Models & Views

- LEVEL 4 -

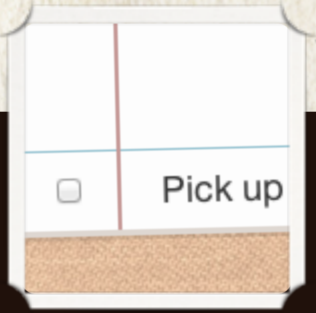
Review our Model View

```
var TodoView = Backbone.View.extend({
  template: _.template('<h3><%= description %></h3>'),

  render: function(){
    this.$el.html(this.template(this.model.toJSON()));
  }
});
```

```
var todoView = new TodoView({ model: todoItem });
todoView.render();
console.log(todoView.el);
```

```
<div>
  <h3>Pick up milk</h3>
</div>
```

Adding a checkbox

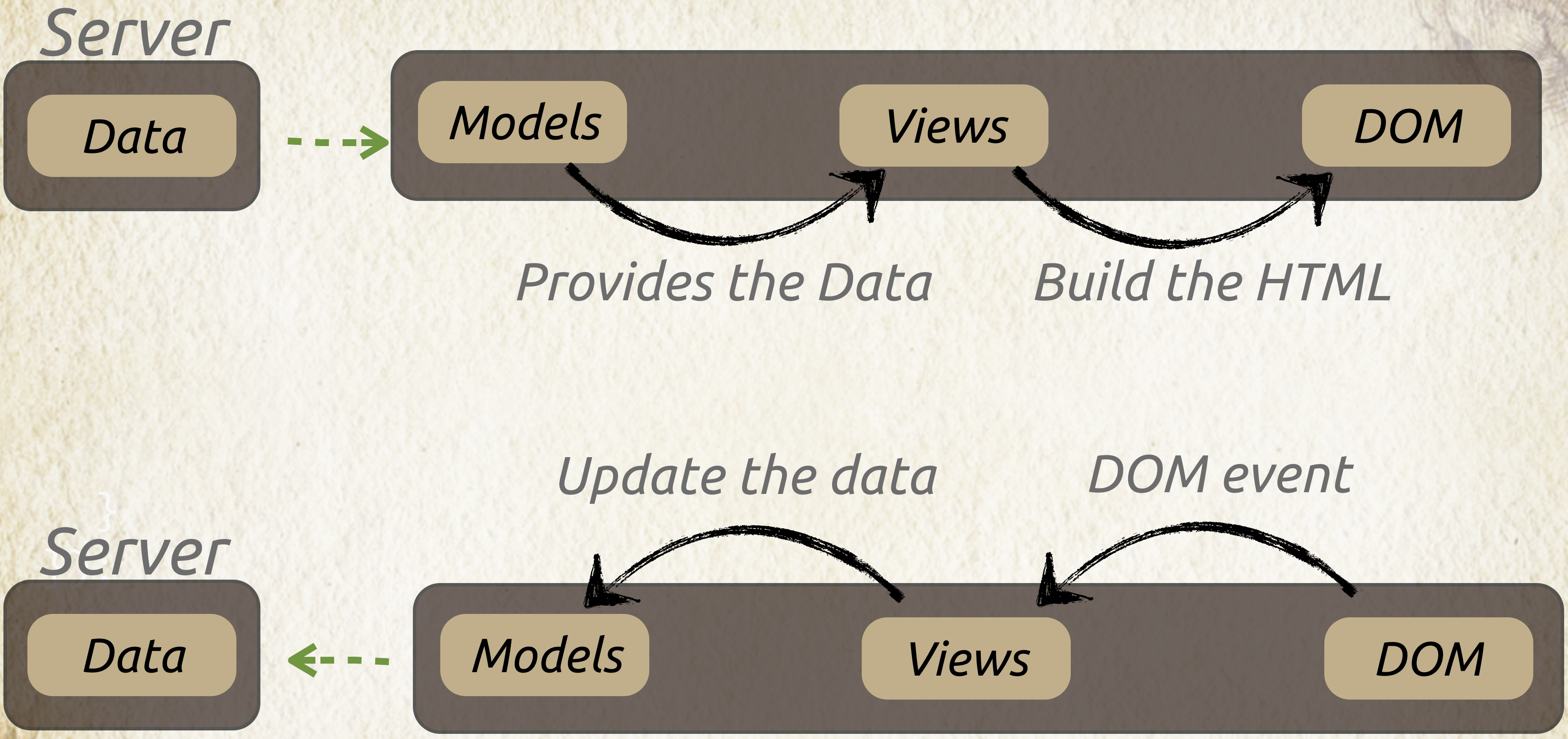


```
var TodoView = Backbone.View.extend({
  template: _.template('<h3>' +
    '<input type=checkbox ' +
    '<% if(status === "complete") print("checked") %>/>' +
                                '<%= description %></h3>'),

  render: function(){
    this.$el.html(this.template(this.model.toJSON()));
  }
});
```

