General Information

Modify the given application to display a list of developers in a Table View with self sizing cells.

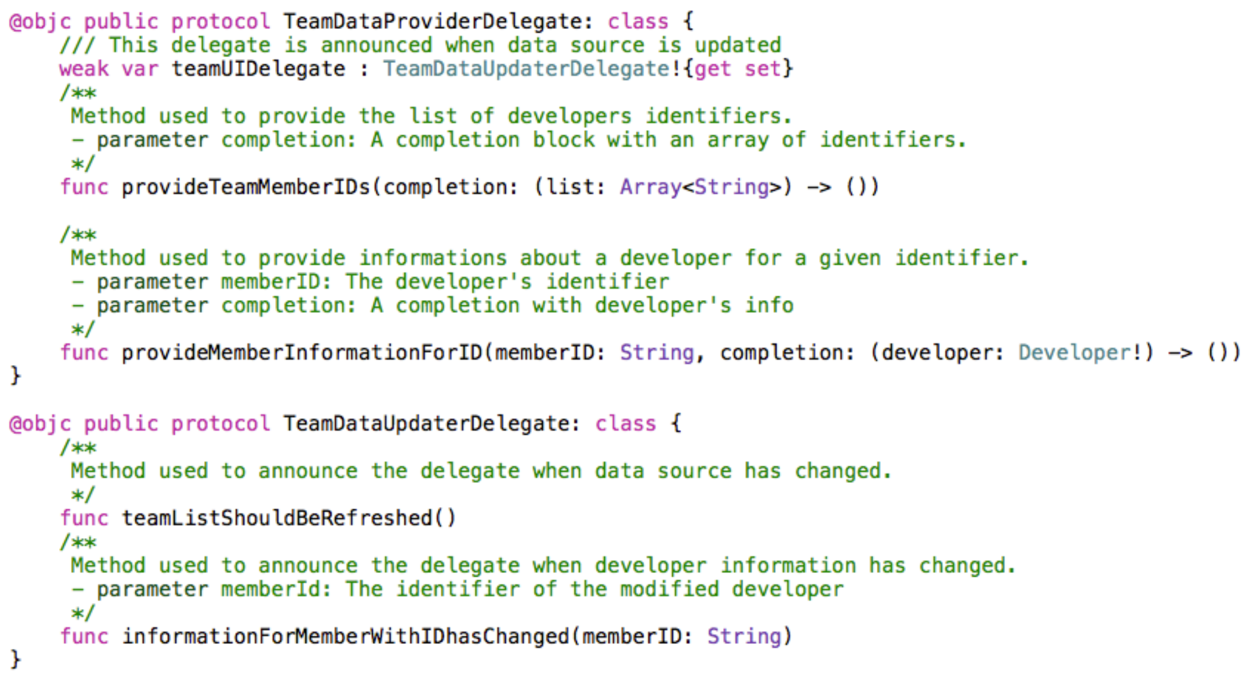
Data Source

Use BDataProvider framework to obtain the information about the developers.

A developer has the fallowing properties:

```
public private(set) var name: String

public private(set) var address: String

public private(set) var codingLevel: String

public private(set) var photoURL: String?

public var accessCode: String?

public var encryptionKey: String
```

Informations are provided by a TeamProvider witch implements TeamDataProviderDelegate protocol. Use TeamDataUpdaterDelegate protocol to update the UI when changes in data source appear.



