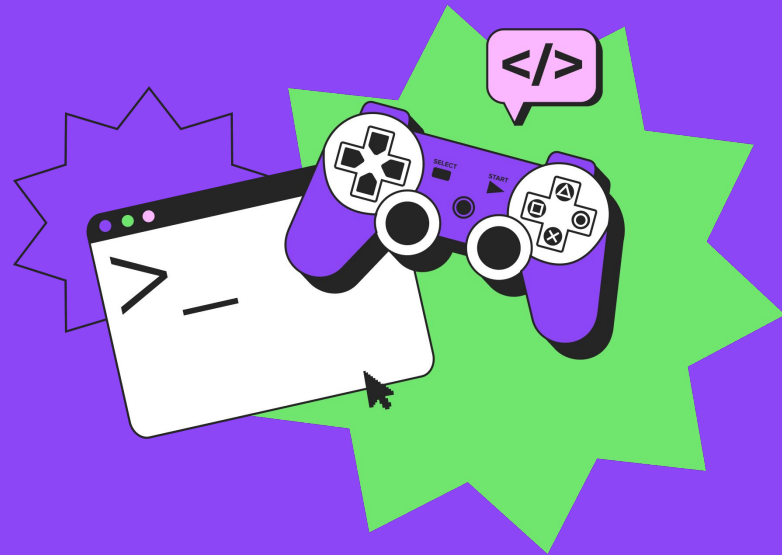


Работа с сетью в Objective – C

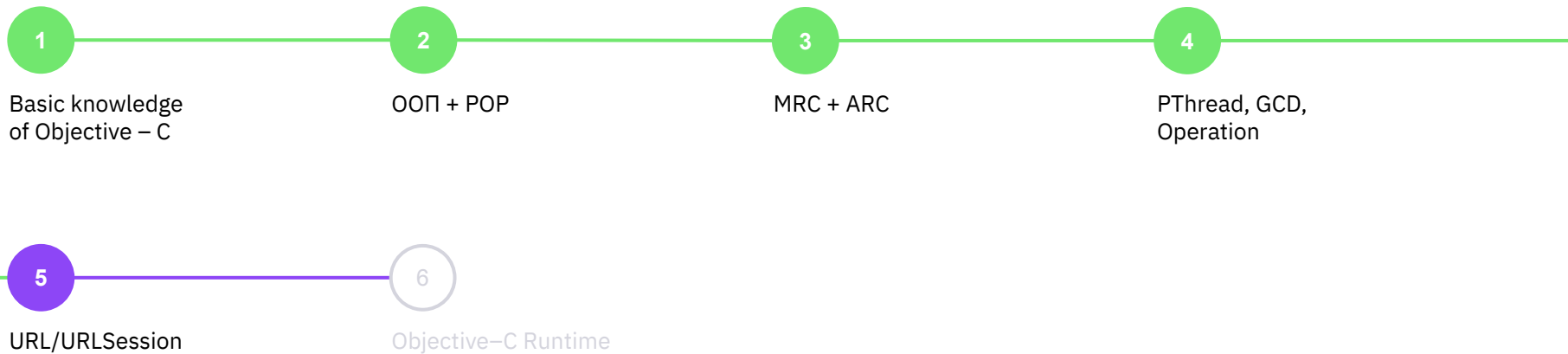
Урок 5

Знакомство с работой с сетью в языке Objective – C. Изучим некоторые базовые понятия для работы с сетью. Поговорим про базовые классы NSURLSession и другие, а также про форматы данных и передачу непосредственно на сервер, также научимся отлаживать нашу программу и отображать WebView.





План курса





Что будет на уроке сегодня

- 📌 Basic guidelines
- 📌 APIs, HTTP, Sync / Async
- 📌 NSURLConnection , NSURLRequest and NSURLConnectionDelegate
- 📌 Synchronous and Asynchronous requests
- 📌 Data formats
- 📌 JSON
- 📌 UIWebView
- 📌 Debugging
- 📌 3rd party





HTTP

HTTP (протокол передачи гипертекста) — протокол прикладного уровня для связи между распределенными системами.

HTTP позволяет осуществлять связь между различными хостами и клиентами.





