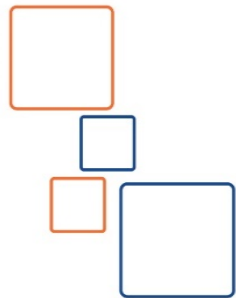


## Lecture 6



Presented By  
Esraa Eid Hassan



Java™ Education  
and Technology Services

Invest In Yourself,  
Develop Your Career

# Agenda

- Networking using URLSession
- NetworkActivityIndicator





# Networking using URLSession



# URLSession

- URLSession is a replacement for NSURLConnection, and similarly it is both a distinct class and also a group of related APIs.
- The URLSession class and related classes provide an API for downloading content.
- It able to do everything NSURLConnection could do but also adds improvements on top.

## NSURLSession Cont.



# URLSession Vs NSURLConnection

- URLSession and NSURLConnection have a lot in common in terms of how they work, at a fundamental level, they are quite different
- Apple created URLSession to resemble the general concepts of NSURLConnection

# Why Use URLSession?

- Background uploads and downloads
  - ✓ You can start a download of a large image or file, close the app and the download will continue until it completes.
- Ability to pause and resume networking operations.
- Uploads and downloads through the file system.
- Improved authentication handling.

## Why Use URLSession? Cont.

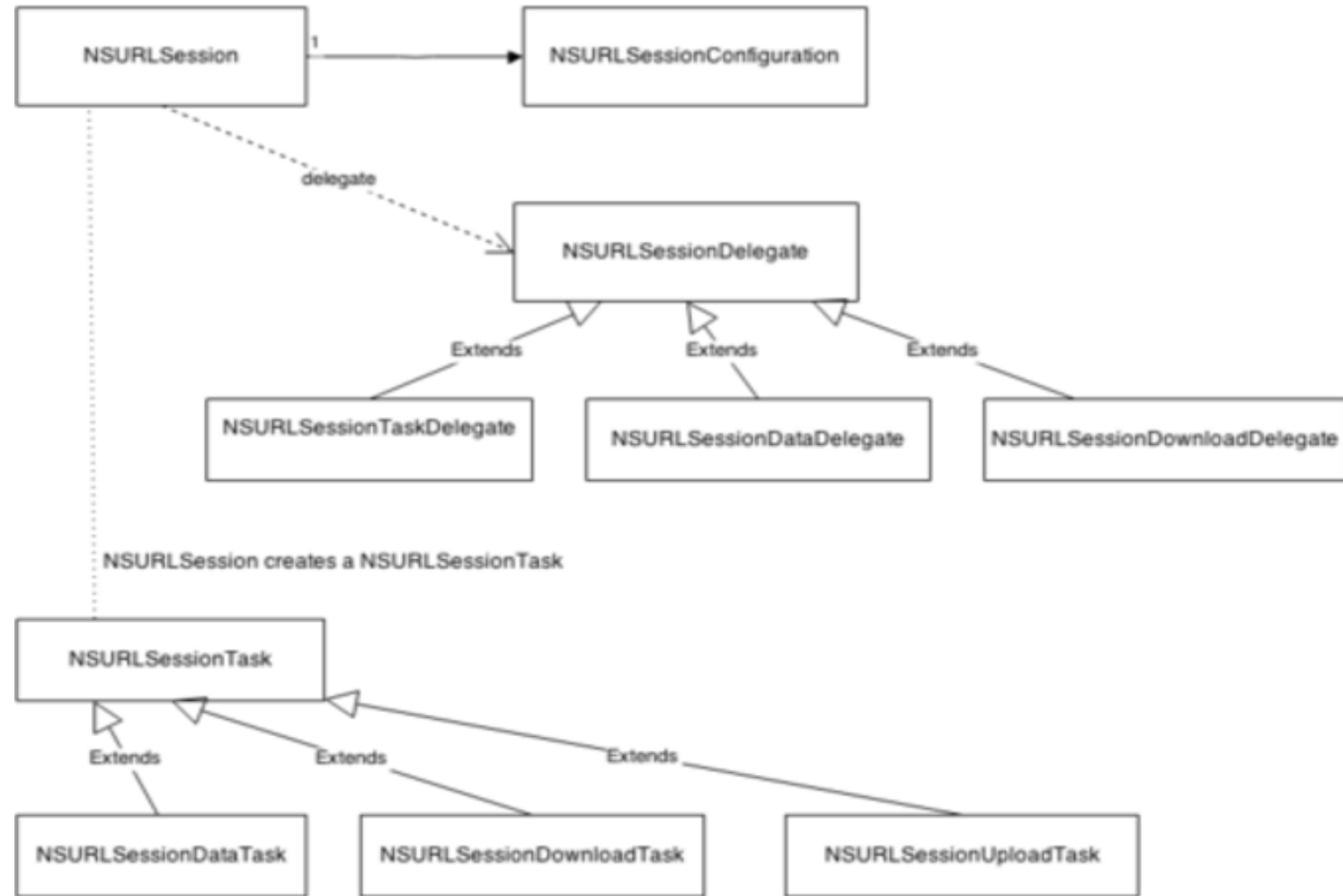
- Configurable container
  - ✓ URLSessionConfiguration is the first class you'll probably work with in the URLSession API.
  - ✓ It is a class that allows you to do exactly what its name suggests: configure your URLSession with connection/HTTP policies, caching behavior, maximum number of connections, custom headers, etc.



