

استكشاف

4. بواسطة المدونة هذه مساعدة تمت

مقدمة

سابق ومهندس، لمنصة التقني والمدير المؤسس بصفتي. أنا بالجميع، مرحبًا
الفورية للرسائل لتطوير أثناء خاصة، مجال في واسعة خبرة لدي، في

أهمية

نطاق على استخدامه 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، التحريز للعابيين، متعددة الألعاب الفورية،

ل- الحديثة التطبيقات

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

تطور

الاستطلاع التحديثات. على للحصول متكرر بشكل الخادم بطلب العميل يقوم: الاستطلاع
اتصال جديدة. مع لومات تتوفر حتى مفتوحًا بالطلب الخادم يحتفظ: التطوير
اتصال. رؤوس على طلب لكل ويحتوي والاستقبال، للإرسال متعددة اتصالات يتطلب الاتجاه: ثنائي
الفعلي الوقت في قدرات ويوفر الاتجاه، ثنائي 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 تطبيقات في تنفيذ ليفي على سنتعرف المقال، هذا في وكفاءة. بسرعة
.

مشروعنا. إلى مكتبة إضافة إلى نحتاج أولاً، مكتبة إضافة 1 الخطوة
في مكتبة لتنفذ في مكتبة هي.

1. التالي: السطر وأضف Podfile ملف افتح

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

2. المكتبة: لتثبيت في التالي الأمر بتشغيل قم

```
pod install
```

اتصال إنشاء في البدء يمكننا المكتبة، تثبيت بعد الاتصال 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 نشأن أن بعد الآن `WebSocketManager` استخدمنا `3` الخطوة
الرسائل. واستقبل لإرسال التطبيق من جزء

```

let websocketManager = WebSocketManager()

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

//
websocketManager.disconnect()

```

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

الاتصال إعادة مثل الميزات من المزيد ليشمل المثل هذا توسيع يمكن وكفاءة. بسرعة البيانات بتبادل الأخطاء. وإدارة التلقائي

نجوم 4910، لنظام الشهيرة مكتبات، نجوم 435، نجوم 1714،

استخدام

1. والاتصال: التهيئة

```
SRWebSocket *webSocket = [[SRWebSocket alloc] initWithURLRequest:[NSURLRequest requestWithURL:[NSURL URLWithString:@"ws://example.com"]]];
webSocket.delegate = self;
[webSocket open];
```

2. الرسالة: إرسال

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

3. والأحداث. الواردة الرسائل مع للتعامل SRWebSocketDelegate طرق بتنفيذ قم الرسائل: استقبل

4. مشكل بأي المستخدم وأخبر مناسب بشكل الأخطاء بمعالجة قم الأحداث: وإشعارات الأخطاء مع معالجة الاتصال. في

لبروتوكول مفصل شرح

الاتصال عبر ثنائي الاتصال قناة يوفر اتصالات بروتوكول هو تطبيق أي في أيضاً استخدمه يمكن ولكنه الوب، وخوادم المتصفحات في تنفيذ هذه ليتم تصميجه تم واحد. الحاجة دون الوقت في والخدم العميل بين البيانات إرسال بإمكانية يميز عميل خادم. متكررة. طلبات إجراء إلى

؟

1. اليد مصافحة:

إلى طلب العميل يرسل. ب- تُعرف بـ بمصافحة الاتصال يبدأ إلى الاتصال ترقيية طلب مع الخادم. إلى الاتصال ترقيية يرده إنّه الطلب، على الخادم وافق إذا

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



```

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

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

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

// (Node.js ws)
const WebSocket = require('ws');
const wss = new WebSocket.Server({ port: 8080 });

wss.on('connection', function connection(ws) {
    ws.on('message', function incoming(message) {
        console.log('received: %s', message);
    });

    ws.send('Hello Client!');
});

```

الأداء في كبرية تحسينات ويوفر الفعلي، الوقت في تفاعلية تطبيقات لبناء قوية أداة هو `ws` مقارنة `socket.io`.

