Intro do Swifta na pełnej bombie

O czym pogadamy?

1. MVC jako pattern lansowany przez Apple
2. Jak tworzyć views za pomocą UIKit?
3. Stack View
4. Table View
5. Container View

MVC

Model

View

Controller

Model

- odpowiada za funkcjonalność naszej klasy / struktury

- komunikuje się JEDYNIE z Controllerem

- NIE MOŻE komunikować się z View

View

Nie ma co za dużo opowiadać, jest to po prostu fragment kodu odpowiedzialny za wyświetlanie naszego View.

Controller

- jest niejako ‚łącznikiem' pomiędzy View oraz Modelem, stąd zazwyczaj zapisujemy taki plik jako np. ‚MyViewController.swift'
- za pośrednictwem Controllera w View wyświetlane jest to, co przestawić chce nam Model

UIKit - tworzenie Views

Views możemy tworzyć na trzy sposoby:

1. poprzez ‚Storyboards'
2. za pomocą pliku .xib (.nib)
3. bezpośrednio z kodu

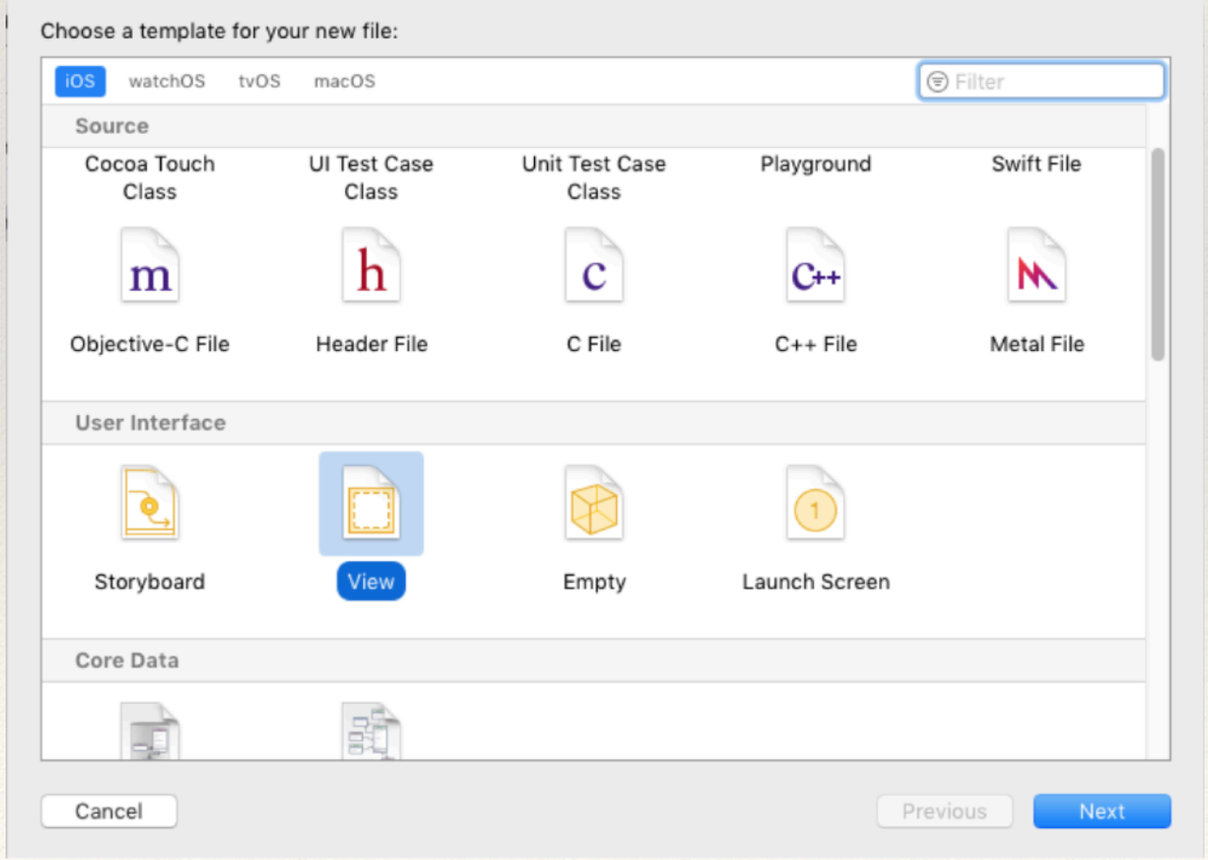
Jak ktoś nie ogarnia metody 1, to robimy live demo

2. Z .xib (.nib)

Co nam to daje?

