

Rendering arrays using EJS

One of the most common tasks when using a template language is to render a list of items. Since EJS is based on escaping to JavaScript, rendering lists can be done using the loop constructs in the language.

In this recipe, we're going to render a list of message objects. Each message object will have an author, arrival time, body, and read status. We're going to use a different style to distinguish between read and unread messages.

Getting ready

We need to download EJS from <http://embeddedjs.com/> and extract `ejs_production.js` in our `recipe` folder.

How to do it...

Let's get started.

1. Create `index.html`, it will contain a header, the EJS template, the placeholder to render the message list, and some styles for the list:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Rendering an array with EJS</title>
<style type="text/css">
.message {
  border-bottom:solid 1px #ccc;
  width: 250px;
  padding: 5px; }
.message p { margin: 0.5em 0; }
.message.unread { font-weight:bold; }
.message .date {
  float: right;
```

```
    font-style: italic;
    color: #999; }
</style>
</head>
<body>
<h2>Messages</h2>
<div id="list">
</div>
<script id="template" type="text/ejs">
<% for (var k = 0; k < list.length; ++k) {
    var message = list[k];  %>
    <div class="message"><%= message.status %>">
        <p><span class="name"><%= message.name %></span>
        <span class="date"><%= message.date %></span></p>
        <p class="text"><%= message.text %></p>
    </div>
<% } %>
</script>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></scrip
t>
<script type="text/javascript" src="ejs_production.js"></script>
<script type="text/javascript" src="example.js"></script>
</body>
</html>
```

2. Call the `render` function from `example.js` with some text data:

```
$(function() {
    var template = new EJS({
        text: $('#template').html()
    });

    $('#list').html(template.render({list:[
        { status: 'read',   name: 'John', date: 'Today',
          text: 'just got back, how are you doing?' },
        { status: 'unread', name: 'Jenny', date: 'Today',
          text: 'please call me asap' },
        { status: 'read',   name: 'Jack', date: 'Yesterday',
          text: 'where do you want to go today?' },
    ]}));
});
```

How it works...

In the `render` function we pass a model object containing an array of messages to the renderer.

To render the array we use a standard JavaScript `for` loop. We can add any valid JavaScript code between the opening and closing tags. In our example we assign a variable inside the body of the loop, and then use it throughout the template.

From the example it's clear that EJS allows you to escape to JavaScript at any point in the template text. Even escaping in HTML attributes is allowed (we are adding a class to our message that corresponds to the message status, read or unread) by escaping inside the `class` attribute.