

NETWORKING WITH URLSESSION



PART 5: URLSESSION COOKBOOK 2

TASK WITH URL / REQUEST

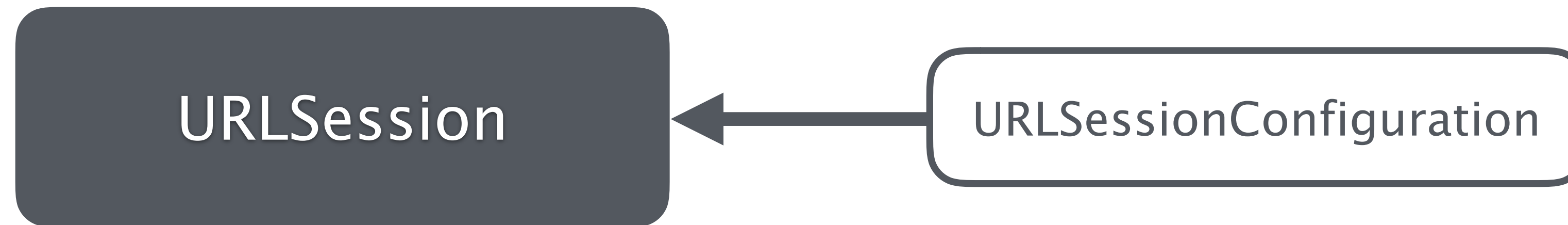


TASK WITH URL / REQUEST

