Alamofire(5.2)

Session

- URLSession
- SessionDelegate
- ServerTrustManager
- RequestInterceptor

Session负责创建和管理Request及其底层URLSession

- 初始化
  - 1，初始化默认的 URLSessionConfigure
  - 2，中介者模式：初始化URLSession对象，并将其代理通过 Session对象转移至SessionDelegate对象

通过中介者模式，通过 Session对象将URLSession的代理回调至SessionDelegate

Request

- 所有请求的超类
  - DataRequest — 通过将服务器响应下载到存储在内存中的数据来封装 URLSessionDataTask
  - DataStreamRequest — 封装 URLSessionDataTask并随时间流化来自HTTP连接的数据
  - UploadRequest — 封装URLSessionDataTask并将数据上传到远程服务器
  - DownloadRequest — 将响应数据下载到磁盘来封装URLSessionDownloadTask

主要的请求类

- 构建请求 — RequestConvertible
  - 1，生成URLRequest
  - 2，对请求参数进行编码 — ParameterEncoding
- url-Request-task的关系
  - 在创建请求时didCreateURLRequest，通过键值对的形式对 Request -Task 进行存储，存放在 Session对象中，建立对应关系之后，启动任务
  - 在SessionDelegate回调中，通过 task获取Request对象

```
//Session
func request(for task: URLSessionTask) -> Request? {
    dispatchPrecondition(condition: .onQueue(rootQueue))

    return requestTaskMap[task]
}
```

request-task一一对应

- 处理请求结果 — 在 SessionDelegate 中通过 task获取request对象，将请求结果存放在Requst对象中 — request.didReceive(data: data)
- 对每个请求进行检查和更改 — RequestAdapter

```
func adapt(
  _ urlRequest: URLRequest,
  for session: Session,
  completion: @escaping (Result<URLRequest, Error>) -> Void
) {
  var urlRequest = urlRequest
  if let token = TokenManager.shared.fetchAccessToken() {
    urlRequest.setValue("token \(token)", forHTTPHeaderField: "Authorization")
  }
  completion(.success(urlRequest))
}
```

- 对请求进行重试 — RequestRetrier

```
func retry(
  _ request: Request,
  for session: Session,
  dueTo error: Error,
  completion: @escaping (RetryResult) -> Void
) {
  let response = request.task?.response as? HTTPURLResponse
  //Retry for 5xx status codes
  if
    let statusCode = response?.statusCode,
    (500...599).contains(statusCode),
    request.retryCount < retryLimit {
      completion(.retryWithDelay(retryDelay))
  } else {
    return completion(.doNotRetry)
  }
}
```

- 可用url协议 — URLRequestConvertible — 有助于抽象并确保请求的端点的一致性。

Response

- 1，Request链式调用，每个方法都返回Request对象
- 2，response 方法，从Request中获取保存的网络数据。

NetworkReachabilityManager

检查网络是否可达

EventMonitor

记录网络请求和响应