten erfassen Daten mit den Daten auf einem Server zu synchronisieren. Dabei soll sowohl die Datenbank, als auch Bilddateien betrachtet werden. Als Ausgangspunkt dienen die Vor- und Nachteile, die es bei nativen Apps, Web Apps und einer Hybrid App Variante gibt und wie sich daraus entstehenden Einschränkungen auf eine Synchronisation der Daten auswirken.

In dieser Arbeit liegt der Fokus nicht auf der Programmierung einer nativen App oder Web App. Stattdessen werden Überlegungen zur Umsetzung einer erfolgreichen Synchronisation der Daten angestellt, welche die Wahl einer mobilen Variante der Barrierefreiheitsdatenbank beeinflussen können.

## 1.3 Untersuchungsgegenstand

Im folgenden Abschnitt werden unterschiedliche Methoden der Datensynchronisation betrachtet und die Vor- und Nachteile miteinander verglichen, um für das Kundenprojekt einen Lösungsansatz zu finden. Die Ergebnisse sollen dabei helfen, die nachfolgende Forschungsfrage zu widerlegen oder zu bestätigen.

### 1.3.1 Forschungsfrage

Aus der Zielstellung ergibt sich eine Forschungsfrage, die mit Hilfe der zu erarbeitenden Ergebnisse am Ende dieser Arbeit beantwortet wird.

**Gibt es eine Möglichkeit alle Daten, die mit mobilen Endgeräten offline erfasst werden, bei bestehender Internetverbindung mit dem Server abzugleichen?**

### 1.3.2 Hypothesen

Daraus ergeben sich folgende Hypothesen, die sich bei der Untersuchung der Methoden der Datenbank- und Dateisynchronisation von mobilen Endgeräten mit einem Server als richtig oder falsch herausstellen werden.

#### 1. Hypothese

Es lassen sich alle mit einem mobilen Endgerät erfassten Daten uneingeschränkt per Internet mit dem Server synchronisieren.

#### 2. Hypothese

Die Entwicklung einer nativen App ist die beste Lösung für die Nutzung der Hardwarefunktionen und des Speicherbedarfs der Kombination aus Datenbank und Bildern.

#### 3. Hypothese

Die Umsetzung einer Web App bietet sich aufgrund der bestehenden Website an. Dadurch verringern sich Aufwand und Kosten, ohne Einschränkung der Funktionen.

# Theorie/Begriffsbestimmung

„ *Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.*

— **Linus Torvalds**

(Finnish American, software engineer and hacker)

Zur Gewährleistung des allgemeinen Verständnisses, werden einige grundlegende Begriffe definiert, welche die Grundlage für die folgende Untersuchung bilden.

Die Entscheidung, welche mobile Variante für die Umsetzung verwendet werden soll, ist abhängig von den jeweiligen Eigenschaften der Anwendungen:

- Wieviel Speicher steht der App zur Verfügung?
- Welche Handyfunktionen werden benötigt?
- Lässt sich feststellen wann eine Internetverbindung besteht?
- Lassen sich die Daten, abhängig von einer bestehenden Internetverbindung, mit dem Server abgleichen.

Die nachfolgenden Vor- und Nachteile von Nativen Apps, Web Apps und Hybrid Apps sind auf der Website [www.app-entwickler-verzeichnis.de](http://www.app-entwickler-verzeichnis.de) beschrieben.[@App14]

## 2.1 Native Apps

Die Nativen Apps werden speziell für das jeweilige Betriebssystem entwickelt z.B. iOS oder Android. Diese laufen dann auch ausschließlich auf iOS Geräten wie dem iPhone und dem iPad, oder Android Geräten wie dem Samsung Galaxy S4.

Dadurch wird eine optimale Nutzung der Ressourcen und einheitlich funktionierende Hardware-schnittstellen sichergestellt.



### 2.1.1 Vorteile

- Native Apps nutzen die Leistung des Betriebssystems und des verwendeten Gerätes voll aus, da sie speziell für das Betriebssystem angepasst sind. Dadurch lassen sich sehr gut komplexere und rechenintensivere Apps umsetzen.
- Durch die Installation der Apps auf dem Endgerät können Hardwarefunktionen wie Kamera, Beschleunigungssensor oder GPS<sup>1</sup> benutzt werden. Das ist in der Regel nur nativen Apps vorbehalten.
- Daten können auf dem Endgerät in beliebiger Menge gespeichert werden.
- Da Native Apps über einen Appstore vertrieben werden, wirken sich positive Bewertungen stark auf den Verkauf aus.
- Die App lässt sich sehr einfach über den Appstore installieren. Nach der Installation wird automatisch ein Icon zum Starten auf dem Homescreen angelegt.
- Der Vertriebsaufwand ist sehr gering, da die Appstores verbreitete Bezugsquellen für Native Apps sind. Ist die App erfolgreich, findet sich diese in den Top-Listen der App Stores wieder und erreicht dadurch sehr hohe Downloadzahlen.