HTTP





HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Conntection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)



HTTP

Стартовая линия

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Conntection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)



HTTP

HTTP Headers

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Conntection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Connnection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

Пустая линия

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Conntection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

Тело

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Conntection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

1. HTTP Method
(like GET, POST...)



HTTP

Методы запросов

GET

HEAD

POST

PUT

PATCH

DELETE

OPTIONS

TRACE

CONNECT



HTTP

Методы запросов

GET

HEAD

POST

PUT

PATCH

DELETE

OPTIONS

TRACE

CONNECT



HTTP

Методы запросов

GET

Create

PUT

Read

POST

Update

DELETE

Delete



HTTP

Методы запросов

GET

PUT

POST

DELETE

CRUD



HTTP

Методы запросов

POST

Метод POST используется для отправки сущности на указанный ресурс, что часто вызывает изменение состояния или побочные эффекты на сервере.



HTTP

Методы запросов

GET

Метод GET запрашивает представление указанного ресурса.
Запросы с использованием GET должны извлекать только данные



HTTP

Методы запросов

PUT

Метод PUT заменяет все текущие представления целевого ресурса полезной нагрузкой запроса.



HTTP

Методы запросов

DELETE

Метод DELETE удаляет указанный ресурс.



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Connnection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

1. Метод HTTP (например, GET, POST...)



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Connnection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

1. Метод HTTP
(например, GET, POST...)

2. Запросить цель



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Connnection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

1. Метод HTTP
(например, GET, POST...)

2. Запросить цель

3. HTTP-версия



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Connnection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

**Заголовки запросов можно
разделить на несколько групп:**
общие заголовки, заголовки
запросов и заголовки сущностей.



HTTP

Request

POST / HTTP 1.1

Host: *example.com*

User-Agent: *Mozilla/5...Safari/605.1.15)*

Accept: *application/json*

Accept-language: *en-US*

Connnection: *keep-alive*

Upgrade-Insecure-Requests: *1*

Content-Type: *multipart/form-data*

Content-Length: *345*

-12345678

(more data)

Не все запросы имеют его: запросы на выборку ресурсов, такие как GET, HEAD, DELETE, обычно не нуждаются в нем.



HTTP

Response

HTTP/1.1 200 OK

Server: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)



HTTP

Response

HTTP/1.1 200 OK

Server: Apache

Set-Cookie: csrftoken=.....

Vary: Cookie, Accept-Encoding

Connection: Keep-Alive

Date: Tue, 23 June 2020 19:00:00

Keep-Alive: timeout=5, max=999

Transfer-Encoding: chunked

Content-Encoding: gzip

Content-Type: application/json

Last-Modified: Tue, 23 June 2020...

(body)

1. Версия протокола



HTTP

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)

1. Версия протокола
2. Код состояния, указывающий на успех или неудачу запроса.



HTTP

Response

HTTP/1.1 200 OK

Server: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)



HTTP

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)

**Заголовки ответов можно
разделить на несколько групп:**
общие заголовки, заголовки
ответов и заголовки сущностей.



HTTP

Response

HTTP/1.1 200 OK

Sever: *Apache*

Set-Cookie: *csrftoken=.....*

Vary: *Cookie, Accept-Encoding*

Connection: *Keep-Alive*

Date: *Tue, 23 June 2020 19:00:00*

Keep-Alive: *timeout=5, max=999*

Transfer-Encoding: *chunked*

Content-Encoding: *gzip*

Content-Type: *application/json*

Last-Modified: *Tue, 23 June 2020...*

(body)