How do we update the model when checkbox changes?

Backbone.js

View events update the Model



Backbone.js

Update model on UI event

```
var TodoView = Backbone.View.extend({

  events: {
    'change input': 'toggleStatus'
  },

  toggleStatus: function(){
    if(this.model.get('status') === 'incomplete'){
      this.model.set({'status': 'complete'});
    }else{
      this.model.set({'status': 'incomplete'});
    }
  }

});
```

✖ Model logic in view

Refactor to the Model

```
var TodoView = Backbone.View.extend({
  events: {
    'change input': 'toggleStatus'
  },
  toggleStatus: function(){
   this.model.toggleStatus();
  }
});

var TodoItem = Backbone.Model.extend({
  toggleStatus: function(){
    if(this.get('status') === 'incomplete'){
      this.set({'status': 'complete'});
    }else{
      this.set({'status': 'incomplete'});
    }
  }
});
```

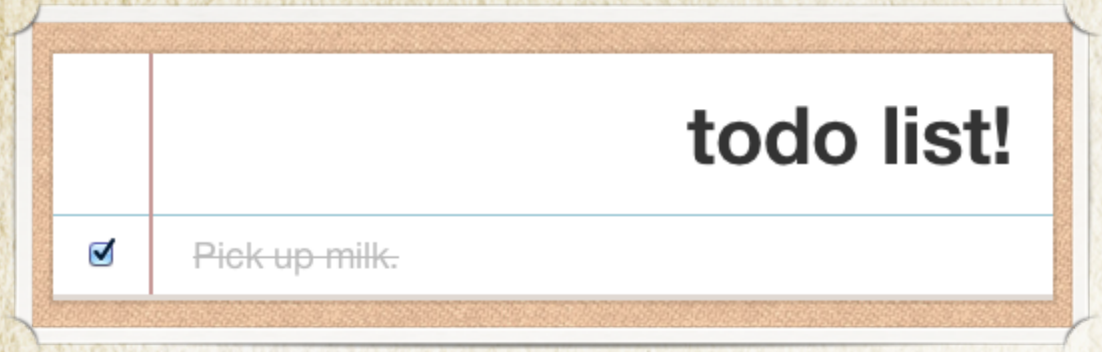
✓ Model logic in Model

Sync changes to server

```
var TodoItem = Backbone.Model.extend({
  toggleStatus: function(){
    if(this.get('status') === 'incomplete'){
      this.set({'status': 'complete'});
    }else{
      this.set({'status': 'incomplete'});
    }

    this.save();  ✓
  }
});
```

```
PUT /todos/1
```

Update view to reflect changes



todo list!

~~Pick up milk.~~

```
.complete {
  color: #bbb;
  text-decoration: line-through;
}
```

update TodoView template:

```
template: _.template('<h3 class="<%= status %>">' +
          '<% if(status === "complete") print("checked") %>/>' +
          ' <%= description %></h3>')
```

