Intrinio IEX Real-Time Price Feed

The Intrinio IEX Real-Time Price Feed uses the **WebSockets** protocol to push out prices to clients in real time. For more information on WebSockets, visit:

https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

JavaScript Integration

Step 1: Download the Phoenix WebSocket client library at https://s3.amazonaws.com/intrinio-iex-realtime/phoenix.js. Include this file before your application code.

```
<script type="text/javascript" src="/phoenix.js"></script>
```

Step 2: In your application code, make a GET request to https://realtime.intrinio.com/auth, providing as Basic Authorization you Intrinio *API Username* and *API Password* (which you can retrieve at http://data.intrinio.com). The response will be the authorization token that you will need in order to connect to the Intrinio WebSocket.

```
var username = "YOUR_API_USERNAME";
var password = "YOUR_API_PASSWORD";
var auth_url = "https://realtime.intrinio.com/auth";

$.ajax({
  type: "GET",
  url: auth_url,
  beforeSend: function(xhr) {
    xhr.setRequestHeader("Authorization", "Basic " + btoa(username + ":" + password)
  },
  success: function(token) {
    connect(token);
  },
  error: function(xhr, status, error) {
    console.error("Error connecting: ", error);
  }
});
```

Step 3: Use the provided auth token to connect to the Intrinio WebSocket (at wss://realtime.intrinio.com/socket).

```
var socket_url = "wss://realtime.intrinio.com/socket";

var connect = function(token) {
  var socket = new Phoenix.Socket(socket_url, { params: { token: token } });
  socket.connect();
  socket.onClose(function (e) {
    return console.log("CLOSE", e);
  });

  startListening(socket);
};
```

Step 4: Join a channel to start listening for price updates. The name of each channel is "iex:securities:[ticker]", such as "iex:securities:AAPL" or "iex:securities:MSFT".

```
var ticker = "AAPL";

var startListening = function(socket) {
  var channel = socket.channel("iex:securities:" + ticker, {})

  channel.join()
    .receive("ok", function () {
      return console.log("joined ok");
    })
    .receive("ignore", function () {
      return console.warn("auth error");
    })
    .receive("timeout", function () {
      return console.error("connection interruption");
    });
  channel.onError(function (e) {
    return console.error("channel error", e);
  });

  channel.onClose(function (e) {
    return console.log("channel closed", e);
  });

  channel.on("quote", function (msg) {
    var ticker = msg.ticker;
    var type = msg.type;
    var timestamp = msg.timestamp;
    var price = msg.price;
    var size = msg.size;

    console.log(ticker, type, timestamp, price, size);
  });
};
```

Note: For easy parsing of timestamps and prices, we recommend using Moment and Numeral:

```
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/moment.js
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/numeral.j
```

```
var timestamp = moment.unix(parseFloat(msg.timestamp)).format('h:mm:ss.SSS');
var price = numeral(parseFloat(msg.price)).format('$0,0.00');
```

Note: There is also an "iex:lobby" channel which broadcasts price updates for all securities, though this may not be accessible, depending on your subscription plan.