### 2.1.2 Nachteile

- Ein großer Nachteil ist der erforderliche Entwicklungsaufwand, wenn die App in allen Appstores angeboten werden soll, da die App an die jeweiligen Gegebenheiten des Betriebssystems optimiert werden muss.
- Es entstehen zusätzliche Kosten, um die App für den entsprechenden Appstore entwickeln und anbieten zu können.

## 2.2 Web Apps

Web Apps sind im eigentlichen Sinne speziell programmierte HTML5<sup>2</sup> Websites, die automatisch erkennen auf welchem Endgerät sie aufgerufen werden und optimieren den Inhalt entsprechend automatisch. Somit kann jedes mobile Endgerät, welches über einen Webbrowser verfügt, die App nutzen.

---

<sup>1</sup>Global Positioning System

<sup>2</sup>Hypertext Markup Language Version 5

### 2.2.1 Vorteile

- Web Apps sind quasi unabhängig vom Betriebssystem und funktionieren auf allen Smartphones. Dadurch werden mehr potentielle Nutzer bei gleichzeitig geringeren Kosten erreicht.
- In der Regel ist die Entwicklung einer Web App günstiger, als die Entwicklung einer nativen App für nur ein Betriebssystem.
- Durch die Verwendung von HTML5 wird auch die Offline-Speicherung von Daten ermöglicht. Somit kann nach erstmaligem laden die App auch ohne permanente Internetverbindung genutzt werden.
- Über Onlinesuchmaschinen wie beispielsweise Google können Web Apps ohne großen Aufwand gefunden und ohne Installation direkt genutzt werden. Werden diese als Lesezeichen gespeichert, lässt die Web App sich genau wie eine Native App vom Startbildschirm aus starten.
- Die Veröffentlichung und Aktualisierung einer Web App erfolgt in Sekundenschnelle, da sie im Gegensatz zu Nativen Apps keinen Zulassungsprozess durchlaufen muss.
- Vertriebt man die App selbst, entfällt die Provision von üblicherweise 30 Prozent an den Betreiber des App Stores.

### 2.2.2 Nachteile

- Viele Hardwarefunktionen wie beispielsweise die Kamera oder das GPS der mobilen Geräte lassen sich gar nicht oder nur mit spezieller Zustimmung des Nutzers verwenden.
- Komplexe Berechnungen wie beispielsweise 3D Darstellungen, Verschlüsselungen oder Bildbearbeitungen sind mit einer Web App nicht möglich.
- Benötigt die App mehr als 10MB an Datenmaterial auf dem Endgerät, ist von einer Entwicklung als reine Web App abzusehen.
- Geschäftsmodelle, die auf In-App-Käufe oder einem App Store aufbauen, funktionieren zusammen mit der Web App nicht.

## 2.3 Hybridapps

Hybridapps haben das Ziel, die Vorteile der Web App Entwicklung und der Entwicklung nativer Apps in sich zu vereinen. Dabei setzen die Entwickler auf eine große Anzahl von Frameworks. PhoneGap, Corona oder Appelerator Titanium sind Beispiele für Frameworks, mit deren Hilfe Web Apps in eine native App umgewandelt werden kann.

Hybrid Apps besitzen jedoch auch Vor- und Nachteile.

### 2.3.1 Vorteile

- Durch die Verwendung einer Hybrid App lässt sich eine Cross Browser Web App erstellen, die in allen modernen Browsern läuft.
- Da eine Web App mittels Frameworks für verschiedene Betriebssysteme umgewandelt werden kann, bleibt die eigenständige Entwicklung für jedes einzelne Betriebssystem erspart. Es bleiben im schlimmsten Fall nur betriebssystemspezifische Feinheiten, die noch angepasst werden müssen.
- Mit Javascript lassen sich viele Hardwarefunktionen der mobilen Endgeräte nutzen, auf die man bei einer Web App nicht zugreifen konnte.
- Der Verkauf einer Hybrid App erfolgt über den jeweiligen App Store.

