

060010815:
iOS Application Development

CocoaPods and Web Services



Dharmendra Bhatti

1

CocoaPods

- According to cocoapods.org,
 - CocoaPods is a dependency manager for Swift and Objective-C Cocoa projects.
- CocoaPods enables us to add other's open source libraries to our iOS, WatchOS, TVOS, and OSX project in a clean and manageable way.

Dharmendra Bhatti

2

CocoaPods

- Why use such external libraries???
 - Do NOT reinvent the wheel
- Caution
 - Dependent on other to update pods

Dharmendra Bhatti

3

CocoaPods

- What are the alternatives of CocoaPods?
 1. **Manually setup** a project by dragging in Frameworks, static libraries (libXML.a), dynamic libraries (swiftCore.dylib) ... and then setting up all the compiler paths to the right header files (.h)
 2. Carthage
 3. Swift Package Manager

Dharmendra Bhatti

4

CocoaPods

- What are the alternatives of CocoaPods?
 1. Manually setup
 2. Use **Carthage** to pull down your dependencies, but you'll manually include the build products into your Xcode project.
 3. Swift Package Manager

Dharmendra Bhatti

5

CocoaPods

- What are the alternatives of CocoaPods?
 1. Manually setup
 2. Carthage
 3. Use the **Swift Package Manager** (Xcode 9.3) for Swift Mac/Linux projects (iOS not officially supported ... yet)

Dharmendra Bhatti

6

CocoaPods

- CocoaPods is built with Ruby and is installable with the default Ruby available on macOS.
- \$ sudo gem install cocoapods
- \$ pod setup --verbose

Dharmendra Bhatti

7

CocoaPods

- Create a NEW project “CocoaPods1” and close the project.
- Open terminal and navigate to project directory
 - cd Desktop/CocoaPods1
- \$ pod init
- \$ open -a Xcode Podfile

Dharmendra Bhatti

8

CocoaPods

- Search for pods “cocoapods.org” and list the dependencies in a text file named Podfile in your Xcode project directory:

- platform :ios, '10.0'
- use_frameworks!
- target 'CocoaPods1' do
 - pod 'Alamofire', '~> 4..8.1'
- end

Dharmendra Bhatti

9

CocoaPods

- Now you can install the dependencies in your project:
- \$ pod install

Dharmendra Bhatti

10

CocoaPods

- Make sure to always open the Xcode workspace instead of the project file when building your project:
- \$ open CocoaPods1.xcworkspace

Dharmendra Bhatti

11

CocoaPods

- Now you can import your dependencies in ViewController.swift file e.g.:
- import UIKit
- **import Alamofire**
- class ViewController: UIViewController {
- }

Dharmendra Bhatti

12

CocoaPods

- ViewController.swift => viewDidLoad()

```
guard let url = URL(string:
    "http://localhost/studentService.php") else {
    return
}

var request = URLRequest(url: url)
request.httpMethod = "GET"
request.addValue("application/json",
    forHTTPHeaderField: "Content-Type")
```

Dharmendra Bhatti

13

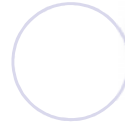
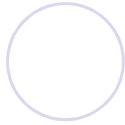
CocoaPods

- ViewController.swift => viewDidLoad()

```
Alamofire.request(request).responseJSON() {response in
    switch response.result {
    case .success(let data):
        print("Successfully received data", data)
    case .failure(let error):
        print("Request failed with error \(error)")
    }
}
```

Dharmendra Bhatti

14



RESTFUL WEB SERVICES

Dharmendra Bhatti

15

What is Web Services?

- An approach to using the web for distributed applications
- A software system designed to support interoperable machine-to-machine interaction over a network (W3C)
- An API that can be accessed over web and executed on a remote system hosting the requested service

Dharmendra Bhatti

16

What is Web Services?



- Characteristics of web services
 - communicate using open protocols such as XML, HTTP, etc.

Dharmendra Bhatti

17

What is REST?



- REpresentational State Transfer (REST) – an approach to the design of software architectures for distributed hypermedia systems.
- Introduced in 2000 by Roy Fielding (one of the principal authors of HTTP) in his doctoral dissertation

Dharmendra Bhatti

18

What is REST?

- Representational
 - Clients possess the information necessary to identify, modify, and/or delete a web resource.
- State
 - All resource state information is stored on the client.
- Transfer
 - Client state is passed from the client to the service through HTTP.

Dharmendra Bhatti

19

What is REST?

- REST is about resources and how to represent resources in different ways.
- REST is about client-server communication.
- REST is about how to manipulate resources.
- REST offers a simple, interoperable and flexible way of writing web services that can be very different from other techniques.

Dharmendra Bhatti

20

REST is NOT!

