Technisches Handbuch Login für iOS & Android



DocCheck



Inhaltsverzeichnis

1 Der DocCheck App Login
1.1 App Login Basic – Der Klassiker für Apps
1.2 App Login Economy – Sortieren Sie Ihre User
1.3 App Login Business – Lernen Sie Ihre User kennen
2 Implementierung des DocCheck Logins 5
2.1 Implementierung innerhalb einer nativen iOS App
2.1.1 Antwerpes.framework hinzufügen 6
2.1.2 Controller anpassen
2.1.4 Login-Schnittstelle anpassen10
2.2 Implementierung innerhalb einer nativen Android App
2.2.1 Library Project hinzufügen 10
2.2.2 DocCheck Login in Applikation integrieren 12
2.3 Implementierung innerhalb einer Hybrid App
2.3.1 Installation 2.3.2 Konfiguration für iOS9+ 14
2.3.3 Aufruf des DocCheck Login Formulars
2.3.4 Senden eines Requests von einem eigenen Formular
2.4 Entwicklungs- und Testumgebung 16
2.4.1 iOS (Xcode) – Geräte/Versionen 16
2.4.2 Anarola – Gerate/ Versionen
Ihre Ansprechpartner





1 Der DocCheck App Login

Mit dem DocCheck App Login können Sie Ihre medizinischen Applikationen ganz bequem HWG-konform gestalten.

Der DocCheck App Login kann unter iOS oder Android zur Authentifizierung der Berufszugehörigkeit eingesetzt werden – Sie steuern dabei den Zugang verschiedener Berufsgruppen zu Ihrer App über unser CRM DocCheck CReaM (https://crm.doccheck.com/).

Hinweis: Zur Einbindung des DocCheck Logins benötigen Sie das Mobile Login Framework. Bitte kontaktieren Sie uns hierzu.

Den DocCheck App Login bieten wir in drei Varianten an:



1.1 App Login Basic – Der Klassiker für Apps

- Einbindung des Logins über das von uns gelieferte Framework
- Berufsgruppen-Check und -Filterung über DocCheck CReaM



+ 250 € Einrichtungsgebühr



DocCheck



1.2 App Login Economy – Sortieren Sie Ihre User

- Alle Bestandteile des Basic App Logins
- Mit dem Unique Key erkennen Sie Ihre User beim nächsten Besuch anonymisiert wieder
- Das Routing ermöglicht Ihnen das Sortieren der User nach Beruf, Sprache oder Land und somit das Weiterleiten auf unterschiedliche Bereiche innerhalb Ihrer App

1. bis 3. App: 1.500 €*
4. bis 6. App: 1.350 €*
7. bis 9. App: 1.200 €*

+ 1.200 € Einrichtungsgebühr



1.3 App Login Business – Lernen Sie Ihre User kennen

- Alle Bestandteile der Basic und Economy App Logins
- Übergabe persönlicher Daten nach Zustimmung des Nutzers, damit Sie eine eigene Nutzerdatenbank aufbauen können
- Einfacher Zugang für Mitarbeiter Ihrer Firma bei gleichzeitigem Ausschluss von Wettbewerbern



+ 4.200 € Einrichtungsgebühr







2 Implementierung des DocCheck Logins

Hinweis: Bitte beachten Sie, dass zur Implementierung des DocCheck Logins in Ihrer App das jeweilige Framework für iOS oder Android benötigt wird. Bitte kontaktieren Sie uns hierfür.

Für eine reibungslose Einbindung und Funktion muss sich in jedem Fall an das jeweilige Framework gehalten werden.

Bevor Sie den DocCheck Login in Ihre App implementieren können, müssen Sie in CReaM (https://crm.doccheck.com/) einen neuen Login erstellen. Für eine detaillierte Anleitung rufen Sie bitte unter http://biz.doccheck.com/de/services/passwordprotection/ das technische Handbuch auf (Kapitel 2.2.3).

Ist ein neuer Login angelegt, fügen Sie bitte den Bundle-Identifier (für iOS) oder den Package-Name (für Android) in die Ziel-URL ein. Dieser setzt sich in der Regel wir folgt zusammen: **topleveldomain.firmenname.appname**

	2. Zugangsgruppen	3. Codegenerierung	4. Spec
Login ID			
200000007350			
Name des Logins			
AppLoginTest			
URL des Logins			
doccheck://login?	appid=bundleidentifier		
Ziel URL			
doccheck://login?	appid=bundleidentifier		
Sprache	La	and	
Deutsch	-	Deutschland	•
Login-Status			
			-



2.1 Implementierung innerhalb einer nativen iOS App

Für eine iOS-App empfehlen wir immer die Verwendung der aktuellsten XCode-Version des Frameworks.