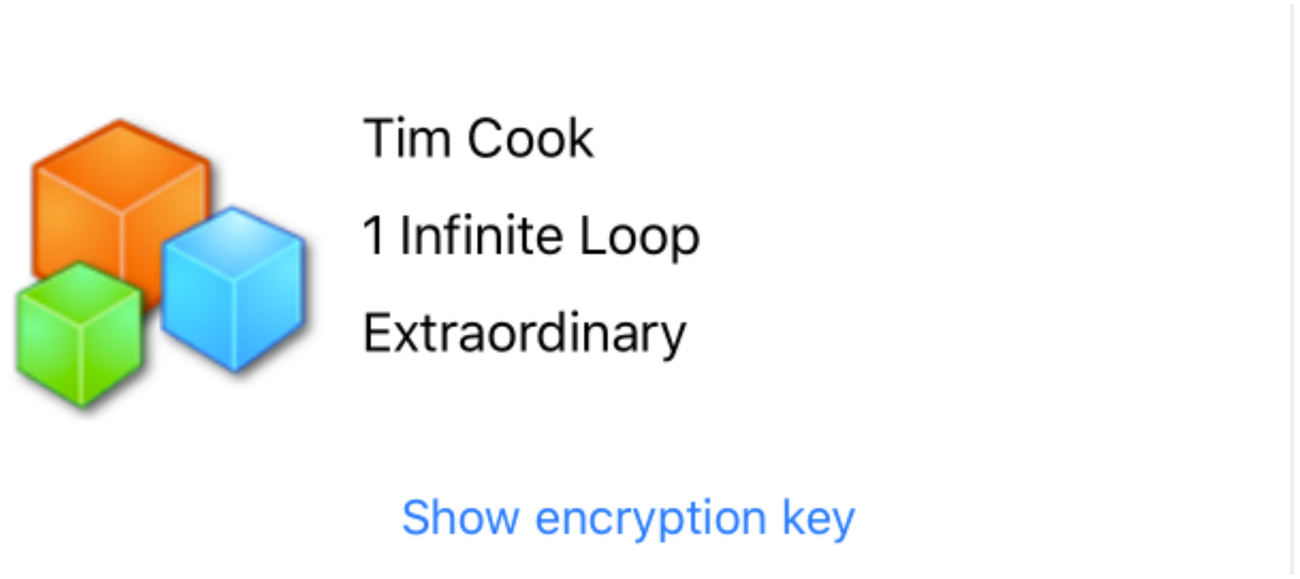
```swift
@objc public protocol TeamDataProviderDelegate: class {
    /// This delegate is announced when data source is updated
    weak var teamUIDelegate : TeamDataUpdaterDelegate!{get set}
    /**
     Method used to provide the list of developers identifiers.
     - parameter completion: A completion block with an array of identifiers.
     */
    func provideTeamMemberIDs(completion: (list: Array<String>) -> ())

    /**
     Method used to provide informations about a developer for a given identifier.
     - parameter memberID: The developer's identifier
     - parameter completion: A completion with developer's info
     */
    func provideMemberInformationForID(memberID: String, completion: (developer: Developer!) -> ())
}

@objc public protocol TeamDataUpdaterDelegate: class {
    /**
     Method used to announce the delegate when data source has changed.
     */
    func teamListShouldBeRefreshed()
    /**
     Method used to announce the delegate when developer information has changed.
     - parameter memberId: The identifier of the modified developer
     */
    func informationForMemberWithIDhasChanged(memberID: String)
}
```

User interface

Every table cell will have an image view in the left. The aspect ration of the image view is 1:1. The width of the image view is 25% of the cell width. The image view is aligned 16 points to left margin and 16 points to top.

On the right we have 3 labels. First for the developer’s name, second for the address and third for coding level. The coding level label should be multi-line. Distance between the 3 labels is 8 points. The horizontal distance between the image view and labels is 16 points. First label is aligned 16 points to top.

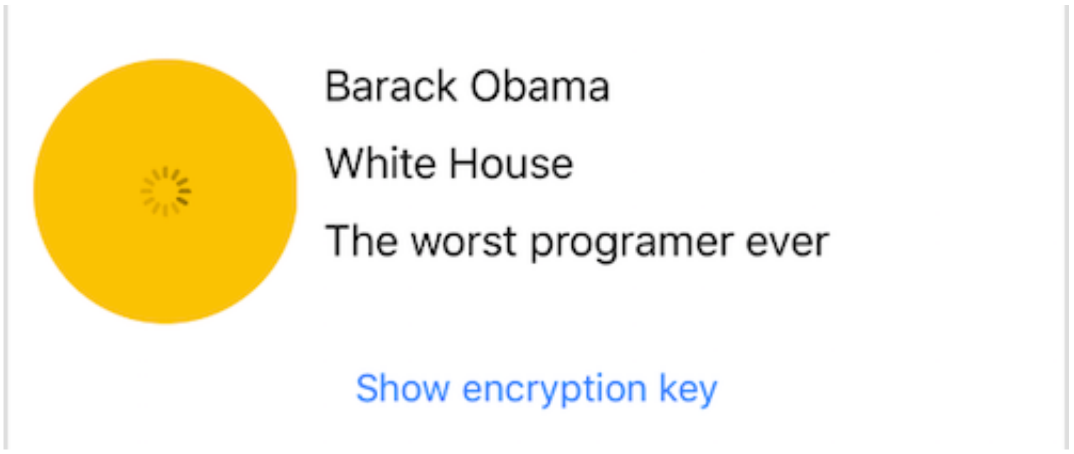
On the bottom of the cell is displayed a button with “Show encryption key” title. The buttons is alined 8 points to bottom.


