

Sed sed orci. Nulla
Donec a turpis in

Models

- LEVEL 2 -



Donec a
sed sed orci.
ipis in erat
ma. tur



Sed sed orci. Nullam a lac
Donec a turpis in erat cur
fringit. Sed sed orci. Nullam

Reviewing Models

Generating a model class

```
var TodoItem = Backbone.Model.extend({});
```

Generating a model instance

```
var todoItem = new TodoItem(  
  { description: 'Pick up milk', status: 'incomplete' }  
);
```

To get an attribute

```
todoItem.get('description');
```

--> 'Pick up milk'

To set an attribute

```
todoItem.set({status: 'complete'});
```


fetching Data from the Server

```
var todoItem = new TodoItem();
```

URL to get JSON data for model

```
todoItem.url = '/todo';
```

To populate model from server

```
todoItem.fetch();
```

```
--> { id: 1, description: 'Pick up milk', status: 'incomplete' }
```

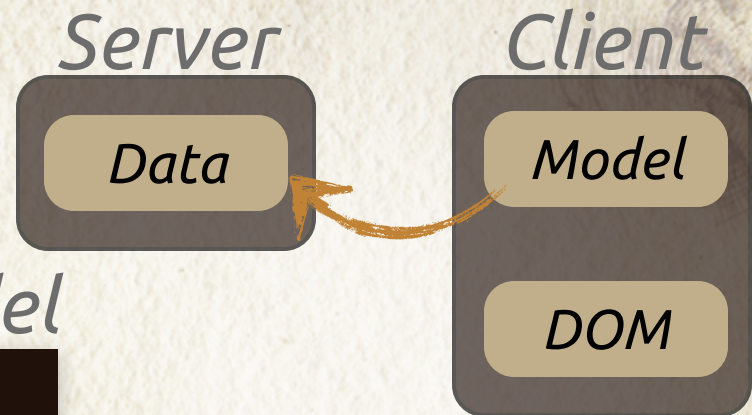
```
todoItem.get('description');
```

```
---> 'Pick up milk'
```



/todo isn't a good URL

Models



Backbone.js

Fetching Data from the Server

```
var TodoItem = Backbone.Model.extend({urlRoot: '/todos'});
```

Fetch todo with id = 1

```
var todoItem = new TodoItem({id: 1})
```

*RESTful web service
(Rails flavor)*

```
todoItem.fetch();
```

GET /todos/1

--> { id: 1, description: 'Pick up milk', status: 'incomplete' }

Update the todo

```
todoItem.set({description: 'Pick up cookies.'});
```

```
todoItem.save();
```

PUT /todos/1
with JSON params

Creating & Destroying a New Todo

```
var todoItem = new TodoItem();
```

```
todoItem.set({description: 'Fill prescription.'});
```

```
todoItem.save();
```

POST /todos
with JSON params

```
todoItem.get('id');
```

--> 2

```
todoItem.destroy();
```

DELETE /todos/2

Get JSON from model

```
todoItem.toJSON();
```

--> { id: 2, description: 'Fill prescription', status: 'incomplete' }

Default Values

```
var TodoItem = Backbone.Model.extend({  
  defaults: {  
    description: 'Empty todo...',  
    status: 'incomplete'  
  }  
});
```

```
var todoItem = new TodoItem();
```

```
todoItem.get('description');
```

--> 'Empty todo...'

```
todoItem.get('status');
```

--> 'incomplete'

Models Can Have Events

To listen for an event on a model

```
todoItem.on('event-name', function(){  
  alert('event-name happened!');  
});
```

Run the event


```
todoItem.trigger('event-name');
```


Special Events

To listen for changes

```
todoItem.on('change', doThing);
```

```
var doThing = function() {  
  ...  
}
```



Event triggered on change

```
todoItem.set({description: 'Fill prescription.'});
```

Set without triggering event

```
todoItem.set({description: 'Fill prescription.'},  
             {silent: true});
```

Remove event listener

```
todoItem.off('change', doThing);
```


Special Events

```
todoItem.on(<event>, <method>);
```

Built-in events

change	<i>When an attribute is modified</i>
change:<attr>	<i>When <attr> is modified</i>
destroy	<i>When a model is destroyed</i>
sync	<i>Whenever successfully synced</i>
error	<i>When model save or validation fails</i>
all	<i>Any triggered event</i>