- A protocol.
- A standard.
- A replacement for SOAP.

Dharmendra Bhatti

21

What is REST?

- An architectural style centered around two basic principles:
 - resources as URIs
 - operations as HTTP methods

Dharmendra Bhatti

22

What is REST?

- Not a standard, but based on web standards
 - URIs, HTTP, HTML, XML, JSON, Atom, RDF, etc.
- Conforming to the REST constraints is often referred to as being “RESTful”.

Dharmendra Bhatti

23

RESTful Architecture

- Client-server – separation of concerns (e.g., storage versus user interface)
- Stateless – each client request contains all the information necessary to service the request
- Cacheable – improves scalability and performance
(Responses can define themselves as being cacheable or not.)

Dharmendra Bhatti

24

RESTful Architecture



- Uniform interface
- Layered system – intermediary servers can improve scalability via load balancing and shared caches
- Code on demand (optional) – servers can transfer logic to the client (e.g., JavaScript or Java applets)

Dharmendra Bhatti

25

Components of RESTful Web Service



- A base URI for the web service
 - e.g., <http://example.com>
- A set of resources with URI names relative to the base
 - e.g., <http://example.com/order/1234>

Dharmendra Bhatti

26

Components of RESTful Web Service



- The MIME types of the data supported by the web service
 - e.g., XML, JSON, plain text, etc.
- The set of operations supported by the web service using HTTP methods
 - e.g., POST, GET, PUT, DELETE

Dharmendra Bhatti

27

Resources vs Representations



- A resource can be essentially any coherent and meaningful abstraction that may be addressed.
- A representation of a resource is typically a document that captures the current or intended state of a resource.
- A resource can have more than one representation – XML, JSON, PDF file, JPEG image, etc.

Dharmendra Bhatti

28

Examples: Different Representations

- XML

```
<?xml version="1.0"?>
<customer>
  <name>Dinesh Patel</name>
  <phone>123-456-7890</phone>
  <city>Bardoli</city>
  <state>Gujarat</state>
</customer>
```

- JSON

```
{ "customer" :
  {
    "name" : "Dinesh Patel",
    "phone" : "123-45-6789",
    "city" : "Bardoli",
    "state" : "Gujarat"
  }
}
```

Dharmendra Bhatti

29

Using URI's to Name Resources

- <http://example.com/customers>
 - all customers
- <http://example.com/orders/>
 - all orders
- <http://example.com/customers/1234>
 - customer with id 1234
- <http://example.com/orders/5678>
 - order with id 5678
- <http://example.com/customers/1234/orders>
 - all orders for customer with id 1234
- <http://example.com/customers?last-name=moore>
 - list of all customers with a last name of moore

Dharmendra Bhatti

30

Example API Mapping



Route	HTTP Verb	Description
/api/user	GET	Get all the users.
/api/user	POST	Create a new user.
/api/user/{id}	GET	Get a single user.
/api/user/{id}	PUT	Update a user with new info.
/api/user/{id}	DELETE	Delete a user.

Dharmendra Bhatti

31

JSON



- JSON is a simple way to represent JavaScript object as strings.
- JSON stands for JavaScript Object Notation.
- A web application and a server communicate easily using the JSON data format.

Dharmendra Bhatti

32

JSON – How data is stored.



- Each object in JSON is represented as a list of property names, called Keynames and their values, in the following format:

```
(  
  "keyname1" => "value1",  
  "keyname2" => "value2"  
)
```

NOTE: => is an association symbol

Dharmendra Bhatti

33

JSON Data Structure Requirements

JSON relies on two data structures: **Strings** and **Arrays**

- Strings are used to store both the Keyname and the value.
Example:

Keyname value
 ↓ ↓
 "FirstName" => "Dinesh"

- An array is used to store the list of Keyname/ Value pairs.
Example:

```
$json = array (  
  "FirstName"     => "Dinesh",  
  "LastName"     => "Patel",  
  "IdNumber"     => "2014001"  
);
```

Dharmendra Bhatti

34

How is JSON used?

- When a web application interacts with a web service it can obtain secure data using a PHP script.
- A PHP script can return this data to JavaScript using a JSON object.
- JSON strings are converted into JavaScript objects with JavaScript's ***JSON.parse*** function.

