Unit 5—Lesson 4: Working with the Web: HTTP and URL Session

The Basics

Internet addresses

The Internet or web address is the Uniform Resource Locator (URL)

Basic URL

```
http://www.pretendco.com
```

Complex URL

```
https://sales.pretendco.com/orders/strack?order=233282&api_key=QREPORT
```

The Basics

Protocols

Protocols specify how the browser and the server communicate with each other

HTTP (Hypertext Transfer Protocol)

```
http://www.pretendco.com
```

HTTPS (HTTP Secure)

```
https://support.pretendco.com
```

The Basics

Subdomain

The subdomain (optional) points to a specific server or redirect to another URL

```
http://www.finance.pretendco.com:80/receivables/ar_january2017.pdf
```

The Basics

Domain name

The domain name is a unique reference to a specific website

Includes a required host name and top-level domain such as `.com` or `.org`

```
http://www.finance.pretendco.com:80/receivables/ar_january2017.pdf
```

The Basics

Port number

The port number used for HTTP (default port 80) or HTTPS (default port 443)

Browsers check the default port if the URL does not specify a port number

```
http://www.finance.pretendco.com:80/receivables/ar_january2017.pdf
```

The Basics

Path

The URL path specifies a file or subdirectory on a server

```
http://www.finance.pretendco.com:80/receivables/ar_january2017.pdf
```

Query parameters

Key-value pairs included in a URL are used to identify a specific request to a web server

```
http://domain/path/file?name=value&name=value
```

`?` starts the query string

`&` separates key-value pairs in the query string

Request type
HTTP methods

Methods identify the type of request

Method	Description
GET	Requests information from a server
POST	Sends information to a server
PUT	Updates information from a server
DELETE	Deletes information from a server

HTTP headers

Allows the client and the server to exchange information

- Used for authentication
- Sends information such as the computer or browser type to the server
- Responds with information such as the server type and software used to handle the request

HTTP body

Includes the data sent from the client or server following the HTTP headers

- Sends form data to the server
- Responds with a web page content and images

Create a URL

```
let url = URL(string: "https://www.apple.com")!
```

Create and execute a network request

Create a data task

```
let url = URL(string: "https://www.apple.com")!
let task = URLSession.shared.dataTask(with: url) { (data, response, error) in
    if let data = data,
       let string = String(data: data, encoding: .utf8) {
         print(string)
    }
}
task.resume()
```

Work with an API

Create a request to an API

```
let url = URL(string: "https://api.nasa.gov/planetary/apod?date=2005-2-22&api_key=DEMO_KEY")!

let task = URLSession.shared.dataTask(with: url) { (data, response, error) in

    if let data = data,
        let string = String(data: data, encoding: .utf8) {

        print(string)
    }
}
task.resume()
```

Work with an API

API response


```
{
  "date": "2005-02-22",
  "explanation": "Are Saturn's auroras like Earth's?  To help answer this question, the Hubble Space Telescope and the Cassini spacecraft monitored Saturn's South Pole simultaneously as Cassini closed in on the gas giant in January 2004.  Hubble snapped images in ultraviolet light, while Cassini recorded radio emissions and monitored the solar wind.  Like on Earth,
…",
  "hdurl": "http://apod.nasa.gov/apod/image/0502/saturnauroras_hst_big.jpg",
  "media_type": "image",
  "service_version": "v1",
  "title": "Persistent Saturnian Auroras",
  "url": "http://apod.nasa.gov/apod/image/0502/saturnauroras_hst.jpg"
}
```

Modify a URL with URL components

Dynamically add query items to a URL using `URLComponents`

```
extension URL {
    func withQueries(_ queries: [String: String]) -> URL? {

        var components = URLComponents(url: self, resolvingAgainstBaseURL: true)
        components?.queryItems = queries.flatMap { URLQueryItem(name: $0.0, value: $0.1) }
        return components?.url
    }
}
```

```swift
let baseURL = URL(string: "https://api.nasa.gov/planetary/apod")!
let query: [String: String] = [
    "api_key": "DEMO_KEY",
    "date": "2011-07-13"
]
let url = baseURL.withQueries(query)!
let task = URLSession.shared.dataTask(with: url) { (data, response, error) in
    if let data = data,
        let string = String(data: data, encoding: .utf8) {
        print(string)

    } }
task.resume()
```

```
// URLComponents API response
{
  "date": "2011-07-13",
  "explanation": "For the last time, the US Space Shuttle has approached the International Space Station (ISS).  Following a dramatic launch from Cape Canaveral last week that was witnessed by an estimated one million people, Space Shuttle Atlantis on STS-135 lifted a small crew to a welcome rendezvous three days ago with the orbiting station.  Although NASA is discontinuing the aging shuttle fleet, NASA astronauts in the near future will be able to visit the ISS on Russian space flights.  Pictured above, Atlantis rises toward the ISS with its cargo bay doors open, showing a gleaming metallic Raffaello Multi-Purpose Logistics Module.  Over 200 kilometers below lie the cool blue waters of planet Earth. The much-anticipated last glide back to Earth for the Space Shuttle is currently scheduled for next Thursday, July 21.   Quiz: Can you identify what land masses are visible below the shuttle?",
  "hdurl": "http://apod.nasa.gov/apod/image/1107/atlantisapproach_nasa_4288.jpg",
  "media_type": "image",
  "service_version": "v1",
  "title": "Atlantis' Last Approach",
  "url": "http://apod.nasa.gov/apod/image/1107/atlantisapproach_nasa_900.jpg"
}
```

Unit 5, Lesson 4

Working with the Web: HTTP and URL Session

Learn the basics of how web data is sent and received, how URLs work, and how to fetch data for use in your app.

You'll build a playground that uses a delegate method called URL Session to fetch and display a photo.

Unit 5, Lesson 4

Lab: iTunes Search (Part 1)

Use URLSession to pull data down from the iTunes API and display it in the console.

Use a search query dictionary to configure a URL, which the URL session will use to fetch and print the correct data.