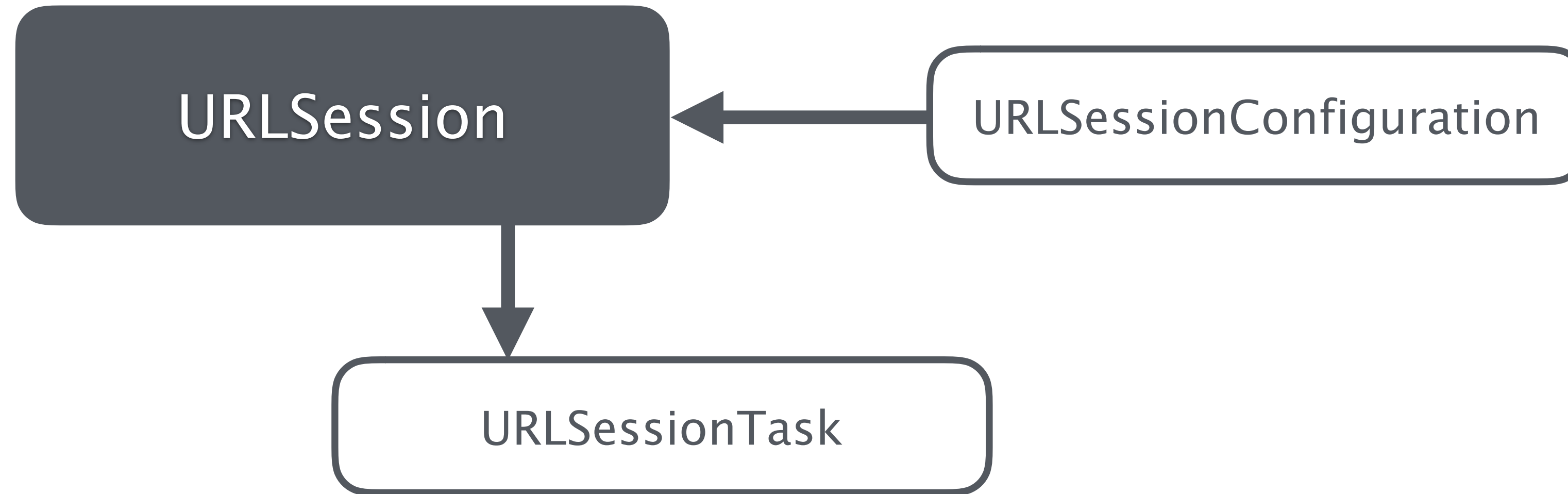
URLSessionConfiguration



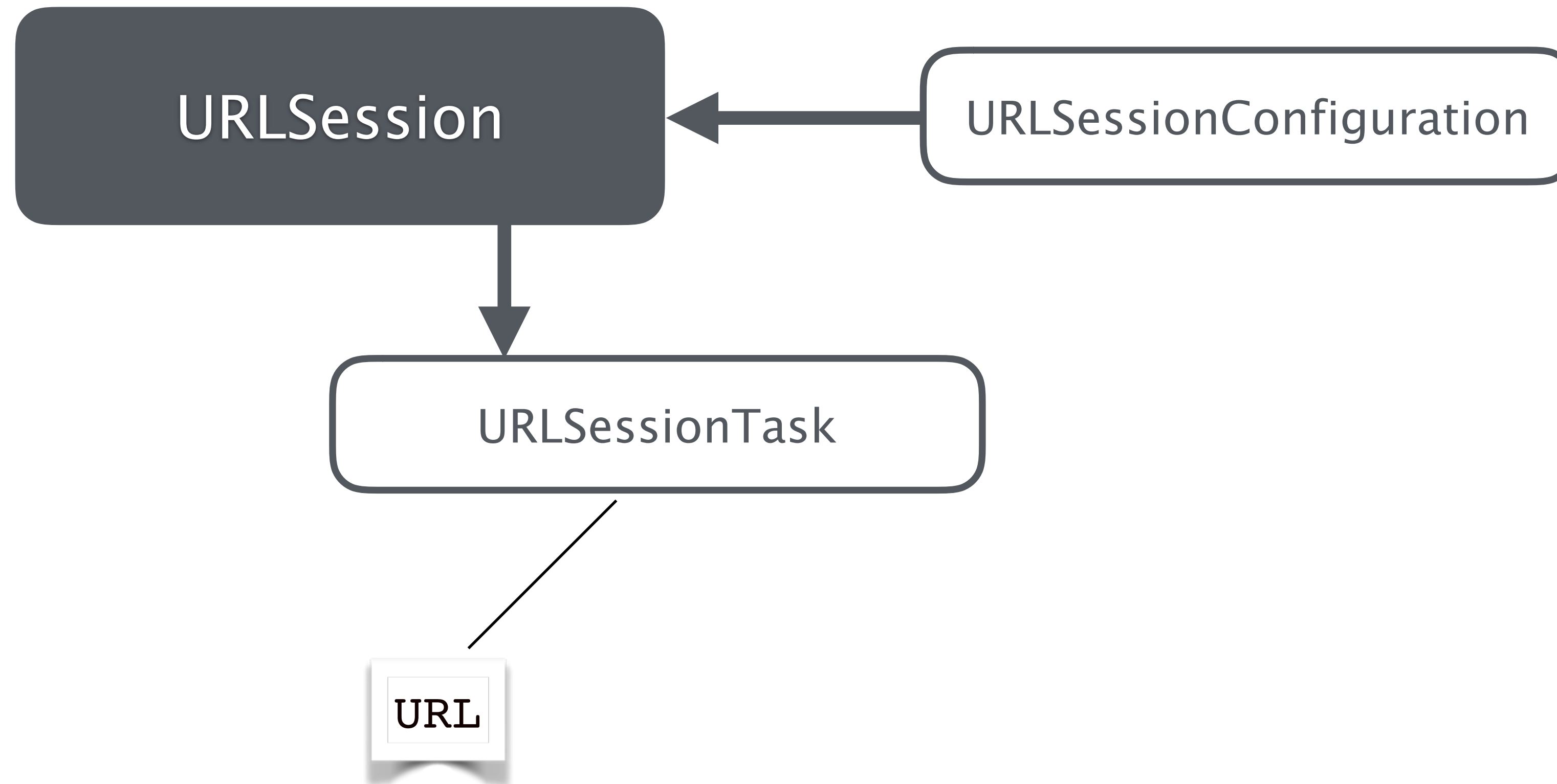
TASK WITH URL / REQUEST



