USE BOWER TO INSTALL PACKAGES

Let's add some order to our list and make it sortable. For this we're going to use Bower to install the AngularUI Sortable module, a directive component for AngularJS.

LIST CURRENT PACKAGES

We can check what packages we have already installed with this command:

```
$ bower list
```

```
addyo at addyo-macbookair3 in ~/projects/mytodo
$ bower list
bower check-new        Checking for new versions of the project depen
dencies..
mytodo#0.0.0 /Users/addyo/projects/mytodo
├── angular#1.2.6 (latest is 1.2.10-build.2145+sha.6a9ccac)
├─┬ angular-cookies#1.2.6 (latest is 1.2.10-build.2145+sha.6a9ccac
)
│ └── angular#1.2.6 (latest is 1.2.10-build.2145+sha.6a9ccac)
├─┬ angular-mocks#1.2.6 (latest is 1.2.10-build.2145+sha.6a9ccac)
│ └── angular#1.2.6
├─┬ angular-resource#1.2.6 (latest is 1.2.10-build.2145+sha.6a9cca
c)
│ └── angular#1.2.6
├─┬ angular-route#1.2.6 (latest is 1.2.10-build.2145+sha.6a9ccac)
│ └── angular#1.2.6
├─┬ angular-sanitize#1.2.6 (latest is 1.2.10-build.2145+sha.6a9cca
c)
```

You should see that you already have packages for angular-cookies, angular-resources, angular-route, plus others. Remember how we selected these as part of Step 3 when we configured our generator?

SEARCH FOR PACKAGES

To verify that there are AngularUI packages available, use Bower to search for "angular-ui-sortable":

```
$ bower search angular-ui-sortable
```

There is one result for "angular-ui-sortable" so let's install it along with jQuery UI as we already have jQuery installed. To save you from searching, the package name for jQuery UI is "jquery-ui".

INSTALL PACKAGES

Use Bower to install both "angular-ui-sortable" and "jquery-ui":

```
$ bower install --save angular-ui-sortable
$ bower install --save jquery-ui
```

The `--save` option updates the **bower.json** file with dependencies on angular-ui-sortable and jquery-ui. This will save you from having to manually add it to *bower.json* yourself.

INSTALL MULTIPLE BOWER PACKAGES AT ONCE

If you have multiple packages that you want to install, you can do it in one line:

```
$ bower install --save angular-ui-sortable jquery-ui
```


CONFIRM INSTALLATION

Take a look at your **bower_components** directory just to check that everything is there as expected. You should see folders for "jquery-ui" and "angular-ui-sortable" alongside the previously installed angular packages:

MAKE TODOS SORTABLE

References to these newly installed dependencies must be added to our *index.html* file. You *could* manually add the AngularUI Sortable module and jQueryUI script files yourself but Yeoman will automate this for you!

Quit your current command line process by using the `Ctrl` + `C` keyboard command.

Then run `grunt serve` again.

You'll see that the `script` section at the bottom of *index.html* has automatically updated to include `jquery-ui/ui/jquery-ui.js` and `angular-ui-sortable/sortable.js`:

```html
<!-- build:js scripts/vendor.js -->
<!-- bower:js -->
<script src="bower_components/jquery/jquery.js"></script>
<script src="bower_components/angular/angular.js"></script>
<script src="bower_components/bootstrap/dist/js/bootstrap.js"></script>
<script src="bower_components/angular-resource/angular-resource.js"></script>
<script src="bower_components/angular-cookies/angular-cookies.js"></script>
<script src="bower_components/angular-sanitize/angular-sanitize.js"></script>
<script src="bower_components/angular-route/angular-route.js"></script>
<script src="bower_components/jquery-ui/ui/jquery-ui.js"></script>
<script src="bower_components/angular-ui-sortable/sortable.js"></script>
<!-- endbower -->
<!-- endbuild -->
```

In order to use the Sortable module, we must load it into our application by updating our Angular module definitions in *scripts/app.js*. Currently it looks like this:

```javascript
angular.module('mytodoApp', [
  'ngCookies',
  'ngResource',
  'ngSanitize',
  'ngRoute'
])
```

Add the `ui.sortable` dependency to the array list in the second paramater. Our complete todo module should now look like this:

```javascript
'use strict';

angular.module('mytodoApp', [
  'ngCookies',
  'ngResource',
  'ngSanitize',
  'ngRoute',
  'ui.sortable'
])
  .config(function ($routeProvider) {
    $routeProvider
      .when('/', {
```

```
      templateUrl: 'views/main.html',
      controller: 'MainCtrl'
    })
    .otherwise({
      redirectTo: '/'
    });
  });
```