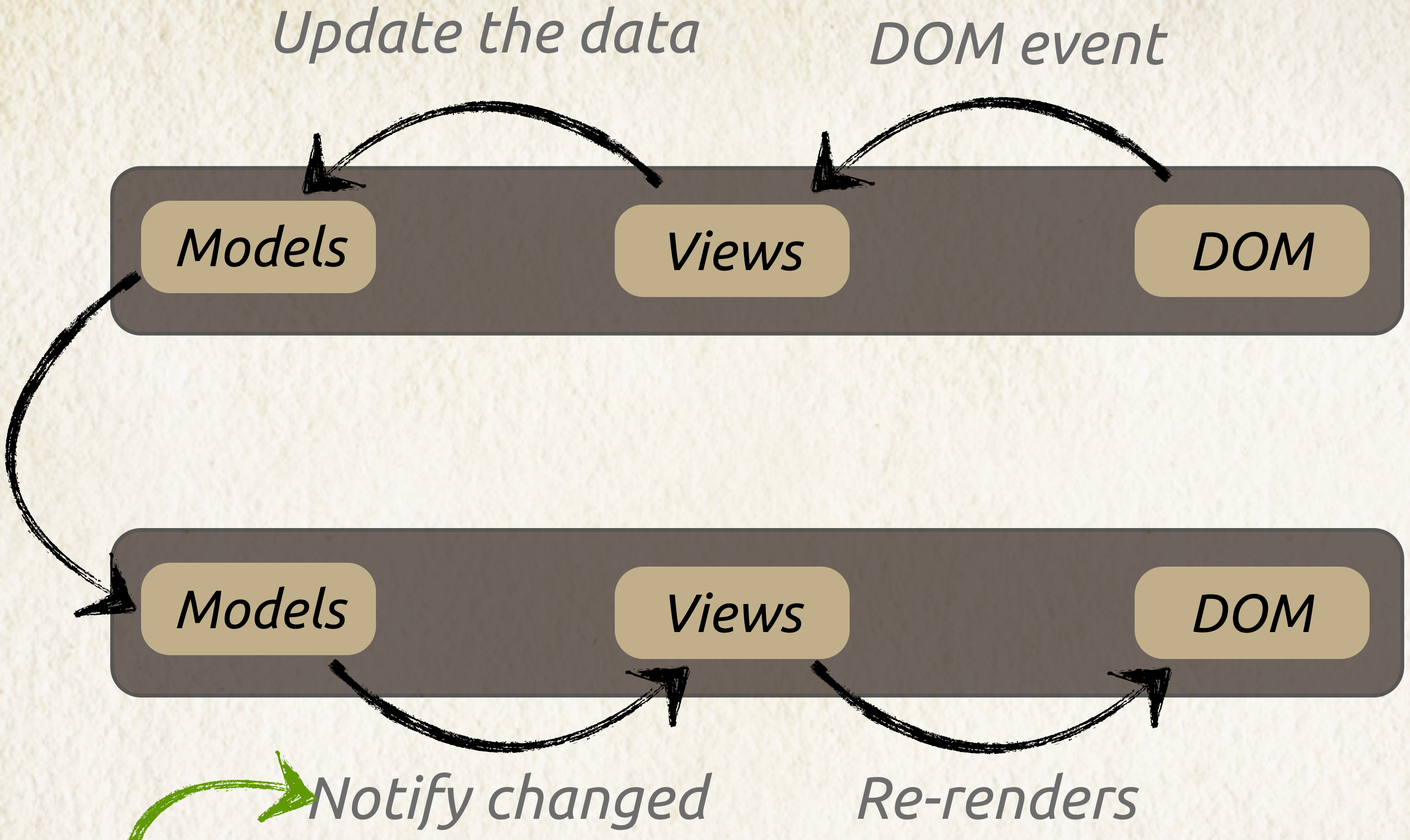
How should we update the view
when the model changes?

Re-render the view

```
var TodoView = Backbone.View.extend({
  events: {
    'change input': 'toggleStatus'
  },
  toggleStatus: function(){
   this.model.toggleStatus();
   this.render();
  },
  render: function(){
  this.$el.html(this.template(this.model.toJSON()));
  }
});
```

