

```
1. $(document).ready(function() {
2.
3.     // key pressed in search field, so filter table
4.     $('#search').on('keyup', function() {
5.
6.         // determine symbol we're searching for
7.         var query = $(this).val();
8.
9.         // iterate over each row in the table
10.        $('#table-portfolio tbody tr').each(function(e) {
11.
12.            // check if the symbol cell contains the query
13.            if (!query || $(this).children().first().text().toLowerCase().indexOf(query) > -1)
14.                $(this).show();
15.
16.            // no match, so hide row
17.            else
18.                $(this).hide();
19.        });
20.    });
21.
22.    // load table into ticker
23.    $('#table-portfolio tbody tr').each(function(e) {
24.
25.        // get the cells in the row
26.        var children = $(this).children();
27.
28.        // add all rows to ticker except the last, which is cash
29.        if (children.length > 2)
30.            $('#ticker').append('<span>' + children.eq(0).text() + ' ' + children.eq(3).text() + '</span>');
31.    });
32.
33.    // animate the ticker
34.    function animateTicker() {
35.
36.        // move ticker text all the way off to the right
37.        $('#ticker').css({ marginLeft: '100%' });
38.
39.        // animate ticker text all the way off to the left, and restart animation when done
40.        $('#ticker').animate({ marginLeft: '-100%' }, 10000, 'linear', animateTicker);
41.    }
42.
43.    // start animation loop
44.    animateTicker();
45.});
```

```
1. $(document).ready(function() {
2.
3.     // key pressed in search field, so filter table
4.     $('#search').on('keyup', function() {
5.
6.         // determine symbol we're searching for
7.         var query = $(this).val();
8.
9.         // iterate over each row in the table
10.        $('#table-portfolio tbody tr').each(function(e) {
11.
12.            // check if the symbol cell contains the query
13.            if (!query || $(this).children().first().text().toLowerCase().indexOf(query) > -1)
14.                $(this).show();
15.
16.            // no match, so hide row
17.            else
18.                $(this).hide();
19.        });
20.    });
21.
22.    // load table into ticker
23.    $('#table-portfolio tbody tr').each(function(e) {
24.
25.        // get the cells in the row
26.        var children = $(this).children();
27.
28.        // add all rows to ticker except the last, which is cash
29.        if (children.length > 2)
30.            $('#ticker').append('<span>' + children.eq(0).text() + ' ' + children.eq(3).text() + '</span>');
31.    });
32.
33.    // timer properties
34.    var speed = 30;
35.    var margin = 100;
36.
37.    setInterval(function() {
38.
39.        // move ticker towards the left
40.        margin--;
41.
42.        // once ticker goes all the way left, reset to the right
43.        if (margin < -100)
44.            margin = 100;
45.
46.        // set ticker position
47.        $('#ticker').css({ marginLeft: margin + '%' });
48.    }, speed);
```

49. });

```
1. <?php
2.
3.     // configuration
4.     require("../includes/config.php");
5.
6.     // if form was submitted
7.     if ($_SERVER["REQUEST_METHOD"] == "POST")
8.     {
9.         // validate submission
10.        if (empty($_POST["symbol"]))
11.        {
12.            exit;
13.        }
14.
15.        // look up quote
16.        $stock = lookup($_POST["symbol"]);
17.        if ($stock === false)
18.        {
19.            exit;
20.        }
21.
22.        $price = number_format($stock['price'], 2);
23.        echo "A share of {$stock['name']} costs $price.";
24.    }
25.    else
26.    {
27.        // else render form
28.        render("quote_form.php", ["title" => "Get Quote"]);
29.    }
30.
31. ?>
```

```
1. $(document).ready(function() {
2.
3.     // load data via ajax when form is submitted
4.     $('#form-quote').on('submit', function() {
5.
6.         // determine symbol
7.         var symbol = $('#form-quote input[name=symbol]').val();
8.
9.         // send request to quote.php
10.        $.ajax({
11.            url: 'quote.php',
12.            type: 'POST',
13.            data: {
14.                symbol: symbol
15.            },
16.            success: function(response) {
17.                $('#price').text(response);
18.            }
19.        });
20.
21.        // since we're overriding form submission, make sure it doesn't submit
22.        return false;
23.    });
24.});
```

```
1. $(document).ready(function() {
2.
3.     // load data via ajax when form is submitted
4.     $('#form-quote input[name=symbol]').on('keyup', function() {
5.
6.         // determine symbol
7.         var symbol = $('#form-quote input[name=symbol]').val();
8.