### 2.3.2 Nachteile

- Ein großer Nachteil einer Hybrid App kann entsteht, wenn sehr rechenintensive Anwendungen verwendet werden. Dadurch erreichen Hybrid Apps sehr schnell ihr Leistungsmaximum reagieren träge. Dies ist sehr stark vom verwendeten Framework abhängig. Dieser Nachteil kann in Zukunft durch merklich effizienter werdende Frameworks behoben werden.
- Die Programmierung einer Hybrid App kann mit zunehmendem Komplexitätsgrad sehr aufwendig werden und dadurch eine Umsetzung mittels nativer App empfehlenswerter machen.

## 2.4 Wahl der Appvariante

Die Wahl der Appvariante ist nicht allein abhängig von den Vor- und Nachteilen der einzelnen Varianten. Hierfür muss auch die bestehende Website "<http://www.thueringentourismus.de/barrierefrei>" betrachtet und die Umsetzungsaufwände gegeneinander abgewägt werden.

Ein nicht zu unterschätzender Vorteil der Website ist die bereits responsive Umsetzung. (Abb.2.1-2.3) Mit dieser passt sich der dargestellte Inhalt an die Auflösung des Endgeräts an, wodurch die optimale Darstellung auch auf mobilen Geräten gewährleistet wird. Es können somit bereits neue Daten für die Barrierefreiheitsdatenbank mobil erfasst werden.

Thüringen entdecken.de Einrichtungen Administration Einrichtungen suchen ... hantschack@webit.de

Erfolgreich angemeldet

## Einrichtungen

← zurück 1 2 weiter →

Bezeichnung	Buchungscode	Adresse	Formulare	Letzte Änderung
Hotel "Zur Erholung"	EIC46009	Brochthäuserstr. 65 37115 Duderstadt	7	vor 4 Tagen
Ferienhaus Altes Schlachthaus		Schmiedfelder Straße 6 98711 Vesser	15	vor 8 Tagen
co.med - Das Gesundheitshotel ****	SLF31539	Rainweg 68 b 07318 Saalfeld	33	vor 8 Tagen
City-Apartments Mühlhausen	MHL45666	Untermarkt 12 99974 Mühlhausen / Th	15	vor 8 Tagen
Hotel Thüringen Suhl	CCS62010	Platz der Deutschen Einheit 2 98527 Suhl	27	vor 10 Tagen
Hotel an der Therme - Haus 3	WMR47284	Rudolf-Gröschner-Str. 11 99518 Bad Sulza	26	vor 17 Tagen
Hotel Goldener Hirsch	CCS62005	An der Hasel 91-93 98527 Suhl-Neundorf	12	vor 22 Tagen
Ringberg Hotel	CCS62007	Ringberg 10 98527 Suhl	74	vor 23 Tagen
egapark			0	vor 23 Tagen
Bio - Seehotel Zeulenroda	GRZ33026	Rudolf-Gröschner-Str. 11 99518 Bad Sulza	31	vor 25 Tagen

← zurück 1 2 weiter →

+ Neue Einrichtung anlegen

Abb. 2.1: Die Abbildung zeigt den Aufbau der Website <http://www.thueringen-tourismus.de/barrierefrei> auf der Daten für die Barrierefreiheitsdatenbanke eingepflegt und abgerufen werden können.

Thüringen entdecken.de Einrichtungen Administration Einrichtungen suchen ... hantschack@webit.de

## Neue Einrichtung anlegen

Bezeichnung

TomasID

OpenTextID

Buchungscode

Beschreibung

Art der Einrichtung

- ☐ Beherbergungsbetrieb
- ☐ Gastronomie
- ☐ Bahnhof
- ☐ Shopping
- ☐ Freizeiteinrichtung

Zertifikate

Speichern Abbrechen

Abb. 2.2: In der Abbildung ist ein Standardformular zu sehen, welches grundlegende Eigenschaften für die Datenerfassung einer Einrichtung erhält. Wie beispielsweise die Art der Einrichtung.

## Einrichtungen

← zurück 1 2 weiter →

Bezeichnung	Buchung
Hotel "Zur Erholung"	EIC4600
Ferienhaus Altes Schlachthaus	
co.med - Das Gesundheitshotel ****	SLF315
City-Apartments Mühlhausen	MHL456
Hotel Thüringen Suhl	CCS620
Hotel an der Therme - Haus 3	WMR470
Hotel Goldener Hirsch	CCS620
Ringberg Hotel	CCS620
egapark	
Bio - Seehotel Zeulenroda	GRZ330