# Networking with URLSession API

- **URLSession** is the key component sending requests and receiving responses.
- The configuration of the session object, is handled by an instance of the **URLSessionConfiguration** class
- The basic unit of work when working with NSURLSession is the task an instance of **URLSessionTask**

# Networking with URLSession API Cont.



# Networking with URLSession API Cont.

- The basic unit of work when working with URLSession is the **task**.
- There are three types of tasks
  - ✓ URLSession**Data**Task
  - ✓ URLSession**Upload**Task
  - ✓ URLSession**Download**Task

# Networking with URLSession API Cont.

- **URLSessionDataTask**: Use this task for HTTP GET requests to retrieve data from servers to memory.
- **URLSessionUploadTask**: Use this task to upload a file from disk to a web service, typically via a HTTP POST or PUT method.
- **URLSessionDownloadTask**: Use this task to download a file from a remote service to a temporary file location.

# Data task example

```

let url = URL(string: "http://api.androidhive.info/
                    json/movies.json")
let request = URLRequest(url: url!)
let session = URLSession(configuration:
                    URLSessionConfiguration.default)
let task = session.dataTask(with: request) { (data,
                    response , error) in
do{
    var json = try JSONSerialization.jsonObject
                (with: data!, options: .allowFragments) as!
                Dictionary<String , String>
    // Manipulate your JSON
}catch{
    print("Error")
}
}
task.resume()
  
```

1

2

3

4

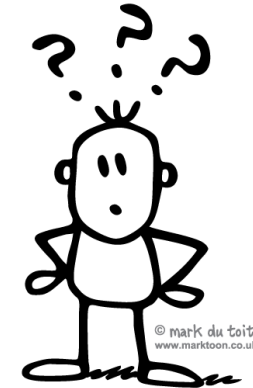
6

5

## Data task example Cont.

**Note:**

You can't do UI things inside your task because you are in a different thread



## Data task example Cont.

- You can simply dispatch back to the main queue

```
DispatchQueue.main.async {  
    //update your GUI here  
}
```



Demo







# NetworkActivityIndicator



## Display Activity Indicator

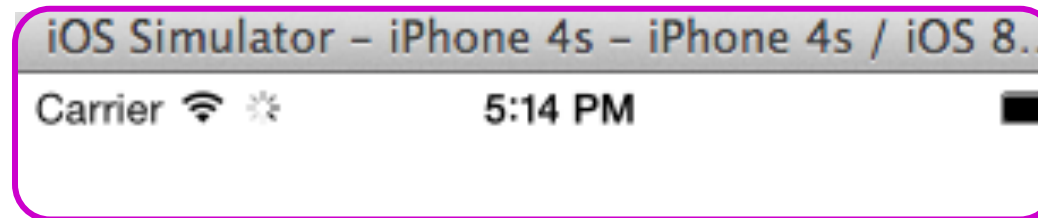
- The Activity Indicator can be displayed in the status bar to inform the user that some kind of network activity is in progress.



## Display Activity Indicator Cont.

- You can simply activate the indicator by single line of code

```
UIApplication.sharedApplication().networkActivityIndicatorVisible = true
```



- To stop the indicator just change the boolean value to **false**

## Display Activity Indicator Cont.

- You can customize coloring , positioning , style of the indicator

```
var networkIndicator = UIActivityIndicatorView  
    (activityIndicatorStyle: UIActivityIndicatorViewStyle.Gray)  
networkIndicator.center = view.center  
networkIndicator.startAnimating()  
view.addSubview(networkIndicator)
```

- To stop the activity indicator

```
networkIndicator.stopAnimating()
```



# Assignments



# 1.Movie List

- Update your Movie list app to get movies from the following webservice:

[https://imdb-api.com/en/API/BoxOffice/\[YourKey\]](https://imdb-api.com/en/API/BoxOffice/[YourKey])

Display the movie detail in another view controller when select it in the table view.