

DELETE Requests

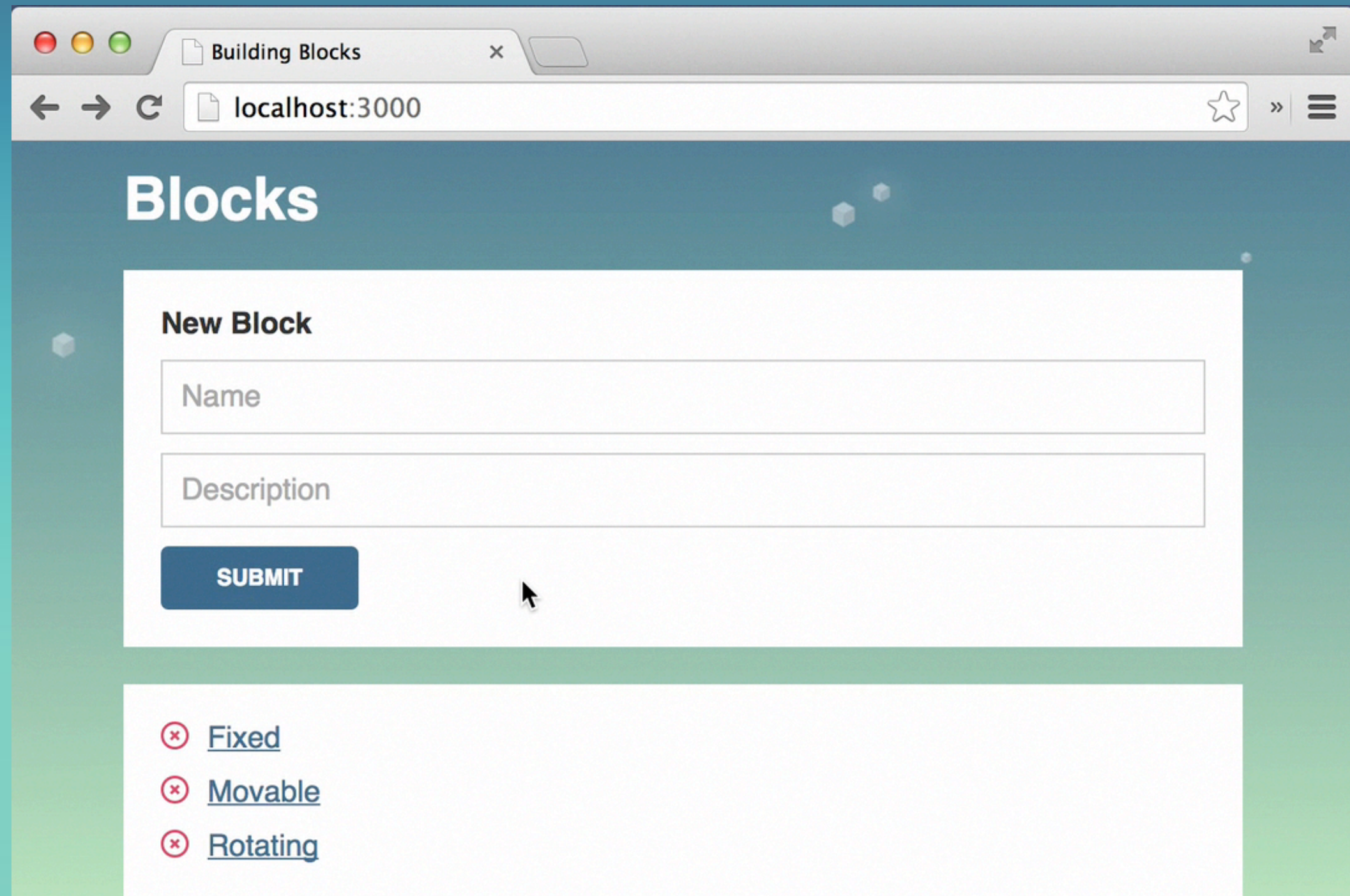
Level 4 - Part II



Deleting existing Blocks

This is what we are going to do:

1. Add delete links
2. Create **DELETE** route



Deleting existing Blocks

Client



DELETE to /blocks/Flying

200 Success

"OK"