← zurück 1 2 weiter →

+ Neue Einrichtung anlegen

## Neue Einrichtung anlegen

### Bezeichnung

Bitte eintragen

### TomasID

Bitte eintragen, falls vorhanden

### OpenTextID

Bitte eintragen, falls vorhanden

### Buchungscode

Bitte eintragen, falls vorhanden

### Beschreibung

Tragen Sie hier auf Wunsch eine Beschreibung ein

### Art der Einrichtung

- ☐ Beherbergungsbetrieb
- ☐ Gastronomie
- ☐ Bahnhof
- ☐ Shopping
- ☐ Freizeiteinrichtung

### Zertifikate

Bitte wählen Sie die passenden Zertifikate :

Speichern Abbrechen

**Abb. 2.3:** Auf den beiden oberen Bildern ist sowohl die Ausgabe der erfassten Daten, als auch die Seite zur Datenerfassung auf einem mobilen Endgerät zu sehen. Navigation und Inhaltselemente passen sich an die Auflösung des Endgeräts an.

Nach Betrachtung der Vor- und Nachteile der drei App Varianten und der bestehenden Website, bietet sich die Umsetzung einer reinen nativen App für die Datenerfassung der Barrierefreiheitsdatenbank nicht an. Der Umsetzungsaufwand und die entstehenden Kosten wären unangemessen hoch im Vergleich zu den daraus entstehenden Vorteilen.

Die responsive Variante der Website bietet sich auch nicht an, da sie eine permanente Internetverbindung voraussetzt.

Da eine Web App auf HTML5 aufbaut und die Website sich bereits teilweise an das Endgerät anpasst, baut die Untersuchung im zweiten Teil der Arbeit auf eine Umsetzung als Web App auf. Daher wird bei der Synchronisation der Daten zur Datenbank auf HTML5 und Javascript gesetzt.

Eine Hybrid App baut auf einer Web App auf und bietet sich daher für eine Weiterentwicklung der Lösung an. Dies kann aber zu einem späteren Zeitpunkt erfolgen und wird daher vorerst nicht betrachtet.

# Synchronisation

“ *The most important property of a program is whether it accomplishes the intention of its user.*

— C.A.R. Hoare

(British computer scientist,  
winner of the 1980 Turing Award)

Synchronisation beschreibt den Datenaustausch zwischen einem Sender und einem Empfänger. Dabei werden die Daten in Blöcke aufgeteilt und in einen Übertragungsrahmen eingepasst. Um die Daten aneinander anzugleichen muss festgestellt werden, welches Endgerät welche Daten besitzt und muss kontrolliert werden ob das andere Gerät eine Anforderung für diese Daten besitzt.

Besitzen beide Endgeräte dieselben Daten z.B. in unterschiedlichen Versionen, muss definiert werden, wie mit den Änderungen umgegangen werden soll.[@Ope13]

Aufgrund der Begrenzung des lokalen Gerätespeichers empfiehlt sich die Synchronisation auf zwei separaten Wegen durchzuführen. Als Erstes wird die Datenbank synchronisiert und als Zweites alle Anhänge, wie z.B. Bilder oder Dokumente. So kann das Problem der Speicherbegrenzung bei Web Apps umgangen werden.

## 3.1 Datenbanksynchronisation am Beispiel “WebSqlSync”

WebSqlSync<sup>1</sup> ist eine Javascript Bibliothek zur automatischen Synchronisation einer lokalen WebSql Datenbank mit dem Server. Die Synchronisation erfolgt dabei in beide Richtungen und arbeitet auf dem Prinzip der inkrementellen Synchronisation, was bedeutet, dass nur erforderliche Daten übertragen werden.

WebSqlSync funktioniert auch ohne Internetverbindung. Alle Änderungen der Daten werden dabei verfolgt und mit dem Server abgeglichen, sobald wieder eine Internetverbindung besteht. Es wird auch die Änderung der Daten auf mehreren Geräten unterstützt.

---

<sup>1</sup><http://www.verious.com/code/orbitaloop/WebSqlSync/>

Die Unterstützung von Webapp und der Phonegap App<sup>2</sup> für mobile Betriebssysteme wie z.B. iOS und Android ermöglicht eine einfache Integration ohne den Programmcode anpassen zu müssen.[@orb13]

### Installation und Initialisierung

Um WebSqlSync nutzen zu können, muss nur die Datei webSqlSync.js im Head-Bereich im HTML<sup>3</sup> des Projekts hinzugefügt werden.

```
<script src="lib/webSqlSync.js" type="application/x-javascript" charset="utf-8"></script>
```

Bei Aufruf der Bibliothek, werden automatisch zwei Datenbanktabellen erstellt, falls diese nicht bereits durch einen vorherigen Aufruf existieren. Die erste Tabelle **new\_elem** speichert alle neuen bzw. geänderten Elemente, die zweite Tabelle **sync\_info** das Datum der letzten Synchronisation.

