

استكشاف

4. بواسطة المدونة هذه مساعدة تمت

مقدمة

سابق ومهندس، لمنصة التقني والمدير المؤسس بصفتي. أنا بالجميع، مرحباً
الفعليّة للرسائل تطوّر أثناء خاصّة، مجال في واسعة خبرة لدي، في

أهمية

نطاق على استخداًه يتم واحد. اتصال عبر الاتجاه ثنائية اتصال قناة يوفر بروتوكول هو
التعليقات الفورية، المراسلة مثل الفعلي، الوقت في تفاعل تتطلب التي الحديثة التطبيقات في واسع
الفورية. الأسهم وأسعار التعاوني، التحريّر اللعابين، متعددة الألعاب الفورية،

ل- الحديثة التطبيقات

في التعليقات الفورية المراسلة التالى: المجالات في واسع نطاق على يستخدم
الفعلي الوقت في الأسهم أسعار التعاوني التعتدي اللعابين متعددة الألعاب الفعلي الوقت

تطور

الاستطلاع التحديثات. على للحصول متكرر بشكل الخادم بطلب العميل يقوم: الاستطلاع
اتصال جديدة. مع لومات تتوفر حتى مفتوحاً بالطلب الخادم يحتفظ: التطوير
اتصال. رؤوس على طلب لكل ويحتوي والاستقبال، للإرسال متعددة اتصالات يتطلب الاتجاه: ثنائي
الفعلي الوقت في قدرات ويوفر الاتجاه، ثنائي التطوير اتصال قيود على يتغلب: التطوير واحد
أقل. انتقال وزمن أعلى

على تنفيذ

تطبيقات في والخادم. العميل بين الاتجاه ثنائية اتصال قناة يوفر اتصال بروتوكول هو
البيانات بتبادل يسمح مما الخادم، مع الفعلي الوقت في اتصالات لإنشاء استخدام يمكن،
باستخدام تطبيقي في تنفيذ ليفي على سنن تعرف المقال، هذا في وكفاءة. بسرعة
.

مشروعنا. إلى مكتبة إضافية إلى نحتاج أولاً، مكتبة إضافية: 1 الخطوة
 في لتنفذ شرائعة مكتبة هي .

التالي: السطر وأضف Podfile ملف افتح 1.

```
pod 'Starscream', '~> 4.0.0'
```

المكتبة: لتثبيت في التالي الأمر بتشغيل قم 2.

```
pod install
```

اتصال إنشاء في البدء يمكننا الالمكتبة، تثبتت بعدد اتصال إنشاء 2: الخطوة

```
import Starscream
```

```
class WebSocketManager: WebSocketDelegate {
    var socket: WebSocket!

    init() {
        var request = URLRequest(url: URL(string: "wss://yourserver.com")!)
        request.timeoutInterval = 5
        socket = WebSocket(request: request)
        socket.delegate = self
        socket.connect()
    }

    func didReceive(event: WebSocketEvent, client: WebSocket) {
        switch event {
        case .connected(let headers):
            print("WebSocket is connected: \(headers)")
        case .disconnected(let reason, let code):
            print("WebSocket is disconnected: \(reason) with code: \(code)")
        case .text(let string):
            print("Received text: \(string)")
        case .binary(let data):
            print("Received data: \(data.count)")
        case .ping(_):
            break
        }
    }
}
```

```

        case .pong(_):
            break
        case .viabilityChanged(_):
            break
        case .reconnectSuggested(_):
            break
        case .cancelled:
            print("WebSocket is cancelled")
        case .error(let error):
            print("WebSocket encountered an error: \(String(describing: error))")
    }
}

func sendMessage(_ message: String) {
    socket.write(string: message)
}

func disconnect() {
    socket.disconnect()
}
}

```

أي في استخدامهم يمكننا WebSocketManager أن نشأ أن بعد الآن `WebSocketManager` استخدام: 3 الخطوة
الرسائل. واستقبال لإرسال التطبيق من جزء

```

let webSocketManager = WebSocketManager()

//
webSocketManager.sendMessage("Hello, Server!")

//
webSocketManager.disconnect()

```

مكتبة باستخدام `WebSocketManager` تطبيق في `WebSocketManager` تنفيذ كئي في تعلمنا المقال، هذا في الاختلاصة
يسمح مما الخادم، مع الفعلي الوقت في اتصالات لإنشاء `WebSocketManager` استخدام يمكن. `WebSocketManager`

الاتصال إعادة مثل الميزات من المزيدي ليشمل الممثال هذا توسيع يمكن وكفاءة. بسرعة البيانات بتبادل الأخطاء، وإدارة التلقائي

٥ نجوم، 4910 ،**٦** نجوم
نجوم، 435 ،**٧** نجوم، 1714 ،**٨**

استخدام

1. والاتصال: الهيئة 1.

```
SRWebSocket *webSocket = [[SRWebSocket alloc] initWithURLRequest:[NSURLRequest requestWithURL:[NSURL URLWithString:@"http://localhost:3000/socket.io/?EIO=3&transport=websocket"]]];
webSocket.delegate = self;
[webSocket open];
```

2. الرسالة: إرسال

```
[websocket send:@"Hello, World!"];
```

والأحداث. الواردة الرسائل مع للتعامل SRWebSocketDelegate طرق بتنفيذ ذقم الرسائل: استق بال 3.

