

Aufgabe 5: TwitterFeed

Ziel

In dieser Aufgabe lernen wir, über ein CoreServices Framework Twitter Nachrichten zu lesen und auf die Konsole zu schreiben. Daneben sehen wir neue Objective-C Konstrukte: Blocks.

Aufgabe

1. Erstellen eines neuen Projekts mit der Vorlage *Single View Application*
2. Erstellen einer neuen Klasse *Tweet* mit zwei Properties *author* und *text* vom Typ *NSString*
3. Implementieren Sie in der Klasse *Tweet* die Methode – (*NSString**)*description* in welcher die zwei Properties in einem String zurückgegeben werden.
4. Erstellen Sie eine zweite Klasse mit dem Namen *TweetService* mit einer öffentlichen Methode – (*void*)*requestTweets* (Tipp: “öffentlich” bedeutet, dass die Methode auch in der Headerdatei deklariert wird).
5. Implementieren Sie nun diese Methode, welche die letzten Tweets mit dem Hashtag “FHNW” liest und auf die Konsole ausgibt.

1. Um das Social Framework von CoreServices zu verwenden, müssen wir zuerst dieses importieren:

```
@import Social;  
@import Accounts;
```

Tipp: Accounts wird verwendet um den Twitter Account von iOS zu verwenden. Setzen Sie diesen unbedingt im Simulator unter *Settings > Twitter*.

2. Mit folgender URL kommen wir an die Informationen von Twitter: <https://api.twitter.com/1.1/search/tweets.json?q=%23fhnw>
3. Nun müssen wir zuerst den Account bekommen, welchen wir in iOS konfiguriert haben.

```
ACAccountStore* store = [[ACAccountStore alloc] init];  
ACAccountType *type = [store  
    accountTypeWithAccountTypeIdentifier:ACAccountTypeIdent  
        ifierTwitter];  
[store requestAccessToAccountsWithType:twitterAccountType  
    options:NULL  
    completion:^(BOOL granted, NSError *error) {  
    if (granted) {  
        NSArray *accounts = [store  
            accountsWithAccountType:twitterAccountType];  
  
        // here we create the SLRequest  
    } else {  
        NSLog(@"%@", [error localizedDescription]);  
    }  
}];
```

4. Den Account und die URL müssen wir in den Social Framework SLRequest einbauen.

```
SLRequest* request =
[SLRequest
 requestForServiceType:SLServiceTypeTwitter
 requestMethod:SLRequestMethodGET
 URL:[NSURL URLWithString:@"..."]
 parameters:@{@"<paramName>" : @""}];
[request setAccount:[accounts lastObject]];
```

Tipp: Auf die URL Parameter achten!

Tipp2: In einer URL wird der Hashtag (#) als %23 codiert.

Tipp3: Um NSDictionary direkt zu erstellen gibt es das Literal @{ <key> : <value> }

Tipp4: Nicht vergessen im Code Snippet <paramName> zu ersetzen

5. Um diesen Request nun abzuschicken müssen wir die Methode -performRequestWithHandler: aufrufen. Dieser übergeben wir einen Block, in welchem wir die Antwort behandeln werden:

```
[request performRequestWithHandler:
 ^(NSData *responseData,           // Block mit drei
  NSHTTPURLResponse *urlResponse, // Parametern
  NSError *error) {
  ...
}];
```

6. Die Antwortdaten müssen zuerst von den reinen Binärdaten in ein JSON Format serialisiert werden mit [NSJSONSerialization JSONObjectWithData:...]. Die Daten sind nun als NSDictionary und NSArray verwendbar.
7. Erstellen Sie eine lokale Instanz von NSMutableArray um die ausgelesenen Tweets darin zu speichern.
8. Lesen Sie nun das JSON Dictionary aus und erstellen Sie aus jeder Nachricht ein Tweet Objekt welches Sie zu unserem Array hinzufügen:

```
NSArray* results = [jsonDict valueForKey:@"results"];
NSMutableArray* tweets = [[NSMutableArray alloc] init];

for (NSDictionary* tweetDict in results) {
    NSString* author = [tweetDict
                        valueForKey:@"from_user_name"];
    NSString* text = [tweetDict
                     valueForKey:@"text"];

    Tweet* tweet = [[Tweet alloc] init];
    tweet.author = author;
    tweet.text = text;
    [tweets addObject:tweet];
}
```

9. Geben Sie am Ende des Blocks das Array mit Hilfe von NSLog() aus.
6. In der Methode -didFinishLaunchingWithOptions: in der Klasse AppDelegate erstellen Sie nun eine Instanz von TweetService und rufen darauf die Methode -requestTweets auf.
7. Applikation starten im Simulator. Auf der Konsole sollten Sie nun eine Reihe von Twitter-Nachrichten sehen welche mit #fhnw versehen waren.

Tipps:

-

Fazit

Folgende Punkte haben wir gelernt:

- Über das Social Framework einen Request an Twitter zu senden
- Mit Blocks die Antwort zu verarbeiten
- JSON Daten mithilfe von Dictionaries und Arrays auszulesen
- Verarbeitete Antwort mit Protocols zurück zu senden