Dharmendra Bhatti

35

Demo – Student Web Service

The screenshot shows the phpMyAdmin interface with a table named 'enrollment' selected. The table contains columns for 'enrollment', 'name', and 'imageName'. The data is displayed in a table with 25 rows. The first row shows enrollment '201406100110070' for 'Jainankumar' with image '201406100110070.png'. The second row shows enrollment '201406100110136' for 'Abhinav' with image '201406100110136.png'. The third row shows enrollment '201506100110001' for 'Khushi' with image '201506100110001.png'. The fourth row shows enrollment '201506100110006' for 'Umesh' with image '201506100110006.png'. The fifth row shows enrollment '201506100110007' for 'Vinit' with image '201506100110007.png'. The sixth row shows enrollment '201506100110008' for 'Anish' with image '201506100110008.png'. The seventh row shows enrollment '201506100110009' for 'Anshuman' with image '201506100110009.png'. The eighth row shows enrollment '201506100110011' for 'Raj' with image '201506100110011.png'. The ninth row shows enrollment '201506100110013' for 'Darshan' with image '201506100110013.png'. The tenth row shows enrollment '201506100110016' for 'Harvi' with image '201506100110016.png'. The eleventh row shows enrollment '201506100110016' for 'Kajalben' with image '201506100110016.png'. The twelfth row shows enrollment '201506100110019' for 'Mansi' with image '201506100110019.png'. The thirteenth row shows enrollment '201506100110023' for 'Preet' with image '201506100110023.png'. The fourteenth row shows enrollment '201506100110025' for 'Naghar' with image '201506100110025.png'. The fifteenth row shows enrollment '201506100110027' for 'Jagdish' with image '201506100110027.png'. The sixteenth row shows enrollment '201506100110028' for 'Jugal' with image '201506100110028.png'. The seventeenth row shows enrollment '201506100110030' for 'Raj' with image '201506100110030.png'.

enrollment	name	imageName
201406100110070	Jainankumar	201406100110070.png
201406100110136	Abhinav	201406100110136.png
201506100110001	Khushi	201506100110001.png
201506100110006	Umesh	201506100110006.png
201506100110007	Vinit	201506100110007.png
201506100110008	Anish	201506100110008.png
201506100110009	Anshuman	201506100110009.png
201506100110011	Raj	201506100110011.png
201506100110013	Darshan	201506100110013.png
201506100110016	Harvi	201506100110016.png
201506100110016	Kajalben	201506100110016.png
201506100110019	Mansi	201506100110019.png
201506100110023	Preet	201506100110023.png
201506100110025	Naghar	201506100110025.png
201506100110027	Jagdish	201506100110027.png
201506100110028	Jugal	201506100110028.png
201506100110030	Raj	201506100110030.png

36

http://localhost/studentService.php

```
studentService.php — Edited
studentService.php > No Selection

1 <?php
2 $con=mysqli_connect("localhost","root","bhatti","mydb");
3 if (mysqli_connect_errno())
4 {
5     echo "Failed to connect to MySQL: " . mysqli_connect_error();
6 }
7
8 $sql="SELECT * FROM student";
9 $rows = array();
10
11 if ($result=mysqli_query($con,$sql))
12 {
13     while ($row=mysqli_fetch_assoc($result))
14     {
15         array_push($rows, $row);
16     }
17     print json_encode($rows);
18     mysqli_free_result($result);
19 }
20 mysqli_close($con); Dharmendra Bhatti
21 ?>
```