مشاكل بأي المستخدمين وأخبر مناسب بشكل الأخطاء بمعالجة قم الأحداث: وإشعارات الأخطاء معالجة 4. الاتصال. في

لبروت وکول مفصل شرح

اتصال عبر **الاتجاه ثنائية اتصال** قناة يوفر اتصالات بروتوكول هو **تطبيقات** أي في أي صا استخدا مكن ولكن له الوب، وخوادم الم تصفحات في تنفيذه لي تم تصمي م واحد. الحاجة دون الفعلي الوقت في والخدم العميل بين البيانات إرسال بإمكانية **يتميز** عميل خادم. مكررة. **طلبات** إجراء إلى

کیا یہ عمل درست ہے؟

1. الى د م ص اف حة : □□□□□□□□□□

إلى [] طلب العميل إرسال []. ب- تعرف [] بمصافحة الاتصال يبدأ []
[]. إلى الاتصال ترقية طلب مع الخادم
[]. إلى الاتصال ترقية تأنيدي برسالة يرد فإنه الطلب، على الخادم وافق إذا []

```
GET /chat HTTP/1.1
Host: server.example.com
```

```

Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Version: 13

HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+x0o=

```

2. البيانات: نقل

- بين البيانات إرسال خلالها من يمكن الاتجاه ثنائي اتصال قناة إنشاء يتم المصافحة، إتمام بعد والخدم. الاعميل
- أو نصوص على تحتوي أن يمكن والتي صغيرة، إطارات شكل في البيانات إرسال يتم ثنائي. بيانات

3. الاتصال: إغلاق

- إغلاق إطار بإرسال الاتصال إغلاق الخادم أو الاعميل الطرفيين من لأي يمكن

ميزات:

- على يحافظ فإن فردية، طلبات على يعتمد الذي عكس على المسمتر: الاتصال التفاع. فترة طوال مفتوح اتصال
- المتكررة. بطلبات مقارنة العامة النفاقات تقليل يتم الكفاءة:
- النصوص. إلى بإضافة الثنائي البيانات إرسال يمكن الثنائي: للبيانات الدعم

استخدامات:

- الرددشة. تطبيقات مثل الفعلي الوقت في تحديثات تتطلب التي التطبيقات
- الإنترنت. عبر اللأعبيين متعددة الأعاب
- للأسعار. سرية تحديثات تتطلب التي المالية التطبيقات

بإستخدام بسيط مثال:

```

//      WebSocket
const socket = new WebSocket('ws://example.com/socket');

```


البروتوكول أساسيات

الاتصال عبر `WebSocket` الاتجاه ثنائي الاتصال قناة يوفر اتصالات بروتوكول هو `WebSocket` تطبيقي أي في أيضا استخداه يمكنه الوب، وخواص المصفحات في تنفيذه ليتم تصميده تم واحد. والخدم العميل بين الاتصال تحسني على العمل وهو `5` من كجزء `WebSocket` تقديم تم عميل خدم. `WebSocket` `WebSocket` `WebSocket` مثل القديمة بالتقنيات مقارنة.

البروتوكول: ل-الرئيسية الميزات

- الحاجة دون الوقت نفس في البيئات إرسال والخدم للعميل يمكن: `WebSocket` الاتجاه ثنائية 1. الآخر. الطرف انتظار إلى
- الطرفين. أحد قبل من إغلاقه يتم حتى مفتوحا يبقى الاتصال، إنشاء بمجرد: مستمرات اتصال 2.
- العامدة النفقات من يقلل مما الوزن، خفي بتسويق البيئات إرسال يتم: الخفي البيئات نقل 3. `WebSocket` مثل أخرى بروتوكولات مقارنة `WebSocket`.
- عبر والاتثنائية النصية البيئات إرسال يمكن: والاتثنائية النصية للبيئات دعم 4. `WebSocket`.

عمل كيفي:

- حيث، `WebSocket` `WebSocket` باسم تعرف `WebSocket` بمصافحة الاتصال يبدأ: `WebSocket` مصافحة 1. `WebSocket` بروتوكول إلى الاتصال ترقيّة الخدم من العميل يطلب.