Doesn’t work for other model changes

Model updates change the View



✓ Use Model Events

Backbone.js

Re-render the view

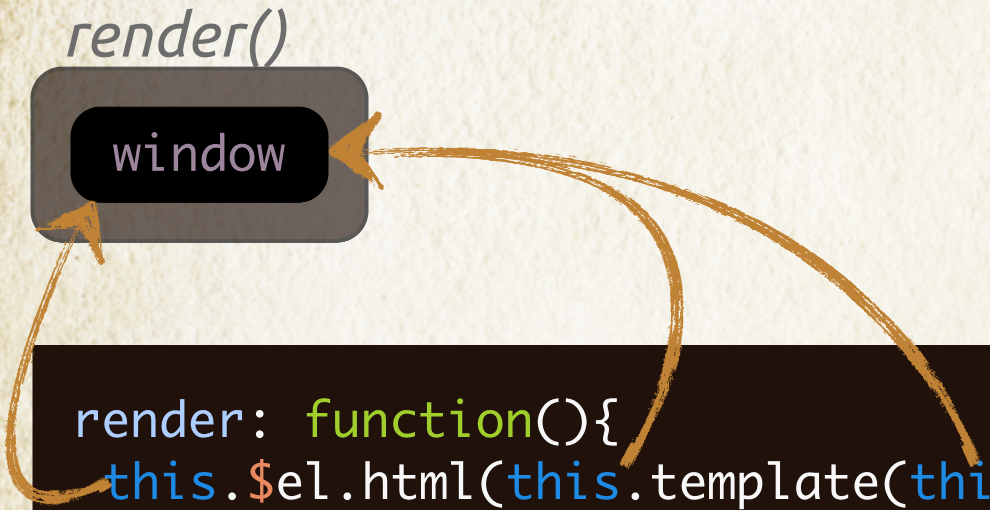
```
var TodoView = Backbone.View.extend({
  events: {
    'change input': 'toggleStatus'
  },
  initialize: function(){
    this.model.on('change', this.render, this);
  },
  toggleStatus: function(){
   this.model.toggleStatus();
  },

  render: function(){
   this.$el.html(this.template(this.model.toJSON()));
  }
});
```

✓

Why the third argument?

What is this?

```
this.model.on('change', this.render);
```

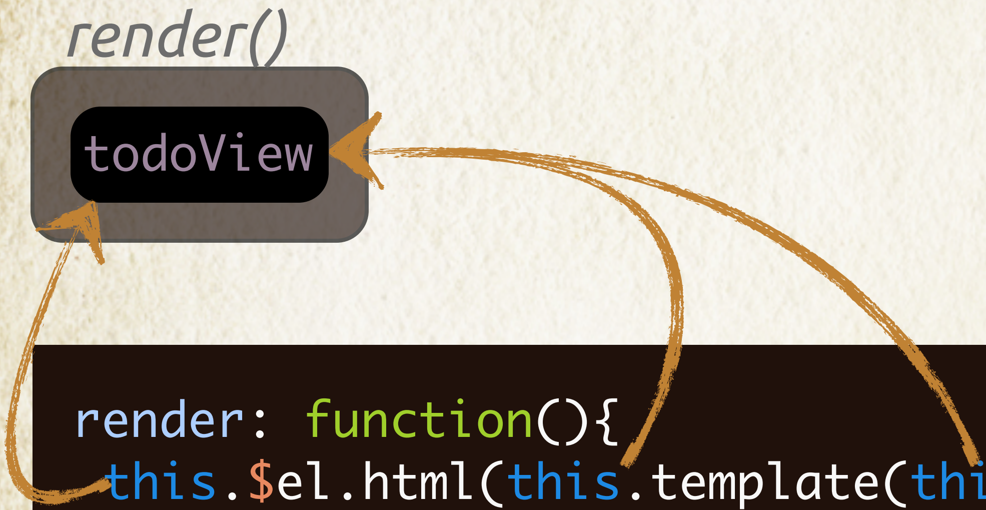


```
render: function(){
 this.$el.html(this.template(this.model.toJSON()));
}
```