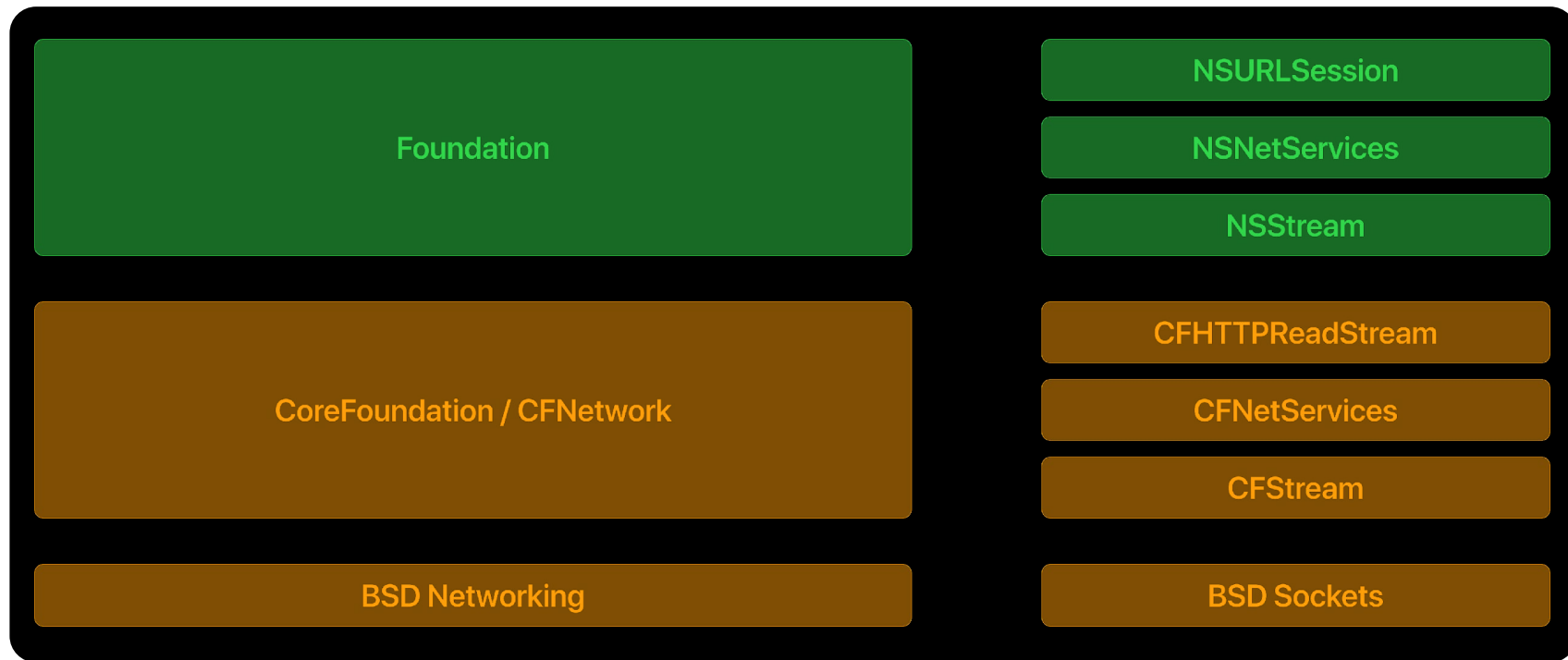
Не все ответы имеют один: ответы с кодом состояния, который в достаточной степени отвечает на запрос без необходимости соответствующей полезной нагрузки



Немного истории



Cocoa networking





NSURLSession концепции

- 📌 Session and configuration
- 📌 Session tasks
- 📌 Session delegate
- 📌 Credentials (credential storage)
- 📌 Cookies (cookie storage)
- 📌 Cache





NSURLSession configuration

Объект **NSURLSessionConfiguration** определяет поведение и политики, используемые при загрузке и выгрузке данных с помощью объекта **NSURLSession**.





NSURLSession configuration

Создание конфигурации

```
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration  
                                           defaultSessionConfiguration];
```

Возвращает стандартную конфигурацию, которая фактически аналогична сетевому стеку `NSURLConnection`, с тем же общим `NSHTTPCookieStorage`, общим `NSURLCache` и общим `NSURLCredentialStorage`.



NSURLSession configuration

Создание конфигурации

```
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration  
                                             ephemeralSessionConfiguration];
```

Возвращает предустановку конфигурации без постоянного хранилища для кешей, файлов cookie или учетных данных. Это было бы идеально для такой функции, как приватный просмотр.



NSURLSession configuration

Создание конфигурации

```
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration  
    backgroundSessionConfigurationWithIdentifier:@"com.rsschool.background" ]  
;
```

Возвращает предустановку конфигурации без постоянного хранилища для кешей, файлов cookie или учетных данных. Это было бы идеально для такой функции, как приватный просмотр.



NSURLSession configuration

Свойства, влияющие на перевод данных

TLS

Cookie polices

Cellular usage

Cache polices

Network service type

etc.



NSURLSession configuration

Свойства, влияющие на перевод данных

TLS

Cookie polices

Cellular usage

Cache polices

Network service type

etc.



