Grand Central Dispatch and UIActivityIndicatorView

Multi-threading in iOS

- iOS Apps have **multiple threads of execution** (inside the **one process** of the app)
- Multithreaded applications still **share the same memory space in the process**, so you do have to be careful at times
- The **Main Thread** is where all updates to the User Interface should occur, in order to avoid errors, visual defects, corruptions, and crashes
- So if we make a network call, and then want to update our UI with some new information, we should put the instructions **back onto the Main Thread**
- We do this using an Apple library called **Grand Central Dispatch (GCD)**

Grand Central Dispatch

- High level overview: GCD primarily works through groups of Dispatch Queues
- A **Dispatch Queue** is an object-like structure that manages the tasks you submit to it. All dispatch queues are first-in, first-out data structures
- The **Main Dispatch Queue** is a globally available serial queue that executes tasks on the application's main thread
- **Dispatch queues** execute their tasks **concurrently with respect to other dispatch queues**. The serialization of tasks is limited to the tasks in a single dispatch queue
- We will just use: **DispatchQueue.main.async** to add instructions back on to the main thread before and after a network call

UIActivityIndicatorView

- Colloquially called a “Spinner”
- Shows that a task is in progress
- Can be used to block UI while visible
- Animating can be controlled by startAnimating() and stopAnimating() methods
- Automatically hide the activity indicator when animation stops by setting the hidesWhenStopped property to `true`