9.         // send request to quote.php
10.        $.ajax({
11.            url: 'quote.php',
12.            type: 'POST',
13.            data: {
14.                symbol: symbol
15.            },
16.            success: function(response) {
17.                $('#price').text(response);
18.            }
19.        });
20.    });
21.});
```

```
1. $(document).ready(function() {
2.
3.     // create autocomplete from list of stock symbols
4.     $('#form-quote input[name=symbol]').typeahead({
5.         source: SYMBOLS
6.     });
7.
8.     // load data via ajax when form is submitted
9.     $('#form-quote').on('submit', function() {
10.
11.         // determine symbol
12.         var symbol = $('#form-quote input[name=symbol]').val();
13.
14.         // send request to quote.php
15.         $.ajax({
16.             url: 'quote.php',
17.             type: 'POST',
18.             data: {
19.                 symbol: symbol
20.             },
21.             success: function(response) {
22.                 $('#price').text(response);
23.             }
24.         });
25.
26.         // since we're overriding form submission, make sure it doesn't submit
27.         return false;
28.     });
29. });
```

```
1. $(document).ready(function() {
2.
3.     // create autocomplete
4.     $('#form-quote input[name=symbol]').typeahead({
5.
6.         // load autocomplete data from suggest.php
7.         source: function(query, callback) {
8.             $.ajax({
9.                 url: 'suggest.php',
10.                type: 'POST',
11.               dataType: 'json',
12.               data: {
13.                   symbol: query
14.               },
15.               success: function(response) {
16.                   callback(response.symbols);
17.               }
18.           });
19.       }
20.   });
21.
22.     // load data via ajax when form is submitted
23.     $('#form-quote').on('submit', function() {
24.
25.         // determine symbol
26.         var symbol = $('#form-quote input[name=symbol]').val();
27.
28.         // send request to quote.php
29.         $.ajax({
30.             url: 'quote.php',
31.             type: 'POST',
32.             data: {
33.                 symbol: symbol
34.             },
35.             success: function(response) {
36.                 $('#price').text(response);
37.             }
38.         });
39.
40.         // since we're overriding form submission, make sure it doesn't submit
41.         return false;
42.     });
43. });
```

```
1. // object expected by yahoo finance autosuggest "api"
2. var YAHOO = {
3.     util: {
4.         ScriptNodeDataSource: {
5.             callbacks: function(data) {
6.
7.                 // create array of symbols
8.                 var results = [];
9.                 for (var i in data.ResultSet.Result)
10.                     results.push(data.ResultSet.Result[i].symbol);
11.
12.                 // display the array in the autocomplete
13.                 window.callback(results);
14.             }
15.         }
16.     }
17. };
18.
19. $(document).ready(function() {
20.
21.     // create autocomplete
22.     $('#form-quote input[name=symbol]').typeahead({
23.         source: function(query, callback) {
24.
25.             // remember how to display results in the autocomplete
26.             window.callback = callback;
27.
28.             // send cross-domain request to yahoo finance
29.             $.ajax({
30.                 url: 'http://d.yimg.com/aq/autoc',
31.                 type: 'GET',
32.                 dataType: 'jsonp',
33.                 jsonpCallback: 'YAHOO.util.ScriptNodeDataSource.callbacks',
34.                 data: {
35.                     query: query,
36.                     region: 'US',
37.                     lang: 'en-US',
38.                 }
39.             });
40.         }
41.     });
42.
43.     // load data via ajax when form is submitted
44.     $('#form-quote').on('submit', function() {
45.
46.         // determine symbol
47.         var symbol = $('#form-quote input[name=symbol]').val();
48.
```

```
49.     // send request to quote.php
50.     $.ajax({
51.         url: 'quote.php',
52.         type: 'POST',
53.         data: {
54.             symbol: symbol
55.         },
56.         success: function(response) {
57.             $('#price').text(response);
58.         }
59.     });
60.
61.     // since we're overriding form submission, make sure it doesn't submit
62.     return false;
63. });
64. });
```

```
1. $(document).ready(function() {
2.
3.     // create autocomplete
4.     $('#form-quote input[name=symbol]').on('keyup', function() {
5.
6.         // load autocomplete data from suggest.php
7.         $.ajax({
8.             url: 'suggest.php',
9.             type: 'POST',
10.            dataType: 'json',
11.            data: {
12.                symbol: $(this).val()
13.            },
14.            success: function(response) {
15.                // build html string for a list of suggestions
16.                var suggestions = '<ul>';
17.                for (var i in response.symbols)
18.                    suggestions += '<li><a href="#" class="suggestion">' + response.symbols[i] + '</a></li>';
19.