NSURLSession configuration

Свойства, влияющие на перевод данных

TLS

Cookie polices

Cellular usage

Cache polices

Network service type

etc.



NSURLSession configuration

Свойства, влияющие на перевод данных

TLS

Cookie polices

Cellular usage

Cache polices

Network service type

etc.



NSURLSession configuration

Свойства, влияющие на перевод данных

TLS

Cookie polices

Cellular usage

Cache polices

Network service type

etc.



NSURLSession configuration

Свойства, влияющие на перевод данных

TLS

Cookie polices

Cellular usage

Cache polices

Network service type

etc.



NSURLSession configuration

Свойства, влияющие на перевод данных

TLS

Cookie polices

Cellular usage

Cache polices

Network service type

etc.



NSURLSession

Создание сеанса

```
NSURLSession *session = [NSURLSession sharedSession];
```

Для простых асинхронных запросов без делегатов



NSURLSession

Создание сеанса

```
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration
                                           defaultSessionConfiguration];

NSURLSession *session = [NSURLSession sessionWithConfiguration :configuration];
```

Для создания сеанса с указанной конфигурацией сеанса, но без делегата



NSURLSession

Создание сеанса

```
NSOperationQueue *queue = [NSOperationQueue new];

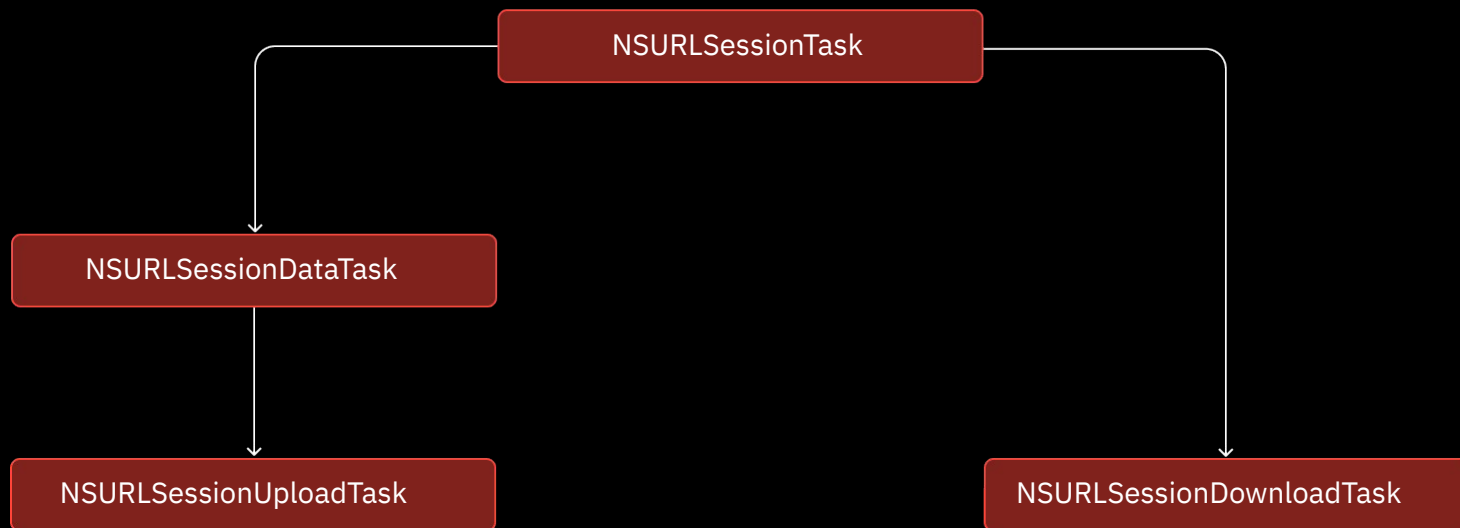
NSURLSessionConfiguration *configuration = [NSURLSessionConfiguration
                                           defaultSessionConfiguration];

NSURLSession *session = [NSURLSession sessionWithConfiguration :configuration
                                                             delegate:self
                                                             delegateQueue:queue];
```

Для создания сеанса с указанной конфигурацией сеанса, делегатом и очередью операций