2.1.1 Antwerpes.framework hinzufügen



Um den DocCheck Login innerhalb Ihrer App verwenden zu können, müssen Sie zunächst das **Antwerpes.framework** zu Ihrem Projekt hinzufügen.

Klicken Sie dazu mit der rechten Maustaste auf "Frameworks" \rightarrow "Add" \rightarrow "Existing Files...".



Navigieren Sie in das Verzeichnis, in das Sie den Ordner **Antwerpes.framework** entpackt haben. Setzen Sie den Haken bei **"Copy items if needed"**. Wählen Sie anschließend das Framework aus und klicken Sie **"Add"**.

	Photos	Name
	Movies	Antwerpes.framework
TAGS		docCheckLoginTest Key App xcodeproi
•	Rot	E with the production
•	Orange	
•	Gelb	
٠	Grün	
•	Blau	
•	Lila	
۰	Grau	
0	All Tags	
	Added fo	olders: ⊙ Create groups ○ Create folder references argets: ☑ 赨 Key App

2.1.2 Controller anpassen

Erstellen Sie zunächst in Ihrem CReaM-Account einen neuen Login (s. Punkt 2 bzw. technisches Handbuch). Während der Konfiguration erhalten Sie die Login-ID. Anschließend wird die App-ID in die Start- und Ziel-URL Felder eingeben.

Importieren Sie in der .h-Datei einer Klasse (z.B. einem View Controller) das **Antwerpes.framework** und fügen Sie ihr das **DocCheckLoginDelegate**-Protokoll hinzu.

Beispiel der Klasse DocCheckLoginTestViewController.h:

<pre>#import <antwerpes apdocchecklogin.h=""></antwerpes></pre>
@interface
DocCheckLoginTestViewController :
UIViewController <docchecklogindelegate> {</docchecklogindelegate>
}
@end



Um das DocCheck Login Formular anzuzeigen, muss folgender Code verwendet werden (z.B. in einer **IBAction**):

```
- (IBAction)
startLogin {
    APDocCheckLogin*dcl = [APDocCheckLogin sharedInstance];
    dcl.loginId= @"1234567890123"; // Ihre DocCheck Login ID
    dcl.delegate=self;
    // Erlaubte Sprachen: de, com, es, fr, nl, it
    dcl.language = @"de";
    // Fügt den DocCheck Login ihrer View hinzu
    self.view addSubview:dcl.view];
}
```

Zusätzlich müssen die folgenden Methoden des DocCheckLoginDelegates implementiert werden:

- (void)docCheckLoginSuccessful;

→ Wird aufgerufen, sobald der Login erfolgreich abgeschlossen ist. Nun kann das Programm normal weiter ausgeführt werden, die View des DocCheck Logins entfernt sich eigenständig.

- (void)docCheckLoginFailedWithError:(NSError *) error;

→ Wird aufgerufen, wenn der Login aufgrund einer nicht bestehenden Internetverbindung fehlschlägt. Der Nutzer sollte über den Fehler informiert und der Fehler angemessen behandelt werden.

Die folgenden Methoden des DocCheckLoginDelegates können optional implementiert werden:

- (void) docCheckLoginCanceled;

→ Wird aufgerufen, sobald der Login von dem Benutzer manuell durch Bestätigung des "Zurück"-Pfeiles abgebrochen wurde.

- (void) docCheckLoginReceivedUserInfo: (NSDictionary *)userInfo;

→ Wird aufgerufen, nachdem der Login erfolgreich abgeschlossen wurde. **userInfo** beinhaltet detaillierte Informationen des Benutzerprofils, falls der Benutzer zuvor der Verwendung entsprechender Daten in seinen Profileinstellungen zugestimmt hat.