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Version: 13
```

الترقية: على بالموافقة بالرد الخدم يقوم

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+x0o=
```

- تقسيمة يتم مستقل. بشكل البيئات إرسال والخدم للعميل يمكن الاتصال، إنشاء بعد: البيئات نقل 2. الاتصال. عبر إرسالها يتم صغيرة `WebSocket` إطارات إلى البيئات
- إغلاق إطار بإرسال الاتصال إغلاق الطرفين من لأي يمكن: الاتصال إغلاق 3. `WebSocket` `WebSocket`.

المتصفح: باستخدام بسيط مثال

```
//      WebSocket

const socket = new WebSocket('ws://example.com/socket');

//

socket.onopen = function(event) {
    console.log('Connection opened');
    socket.send('Hello Server!');
};

//

socket.onmessage = function(event) {
    console.log('Message from server:', event.data);
};

//

socket.onclose = function(event) {
    console.log('Connection closed');
};

//

socket.onerror = function(error) {
    console.error('WebSocket error:', error);
};
```

المتصفح: استخدامات

- اللعابون. متعددة الألعاب الحية، الدردشة مثل: التفاعلية التطبيقات
- الفورية. الإشعارات الأسهم، تحديثات مثل: الفورية التحديثات
- بعد. عن الأجهزة في التحكم مثل: الحقيقي الوقت في التحكم

المتصفح يوفر طريقة يوفر
الحدث. التطبيقات من للعدد مثال

http العميل: طلب المصافحة: لإتمام ب. الخاصة الترقية آلي تستخدم المصافحة: 1.
GET /chat HTTP/1.1 Host: server.example.com Upgrade: websocket Connection:


```
HTTP/1.1 101 websocket :
Sec-WebSocket-Accept: s3pPLMBiTraQ9kYGzzhZRbK+x0o= Sec-WebSocket-Protocol: chat
```

الآخرين. العملاء قبل من تزويره يمكن لا والذي Origin، رأس بإضافة تلقائيًا الدم تصرفح يقوم الأمن: 3.

```

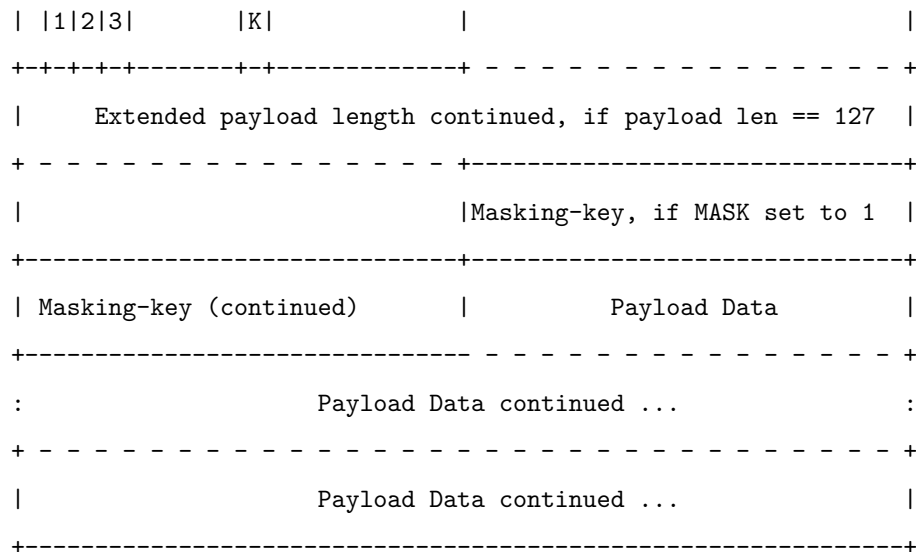
[] [][]:ws://host:port/path?query
[] [][]:wss://host:port/path?query

```

المرسلة البيانات تقسيم يتم واحد. **اتصال** فوق يعمل الاتجاه ثنائي اتصال بروتوكول هو **اتصال** بروتوكول يتكون مستقل. بشكل إطار لكل إرسال يتم حيث صغيرة. **اتصال** إطارات إلى **اتصال** عبر رؤى سية: أجزاء عدة من **اتصال** إطارات

- الرسالة. في الأخير الإطار هي هذه كانت إذا ما إلى يشرى: الأول البت
- الخ. إغلاق، ثنائىة، بيانات نص، مثل الإطار نوع تحدد: التحكم بتات
- مخفية. البيانات كانت إذا ما إلى يشرى: الإخفاء بت
- المرسلة. البيانات طول يحدد: المحمولة طول
- مضبوطة. الإخفاء بت كان إذا البيانات إخفاء يستخدم: الإخفاء مفتاح
- المرسلة. الفعلية البيانات: البيانات

0		1		2		3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
+--+--+--+-----+--+-----+-----+-----+																					
F R R R	opcode M				Payload len					Extended payload length											
I S S S	(4)	A	(7)					(16/64)													
N V V V	S										(if payload len==126/127)										



استخدام يتم المرسل. والبيانات الإطار خصائص تحدد ممتلئة أجزاء إلى الإطار تقسيم يتم المثال، هذا في `00000000` اتصال عبر وآمن فعال بشكل البيانات نقل لضمان التنسيق هذا.