NSURLSession tasks





NSURLSession tasks

Типы задач

NSURLSessionDataTask

NSURLSessionUploadTask

NSURLSessionDownloadTask



NSURLSession tasks

Типы задач

Data Task

Upload Task

Download Task



NSURLSession tasks

Типы задач

Data Task

WebSocket

Upload Task

Task

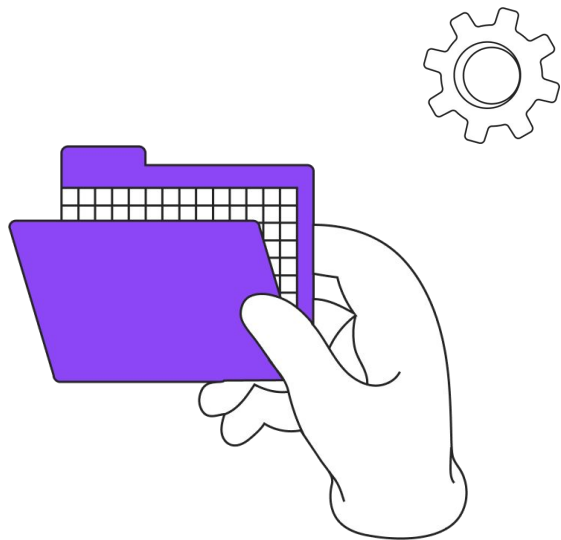
Download Task



NSURLSession tasks

Задача данных

Задачи данных отправляют и получают данные с помощью объектов NSData. Задачи данных предназначены для коротких, часто интерактивных запросов к серверу.





NSURLSession tasks


 Задача данных

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLSession *session = [NSURLSession sharedSession];

NSURLSessionDataTask *dataTask = [session dataTaskWithURL:url];
```



NSURLSession tasks

 Задача данных

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
NSURLSession *session = [NSURLSession sharedSession];

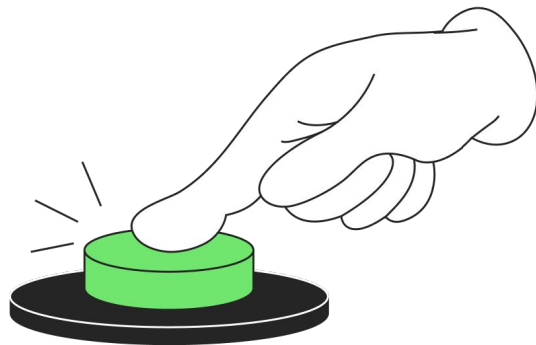
NSURLSessionDataTask *dataTask = [session dataTaskWithRequest:request];
```



NSURLSession tasks

Загрузить задачу

Задачи загрузки аналогичны задачам данных, но они также отправляют данные (часто в виде файла) и поддерживают фоновую загрузку, когда приложение не запущено.





NSURLSession tasks




Загрузить задачу

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
// some streamed additional request setup
NSURLSession *session = [NSURLSession sharedSession];

NSURLSessionUploadTask *uploadTask = [session uploadTaskWithStreamedRequest:request];
```



NSURLSession tasks

 Скачать задачу

Задачи загрузки извлекают данные в виде файла и поддерживают фоновую загрузку и выгрузку, когда приложение не запущено.





NSURLSession tasks



Скачать задачу

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLSession *session = [NSURLSession sharedSession];

NSURLSessionDownloadTask *downloadTask = [session downloadTaskWithURL:url];
```



NSURLSession tasks



Скачать задачу

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
NSURLSession *session = [NSURLSession sharedSession];

NSURLSessionDownloadTask *downloadTask = [session downloadTaskWithRequest:request];
```



NSURLSession tasks



Скачать задачу

```
NSData *data = [NSData new];  
NSURLSession *session = [NSURLSession sharedSession];  
  
NSURLSessionDownloadTask *downloadTask = [session downloadTaskWithResumeData :data];
```



NSURLSession tasks

Общие методы

По умолчанию все задачи приостановлены. Вызовите метод возобновления, чтобы запустить задачу.

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLSession *session = [NSURLSession sharedSession];
NSURLSessionDataTask *dataTask = [session dataTaskWithURL:url];

[dataTask resume];
```




NSURLSession tasks

Common methods

Вызов метода `suspend` для временной приостановки задачи.

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLSession *session = [NSURLSession sharedSession];

NSURLSessionDataTask *dataTask = [session dataTaskWithURL:url];
[dataTask resume];

[dataTask suspend];
```



NSURLSession tasks

Общие методы

Метод отмены вызова для отмены задачи.

