#7 It has to bounce

And some local data storing

Local storage

Let’s get the boring stuff out of the way

Reasons to store data

- Better user experience
  - Show users something while you fetch new data (e.g. Facebook, Twitter, Instagram…)
- Need for a local database
- You need to remember data so you can retrieve it at any time later

Commonly saved types of data

- Primitive values
- Custom objects
- Media (images, videos, etc.)

Storing primitive values

Aka using NSUserDefaults

NSUserDefaults

- Commonly used to store app preferences
- Can store either primitive objects, or custom classes if they implement NSCoder protocol
  - NSCoder is a bit tedious
- Can be synced with iCloud

NSUserDefaults usage

- Storing a value

```
[[NSUserDefaults standardUserDefaults] setObject:@"123456"
forKey:@"token"];
[[NSUserDefaults standardUserDefaults] synchronize];
```

- Retrieving a value

```
NSString *token = [[NSUserDefaults standardUserDefaults]
objectForKey:@“token"];
```

Demo

Storing files

Documents, pics, anything that you can’t bundle with your app

Documents directory

- Something like a user directory on every major OS
- Used to store documents created by users, downloaded, and sometimes, given from other apps
- Synced to iCloud by default which will cause your app to be rejected if you abuse the storage amount

Storing an image to documents directory

- Storing the image

```
UIImage *image;
NSData *pngData = UIImagePNGRepresentation(image);

NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES);
NSString *documentsPath = [paths objectAtIndex:0];
NSString *filePath = [documentsPath
stringByAppendingPathComponent:@“image.png"];

 [pngData writeToFile:filePath atomically:YES];
```

- Retrieving the image

```
NSData *pngData = [NSData dataWithContentsOfFile:filePath];
UIImage *image = [UIImage imageWithData:pngData];
```

Storing complex data

This can get really complicated

CoreData

- Object graph and persistence framework
- Sort of like a relational database (RDBMS)
- Except it's not
- Official documentation:

  https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html

Other options

- Third party libs with persistence management
  - https://github.com/couchbase/couchbase-lite-ios
  - https://realm.io/
  - SQLite (manually mostly)

Animations

Bouncy bouncy

Bring on the sexy

- The visual identity of iOS apps are sexy animations
- Handled by Core Animation where the most of the complex stuff is tucked away from the developers
- Individual frames are left to the OS to handle

Part 1

UIView animations

UIView animatable properties

- frame
- bounds
- center
- transform
- alpha
- backgroundColor
- contentStretch

Animating stuff

```
[UIView animateWithDuration:1.0f animations:^{
    imageView.alpha = 1.0;
}];
```

Demo

pop

pop will save the world

What’s pop?

- Facebook’s library for physics animations
- https://github.com/facebook/pop
- Allows easy organic animations
- Usually there’s no way to time animations

Using pop

```
POPSpringAnimation *animation = [POPSpringAnimation
animationWithPropertyNamed:kPOPLayerCornerRadius];

animation.toValue = @(50.0f);
animation.springBounciness = 16;
animation.springSpeed = 2;

[self.imageView.layer pop_addAnimation:animation forKey:@"corners"];
```

Demo

DZ 7

This one is easy

DZ do 31.7.

Pimp my app.

Neobvezno, možete token nakon logiranja spremiti negdje u NSUserDefaults pa preskočiti login po potrebi. Ovdje morate samo paziti na to da u nekom trenutku taj login mora biti dostupan ako vas API izbaci van pa taj proces morate opet proći.

DZ extra

Dodajte caching postova u aplikaciju :). Tehnologija i princip nisu važni, implementirajte bilo koju nativnu ili third party opciju tako da jednom učitane postove čuvate negdje.

Appendix

- Data storage
  - NSUserDefaults
  - File system overview
  - Core Data programming guide
- Animations
  - UIView animations
  - pop