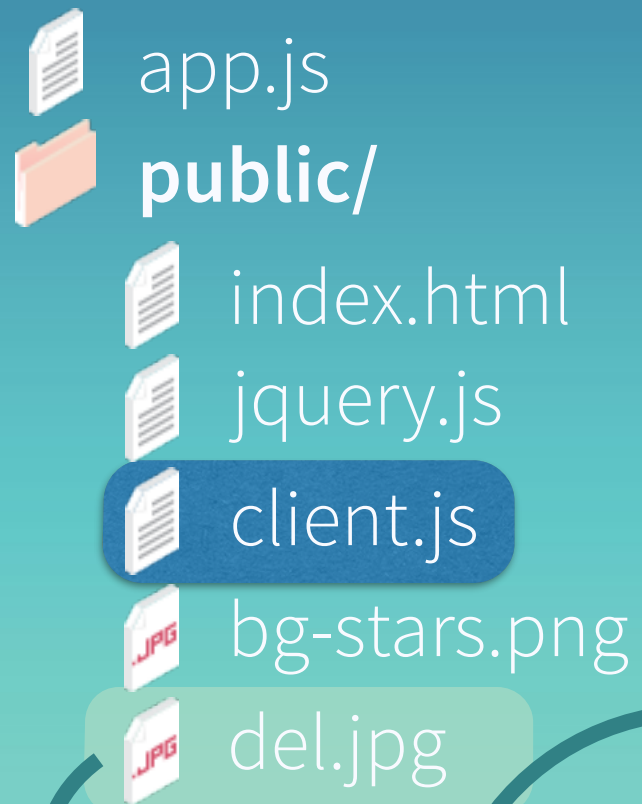
returns proper status code
and "OK" response body

Server



Adding delete links to Blocks list

client.js

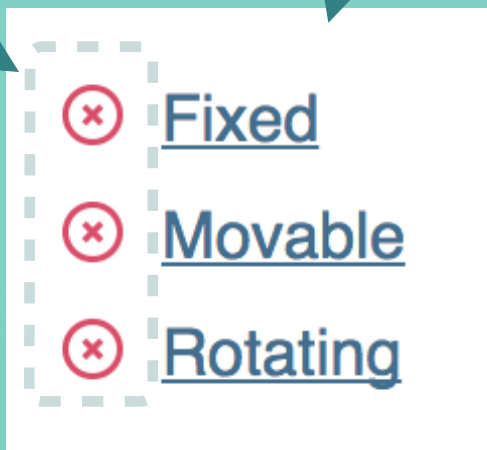


```
$(function(){
    ...

    function appendToList(blocks) {
        var list = [];
        var content, block;
        for(var i in blocks){
            block = blocks[i];
            content = '<a href="/blocks/'+block+'">'+block+'</a> ' +
                '<a href="#" data-block="'+block+'"></a>';

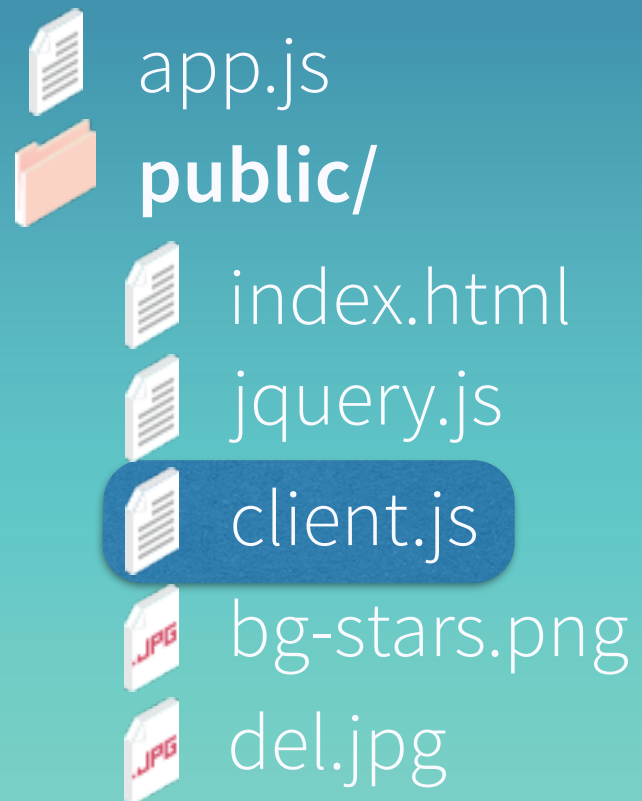
            list.push($('<li>', { html: content }));
        }

        $('<div class="block-list">').append(list)
    }
});
```



Listening for click events

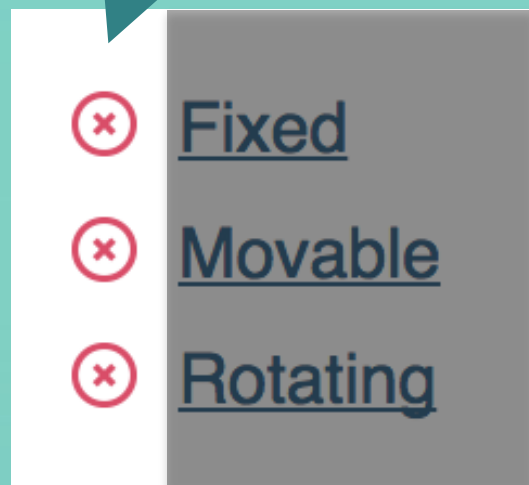
Let's attach an event listener on all links with a **data-block** attribute



```
$(function(){  
  ...  
  $(' .block-list').on('click', 'a[data-block]', function(event){  
    ...  
  });  
});
```