render context is not the view

What is this?

```
this.model.on('change', this.render, this);
```



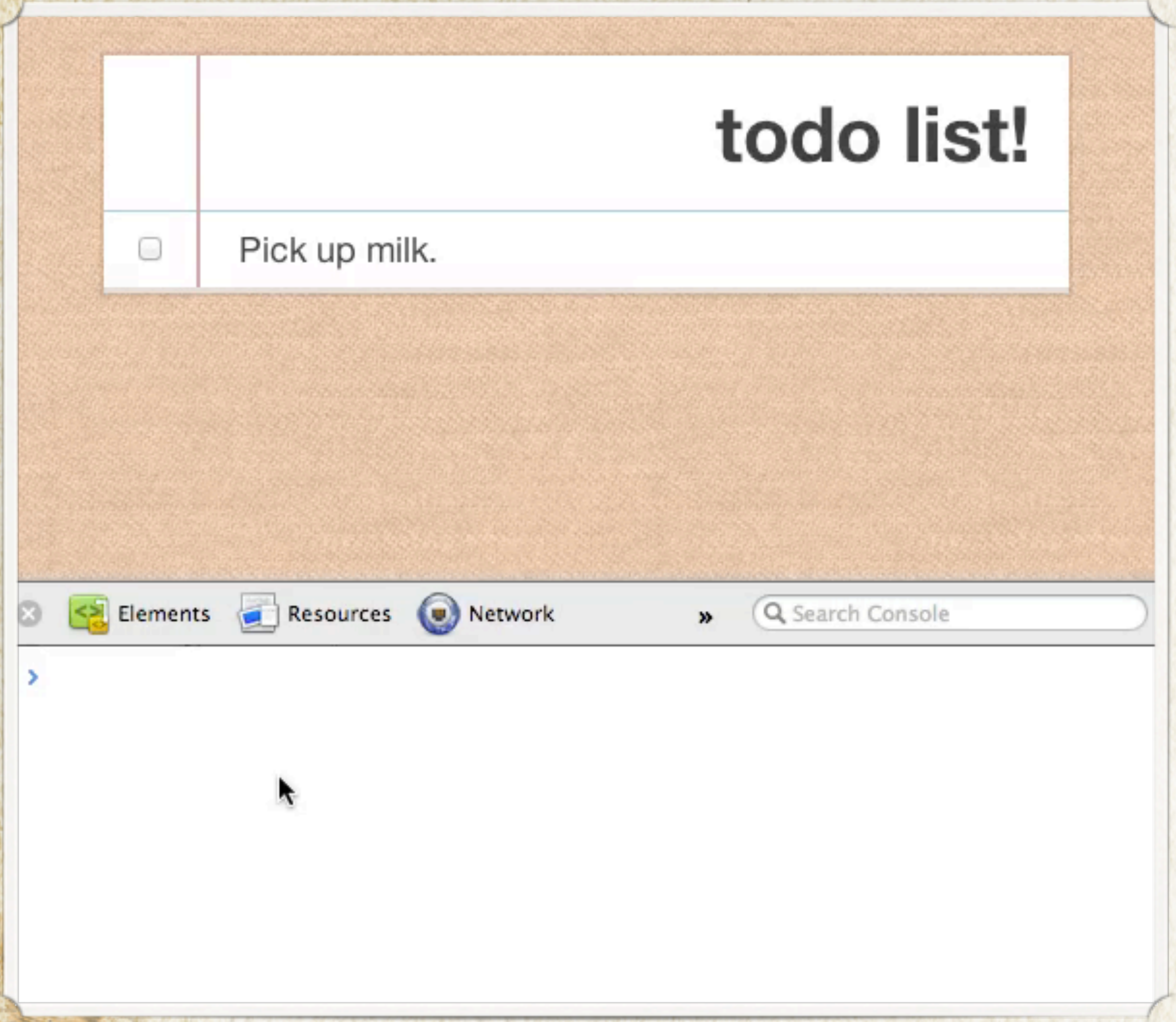
```
render: function(){
 this.$el.html(this.template(this.model.toJSON()));
}
```

✓ **render context is bound to the view**

Remove view on model destroy

```
var TodoView = Backbone.View.extend({
  initialize: function(){
    this.model.on('change', this.render, this);
    this.model.on('destroy', this.remove, this);
  },

  render: function(){
   this.$el.html(this.template(this.model.toJSON()));
  },

  remove: function(){
    this.$el.remove();
  }

});
```

Watch it in action


todo list!

Pick up milk.

Backbone.js