Nasze View jest bardziej uniwersalne, ponieważ w pliku .xib zapisywany jest pojedynczy element UIView. Pozwala nam to na ewentualne dziedziczenie Views.

Jak to wygląda w praktyce?



1. Dodajemy nowy plik View do naszego katalogu.
2. Robimy z nim co chcemy - trochę jak w Storyboardzie
3. Tworzymy nowy plik z rozszerzeniem .swift, który opisuje plik .xib.

Jeśli potrzeba demo, to krzyczeć!

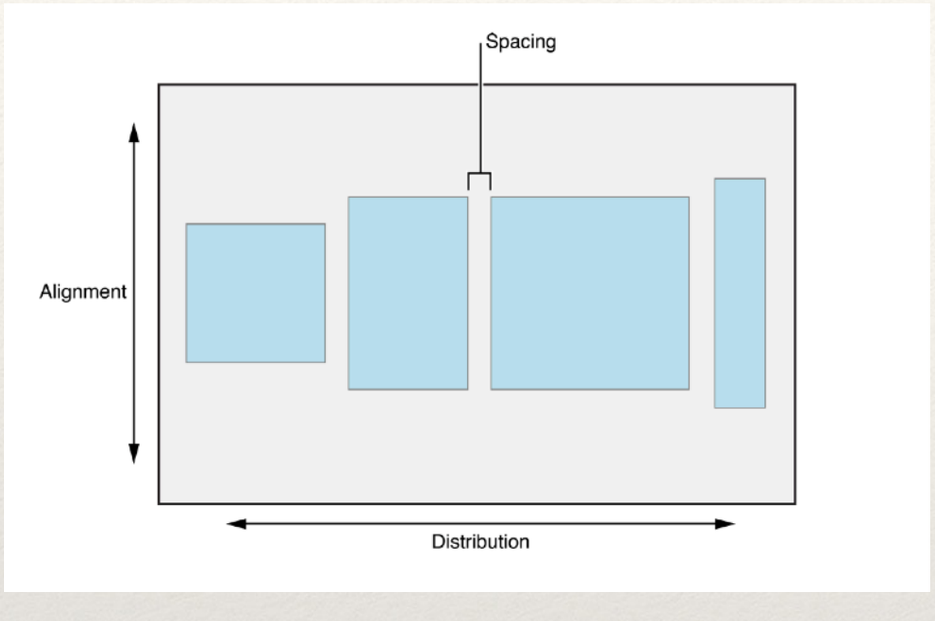
3. Z kodu

Jest tu podobieństwo z metodą .xib, tyle że WSZYSTKO musimy zapisać kodem.

Należy zatem stworzyć sobie najpierw bazowe UIView, a potem kolejno dodawać do niego kolejne elementy.