Tim Cook

1 Infinite Loop

Extraordinary

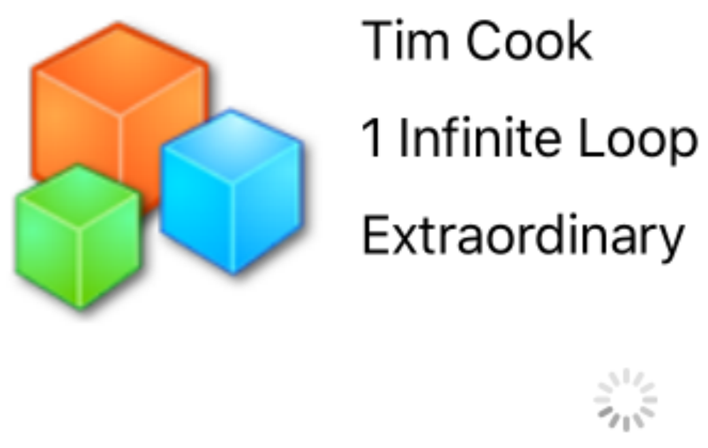
The image used will be downloaded from the URL provided in “photoURL” property. Use a circle view (not a circle image) as a placeholder for the image. Display a spinner while image is downloading.


Barack Obama

White House

The worst programer ever

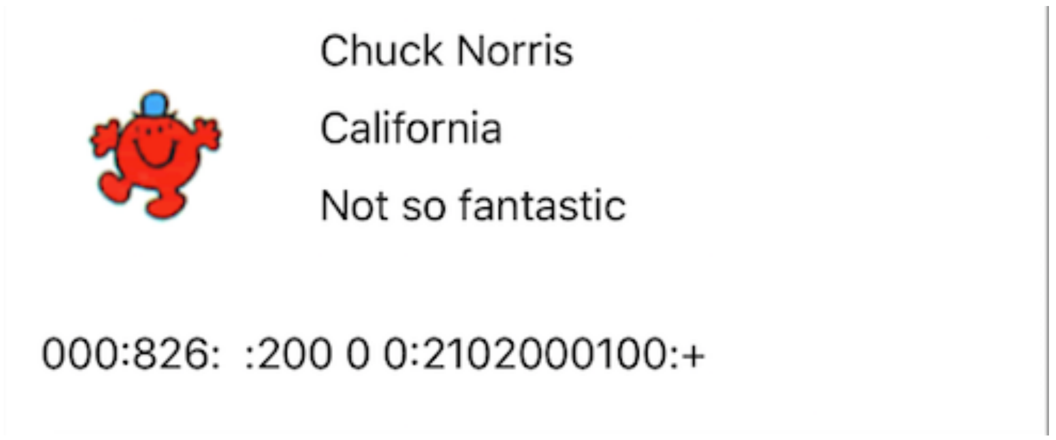
When user taps on the “Show encryption key” button, the button will disappear and a label displaying the encryption key will appear on the left. A spinner will be displayed instead of the button, until the encryption key is generated.


Tim Cook

1 Infinite Loop

Extraordinary

The encryption key's label is aligned 8 points to left margin and the distance between the image view and the label is 8 points.


Chuck Norris

California

Not so fantastic

000:826: :200 0 0:2102000100:+

After 5 seconds, you must hide the generated encryption key and show the "Show encryption key" button again. A new key should be generated every time user taps on the button.