Zusätzlich werden sogenannte SQLite Auslöser erstellt, die überwachen, ob Änderungen per **INSERT** oder **UPDATE** an den Tabellen vorgenommen wurden. SQLite ist eine einfache Datenbankbibliothek, die Befehle der Sprache SQL<sup>4</sup> verwendet.

Geänderte Elemente werden somit automatisch in der Tabelle **new\_elem** eingefügt. (Abb.3.1)

```
DBSYNC.initSync(  
    TABLES_TO_SYNC, webSqlDb, sync_info,  
    'http://www.myserver.com', callBackEndInit  
);
```

**Abb. 3.1:** Codebeispiel für den Aufruf zur Datenbanksynchronisation

Die Tabellen die, mit dem Server synchronisiert werden sollen, werden in der Funktion **TABLES\_TO\_SYNC** angegeben. (Abb.3.2)

```
TABLES_TO_SYNC = [  
    {tableName : 'table1', idName : 'the_id'},  
    {tableName : 'table2'}  
    //if idName not specified, it will assume that it's "id"  
];
```

**Abb. 3.2:** Codebeispiel für die Datenbanktabellen die synchronisiert werden sollen

In der Tabelle **sync\_info** können alle Informationen gespeichert werden, die der Entwickler als nützlich empfindet, beispielsweise die Identifikation des Clients, da Sie mit an den Server

---

<sup>2</sup><http://phonegap.com/>

<sup>3</sup>Hypertext Markup Language

<sup>4</sup>Structured Query Language



gesendet wird. Dafür kann jegliche Information genutzt werden, wie z.B. die Emailadresse, ein Login oder auch eine entsprechende ID<sup>5</sup> des genutzten mobilen Endgeräts.

### Aufruf

Um die Synchronisation zu starten wird die Funktion **syncNow** aufgerufen. Die Synchronisation erfolgt dabei nach einer freiwählbaren Zeitspanne oder aber nach einer festgelegten Anzahl von Datenänderungen. (Abb.3.3)

```
DBSYNC.syncNow(callbackSyncProgress, function(result) {
    if (result.syncOK === true) {
        //Synchronized successfully
    }
});
```

**Abb. 3.3:** Codebeispiel für den Aufruf zur Synchronisation

Bei größeren Datenmengen ist es für den Nutzer hilfreich, wenn dieser eine Fortschrittsanzeige erhält. Während der Synchronisation wird dafür bei jedem Einzelschritt, beispielsweise einzelne Datenpakete, die Funktion **callbackSyncProgress** aufgerufen. (Abb.3.4)

```
callbackSyncProgress: function(message, percent, msgKey) {
    $('#uiProgress').html(message + ' (' + percent + '%)');
},
```

**Abb. 3.4:** Codebeispiel für den Callback zur Erstellung einer Fortschrittsanzeige

### Einschränkungen

Die Bibliothek WebSqlSync besitzt auch ein paar wenige Einschränkungen. Zum Beispiel wird der SQL-Befehl **DELETE** nicht unterstützt. Stattdessen sollte dies mit einem update an der entsprechenden Stelle umgangen werden.

## 3.2 Dateisynchronisation am Beispiel “ownCloud”

ownCloud<sup>6</sup> ist eine Open Source Software zur Einrichtung einer unabhängigen serverseitigen Datenspeicherlösung. Sogenannte Cloudspeicher gibt es mittlerweile sehr viele. Die wohl bekanntesten sind Dropbox, Google Drive und OneDrive. Sie ermöglichen einen einfachen Datenzugriff von überall auf der Welt. Dafür lädt man seine Dateien über eine Internetverbindung auf einen speziellen Datenserver. Leider besitzt dies den großen Nachteil, dass keine 100 prozentige Sicherheit besteht, was mit den eigenen Daten passiert.

<sup>5</sup>Ausweis, Kennnummer

<sup>6</sup><https://owncloud.com/>

Um dies sicherzustellen bietet ownCloud die Möglichkeit einen eigenen Cloudserver<sup>3.5</sup> zu erstellen, auf dem der Nutzer alles selbst konfigurieren kann. Per Nutzer- und Rechteverwaltung lässt sich wie auf einem normalen Server festlegen, welcher Nutzer Dateien verändern kann.

