[Classes](Classes.html)

* [AppDelegate](Classes/AppDelegate.html)
* [Planets](Classes/Planets.html)
* [PlanetsResult](Classes/PlanetsResult.html)
* [PlanetsRootClass](Classes/PlanetsRootClass.html)
* [ViewController](Classes/ViewController.html)

AppDelegate

**@UIApplicationMain**

**class** AppDelegate : UIResponder, UIApplicationDelegate

A set of methods that are called by the singleton UIApplication object in response to important events in the lifetime of your app.

* [window](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)AppDel)

* [application(\_:didFinishLaunchingWithOptions:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)AppDel)
* [applicationWillResignActive(\_:)](#/c:@M@PlanetsList_Ramesh@objc(cs)AppDel)

* [applicationDidEnterBackground(\_:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)AppDel)
* [applicationWillEnterForeground(\_:)](#/c:@M@PlanetsList_Ramesh@objc(cs)AppDel)

* [applicationDidBecomeActive(\_:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)AppDel)

* [applicationWillTerminate(\_:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)AppDel)

[Core Data stack](" \l "/Core%20Data%20stack)

* [persistentContainer](" \l "/s:18PlanetsList_Ramesh11AppDelegateC19)

[Core Data Saving support](" \l "/Core%20Data%20Saving%20support)

* [saveContext()](" \l "/s:18PlanetsList_Ramesh11AppDelegateC11)

PlanetsResult

**class** PlanetsResult : NSObject, NSCoding

PlanetsResult are getting data from planet service Instantiate the instance using the passed dictionary values to set the properties values Returns all the available property values in the form of [String:Any]

* [climate](" \l "/s:18PlanetsList_Ramesh0A6ResultC7clima)

* [created](" \l "/s:18PlanetsList_Ramesh0A6ResultC7creat)

* [diameter](" \l "/s:18PlanetsList_Ramesh0A6ResultC8diame)

* [edited](" \l "/s:18PlanetsList_Ramesh0A6ResultC6edite)

* [films](" \l "/s:18PlanetsList_Ramesh0A6ResultC5films)

* [gravity](" \l "/s:18PlanetsList_Ramesh0A6ResultC7gravi)

* [name](" \l "/s:18PlanetsList_Ramesh0A6ResultC4nameS)

* [orbitalPeriod](" \l "/s:18PlanetsList_Ramesh0A6ResultC13orbi)

* [population](" \l "/s:18PlanetsList_Ramesh0A6ResultC10popu)

* [residents](" \l "/s:18PlanetsList_Ramesh0A6ResultC9resid)

* [rotationPeriod](" \l "/s:18PlanetsList_Ramesh0A6ResultC14rota)

* [surfaceWater](" \l "/s:18PlanetsList_Ramesh0A6ResultC12surf)

* [terrain](" \l "/s:18PlanetsList_Ramesh0A6ResultC7terra)

* [url](" \l "/s:18PlanetsList_Ramesh0A6ResultC3urlSS)

* [init(fromDictionary:)](" \l "/s:18PlanetsList_Ramesh0A6ResultC14from)

* [toDictionary()](" \l "/s:18PlanetsList_Ramesh0A6ResultC12toDi)

* [init(coder:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)Planet)

* [encode(with:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)Planet)

PlanetsRootClass

**class** PlanetsRootClass : NSObject, NSCoding

PlanetsRootClass are getting data from planet service Instantiate the instance using the passed dictionary values to set the properties values Returns all the available property values in the form of [String:Any]

* [count](" \l "/s:18PlanetsList_Ramesh0A9RootClassC5co)

* [next](" \l "/s:18PlanetsList_Ramesh0A9RootClassC4ne)

* [previous](" \l "/s:18PlanetsList_Ramesh0A9RootClassC8pr)

* [results](" \l "/s:18PlanetsList_Ramesh0A9RootClassC7re)

* [init(fromDictionary:)](" \l "/s:18PlanetsList_Ramesh0A9RootClassC14f)

* [toDictionary()](" \l "/s:18PlanetsList_Ramesh0A9RootClassC12t)
* [init(coder:)](#/c:@M@PlanetsList_Ramesh@objc(cs)Planet)
* [encode(with:)](#/c:@M@PlanetsList_Ramesh@objc(cs)Planet)

ViewController

**class** ViewController : UIViewController, UITableViewDelegate, UITableViewDataSource

* List the planets data to show in table view in this class
* The UIViewController class defines the shared behavior that is common to all view controllers. You rarely create instances of the UIViewController class directly. Instead, you subclass UIViewController and add the methods and properties needed to manage the view controller’s view hierarchy. A view controller’s main responsibilities include the following: Updating the contents of the views, usually in response to changes to the underlying data. Responding to user interactions with views. Resizing views and managing the layout of the overall interface. Coordinating with other objects—including other view controllers—in your app.

* [plantesModel](" \l "/s:18PlanetsList_Ramesh14ViewController)

* [planetsNames](" \l "/s:18PlanetsList_Ramesh14ViewController)

* [listTableView](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)ViewCo)

* [appDelegate](" \l "/s:18PlanetsList_Ramesh14ViewController)

* [viewDidLoad()](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)ViewCo)

* [viewWillAppear(\_:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)ViewCo)

* [getDataFromUrl()](" \l "/s:18PlanetsList_Ramesh14ViewController)

Get data from URl and Reload Tableview Discussion\*\* Get the resluts from service and mapping with model

1. Use JSONSerialization class to convert JSON to Foundation objects and convert Foundation objects to JSON. to assaign the data in jsonResult
2. Parse into model
3. save the data offline using core data
4. Reloading the table view with that data. Call this method to reload all the data that is used to construct the table, including cells, section headers and footers, index arrays, and so on. For efficiency, the table view redisplays only those rows that are visible. It adjusts offsets if the table shrinks

* [deleteAllData(\_:)](" \l "/s:18PlanetsList_Ramesh14ViewController)

**deleteAllData** **Discussion** If any data already stored in entitiy before adding delete old data . When changes are committed, object will be removed from the uniquing tables. If object has not yet been saved to a persistent store, it is simply removed from the receiver

#### Declaration

SWIFT

**func** deleteAllData(\_ entity: String)

* [saveNameToCoreData(name:)](" \l "/s:18PlanetsList_Ramesh14ViewController)

**Discussion** Save name into core data and its mangedobject array

1. An NSEntityDescription object is associated with a specific class whose instances are used to represent entries in a persistent store in applications using the Core Data Framework. Minimally, an entity description should have: A name The name of a managed object class (If an entity has no managed object class name, it defaults to NSManagedObject.)
   1. A managed object is associated with an entity description (an instance of NSEntityDescription) that provides metadata about the object
   2. The value for the property identified by keyPath.
   3. Append the data into NSManagedObject

#### Declaration

SWIFT

**func** saveNameToCoreData(name: String)

[Table view data source](" \l "/Table%20view%20data%20source)

* [numberOfSections(in:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)ViewCo)

* [tableView(\_:numberOfRowsInSection:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)ViewCo)

* [tableView(\_:cellForRowAt:)](" \l "/c:@M@PlanetsList_Ramesh@objc(cs)ViewCo)