2.1.3 Konfiguration für iOS9+

Aufgrund neuer Restriktionen unter iOS9 ist es notwendig, eine passende Ausnahme zu definieren, damit die Verbindung zu DocCheck hergestellt werden kann. Stellen Sie hierfür sicher, dass die nachfolgenden Einträge in der **Info.plist** Datei Ihres iOS Projektes vorhanden sind:

```
<key>NSAppTransportSecurity</key>
<dict>
 <key>NSExceptionDomains</key>
 <dict>
  <key>dccdn.de</key>
  <dict>
    <key>NSExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSExceptionRequiresForwardSecrecy</key>
    <false/>
    <key>NSIncludesSubdomains</key>
    <true/>
    <key>NSRequiresCertificateTransparency</key>
    <false/>
    <key>NSThirdPartyExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSThirdPartyExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
    <false/>
   </dict>
   <key>doccheck.com</key>
   <dict>
    <key>NSExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSExceptionRequiresForwardSecrecy</key>
    <false/>
    <key>NSIncludesSubdomains</key>
    <true/>
    <key>NSRequiresCertificateTransparency</key>
    <false/>
    <key>NSThirdPartyExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSThirdPartyExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
    <false/>
  </dict>
 </dict>
</dict>
```



Hinweis: Damit wir Ihnen auch die Login-Schnittstelle an Ihre App anpassen können, muss seitens DocCheck das dazugehörige Special (je nach Lizenzmodell) aktiviert werden. Bitte teilen Sie uns hierfür die ID Ihres DocCheck Logins sowie Ihre AppID (Bundle Identifier) mit.

2.2 Implementierung innerhalb einer nativen Android App

Für die Verwendung des DocCheck Logins muss die aufrufende App das Library Project "DocCheckLogin" einbinden. Ergänzend zur DocCheckLogin Library liegt das Androidprojekt DocCheckLoginTest vor, um exemplarisch die Einbindung und den Aufruf des Logins zu verdeutlichen.

2.2.1 Library Project hinzufügen

Um den DocCheck Login innerhalb Ihrer App verwenden zu können, müssen Sie zunächst das **Library Project** zu Ihrem Projekt hinzufügen.

Klicken Sie dazu im Project Explorer mit der rechten Maustaste auf ihr Projekt und wählen Sie **Properties**.

∎ Package Explorer 🖾		oc
Som.doccheck.login.android [con	n.doccheck.login.android master] 1	p
R > com.doccheck.login.android.testa	pp [d New	۲
	Go Into	
	Open in New Window	
	Open Type Hierarchy	F4
	Show In て第W	•
	🖹 Сору 🖁	С
	Copy Qualified Name	
	The Paste #	٧
	💥 Delete	X
	. Remove from Context ℃企器	Ļ
	Build Path	•
	Source \C#S	۲
	Refactor \C#T	•
	🔤 Import	
	🚵 Export	
	🛷 Refresh	F5
	Close Project	
	Close Unrelated Projects	
	Assign Working Sets	
	Profile As	•
	Debug As	۲
	Kun As	۲
	Team	
	Compare With	5
	Replace With	
	Restore from Local History	-
	Android Tools	•
	C ()	
	Configure	



Im nachfolgenden Fenster wählen Sie das Untermenü **Android**. Klicken Sie im Abschnitt Library auf den **Add...** -Button.

type filter text	Android			⇔ •⇔••
Resource Android Android Lint Preferences Builders Git Java Build Path Java Code Style Java Compiler Java A Compiler Java Editor Javadoc Location Project References Run/Debug Settings Tack Repository	Project Build Target Target Name Android 4.2.2 Google APIs Android 5.0.1 Google APIs Android 5.1.1 Google APIs	Vendor Android Open Source Project Google Inc. Android Open Source Project Google Inc. Android Open Source Project Google Inc.	Platform 4.2.2 5.0.1 5.0.1 5.1.1 5.1.1	API Leve 17 17 21 21 22 22 22
Task Tags ▶Validation WikiText	Library	Project	1.	
	Reference	Hoject		Add
				Remove
				Up
				Down
		Resto	ore Defaults	Apply

lease select a li	br <mark>ary proje</mark> ct	:		
🗐 com.docche	ck.login.and	Iroid		

Wählen Sie das DocCheck Login Library Project aus der Liste. Sollte das Projekt nicht in der Liste erscheinen, stellen Sie sicher, dass sich das Projekt in Ihrem Workspace befindet. Bestätigen Sie Ihre Auswahl.



