

Communications

Mobile Application Development in iOS

School of EECS

Washington State University

Instructor: Larry Holder

Outline

- Already seen
 - Cloud Kit, Remote notifications, Placemarks via Geocoder
- Safari Services and Web Kit
- HTTP requests
- APIs
- Other communications services

Safari Services

- Full browser functionality within app
- Import `SafariServices`
- Create `URL`
- Create `SFSafariViewController` (`URL`)
- Execute `present` (`ViewController`)

Safari Services

```
import SafariServices

class ViewController: UIViewController {

    let urlString = "http://www.eecs.wsu.edu/~holder/courses/MAD/"

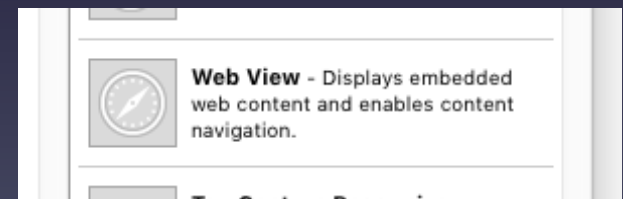
    @IBAction func safariTapped(_ sender: UIButton) {
        let url = URL(string: urlString)
        let vc = SFSafariViewController(url: url!)
        present(vc, animated: true, completion: nil)
    }
}
```

Web Kit

- Import WebKit
- WKWebView vs. UIWebView
 - StoryBoard only supports UIWebView
 - Apple recommends WKWebView in iOS 8+
 - Use programmatically

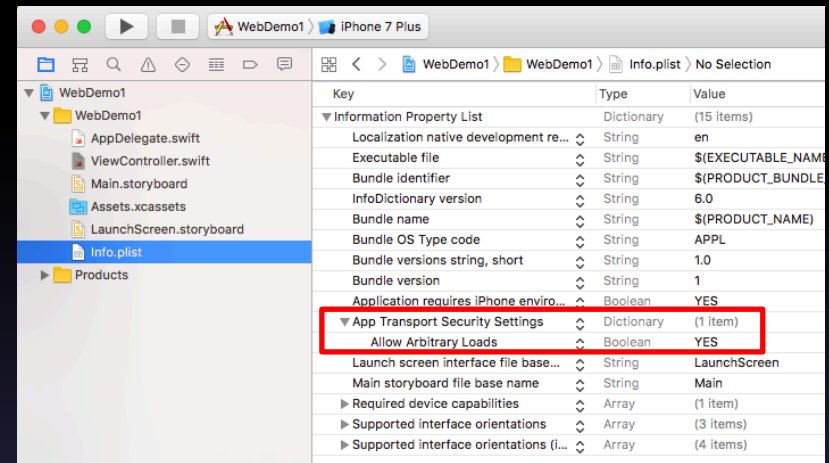
Storyboard UIWebView

```
class ViewController: UIViewController {  
  
    let urlString = "http://www.eecs.wsu.edu/~holder/courses/MAD/"  
  
    @IBOutlet weak var uiWebView: UIWebView!  
  
    @IBAction func uiWebViewTapped(_ sender: UIButton) {  
        let url = URL(string: urlString)  
        let request = URLRequest(url: url!)  
        uiWebView.loadRequest(request)  
    }  
}
```



Web Security

- App Transport Security
 - Only HTTPS by default
- Info settings
 - App Transport Security Settings
 - Allow Arbitrary Loads



Web Kit

```
import WebKit

class ViewController: UIViewController {

    let urlString = "http://www.eecs.wsu.edu/~holder/courses/MAD/"

    @IBOutlet weak var webSubView: UIView!

    @IBAction func wkWebViewTapped(_ sender: UIButton) {
        let wkWebView = WKWebView(frame: webSubView.frame)
        let url = URL(string: urlString)
        let request = URLRequest(url: url!)
        wkWebView.load(request)
        self.view.addSubview(wkWebView)
    }
}
```


HTTP Requests

- Create data task for `URLSession.shared` instance
 - `URLSession.shared.dataTask(with: URL)` *// calls delegates*
 - `URLSession.shared.dataTask(with: URLRequest)` *// calls delegates*
 - `URLSession.shared.dataTask(with: URL, completionHandler: @escaping (Data?, URLResponse?, Error?) -> Void)`
 - `URLSession.shared.dataTask(with: URLRequest, completionHandler: @escaping (Data?, URLResponse?, Error?) -> Void)`
- Call `resume()` on data task