20.                // display list of suggestions
21.                suggestions += '</ul>';
22.                $('#suggestions').html(suggestions);
23.            }
24.        });
25.    });
26.
27.    // set value of symbol field when a suggestion is clicked
28.    $('#suggestions').on('click', '.suggestion', function() {
29.        $('#form-quote input[name=symbol]').val($(this).text());
30.    });
31.
32.    // load data via ajax when form is submitted
33.    $('#form-quote').on('submit', function() {
34.
35.        // determine symbol
36.        var symbol = $('#form-quote input[name=symbol]').val();
37.
38.        // send request to quote.php
39.        $.ajax({
40.            url: 'quote.php',
41.            type: 'POST',
42.            data: {
43.                symbol: symbol
44.            },
45.            success: function(response) {
46.                $('#price').text(response);
47.            }
48.        });

```

```
49.  
50.      // since we're overriding form submission, make sure it doesn't submit  
51.      return false;  
52.  } );  
53. } );
```

```
1. $(document).ready(function() {
2.
3.     // create autocomplete
4.     $('#form-quote input[name=symbol]').on('keyup', function() {
5.
6.         // load autocomplete data from suggest.php
7.         $.ajax({
8.             url: 'suggest_more.php',
9.             type: 'POST',
10.            dataType: 'json',
11.            data: {
12.                symbol: $(this).val()
13.            },
14.            success: function(response) {
15.                // build html string for a list of suggestions
16.                var suggestions = '<ul>';
17.                for (var i in response.symbols)
18.                    suggestions += '<li><a href="#" class="suggestion" data-symbol="' + response.symbols[i].symbol + '">' +
19.                        response.symbols[i].name + '</a></li>';
20.
21.                // display list of suggestions
22.                suggestions += '</ul>';
23.                $('#suggestions').html(suggestions);
24.            }
25.        });
26.    });
27.
28.    // set value of symbol field when a suggestion is clicked
29.    $('#suggestions').on('click', '.suggestion', function() {
30.        $('#form-quote input[name=symbol]').val($(this).attr('data-symbol'));
31.    });
32.
33.    // load data via ajax when form is submitted
34.    $('#form-quote').on('submit', function() {
35.
36.        // determine symbol
37.        var symbol = $('#form-quote input[name=symbol]').val();
38.
39.        // send request to quote.php
40.        $.ajax({
41.            url: 'quote.php',
42.            type: 'POST',
43.            data: {
44.                symbol: symbol
45.            },
46.            success: function(response) {
47.                $('#price').text(response);
48.            }
49.        });
50.    });
51.
```

```
49.    });
50.
51.    // since we're overriding form submission, make sure it doesn't submit
52.    return false;
53.  });
54.});
```

```
1. <?php
2.
3. // configuration
4. require("../includes/config.php");
5.
6. // if form was submitted
7. if ($_SERVER["REQUEST_METHOD"] == "POST")
8. {
9.     // load suggestion data
10.    $data = @file_get_contents("http://d.yimg.com/aq/autoc?query={$_POST['symbol']}
&region=US&lang=en-US&callback=YAHOO.util.ScriptNodeDataSource.callbacks");
11.
12.    // parse yahoo data into a list of symbols
13.    $result = [];
14.    $json = json_decode(substr($data, strlen('YAHOO.util.ScriptNodeDataSource.callbacks('), -1));
15.    foreach ($json->ResultSet->Result as $stock)
16.        $result[] = $stock->symbol;
17.
18.    echo json_encode(['symbols' => $result]);
19. }
20.
21. ?>
```

```
1. <?php
2.
3. // configuration
4. require("../includes/config.php");
5.
6. // if form was submitted
7. if ($_SERVER["REQUEST_METHOD"] == "POST")
8. {
9.     // load suggestion data
10.    $data = @file_get_contents("http://d.yimg.com/aq/autoc?query={$_POST['symbol']}
&region=US&lang=en-US&callback=YAHOO.util.ScriptNodeDataSource.callbacks");
11.
12.    // parse yahoo data into a list of symbols
13.    $result = [];
14.    $json = json_decode(substr($data, strlen('YAHOO.util.ScriptNodeDataSource.callbacks('), -1));
15.    foreach ($json->ResultSet->Result as $stock)
16.        $result[] = $stock;
17.
18.    echo json_encode(['symbols' => $result]);
19. }
20.
21. ?>
```