2.2.2 DocCheck Login in Applikation integrieren

Einleitende Hinweise:

- Der für die Applikation genutzte Package Identifier (z.B. com.example.app) muss demjenigen entsprechen, der bei DocCheck hinterlegt wurde
- Das Library Project beinhaltet eine dcview.xml, dieser Name ist daher für den DocCheck Login reserviert und darf im eigenen Projekt nicht verwendet werden
- Folgende Einträge müssen in der Android Manifest Datei ergänzt werden, um den DocCheck Login zu nutzen:

<uses-permission android:name="android.permission.INTERNET" /> <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/> <activity android:name="com.doccheck.login.android.DocCheckLogin" android:configChanges="orientation|keyboard|keyboardHidden"/>

Die Manifest-Einträge sowie die nachfolgenden Code-Snippets können ebenfalls dem beiliegenden Beispielprojekt entnommen werden.

Definition des Interfaces:

package com.doccheck.login.android.test; public interface DCTestActivityInterface { public static final String loginId = "Your login id"; public static final String templateName = "s_mobile"; public static final int SHOW_DC_LOGIN = 1; public static final String CONST_LOGINID = "loginId"; public static final String CONST_TEMPLATENAME = "templateName"; public static final String CONST_LANGUAGE = "lang"; }

Aufruf des Library Projects aus der App heraus:

Intent i = new Intent(DocCheckLogin2TestActivity.this, DocCheckLogin.class); i.putExtra(CONST_LOGINID, loginId); i.putExtra(CONST_TEMPLATENAME, templateName); i.putExtra(CONST_COUNTRYCODE, selectedLang); startActivityForResult(i, SHOW_DC_LOGIN);

DocCheckLogin2TestActivity.this bezeichnet die aufrufende Activity und muss dementsprechend dem eigenen Projekt angepasst werden. DocCheckLogin.class ist die Login-Activity der library.

Der CountryCode wird im Beispielprojekt über einen Spinner selektiert, kann aber natürlich auch anders übergeben werden. Gültige Sprachkürzel sind **de, com, es, fr, nl, it**.

Um das Ergebnis des Logins auszuwerten muss die Methode onActivityResult implementiert werden. Das Login-Ergebnis wird via data.getBooleanExtra ("LOGIN_RESULT", false) ausgelesen. Zusätzliche Parameter (z.B. Anrede, Vorname, Name, etc.) werden über data.getSerializableExtra ("URLPARAMS") ausgelesen. Der Zugriff auf einzelne Parameter wird folgendermaßen realisiert: String anrede = paramsMap.get("dc_anrede").get(0);



Einer der zusätzlichen Parameter muss gegen Null geprüft werden, da ein valider Login diese Parameter nicht zwangsläufig beinhalten muss. Dies ist der Fall, wenn die Seite anonym betreten wird (siehe Beispielprojekt, Klasse DocCheckLogin2Activity.java, Methode onActivityResult).

Folgende Parameter können ausgelesen werden (Voraussetzung hierfür ist die Business-Lizenz, siehe 1.3):

- Appid
- dc_anrede, dc_gender, dc_titel, dc_vorname, dc_name
- dc_strasse, dc_plz, dc_ort, dc_land
- dc_beruf, dc_fachgebiet, dc_email

2.3 Implementierung innerhalb einer Hybrid App

2.3.1 Installation

Ausgegangen wird in diesem Fall von der hybriden App-Entwicklung mittels Cordova.

Hinweis: Bitte beachten Sie, dass zur Implementierung einer hybriden App das jeweilige Plugin benötigt wird. Hierfür kontaktieren Sie uns bitte.

- 1. Entpacken Sie das Plugin Archiv an einem Ort Ihrer Wahl.
- Der Prozess des Hinzufügens unterscheidet sich nicht von anderen Cordova Plugins. Lediglich ist zu beachten, dass der korrekte lokale Pfad angegeben wird. Wechseln Sie auf Kommandozeilenebene in Ihr Projektverzeichnis und führen Sie folgenden Befehl aus:

cordova plugin add /pfad/zur/lokalen/pluginkopie

3. Nach der Installation muss die dclogin.js in der index.html referenziert werden:

<script type="text/javascript" src="js/dclogin.js"></script>



2.3.2 Konfiguration für iOS9+

Aufgrund neuer Restriktionen unter iOS9 ist es notwendig, eine passende Ausnahme zu definieren, damit die Verbindung zu DocCheck hergestellt werden kann. Stellen Sie hierfür sicher, dass die nachfolgenden Einträge in der **Info.plist** Datei Ihres iOS Projektes vorhanden sind:

```
<key>NSAppTransportSecurity</key>
<dict>
 <key>NSExceptionDomains</key>
 <dict>
  <key>dccdn.de</key>
  <dict>
    <key>NSExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSExceptionRequiresForwardSecrecy</key>
    <false/>
    <key>NSIncludesSubdomains</key>
    <true/>
    <key>NSRequiresCertificateTransparency</key>
    <false/>
    <key>NSThirdPartyExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSThirdPartyExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
    <false/>
   </dict>
   <key>doccheck.com</key>
   <dict>
    <key>NSExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSExceptionRequiresForwardSecrecy</key>
    <false/>
    <key>NSIncludesSubdomains</key>
    <true/>
    <key>NSRequiresCertificateTransparency</key>
    <false/>
    <key>NSThirdPartyExceptionAllowsInsecureHTTPLoads</key>
    <true/>
    <key>NSThirdPartyExceptionMinimumTLSVersion</key>
    <string>TLSv1.0</string>
    <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
    <false/>
  </dict>
 </dict>
</dict>
```



2.3.3 Aufruf des DocCheck Login Formulars

Danach kann der Login aufgerufen werden. Nachfolgend ein beispielhafter Aufruf:

```
window.dclogin("1234567890000", "de", function(result) {
    if(result == 1)
        alert("Eingeloggt!");
    else if(result == 0)
        alert("Nicht eingeloggt!");
});
```

Die Parameter ergeben sich wie folgt:

dclogin(loginId, countryCode, callback)

loginId – Die von DocCheck bereitgestellte Login ID

countryCode – "de", "com", "es", "fr", "nl" oder "it" (waren zumindest in der nativen Implementierung als gültige Werte vermerkt).

callback – Eine Funktion, in der der Kunde seine eigene Logik unterbringen kann. Der Parameter (im obigen Beispiel result genannt) gibt hier Aufschluss darüber, ob der Login erfolgreich war. Ein Wert von 1 repräsentiert einen erfolgreichen Login, 0 einen Fehlschlag.

2.3.4 Senden eines Requests von einem eigenen Formular

Sollte ein eigenes Formular für den DocCheck Login erstellt worden sein, steht für den nötigen Login-Request ebenfalls eine Funktion bereit.

```
window.dcloginrequest("1234567890000", function(result) {
    if(result)
        alert("Eingeloggt!");
    else
        alert("Nicht eingeloggt!");
});
```

Die Parameter ergeben sich wie folgt:

dcloginrequest(loginId, callback <, usernameId, passwordId>)

loginId - Die von DocCheck bereitgestellte Login ID

callback – Dem Callback wird ein Result-Parameter vom Typ Boolean übergeben. Ist das Result true, so war der Login-Versuch erfolgreich, bei false ist der Versuch fehlgeschlagen.

usernameld, passwordld – Werden diese Parameter nicht übergeben, so erwartet das Plugin standardmäßig Formulareingabefelder mit den IDs dc_username bzw. dc_password.

Sollten die Eingabefelder im Formular mit anderen IDs belegt worden sein, dann müssen die entsprechenden IDs hier übergeben werden, damit die Werte vom Plugin ausgelesen und ausgewertet werden können. Statt IDs können auch Namen verwendet werden, um die Eingabefelder zu identifizieren.



2.4 Entwicklungs- und Testumgebung

Xcode 6.2 Eclipse Luna

2.4.1 iOS (Xcode) - Geräte/Versionen

iPhone 4 (iOS 7) iPhone 5 (iOS 8) iPhone 6 (iOS 8) iPhone 6 Plus (iOS 8) iPad Air (iOS 8) iPad 4 (iOS 8)

2.4.2 Android - Geräte/Versionen

HTC Desire (Android 2.3) Samsung Galaxy S4 Mini (Android 4.4) HTC One (Android 5.0) Nexus 9 (Android 5.0)







Ihre Ansprechpartner



Lara Albrecht

Product Managerin

+49 221 92053-595 industry@doccheck.com



David Steiger

Junior Product Manager +49 221 92053-595 industry@doccheck.com





DocCheck Community GmbH Vogelsanger Str. 66, 50823 Köln www.doccheck.com/de/



ĨΞ