Kilka przykładów ‚Views and Controls' z UIKit

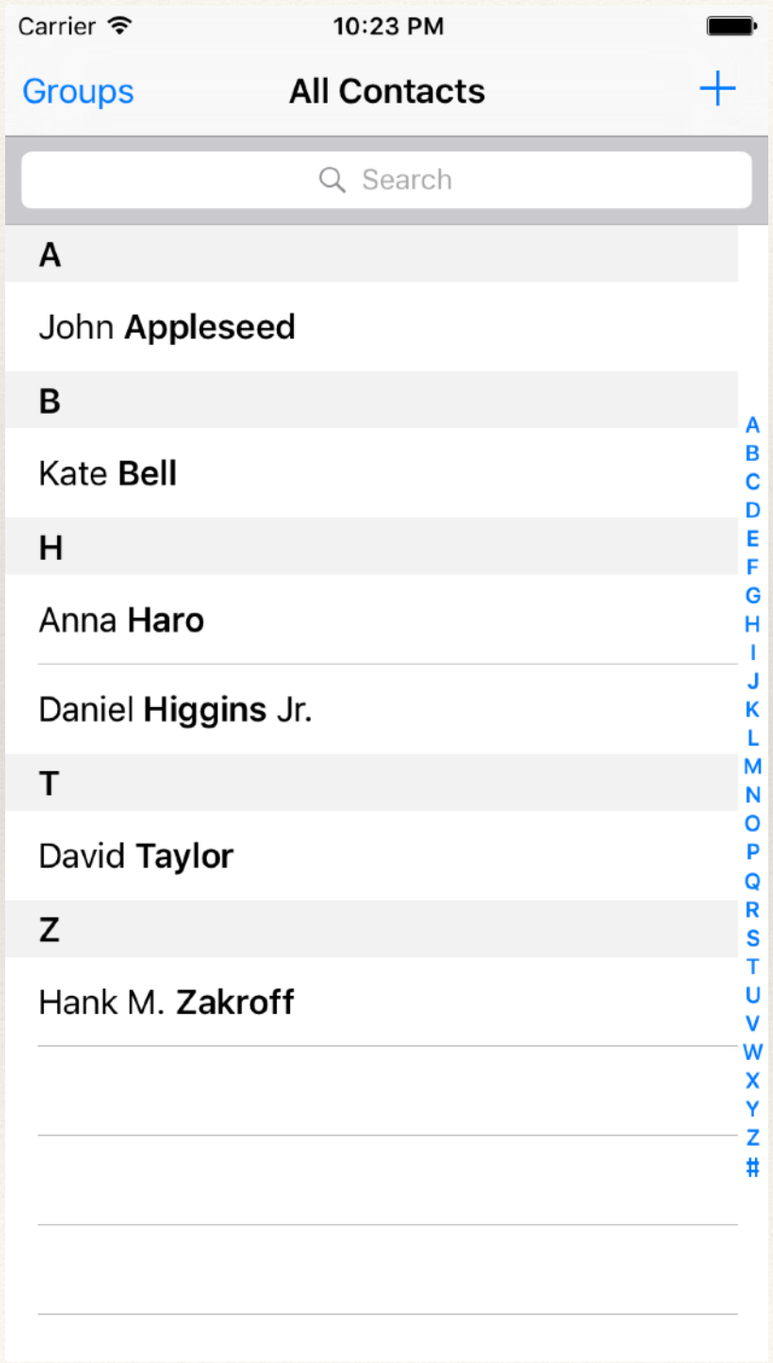
Stack View



Jest to interfejs, który dosyć zgrabnie wykorzystuje możliwości Auto Layout (to temat na zupełnie inną prezentacje).

Dynamicznie dostosowuje się do orientacji urządzenia, a także jego rozmiaru i zmian w obrębie ekranu.