Für die Daten werden Backup Lösungen angeboten, so dass jederzeit Sicherungen der Dateien angelegt werden können. Der wohl wichtigste Punkt ist jedoch die Verschlüsselung der gespeicherten Daten mittels SSL<sup>7</sup>. Dadurch können auch unternehmensrelevante Daten sicher gespeichert werden.

Durch die Open Source Lösung ownCloud lässt sich zu den bereits vorhandenen Plugins neue Software entwickeln und den eigenen Bedürfnissen anpassen. Da bereits Apps für die mobilen Betriebssysteme (Abb.3.6) zu Verfügung stehen, lassen sich relativ einfach Daten zwischen dem Server und dem Client synchronisieren.

Neben der frei verfügbaren “Community Edition” stehen für die volle Unterstützung der ownCloud Entwickler, welche bei Fragen zur Software oder den Apps zur Verfügung stehen, auch eine “Business” und eine “Enterprise Version” zur Verfügung, welche gegen eine jährliche Gebühr angeboten wird.[@Fra14]



**Abb. 3.5:** Architektur ownCloud

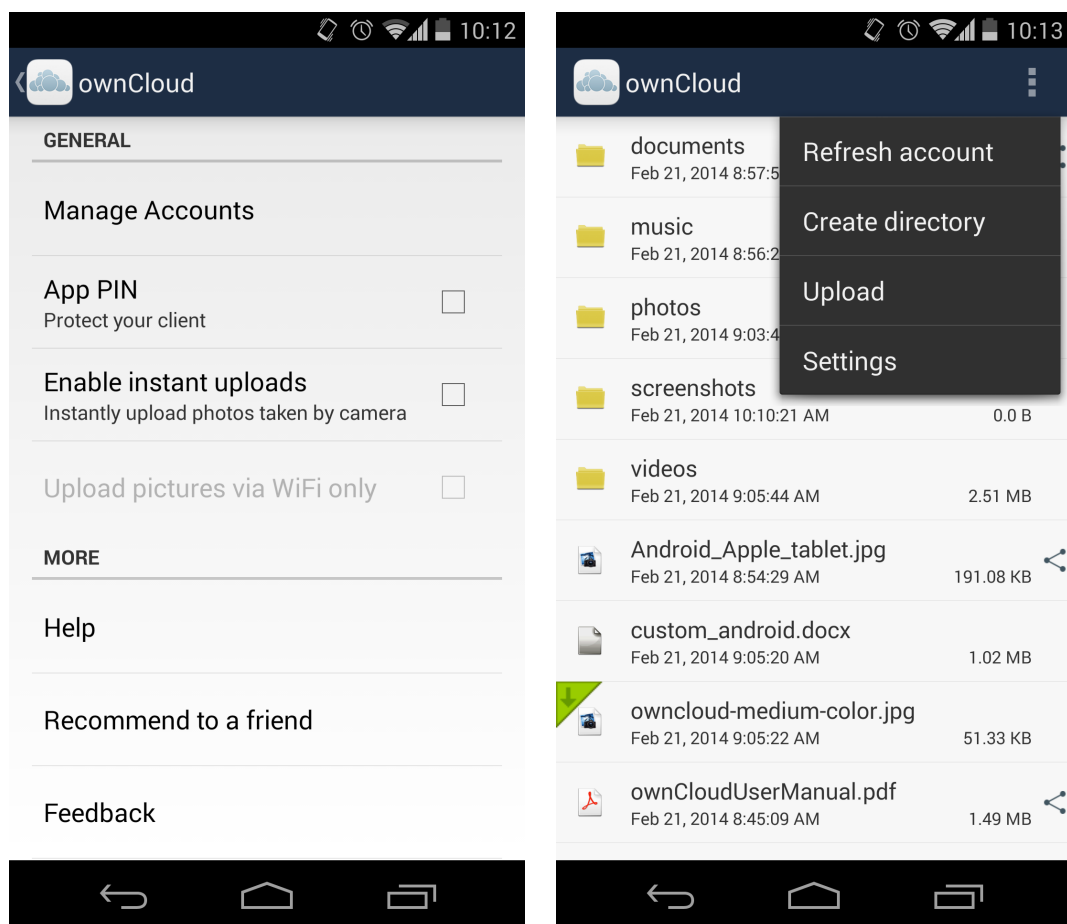
### Vorteile der Android/iOS App von ownCloud

Die nachfolgenden Vor- und Nachteile der ownCloud App sind auf der Website <https://owncloud.com/> beschrieben.[@Fra14]

---

<sup>7</sup>engl. Secure Sockets Layer

- SSL- und HTTP-Verbindungen werden automatisch erkannt, so dass eine einfache und gesicherte Verbindung zu dem Server möglich wird.
- Dateien und Ordner auf dem Server können neu angelegt, durchsucht, umbenannt und gelöscht werden, je nachdem wie die Rechte an den Nutzer vergeben sind.
- Durch das Anlegen von Favoriten lassen sich Daten wie Dokumente, Bilder und Videos automatisch mit dem Server synchronisieren.
- Dateien können auf das mobile Endgerät heruntergeladen werden, um diese Offline nutzen zu können.
- Es werden durch die App mehrere ownCloud Accounts mit einem Device und zusätzlich die Anbindung an mehrere ownCloud Server unterstützt.