Finally, we need to add the `ui-sortable` directive as a `div` wrapper around our `ng-repeat` in *main.html*:

```html
<!-- Todos list -->
<div ui-sortable ng-model="todos">
  <p class="input-group" ng-repeat="todo in todos">
```

Let's also add some inline css in order to show a move cursor so it's clear that we can move todo items:

```html
<p class="input-group" ng-repeat="todo in todos" style="padding:5px 10px; cursor: move;">
```

The full todo list markup looks like this now:

```html
<!-- Todos list -->
<div ui-sortable ng-model="todos">
  <p class="input-group" ng-repeat="todo in todos" style="padding:5px 10px; cursor: move;">
    <input type="text" ng-model="todo" class="form-control">
    <span class="input-group-btn">
      <button class="btn btn-danger" ng-click="removeTodo($index)" aria-label="Remove">X</button>
    </span>
  </p>
</div>
```
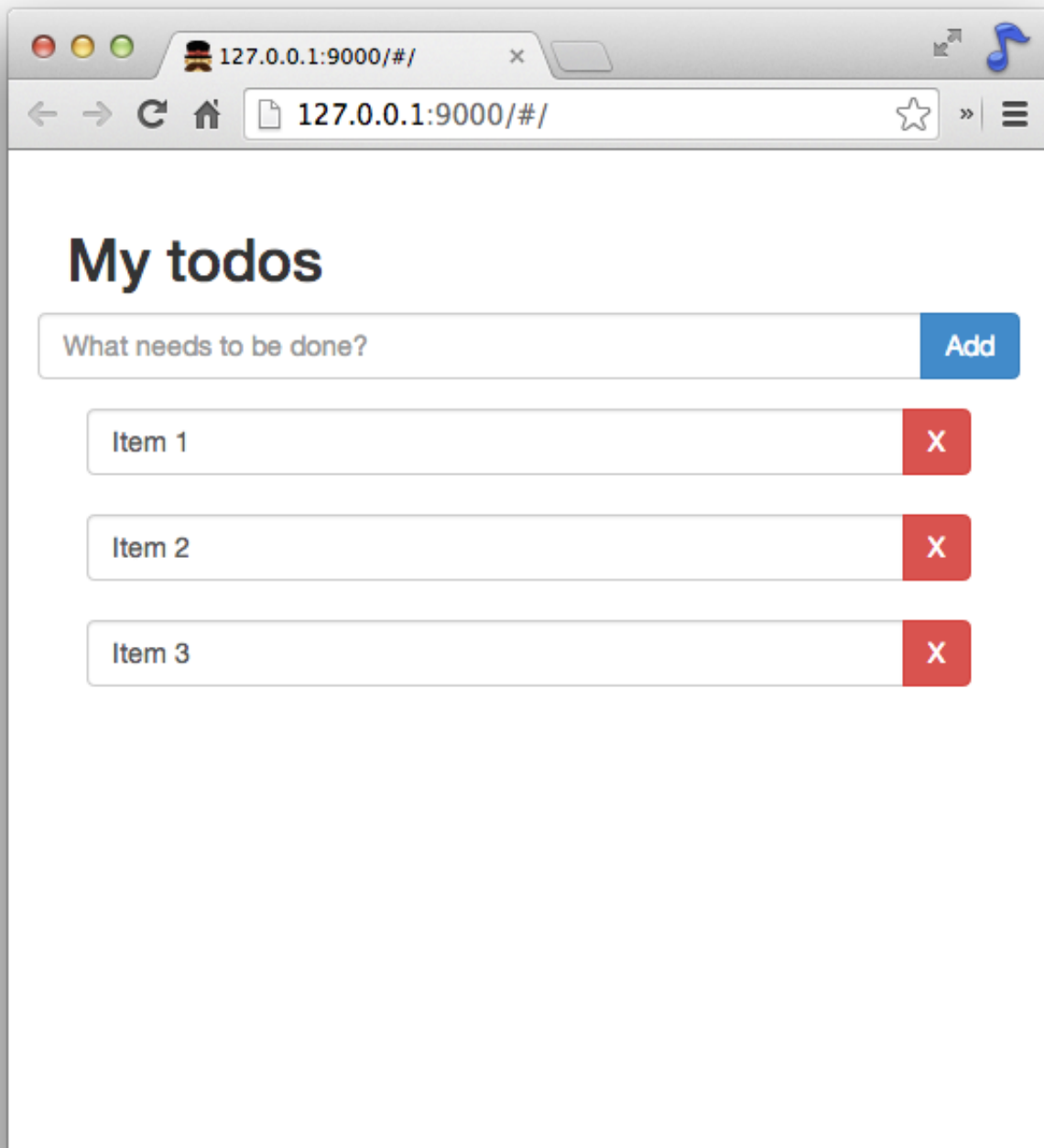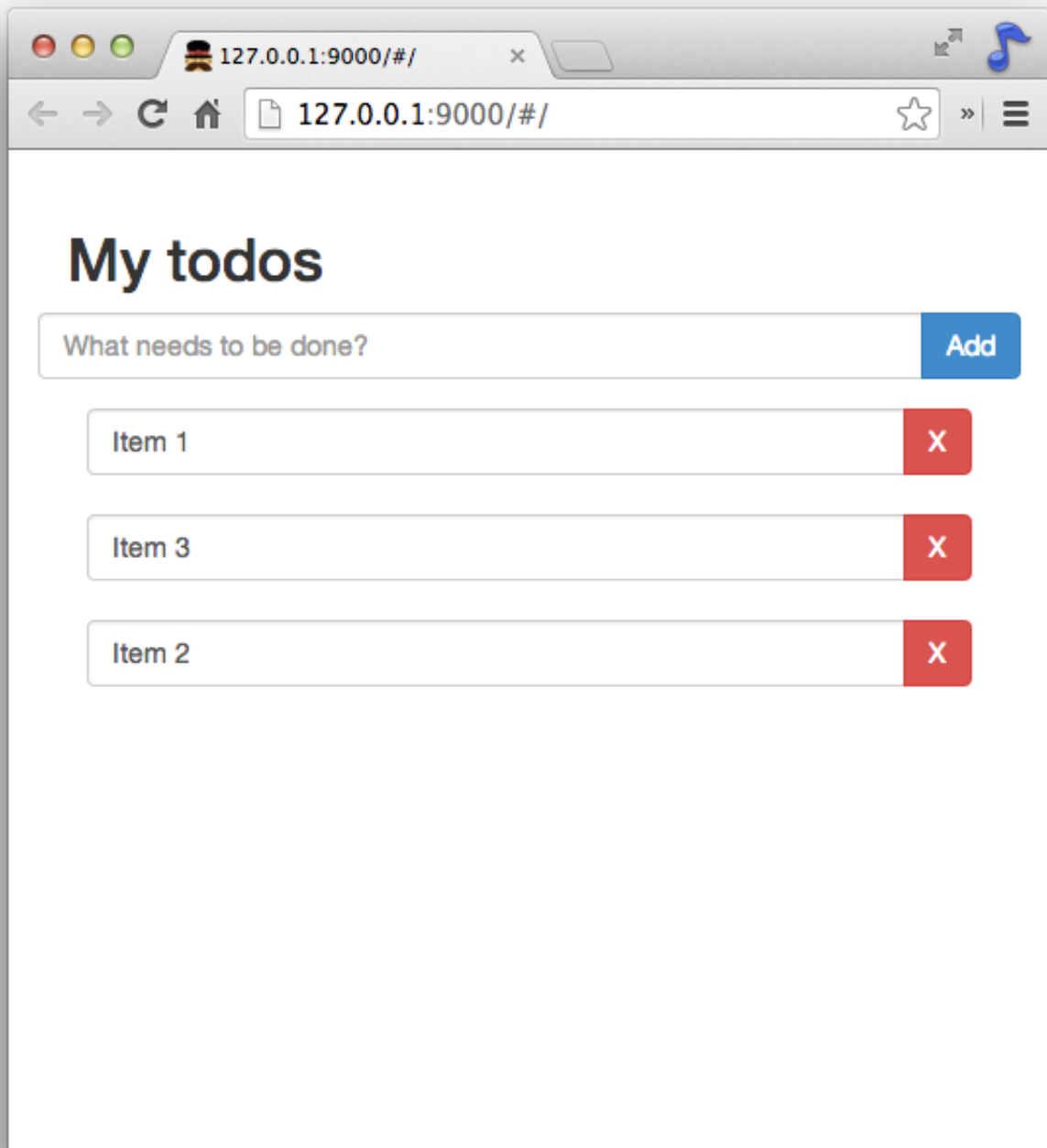
Back in the browser, we can now re-order our list:

My todos

What needs to be done?   Add

Item 1   X

Item 2   X

Item 3   X

Pat yourself on the back! You just used Yeoman to build a snazzy todo app in no time. Can you imagine doing front-end web development in any other way now?

« Return to overview   or   **Go to the next step »**

Tweet 209   g+ **Share** Share this on Google+