يعتمد أمن نموذج إضافة تمت الأمان: نموذج `ws` تحسينات: عدة ويقدّم `ws` بروتوكول فوق `ws` يعمل من فذ على متعددة خدمات يدعم والبروتوكولات: العناوين تسمية `ws` المتصفحات. في المصدّر من التحقق على `ws` لحزم مشابهة إطارات بآلية `ws` تعزيتم الإطارات: آلية `ws` واحد. `ws` عنوان على متعددة نطاقات وأسماء واحد نظيف. بشكل الاتصال إغلاق يضمن الإغلاق: مصافحة `ws` الطول. على قيود دون

بروتوكول أساسيات

الاتصال عبر `WebSocket` الاتجاه ثنائي الاتصال قناة يوفر اتصالات بروتوكول هو `WebSocket` تطبق أي في أيضاً استخدمه يمكن ولكنه الوب، وخواص المتصفحات في تنفيذه ليتم تصميجه تم واحد. والخدم العميل بين الاتصال تحسب على يعمل وهو 5، من كجزء `WebSocket` تقديم تم عميل خادم. `WebSocket` مثل القديمة بالتقنيات مقارنة.

المميزات:

- الحاجة دون الوقت نفس في البيانات إرسال والخدم للعميل يمكن: `WebSocket` الاتجاه ثنائية الآخر. الطرف انتظار إلى
- الطرفين. أحد قبل من إغلاقه يتم حتى مفتوحاً يبقى الاتصال، إنشاء بمجرد: مستمر اتصال
- العامة النفقات من يقلل مما الوزن، خفيف بتسويق البيانات إرسال يتم: الخفيف البيانات نقل `WebSocket` مثل أخرى بروتوكولات مقارنة `WebSocket`.
- عبر والاتجاهية النصية البيانات إرسال يمكن: والاتجاهية النصية للبيانات دعم `WebSocket`.

عملية:

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

```
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 الخادم يقوم

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

- تقسيم يتم مستقر. بشكل البيانات إرسال والخدم للعميل يمكن الاتصال، إنشاء بعد: البيانات نقل الاتصال. عبر إرسالها يتم صغيرة `WebSocket` إطارات إلى البيانات `WebSocket`.
- إغلاق إطار بإرسال الاتصال إغلاق الطرفيين من لأي يمكن: الاتصال إغلاق `WebSocket`.


```
Upgrade      Sec-WebSocket-Key: dGh1IHNhbXBsZSBub25jZQ==      Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat      Sec-WebSocket-Version: 13
```

```
␣ استجابة http      HTTP/1.1 101      :      websocket      :
      Sec-WebSocket-Accept: s3pLMBiTxaQ9kYGzZhZRbK+x0o=      Sec-WebSocket-Protocol: chat
```

2. ثنائيات، وبيانات 8 بت نسقي نصوص على إطارات تحتوي أن يمكن البيانات: نقل و. و. والإغلاق، مثل تحكم وإطارات

3. الآخري. العملاء قبل من تزويره يمكن لا والذي Origin، رأس بإضافة تلاقئها المتصفح يقوم الأمان: .

عنوان

```
␣ :ws://host:port/path?query
␣ :wss://host:port/path?query
```

إطارات بروتوكول

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

1. الرسالة. في الأخير الإطار هي هذه كانت إذا ما إلى يشير: الأول البت
2. الخ. إغلاق، ثنائيات، بيانات نص، مثل الإطار نوع تحدد: التحكم بتات
3. مخفية. البيانات كانت إذا ما إلى يشير: الإخفاء بت
4. المرسلة. البيانات طول يحدد: الحمولة طول
5. مضبوطًا. الإخفاء بت كان إذا البيانات لإخفاء يستخدم: الإخفاء مفتاح
6. المرسلة. الفعلي البيانات: البيانات

إطار على مثال

```

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 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) |
```