**Abb. 3.6:** ownCloud App

Unabhängig von der Anzahl der Nutzer lassen sich mit dieser Cloudlösung auch die für die Datenbank aufgenommenen Bilder unkompliziert speichern. Egal ob ein Internetzugang zur

Verfügung steht oder man Fotos offline aufgenommen und in einem Ordner von ownCloud ablegt werden, synchronisiert ownCloud diese automatisch sobald eine Internetverbindung zur Verfügung steht.

Einziger Nachteil ist dabei der hohe Kostenfaktor für die “Enterprise” Lösung, den man mit dem Kunden abstimmen muss. In Ausblick auf die Erweiterbarkeit durch die Open Source API<sup>8</sup> lässt sich die Software beliebig erweitern und kann somit den Nutzen-Kosten-Faktor verringern.

---

<sup>8</sup>Application-Programming-Interface

# Fazit

Zu Beginn der Arbeit werden die zur Verfügung stehenden Varianten der Umsetzung der Barrierefreiheitsdatenbank auf mobilen Endgeräten betrachtet, da sich diese stark auf die Möglichkeiten der Datenspeicherung und der Synchronisation auswirken. Auch die bereits bestehende Website wird für die Wahl der besten Variante herangezogen und beeinflusst maßgeblich die Wahl der Synchronisation.

Eine kurze Erläuterung der Funktionen der beiden Lösungsansätze soll vermitteln, wie diese für die Barrierefreiheitsdatenbank eingesetzt werden kann und welche Kombinationsmöglichkeiten es gibt.

Die Programmbibliothek WebSqlSync zur Datenbanksynchronisation und die Dateispeicherlösung ownCloud sind nur zwei Beispiele, wie die Synchronisation zwischen einem mobilen Endgerät und einem Server gewährleistet werden kann. Mit der Umsetzung einer Web App und der Möglichkeit diese mit Hilfe eines zusätzlichen Javascript-Frameworks in eine Hybrid App umwandeln zu können, lassen sich beide Lösungen mit etwas mehr Aufwand zu einer Anwendung zusammenführen.

## 4.1 Auswertung

Nach Betrachtung der beiden Beispiele lässt sich sagen, dass Hypothese eins[1.3.2] und drei[1.3.2] sich bewahrheitet haben. Durch die Trennung von Datenbanksynchronisation und Dateisynchronisation lassen sich Beschränkungen, die bei einer Web App an den lokalen Speicher der mobilen Endgeräte bestehen umgehen. Eine Web App bietet sich als Plattform für mobile Geräte an, da sie auf das bereits bestehende Grundgerüst der Barrierefreiheitsdatenbank aufbaut. Gleichzeitig erspart sich so viel Aufwand was zur Folge hat, dass sich anfallende Umsetzungskosten verringern, im Gegensatz zu der Entwicklung einer nativen App.

Beide Lösungen bieten die Möglichkeit auch ohne Internetzugang weiter Daten einpflegen zu können und erst bei bestehender Internetverbindung diese zu synchronisieren. Auch bei einem erneuten Verbindungsabbruch bleiben die Daten erhalten und werden später einfach erneut synchronisiert.

Hypothese zwei[1.3.2] hat sich nach der Betrachtung als eine weniger optimale Lösung herausgestellt. Der Aufwand und die Kosten stehen in keinem Verhältnis zum Vorteil des erhöhten Speichergewinns und der Kombination aus Datei- und Datenbankspeichers.

## 4.2 Ausblick

Im Ausblick auf die Weiterentwicklung der Barrierefreiheitsdatenbank und einer größer werdenden Zielgruppe, bietet der Ansatz der getrennten Synchronisationswege einige Vorteile. Es lassen sich beliebig Nutzer ergänzen, die neue Daten einpflegen können. Der Speicherbedarf für die Datenbank ist begrenzt, da nur Text in der Datenbank gespeichert wird. Bilder werden in einem separaten Ordner im Endgerät gespeichert, dessen Speicherplatz nur durch die Größe der Speicherkarte des Endgeräts begrenzt wird.