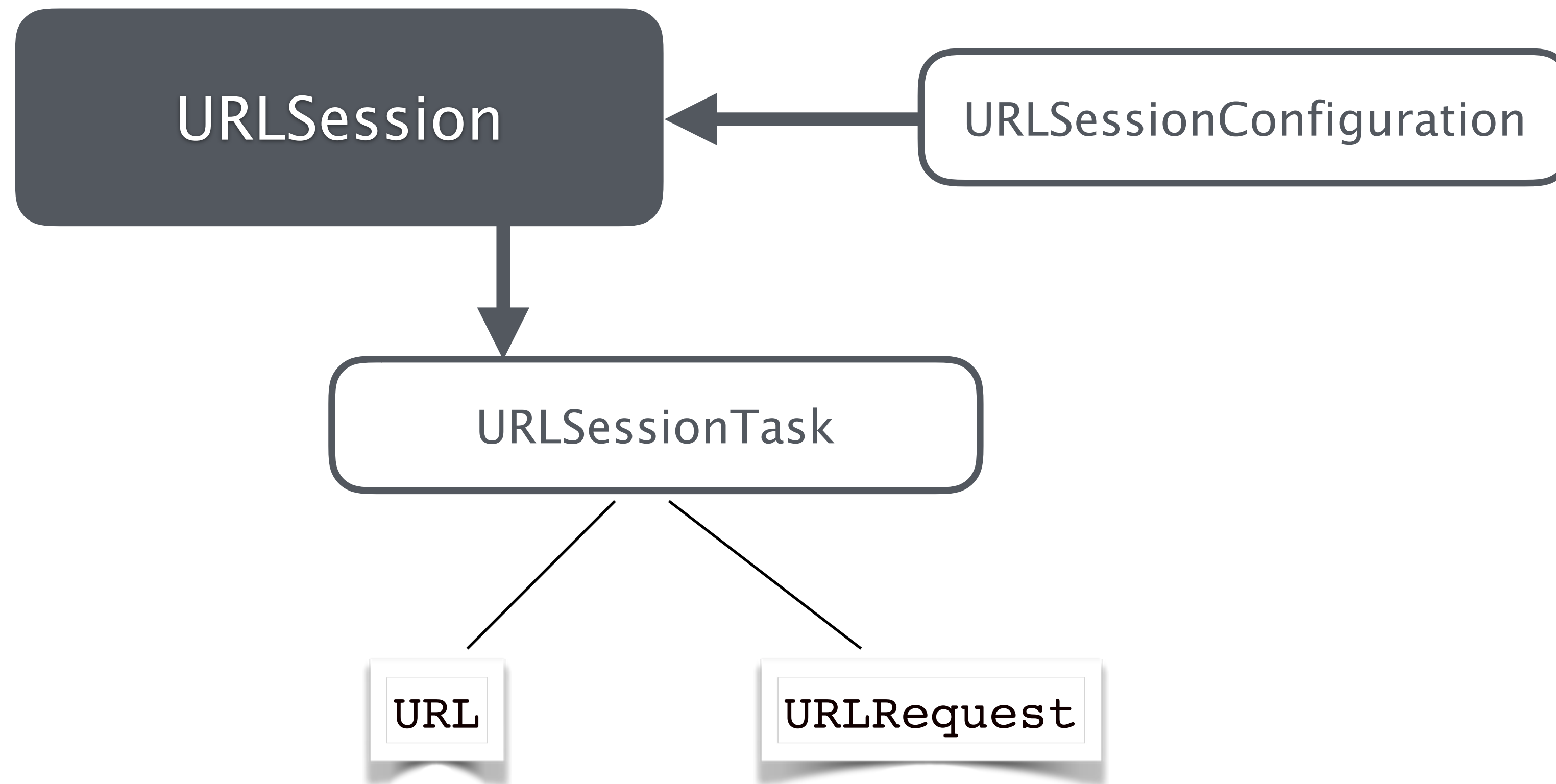
TASK WITH URL / REQUEST



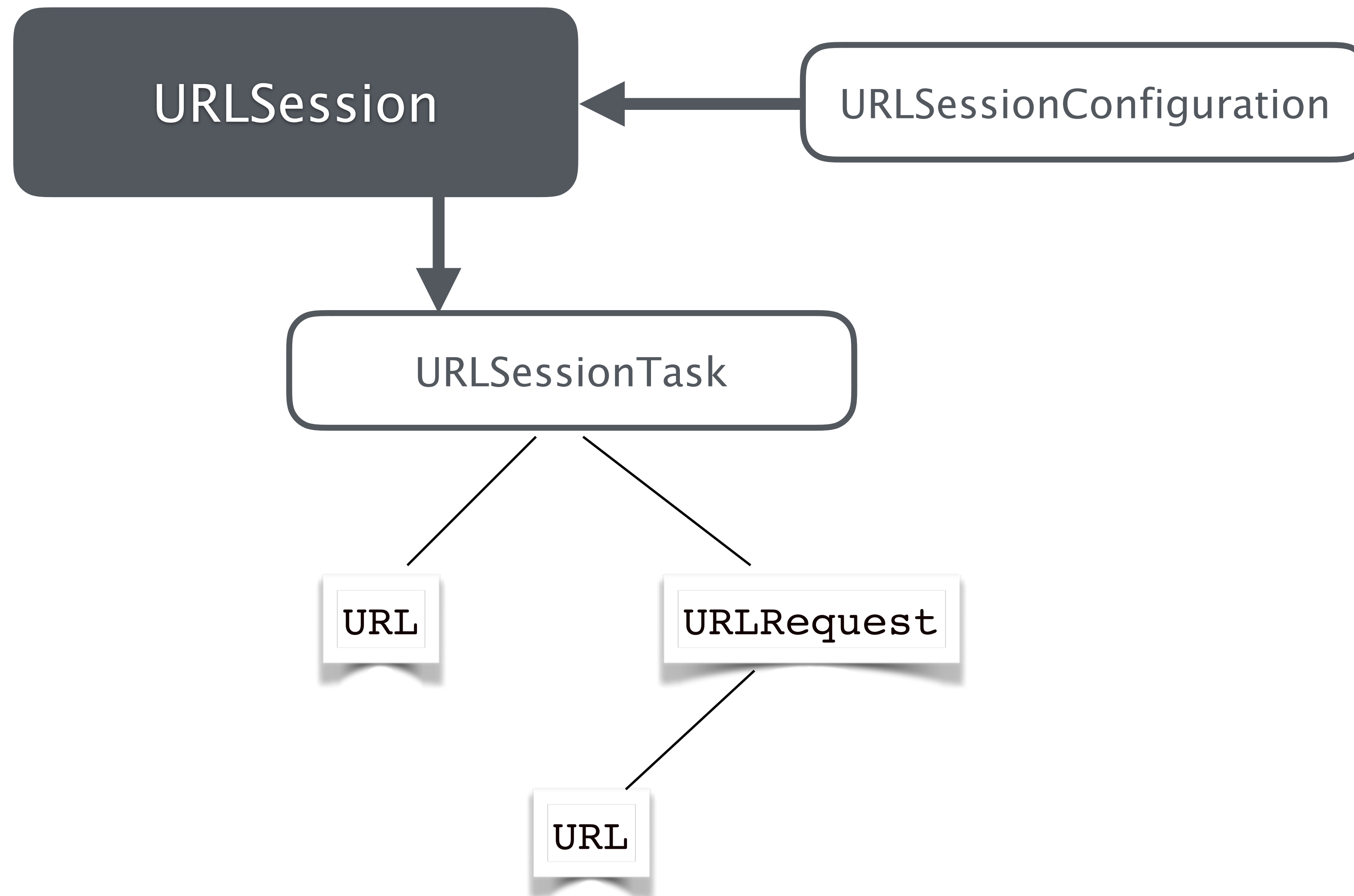
TASK WITH URL / REQUEST



TASK WITH URL / REQUEST



TASK WITH URL / REQUEST



URLRequest

```
init(url: URL)
init(url: URL, cachePolicy: NSURLRequest.CachePolicy,
      timeoutInterval: TimeInterval)
```



