Using StickIt to introduce Data Binding

What we want to do is update the model's attributes each time the user has changed something in the form. To achieve this task, we are going to use StickIt plugin that introduces data binding. If you have used AngularJS before, you know what I mean. If no – stay with me for a couple of minutes and things will become clear.

Include the latest version of the plugin from GitHub:

```
<script src="https://raw.githubusercontent.com/NYTimes/backbone.stickit/master/backbone.stickit.js
```

What we have to do now is to set up data bindings inside the view to instruct which field corresponds to which attribute. Moreover, we need to re-validate the model each time something has changed.

```
bindings: {
  '[name=title]': {
    observe: 'title',
    setOptions: {
      validate: true
    }
  },
  '[name=description]': {
    observe: 'description',
    setOptions: {
      validate: true
    }
  }
},
```

We also have to call `stickit` method on our element inside the `render` function. As long as we are using a base class we have to check if this method should actually be called. To do this simply check if a view has any bindings:

```
if (!_.isUndefined(this.bindings)) {
  this.stickit();
}
```

Configure the Backbone Validation

The last step is to add

```
Backbone.Validation.configure({
  forceUpdate: true
});
```

to the *validation.js* – this is required for validation to work correctly with data binding.

Reload the page and observe the result!