37

http://localhost/studentService.php

```
localhost
[{"enrollment":20150610010079,"name":"Jaimalkumar","imageName":"20150610010079.png"},{"enrollment":20150610010080,"name":"Khyati","imageName":"20150610010080.png"},{"enrollment":20150610010081,"name":"Vikrant","imageName":"20150610010081.png"},{"enrollment":20150610010082,"name":"Aaryakumar","imageName":"20150610010082.png"},{"enrollment":20150610010083,"name":"Raj","imageName":"20150610010083.png"},{"enrollment":20150610010084,"name":"Darshan","imageName":"20150610010084.png"},{"enrollment":20150610010085,"name":"Hesvi","imageName":"20150610010085.png"},{"enrollment":20150610010086,"name":"Kajalber","imageName":"20150610010086.png"},{"enrollment":20150610010087,"name":"Manasi","imageName":"20150610010087.png"},{"enrollment":20150610010088,"name":"Prachi","imageName":"20150610010088.png"},{"enrollment":20150610010089,"name":"Raghav","imageName":"20150610010089.png"},{"enrollment":20150610010090,"name":"Higisha","imageName":"20150610010090.png"},{"enrollment":20150610010091,"name":"Pooja","imageName":"20150610010091.png"},{"enrollment":20150610010092,"name":"Raj","imageName":"20150610010092.png"},{"enrollment":20150610010093,"name":"Jyoti","imageName":"20150610010093.png"},{"enrollment":20150610010094,"name":"Rishabh","imageName":"20150610010094.png"},{"enrollment":20150610010095,"name":"Kishan","imageName":"20150610010095.png"},{"enrollment":20150610010096,"name":"Tanvi","imageName":"20150610010096.png"},{"enrollment":20150610010097,"name":"Komal","imageName":"20150610010097.png"},{"enrollment":20150610010098,"name":"Arbhen","imageName":"20150610010098.png"},{"enrollment":20150610010099,"name":"Nalinihohen","imageName":"20150610010099.png"},{"enrollment":20150610010100,"name":"Jeniben","imageName":"20150610010100.png"},{"enrollment":20150610010101,"name":"Srujan","imageName":"20150610010101.png"},{"enrollment":20150610010102,"name":"Jeni","imageName":"20150610010102.png"},{"enrollment":20150610010103,"name":"Manasi","imageName":"20150610010103.png"},{"enrollment":20150610010104,"name":"Himadri","imageName":"20150610010104.png"},{"enrollment":20150610010105,"name":"Hrisha","imageName":"20150610010105.png"},{"enrollment":20150610010106,"name":"Pallavi","imageName":"20150610010106.png"},{"enrollment":20150610010107,"name":"Farooq","imageName":"20150610010107.png"},{"enrollment":20150610010108,"name":"Monaben","imageName":"20150610010108.png"},{"enrollment":20150610010109,"name":"Mahipalish","imageName":"20150610010109.png"},{"enrollment":20150610010110,"name":"Nikita","imageName":"20150610010110.png"},{"enrollment":20150610010111,"name":"Pradikkumar","imageName":"20150610010111.png"},{"enrollment":20150610010112,"name":"Ankurbea","imageName":"20150610010112.png"},{"enrollment":20150610010113,"name":"Isabehn","imageName":"20150610010113.png"},{"enrollment":20150610010114,"name":"Natali","imageName":"20150610010114.png"},{"enrollment":20150610010115,"name":"Ratna","imageName":"20150610010115.png"},{"enrollment":20150610010116,"name":"Aash","imageName":"20150610010116.png"},{"enrollment":20150610010117,"name":"Mantikumar","imageName":"20150610010117.png"},{"enrollment":20150610010118,"name":"Madhav","imageName":"20150610010118.png"},{"enrollment":20150610010119,"name":"Pinal","imageName":"20150610010119.png"},{"enrollment":20150610010120,"name":"Harsh","imageName":"20150610010120.png"},{"enrollment":20150610010121,"name":"Purth","imageName":"20150610010121.png"},{"enrollment":20150610010122,"name":"Goutamsinh","imageName":"20150610010122.png"},{"enrollment":20150610010123,"name":"Jay","imageName":"20150610010123.png"},{"enrollment":20150610010124,"name":"Eka","imageName":"20150610010124.png"},{"enrollment":20150610010125,"name":"Twinkle","imageName":"20150610010125.png"},{"enrollment":20150610010126,"name":"ANKUR","imageName":"20150610010126.png"}]
```

Dharmendra Bhatti

38

HTTP Methods in REST



- In REST

- GET - retrieve resource
- POST - create new resource
- PUT - update existing resource
- DELETE - delete resource

Dharmendra Bhatti

39

Demo – Student Web Service



- /student/list/
 - Retrieve all student data
- /student/list/1234
 - Retrieve student data whose enrollment number is 1234

Dharmendra Bhatti

40

Demo – Student Web Service



.htaccess

```
.htaccess — Edited
< > .htaccess No Selection
1 # Turn rewrite engine on
2 Options +FollowSymlinks
3 RewriteEngine on
4
5 # map neat URL to internal URL
6 RewriteRule ^student/list/$ studentService.php [nc,qa]
7 RewriteRule ^student/list/([0-9]+)/$ studentService.php?enrollment=$1 [nc,qa]
```

Dharmendra Bhatti

41

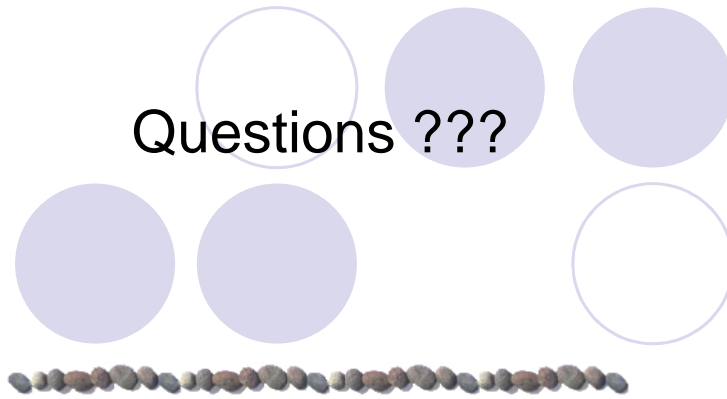
Demo – Student Web Service



Dharmendra Bhatti

42

Questions ???



Dharmendra Bhatti

43