URLREQUEST

```
init(url: URL)
init(url: URL, cachePolicy: NSURLRequest.CachePolicy,
    timeoutInterval: TimeInterval)
```

```
var url: URL?
var httpMethod: String?
func value(forHTTPHeaderField: String)
var httpBody: String?
```



URLRequest

```
init(url: URL)
init(url: URL, cachePolicy: NSURLRequest.CachePolicy,
    timeoutInterval: TimeInterval)
```

```
var url: URL?
var httpMethod: String?
func value(forHTTPHeaderField: String)
var httpBody: String?
```

```
// may override session configuration values
var allowsCellularAccess: Bool
var timeoutInterval: TimeInterval
var networkServiceType: NSURLRequest.NetworkServiceType
var cachePolicy: NSURLRequest.CachePolicy
```



HTTP HEADERS



HTTP HEADERS

- ▶ General

- ▶ Cache-Control: max-age=<seconds>, Connection: keep-alive



HTTP HEADERS

- ▶ General

- ▶ Cache-Control: max-age=<seconds>, Connection: keep-alive

- ▶ Request

- ▶ Accept: application/json, X-stormpath-agent: "stormpath-sdk-ios/2.0.1"



HTTP HEADERS

▶ General

- ▶ Cache-Control: max-age=<seconds>, Connection: keep-alive

▶ Request

- ▶ Accept: application/json, X-stormpath-agent: "stormpath-sdk-ios/2.0.1"

▶ Response

- ▶ Age: <delta-seconds>, Server: Apache/2.4.1 (Unix)



HTTP HEADERS

▶ General

- ▶ Cache-Control: max-age=<seconds>, Connection: keep-alive

▶ Request

- ▶ Accept: application/json, X-stormpath-agent: "stormpath-sdk-ios/2.0.1"

▶ Response

- ▶ Age: <delta-seconds>, Server: Apache/2.4.1 (Unix)

▶ Entity

- ▶ Content-Language: en-US, Content-Encoding: gzip



REQUEST ROUTER ENUM



REQUEST ROUTER ENUM

- ▶ Cases: All possible requests



REQUEST ROUTER ENUM

- ▶ Cases: All possible requests
- ▶ Base endpoint: "http://localhost:3000/posts/"



REQUEST ROUTER ENUM

- ▶ Cases: All possible requests
- ▶ Base endpoint: "http://localhost:3000/posts/"
- ▶ var method: String
 - ▶ switch on case to return "GET", "POST" etc.



REQUEST ROUTER ENUM

- ▶ Cases: All possible requests
- ▶ Base endpoint: "http://localhost:3000/posts/"
- ▶ var method: String
 - ▶ switch on case to return "GET", "POST" etc.
- ▶ func asURLRequest() -> URLRequest
 - ▶ Build the request endpoint
 - ▶ Set up request parameters
 - ▶ Create and return request



DEMO



CHALLENGE TIME!

```
let urlString = "http://localhost:3000/posts/1"
```

```
let dictionaryMod = ["author": "an old friend", "title": "an old title"]  
as? [JSONDictionary]
```

```
putRequest.httpBody = putData
```

```
public enum PostRouter {  
  
    // Possible requests  
    case getAll  
    case get(Int)  
    case create([String: Any])  
    case update(Int, [String: Any])  
    case delete(Int)  
    . . .  
}
```