لكل بت 1 3 2 1 1 الرسالة. من الأخير الجزء هو هذا أن إلى يشير بت 1: الإطار: هيكل
 المحمولة. بيانات تحليل لفي في حد بتات 4: `00000000` المستقبلي. للاستخدام محجوزة من هـ: `0x0`
 0xA: `0x9` الاتصال إغلاق: `0x8` ثنائي إطار: `0x2` نصي إطار: `0x1` متابعة إطار: `0x0`
 بيانات طول بتات 7: المحمولة طول لا. أم مقنعة المحمولة بيانات كانت إذا ما إلى يشير بت 1: `0000`
 المحمولة.

عن `00000000000000000000000000000000` المن تصف في الرجل هجمات لمنع يستخدم: `00000000` القناع مفتاح
 العميل. إطارات إخفاء طريقي

المصافحة إغلاق

الطرفين لكل على يجب الإغلاق. سبب يوضح نص على يحتوي أن يمكن: `00000000 00000000` الإغلاق إطار
 الإغلاق. إطار واستقبال إرسال

مثال

واحد بإطار مقنعة غير نصية رسالة 1 المثال

`0x81 0x05 0x48 0x65 0x6c 0x6c 0x6f`

`00000000` على تحتوي

قناع مع واحد بإطار نصية رسالة الثاني: المثال

0x81 0x85 0x37 0xfa 0x21 0x3d 0x7f 0x9f 0x4d 0x51 0x58

القناع. مفتاح مع 00000000 على تحتوي

قناع بدون مجزأة نصية رسالة الثالث: المثال

0x01 0x03 0x48 0x65 0x6c

0x80 0x02 0x6c 0x6f

000000 و 000000 إطارين: على تحتوي المجزأة الرسالة

المقدمة الموضوعات

0000000000000000 الممن تصرف في الرجل هجمات لمنع التعتيم يستخدم 0 التعتيم: وإزالة التعتيم بشكل إطار لكل التعتيم مفتاح اختياري تم 0 العميل. من يأتي 00000000 إطار لكل تعتيم يجب 0. 00000000. عشوائي.

بإطار التجزئة رسائل تبدأ 0 معروف. غير طول ذات بيانات لإرسال تُستخدم 0: 0000000000000000 التجزئة 1. تساوي 0000 علامة ذو بإطار وتنتهي، 0 تساوي 0000 علامة ذو

0 محددة. تشغيل أكواد على و 000000، 000000، إلحاق 0 مثل التحويلات إطارات تحتوي 0 التحويلات إطارات 000000000000 اتصال حالة لإدارة الإطارات هذه تُستخدم

التوسيع قابلية

إطار لكل في التحويلات يمكن 0 الرسالة: جسم في التتطبيق بيانات قبل الموسعة البيانات وضع يمكن هناك لكانت إذا 0 المستقبلي. للتعريفات العمليات رموز بعض حجز يتم 0 المحجوزة. البتات باستخدام المحجوزة. البتات استخدام يمكن العمليات، رموز من المزيد إلى حاجة

إرسالها اختياري يمكن إطارات، في البيانات تغليف يتم 0. 000000 حالة في الاتصال أن من التأكد يجب 0 الإرسال: بنوع المستقبل الطرف لإعلام صحيحة، الأول الإطار قيمة تكون أن يجب 0 كقيمة. البيانات لكانت إذا مجزأة 1. إلى الأخير للإطار 0000 تعييين يجب 0 ثنائي. أو 0 نص البيانات

بيانات أي إرسال يتم لا الإغلاق، إطار إرسال بعد 0 الإغلاق. إطار إرسال الطرفيين لكل يمكن 0 المصافحة: إغلاق ذلك. بعد تلقيها يتم بيانات أي تجاهل يتم الإغلاق، إطار استلام عند 0 أخرى.

استقبال أو إرسال بعد 0 الأساسي. 0000 الاتصال إغلاق يعني 0000000000 اتصال إغلاق 0 الاتصال: إغلاق تصبح الأساسي، 0000 الاتصال إغلاق عند 0 الإغلاق. طور في 0000000000 اتصال حالة تصبح الإغلاق، إطار مغلقًا. 0000000000 اتصال حالة

المراجع

□ □□□□□□□ □□: □□ 6455

رابطہ: [ہیروئن کی دوا](#)؟

رابطہ:

والتقدير الشكر

معي التواصل في التناقضات، أو الأسئلة من المزيدي لديكم كانت إذا اهتمامكم. على جميعاً لكم شكراً
 . أو .