Generator used: <https://github.com/yeoman/generator-angular>

* Used only for client
* Properly sepeartes files into controller folders etc
* Doesn’t include server-side components in the same project, which is a bad idea.

yo angular:route events

* Scaffold out controller and views and add them to index.html and app.js

yo angular-fullstack:endpoint event

<http://stackoverflow.com/questions/17407024/telling-angular-js-to-ignore-a-specific-route>

Required for setting up separate client and server projects:

<https://github.com/drewzboto/grunt-connect-proxy>

Required for sending email

<http://stackoverflow.com/questions/26196467/sending-email-via-node-js-using-nodemailer-is-not-working>

Mongoose does not have enough validators out of the box. Use:

npm install mongoose-validator --save

<https://github.com/leepowellcouk/mongoose-validator>

Angular Form validation

<https://docs.angularjs.org/guide/forms>

Note that novalidate is used to disable browser's native form validation.

Using CSS classes

To allow styling of form as well as controls, ngModel adds these CSS classes:

* ng-valid: the model is valid
* ng-invalid: the model is invalid
* ng-valid-[key]: for each valid key added by $setValidity
* ng-invalid-[key]: for each invalid key added by $setValidity
* ng-pristine: the control hasn't been interacted with yet
* ng-dirty: the control has been interacted with
* ng-touched: the control has been blurred
* ng-untouched: the control hasn't been blurred
* ng-pending: any $asyncValidators are unfulfilled

A form is an instance of [FormController](https://docs.angularjs.org/api/ng/type/form.FormController). The form instance can optionally be published into the scope using the name attribute.

Similarly, an input control that has the [ngModel](https://docs.angularjs.org/api/ng/directive/ngModel) directive holds an instance of [NgModelController](https://docs.angularjs.org/api/ng/type/ngModel.NgModelController). Such a control instance can be published as a property of the form instance using the name attribute on the input control. The name attribute specifies the name of the property on the form instance.

* Custom error message displayed after the user interacted with a control (i.e. when $touched is set)
* Custom error message displayed upon submitting the form ($submitted is set), even if the user didn't interact with a control

By default, any change to the content will trigger a model update and form validation. You can override this behavior using the[ngModelOptions](https://docs.angularjs.org/api/ng/directive/ngModelOptions) directive to bind only to specified list of events. I.e. ng-model-options="{ updateOn: 'blur' }" will update and validate only after the control loses focus. You can set several events using a space delimited list. I.e.ng-model-options="{ updateOn: 'mousedown blur' }"

Angular provides basic implementation for most common HTML5 [input](https://docs.angularjs.org/api/ng/directive/input) types: ([text](https://docs.angularjs.org/api/ng/input/input%5Btext%5D), [number](https://docs.angularjs.org/api/ng/input/input%5Bnumber%5D), [url](https://docs.angularjs.org/api/ng/input/input%5Burl%5D), [email](https://docs.angularjs.org/api/ng/input/input%5Bemail%5D), [date](https://docs.angularjs.org/api/ng/input/input%5Bdate%5D), [radio](https://docs.angularjs.org/api/ng/input/input%5Bradio%5D), [checkbox](https://docs.angularjs.org/api/ng/input/input%5Bcheckbox%5D)), as well as some directives for validation (required, pattern, minlength, maxlength, min, max).