```
NSURL *url = [NSURL URLWithString:@"https://rs.school"];
NSURLSession *session = [NSURLSession sharedSession];
NSURLSessionDataTask *dataTask = [session dataTaskWithURL:url];
[dataTask resume];

[dataTask cancel];
```



NSURLSession tasks

Общие свойства

state

Текущее состояние задачи – активна, приостановлена, в процессе отмены или завершена.



NSURLSession tasks

Общие свойства

priority

Относительный приоритет, с которым вы хотите, чтобы хост обрабатывал задачу, указанный в виде значения с плавающей запятой от 0,0 (самый низкий приоритет) до 1,0 (самый высокий приоритет).



NSURLSession tasks

Общие свойства

`currentRequest`

Объект запроса URL, который в данный момент обрабатывается задачей.



NSURLSession tasks

Общие свойства

`originalRequest`

Исходный объект запроса, переданный при создании задачи.



NSURLSession tasks

Общие свойства

response

Ответ сервера на текущий активный запрос.



NSURLSession delegate

NSURLSessionDelegate

NSURLSessionDataDelegate

NSURLSessionTaskDelegate

NSURLSessionDownloadDelegate



NSURLSession delegate



NSURLSessionDelegate



NSURLSessionDelegate

- Сообщения делегата, связанные с сеансом
- Обработка аутентификации соединения
- Аннулирование сеанса/ошибки



NSURLSession delegate



NSURLSessionDelegate



NSURLSessionDelegate

NSURLSessionTaskDelegate

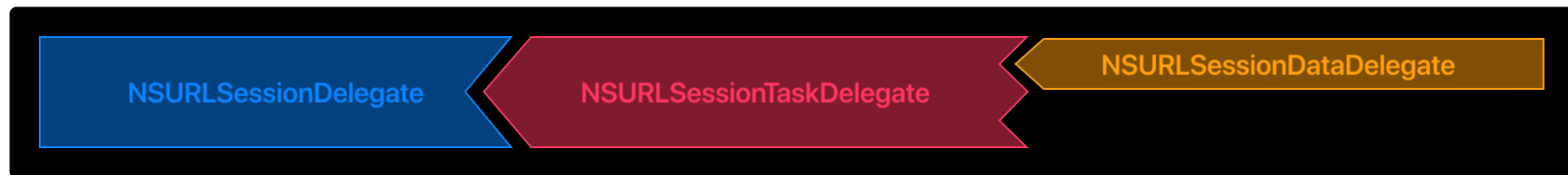
- Расширяет NSURLSessionDelegate
- Запрос обработки аутентификации
- Выполнение задачи/ошибки



NSURLSession delegate



NSURLSessionDelegate



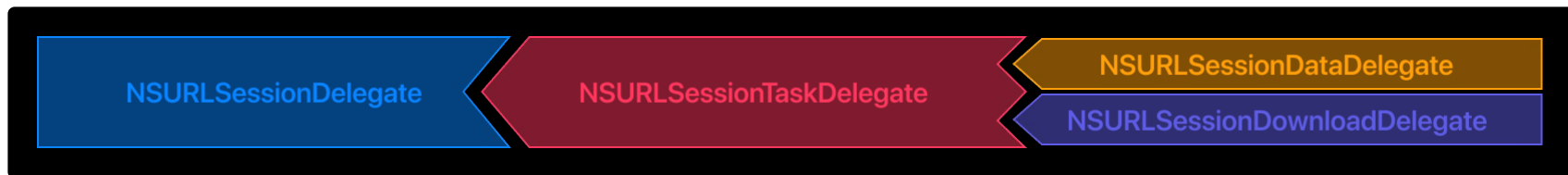
- Расширяет NSURLSessionTaskDelegate
- Доставляет байты по мере их передачи



NSURLSession delegate



NSURLSessionDelegate



- Расширяет NSURLSessionTaskDelegate
- Показывает прогресс во время передачи
- Предоставляет URL-адрес локального файла переданных ресурсов.



NSURLSession delegate

NSURLSessionDataTask



NSURLSession delegate

NSURLSessionDataTask

State:
Suspended

GET /foo.html HTTP/1.1



NSURLSession delegate

NSURLSessionDataTask

State:
Running

GET /foo.html HTTP/1.1



NSURLSession delegate

NSURLSessionDataTask

State:
Running

GET /foo.html HTTP/1.1

HTTP/1.1 200 OK

:didReceiveResponse:



NSURLSession delegate

NSURLSessionDataTask

State:
Running

GET /foo.html HTTP/1.1

HTTP/1.1 200 OK

