

QUICK TIPS : iOS



# TUTORIAL ON URLSESSION WITH RESULT TYPE AND GENERICS IN SWIFT.



Rohit Saini  
iOS Developer

QUICK TIPS : iOS

## URLSession:-

➤ URLSession is a class provide an API for performing network related tasks.

## Requirements:-

```
//MARK:- Protocol
protocol EndpointKind {
    //using associatedtype we can request any kind of data
    //Eg: Int,String,Data etc.
    associatedtype RequestData

    //prepare method is used to prepare our URLRequest
    //inout parameter is used so that can modify the original URLRequest outside the
    function.
    static func prepare(_ request: inout URLRequest,
                        with data: RequestData)
}
```

## QUICK TIPS : iOS

```
//MARK:- Enum
//We can add different kind of methods as per our requirements.
enum EndpointKinds {
    //Get Request method
    enum GET: EndpointKind {
        static func prepare(_ request: inout URLRequest,
                           with _: Void) {
            // Here we can do things like assign a custom cache
            // policy for loading our publicly available data.
            // In this example we're telling URLSession not to
            // use any locally cached data for these requests:
            request.cachePolicy = .reloadIgnoringLocalCacheData
        }
    }
    //Post Request method
    enum POST: EndpointKind {
        typealias RequestData = Data
        static func prepare(_ request: inout URLRequest, with data: RequestData) {
            request.httpBody = data
            request.httpMethod = "POST"
            request.addValue("application/json", forHTTPHeaderField: "content-type")
        }
    }
}
```

## QUICK TIPS : iOS

```
//MARK:- Endpoint
struct Endpoint<Kind: EndpointKind, Response: Decodable> {
    var path: String
    var queryItems = [URLQueryItem]()
}
//Extending the Endpoint for TypeSafety
extension Endpoint where Kind = EndpointKinds.POST, Response = LoginResponse {
    static var login: Self {
        Endpoint(path: "api/login")
    }
}
```

## QUICK TIPS : iOS

```
//Using URLComponents for building URLRequest
extension Endpoint {
    func makeRequest(with data: Kind.RequestData) → URLRequest? {
        var components = URLComponents()
        components.scheme = "https"
        components.host = "promotebio.com"
        components.path = "/housbrandt/" + path
        components.queryItems = queryItems.isEmpty ? nil : queryItems

        // If either the path or the query items passed contained
        // invalid characters, we'll get a nil URL back:
        guard let url = components.url else {
            return nil
        }

        var request = URLRequest(url: url)
        Kind.prepare(&request, with: data)
        return request
    }
}
```

## QUICK TIPS : iOS

//MARK:- URLSession

//Using Result Type and Generics for better Reusability

```
extension URLSession{
    typealias Handler<Response> = (Result<Response, Error>) → Void
    @discardableResult
    func request<Kind, Response>(_ endpoint:Endpoint<Kind, Response>,using
requestData: Kind.RequestData,
                                decoder: JSONDecoder = .init(),then handler:
@escaping Handler<Response>) → URLSessionDataTask{
    guard let request = endpoint.makeRequest(with: requestData) else {
        preconditionFailure("Error in making request")
    }
    let task = dataTask(with: request){ data,_,error in
        if let err = error{
            handler(.failure(err))
        }
        else{
            let response = try! decoder.decode(Response.self, from: data!)
            handler(.success(response))
        }
    }
    task.resume()
    return task
}
```

## QUICK TIPS : iOS

```
extension Encodable {  
    //MARK:- Converting object to bodyType of URLRequest  
    public func toJSON(_ encoder: JSONEncoder = JSONEncoder()) → Data {  
        guard let data = try? encoder.encode(self) else { return Data() }  
        return data  
    }  
}
```

## HOW TO USE:-

```
private func login(using session: URLSession = .shared){  
    let requestdata = LoginRequest(userId: "rohitsainier@gmail.com", password:  
"123456", latitude: 47, longitude: 57).toJSON()  
    session.request(.login, using: requestdata) { (result) in  
        switch result{  
            case .success(let response):  
                print(response)  
            case .failure(let err):  
                print(err)  
        }  
    }  
}
```

QUICK TIPS : iOS



LIKE, SHARE, ENJOY!



Rohit Saini  
iOS Developer