Table View



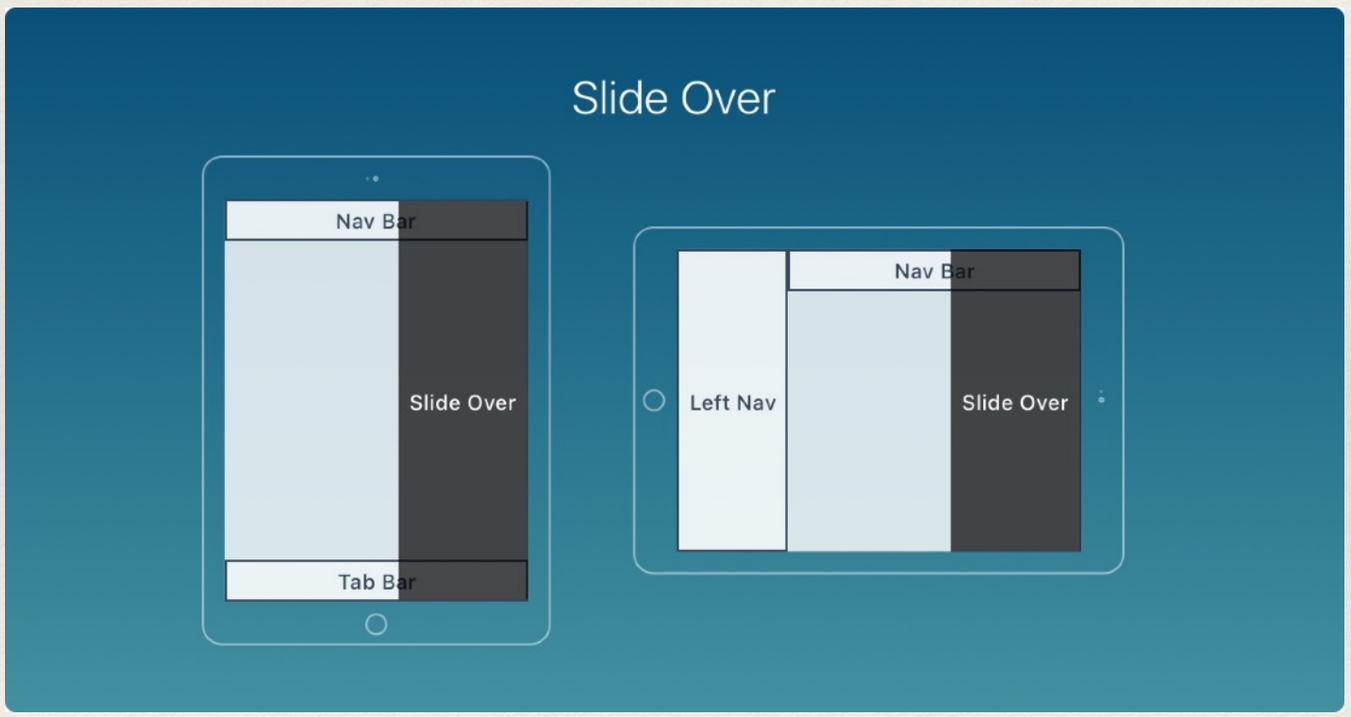
Często wykorzystywana struktura, która organizuje dane w formie listy. Przykładowe zastosowanie: aplikacja z kontaktami.

Ciekawostka: Choć często wykorzystuje się TableView, to jest coś takiego jak ‚Collection View'. Czym się różnią? W skrócie - Collection View jest trochę bardziej ‚elastyczne'.

Zdanie które zapadło mi w pamięć:
„CollectionView jest lepsze od TableView, bo z CollectionView można zrobić tabelę, a w drugą stronę to już nie działa."

Container View

Pod hasłem ContainerView kryje się kilka struktur - UINavigationBar, UITabBarController, UISplitViewController.
O dwóch z nich będzie potem, więc teraz krótko o SplitViewControllerze.



Zarządza on layoutem, biorąc pod uwagę orientacje i rozmiar urządzenia. W zależności od tego, czy korzystamy z iPhone'a 8, 8 Plus czy iPada 10.5, aplikacja może prezentować się na ekranie jako widok ogólny, szczegółowy, bądź też mieszanka dwóch powyższych.

CodeBlocks

PlayOnMac

en_windows_10_ed
ucation_...9297.iso

Duperele na stu

MarioToLama.m

Kate

iMessage

Pay

iPhone X - 11.2

Messages

Kate

Kate Bell

John Appleseed

iMessage

Pay

Od razu PROTIP:

Z czego się uczyć?

1. Dokumentacja Apple - naprawdę postarali się przy jej tworzeniu
2. raywenderlich.com - prawdopodobnie najbardziej ozłocone złoto w temacie iOS, genialne tutoriale i artykuły
3. stackoverflow.com - chyba nie ma tu jeszcze pytania, na które już ktoś by kiedyś nie odpowiedział

To by było na tyle, teraz czas na pytania!