First struct File

This is a way to access / parse the Jason file

It will be retrieved as a json object not javascript object :

**import** Foundation

**struct** Flights {

**let** flightaPIaddress = // exapmle

<https://api.openweathermap.org/data/2.5/weather?&appid=b2bf7eb27ddb3765712dd03c280ef3c1&>

// following function will look for a particular flight

**func** fetchflight (cityName : String) {

**let** apilString = "\(weatherURL)&q=\(cityName)"

// print(apilString)

peformRequest(apilsString: apiString)

}

// here we will create NETWORKING ( steps to access / retrieve , parse and output

**func** peformRequest(apisString : String){

//1- create a apiURL

**if** **let** api = URL(string : apisString)

{

// create a url session

**let** session = URLSession(configuration: .default)

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

**if** error != **nil**

{

print(error!)

**return**

}

// this part will be parsed

**if** **let** safeData = data

{

parseJSON(flightData: safeData)

}

}

task.resume()

}

**func** parseJSON( flightData : Data)

{

// this is an object that can decode json object

**let** decoder = JSONDecoder()

**do**{

**let** decodedData = **try** decoder.decode(FlightData.**self**, from: flightData)

print(decodedData.aircraftCategory)

print(decodedData. quotedAmount)

print(decodedData.taxAmount)

}**catch** { print(error)}

}

}

}

Second struct file

THIS IS WHERE WE DEFINE THE KEYWORDS OF JSON FILE IN A STRUCT which they must match the ones accessed by decodedData

**import** Foundation

**struct** FlightData : Decodable

{

**let** aircraftCategory : String

**let** taxamount : Double

let createdDate: String

let salesperson : String

let status. : String

let Depature. : Double

let [legs] // this is array type in JSON

// we can access the array based on

decodedData.leg[o].id and so forth

id

orgin

dest

scheduledDeparture04:00

estimatedFlightTime