HTTP Requests

- Delegates
 - URLSessionDelegate
 - URLSessionTaskDelegate
 - URLSessionDataDelegate
 - URLSessionDownloadDelegate
 - URLSessionStreamDelegate

HTTP Requests

```
let url = URL(string: "http://www.eecs.wsu.edu/~holder/tmp/date.php")
let dataTask = URLSession.shared.dataTask(with: url!,
                                           completionHandler: handleResponse)
dataTask.resume()

func handleResponse (data: Data?, response: URLResponse?, error: Error?) {
    if (error != nil) {
        print("error: \(error!.localizedDescription)")
    } else {
        let httpResponse = response as! HTTPURLResponse
        let statusCode = httpResponse.statusCode
        if statusCode != 200 {
            let msg = HTTPURLResponse.localizedString(forStatusCode: statusCode)
            print("HTTP \(statusCode) error: \(msg)")
        } else {
            // response okay, do something with data
            let dataStr = String(data: data!, encoding: .utf8)
            print("data = \(dataStr)")
        }
    }
}
```

HTTP Requests: Server Side

```
<?php
// date.php – return current date and local time in JSON format

date_default_timezone_set("America/Los_Angeles");
$myDate = date("Y-m-d");
$myTime = date("H:i:s");
$json = '{"date":"' . $myDate . '","time":"' . $myTime . '"}';
print $json;

?>
```

Handling Responses

- If JSON data, then use JSONSerialization

```
func handleResponse (data: Data?, response: URLResponse?, error: Error?) {  
    ...  
    // response okay, do something with data  
    if let jsonObj = try? JSONSerialization.jsonObject(with: data!,  
                                                         options: []) {  
        let jsonDict = jsonObj as! [String: AnyObject]  
        let dateStr = jsonDict["date"] as! String  
    } else {  
        print("error: invalid JSON data")  
    }  
}
```

Handling Responses

- If need to change view, dispatch to main queue

```
func handleResponse (data: Data?, response: URLResponse?, error: Error?) {  
    ...  
    // response okay, do something with data  
    let dataStr = String(data: data!, encoding: .utf8)  
    DispatchQueue.main.async {  
        self.responseLabel.text = dataStr  
    }  
}
```

```
2017-03-20 23:05:58.922 CommDemo1[46277:3142594] This application is modifying  
the autolayout engine from a background thread after the engine was accessed  
from the main thread. This can lead to engine corruption and weird crashes.  
Stack:(  
    0   CoreFoundation                                0x000000010c2b4d4b  
    __exceptionPreprocess + 171
```

HTTP POST Requests

```
let jsonStr = "{\"username\":\"\"(username)\",\"password\":\"\"(password)\"}"
let jsonData = jsonStr.data(using: .utf8)
let url = URL(string: "http://www.eecs.wsu.edu/~holder/tmp/login.php")
var request = URLRequest(url: url!)
request.httpMethod = "POST"
request.httpBody = jsonData!
request.setValue("application/json", forHTTPHeaderField: "Content-Type")

let dataTask = URLSession.shared.dataTask(with: request,
                                           completionHandler: handleLoginResponse)
dataTask.resume()
```

HTTP POST Requests: Server Side

```
<?php

// login - Return user ID for given username and password.

$json = file_get_contents('php://input');
$obj = json_decode($json);
$username = $obj->username;
$password = $obj->password;

$users = array ( array("Mario", "peach", 1),
                  array("Bowser", "junior", 2),
                  array("Yoshi", "dragon", 3) );

$userId = 0;
for ($userIndex = 0; $userIndex < count($users); $userIndex++) {
    if (($users[$userIndex][0] == $username) and
        ($users[$userIndex][1] == $password)) {
        $userId = $users[$userIndex][2];
        break;
    }
}
echo $userId;
?>
```


Application Programming Interfaces (APIs)

- News
 - <https://newsapi.org/>
- Weather
 - <https://www.wunderground.com/weather/api/>
- Much, much more
 - <https://www.programmableweb.com/>

Other Communications Services

- Network Services
- Game Kit (peer-to-peer)
- Bluetooth

Resources

- Safari Services
 - developer.apple.com/reference/safariservices
- UIWebView
 - developer.apple.com/reference/uikit/uiwebview
- Web Kit
 - developer.apple.com/reference/webkit
- URLSession
 - developer.apple.com/reference/foundation/urlsession