```
<head><title>weeble</title>
</head><body>
```

:didReceiveResponse:

:didReceiveData:



NSURLSession delegate

NSURLSessionDataTask

State:
Running

GET /foo.html HTTP/1.1

HTTP/1.1 200 OK

```
<head><title>weeble</title>
</head><body>
```

:didReceiveResponse:

:didReceiveData:

:willCacheResponse:



NSURLSession delegate

NSURLSessionDataTask

State:
Finished

GET /foo.html HTTP/1.1

HTTP/1.1 200 OK

<head><title>weeble</title>
</head><body>

:didReceiveResponse:

:didReceiveData:

:willCacheResponse:

:didCompleteWithError:



NSURLSession delegate

NSURLSessionDownloadTask



NSURLSession delegate

NSURLSessionDownloadTask

State:
Suspended

GET /BigFile.tgz



NSURLSession delegate

NSURLSessionDownloadTask

State:
Running

GET /BigFile.tgz



NSURLSession delegate

NSURLSessionDownloadTask

State:
Running

GET /BigFile.tgz

HTTP/1.1 200 OK

:didWriteData:



NSURLSession delegate

NSURLSessionDownloadTask

State:
Running

GET /BigFile.tgz

HTTP/1.1 200 OK

file:///Location/BigFile.tgz

:didWriteData:

:didFinishDownloadingToUrl:



NSURLSession delegate

NSURLSessionDownloadTask

State:
Finished

GET /BigFile.tgz

HTTP/1.1 200 OK

file:///Location/BigFile.tgz

:didWriteData:

:didFinishDownloadingToUrl:

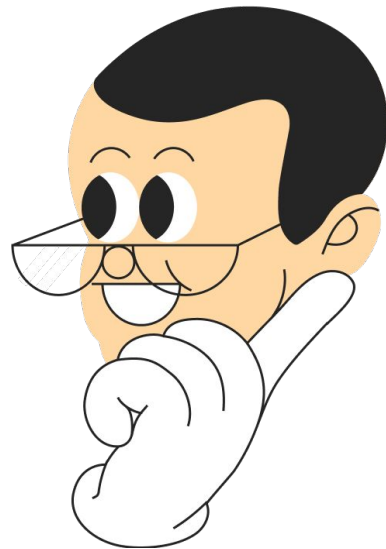
:didCompleteWithError:



NSURLSession delegate

Объект сеанса сохраняет сильную ссылку на делегата до тех пор, пока ваше приложение не выйдет или явно не аннулирует сеанс. Если вы не аннулируете сеанс, ваше приложение будет терять память до тех пор, пока приложение не завершится

Итак, вызовите методы сеанса `invalidateAndCancel` или `finishTasksAndInvalidate`, чтобы аннулировать ненужный сеанс.





Summary

Three-step process

1. Configuration
2. Session
3. Tasks





Summary

Best practice

Task 1 → Session 1

Task 2 → Session 2

Task 3 → Session 3



Task 1 ↘

Task 2 → Session 1

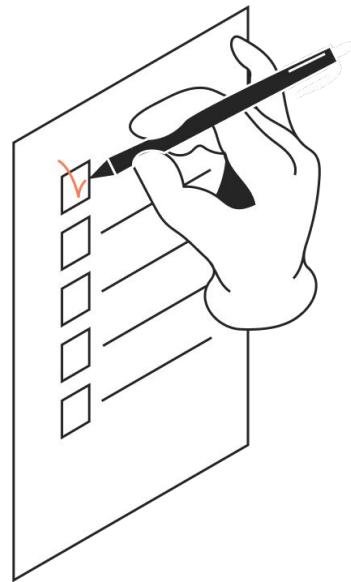
Task 3 ↗





Что почитать?

- 📌 Стивен Кочан. «Программирование на Objective-C».
- 📌 Скотт Кнастер, Вакар Малик, Марк Далримпл. «Objective-C. Программирование для Mac OS X и iOS».
- 📌 <https://www.objc.io/>
- 📌 <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/>





Вопросы?

Вопросы?



Вопросы?

