

Dart How to code a simple web-socket echo app

Asked 7 years, 11 months ago Active 7 years, 11 months ago Viewed 9k times

I've been attempting to learn enough html, css, and Dart to create my first web page and all is going well, **except** that I do not understand how to create a simple page and a server side web-socket server that will just echo it back. The examples that I find tend to illustrate other Dart tools and either connect to echo server on the web or do other things that make their code not simple for a newbie.

I've tried to simplify Seth Ladd's [example](#) "dart-example-web-sockets-client" as the 'best' example. I can receive what is sent from the page, repackage it and *think* i'm sending it back but absolutely nothing happens on the web page. I start the page by clicking on the URL returned when the web-server is run from inside the Dart editor. Since the page is not, AFAIK, run in the debugger I'm hampered in diagnosing the error.

Here is simplified code from Seth's server:

```
void handleEchoWebSocket(WebSocket webSocket) {
  log.info('New WebSocket connection');

  // Listen for incoming data. We expect the data to be a JSON-encoded String.
  webSocket
    .map((string) => JSON.decode(string))
    .listen((json) {
      // The JSON object should contain a 'request' entry.
      var request = json['request'];
      switch (request) {
        case 'search':
          var input = json['input'];
          log.info("Received request '$request' for '$input'");
          var response = {
            'response': request,
            'input': input,
          };
          webSocket.add(JSON.encode(response)); // can't detect page receiving
        this:
          log.info("Echoed request..$request $input"); // correct data
          break;

        default:
          log.warning("Invalid request: '$request'");
      }
    }, onError: (error) {
      log.warning('Bad WebSocket request');
    });
}
```

This example took the user input using it as input to two search engines, packaged the results and returned them to the page for display creating new DOM elements on the fly.

I just need to be pointed to a simple example that will echo what is submitted.

[websocket](#) [dart](#) [dart-webui](#) [dart-editor](#)

Share Edit Follow Flag

edited Mar 16, 2014 at 11:05

 Günter Zöchbauer
542k ● 184 ● 1870 ● 1487

asked Mar 14, 2014 at 23:27

 Nate Lockwood
2,697 ● 5 ● 24 ● 30

1 Answer

Sorted by: Highest Score (default) ▾

Here is a simple websocket client/server echo example. Messages doesn't show in browser window, but they are printed in console window. You have to start server.dart and main.dart separately. Both processes print messages to their own console window. Edit: I added an output div for displaying the message also in browser.

binlws_server.dart:

```
import "dart:convert";
import 'dart:io';
import 'package:route/server.dart' show Router;

void handleWebSocket(WebSocket webSocket) {
  // Listen for incoming data. We expect the data to be a JSON-encoded String.
  webSocket
    .map((string)=> JSON.decode(string))
    .listen((json) {
      // The JSON object should contains a 'echo' entry.
      var echo = json['echo'];
      print("Message to be echoed: $echo");
      var response={"response": "$echo"};
      webSocket.add(response);
    }, onError: (error) {
      print('Bad WebSocket request');
    });
}

void main() {
  int port = 9223;

  HttpServer.bind(InternetAddress.LOOPBACK_IP_V4, port).then((server) {
    print("Search server is running on "
      "http://${server.address.address}:$port/");
    var router = new Router(server);
    // The client will connect using a WebSocket. Upgrade requests to '/ws' and
    // forward them to 'handleWebSocket'.
    router.serve('/ws')
      .transform(new WebSocketTransformer())
      .listen(handleWebSocket);
  });
}
```

weblindex.html:

```
<!DOCTYPE html>

<html>
<head>
  <meta charset="utf-8">
  <title>WebSocket echo</title>
</head>
<body>
  <p>WebSocket test</p>
  <div id="output"></div>

  <script type="application/dart" src="main.dart"></script>
</body>
</html>
```

web/main.dart:

```
library main;

import 'dart:async';
import 'dart:convert';
import 'dart:html';

class WebSocketService {

  WebSocket websocket;

  WebSocketService() {
    connect();
  }

  void connect() {
    websocket = new WebSocket('ws://127.0.0.1:9223/ws');
    websocket.onOpen.first.then((_) {
      onConnected();
      sendws("Hello websocket server");
      websocket.onClose.first.then((_) {
        print("Connection disconnected to ${websocket.url}");
        onDisconnected();
      });
    });
    websocket.onError.first.then((_) {
      print("Failed to connect to ${websocket.url}. "
        "Please run bin/server.dart and try again.");
      onDisconnected();
    });
  }

  void onConnected() {
    websocket.onMessage.listen((e) {
      onMessage(e.data);
    });
  }

  void onDisconnected() {
    print("Disconnected, trying again in 3s");
    new Timer(new Duration(seconds:3), (){
      connect();
    });
  }

  void onMessage(data) {
    var json = JSON.decode(data);
    var echoFromServer = json['response'];
    print("Received message: $echoFromServer");
    var output=querySelector('#output');
    output.text="Received message: $echoFromServer";

    new Timer(new Duration(seconds:3), (){ //Send a new message to server after
3s      String now = new DateTime.now().toString();
      sendws("Time: $now");
    });
  }

  void sendws(String msg){
    var request = '{"echo": "$msg"}';
    print("Send message to server: $request");
    websocket.send(request);
  }
}


void main() {
  WebSocketService ws=new WebSocketService();
}
```

Share Edit Follow Flag

edited Mar 15, 2014 at 10:19

 **Günter Zöchbauer**
542k ● 184 ● 1870 ●
1487

answered Mar 15, 2014 at 8:24

 **grohly**
1,824 ● 1 ● 14 ● 17

▲ I think that this is exactly what I needed. Serendipitously you used 'library main,' which motivated this newbie to re-read 'A Tour of the Dart Libraries' and pick up some details I ignored the first times around. – [Nate Lockwood](#) Mar 15, 2014 at 16:38

▲ @natelockwood Glad I could help, I was fiddling around with websockets when you placed your question. – [grohly](#) Mar 15, 2014 at 17:03

2 ▲ This answer is now out of date because the Dart package 'route' is discontinued. This stack overflow answer has a better websocket server implementation: stackoverflow.com/a/52591092/76295 And if you just want to test the server, you can use the browser-based websocket client on this page websocket.org/echo.html – [Mark](#) Jul 14, 2019 at 6:44