Eine Weiterentwicklung des Lösungsansatzes kann die Umwandlung der Web App in eine Hybrid App sein. Dadurch ergibt sich die Möglichkeit die Cloudfunktionen und die Datenbanksynchronisation in einer App zusammenzuführen und zusätzlich Hardwarefunktionen wie z.B. die Kamera zu integrieren. Der erweiterte Speicher ist für die damit in Verbindung stehende Projektkomplexität von Vorteil.

Der nächste Schritt ist die Erstellung einer Prototyp Anwendung, um die Funktionen am praktischen Beispiel zu testen.

## Webseiten

- [@App14] App Entwickler Verzeichnis. *Native App vs. Web App*. 2014. URL: <http://www.app-entwickler-verzeichnis.de/faq-app-entwicklung/11-definitionen/107-unterschiede-und-vergleich-native-apps-vs-web-apps> (besucht am 2. Mai 2014) (zitiert auf Seite 4).
- [@Fra14] Frank Karlitschek. *ownCloud*. 2014. URL: <http://owncloud.org/> (besucht am 5. Mai 2014) (zitiert auf Seite 14).
- [@Hyp10] HyperJoint GmbH. *DIN-18040*. 2010. URL: <http://nullbarriere.de/din18040-1.htm> (zitiert auf Seite 1).
- [@Ope13] Open Mobile Alliance. *SyncML Spezifikationen*. 2013. URL: [http://technical.openmobilealliance.org/Technical/release\\_program/SyncML\\_v1\\_2\\_2.aspx](http://technical.openmobilealliance.org/Technical/release_program/SyncML_v1_2_2.aspx) (besucht am 3. Mai 2014) (zitiert auf Seite 11).
- [@Sta11] Statistisches Bundesamt Wiesbaden. *7,3 Millionen schwerbehinderte Menschen*. 2011. URL: <https://www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/Gesundheit/Behinderte/Aktuell.html> (besucht am 27. Mai 2011) (zitiert auf Seite 1).
- [@orb13] orbitaloop. *WebSqlSync*. 2013. URL: <http://www.verious.com/code/orbitaloop/WebSqlSync/> (besucht am 3. Mai 2014) (zitiert auf Seite 12).

# Abbildungsverzeichnis

2.1	Die Abbildung zeigt den Aufbau der Website <a href="http://www.thueringen-tourismus.de/barrierefrei">http://www.thueringen-tourismus.de/barrierefrei</a> auf der Daten für die Barrierefreiheitsdatenbanke eingepflegt und abgerufen werden können. . . . .	8
2.2	In der Abbildung ist ein Standardformular zu sehen, welches grundlegende Eigenschaften für die Datenerfassung einer Einrichtung erhält. Wie beispielsweise die Art der Einrichtung. . . . .	8
2.3	Auf den beiden oberen Bilder ist sowohl die Ausgabe der erfassten Daten, als auch die Seite zur Datenerfassung auf einem mobilen Endgerät zu sehen. Navigation und Inhaltselemente passen sich an die Auflösung des Endgeräts an. . . . .	9
3.1	Codebeispiel für den Aufruf zur Datenbanksynchronisation . . . . .	12
3.2	Codebeispiel für die Datenbanktabellen die synchronisiert werden sollen . . . . .	12
3.3	Codebeispiel für den Aufruf zur Synchronisation . . . . .	13
3.4	Codebeispiel für den Callback zur Erstellung einer Fortschrittsanzeige . . . . .	13
3.5	Architektur ownCloud . . . . .	14
3.6	ownCloud App . . . . .	15



# Abkürzungsverzeichnis

<b>GPS</b>	Global Positioning System
<b>HTML</b>	Hypertext Markup Language
<b>HTML5</b>	Hypertext Markup Language Version 5
<b>SQL</b>	Structured Query Language
<b>ID</b>	Ausweis, Kennnummer
<b>SSL</b>	engl. Secure Sockets Layer
<b>API</b>	Aplication-Programming-Interface

# Selbstständigkeitserklärung

Ich, Christian Schramm, Matrikel-Nr. s3001102, versichere hiermit, dass ich meinen Praxistransferbeleg mit dem Thema

*Barrierefreiheitsdatenbank - Erstellung eines Konzeptes zur Synchronisierung von Daten und Bildern zwischen mobilen Endgeräten und einem Server bei unbeständiger Internetverbindung.*

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

*Dresden, 06. Mai 2014*

---

Christian Schramm