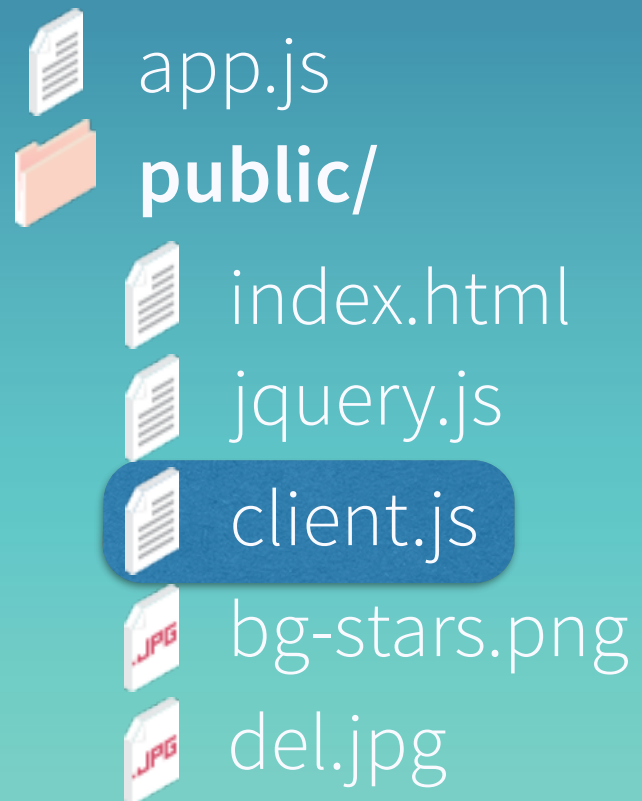
client.js

links with a **data-block** attribute



Making DELETE request to /blocks

client.js



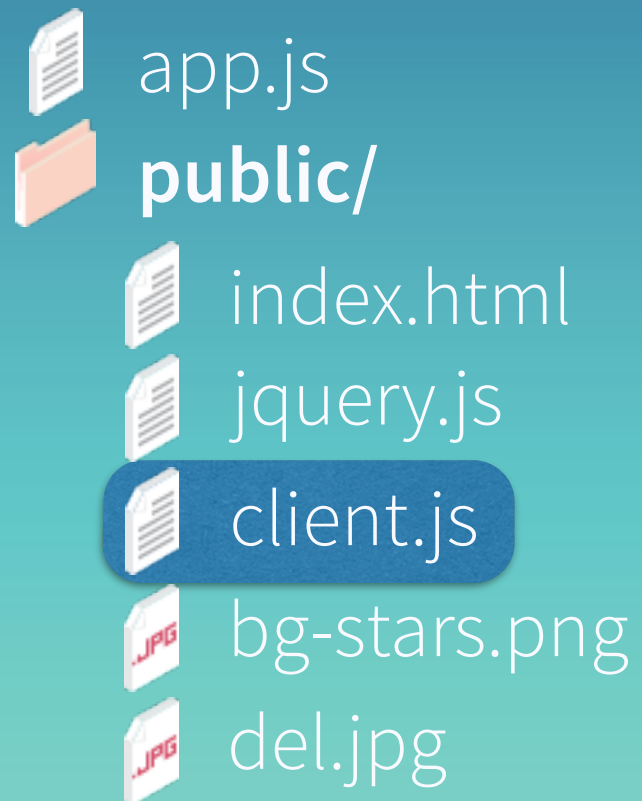
```
$(function(){  
  ...  
  $('.block-list').on('click', 'a[data-block]', function(event){  
    if (!confirm('Are you sure ?')) {  
      return false;  
    }  
  
    var target = $(event.currentTarget);  
  
    $.ajax({  
      type: 'DELETE', url: '/blocks/' + target.data('block')  
    }).done(function() {  
    })  
  });  
});
```

the link element that was clicked

reads the block name from the link's data-block attribute

Removing elements from the page

client.js



```
$(function(){
    ...
    $('.block-list').on('click', 'a[data-block]', function(event){
        if (!confirm('Are you sure ?')) {
            return false;
        }

        var target = $(event.currentTarget);

        $.ajax({
            type: 'DELETE', url: '/blocks/' + target.data('block')
        }).done(function() {
            target.parents('li').remove();
        });
    });
});
```

removes li element
containing the link

Creating a DELETE route

The delete route takes the block **name** as argument



app.js

```
var express = require('express');
var app = express();

var blocks = { ... };

app.delete('/blocks/:name', function(request, response) {

});
...
```


Deleting Blocks

The `delete` operator from JavaScript removes a property from an object

app.js

```
var express = require('express');
var app = express();

var blocks = { ... };

app.delete('/blocks/:name', function(request, response) {
  delete blocks[request.blockName];
});
...
```

removes entry from
the blocks object

in case you don't remember,
this is where we set **blockName**

```
app.param('name', ...)
```

Responding with sendStatus

The `sendStatus` function sets both the status code and the response body


app.js

```
var express = require('express');
var app = express();

var blocks = { ... };

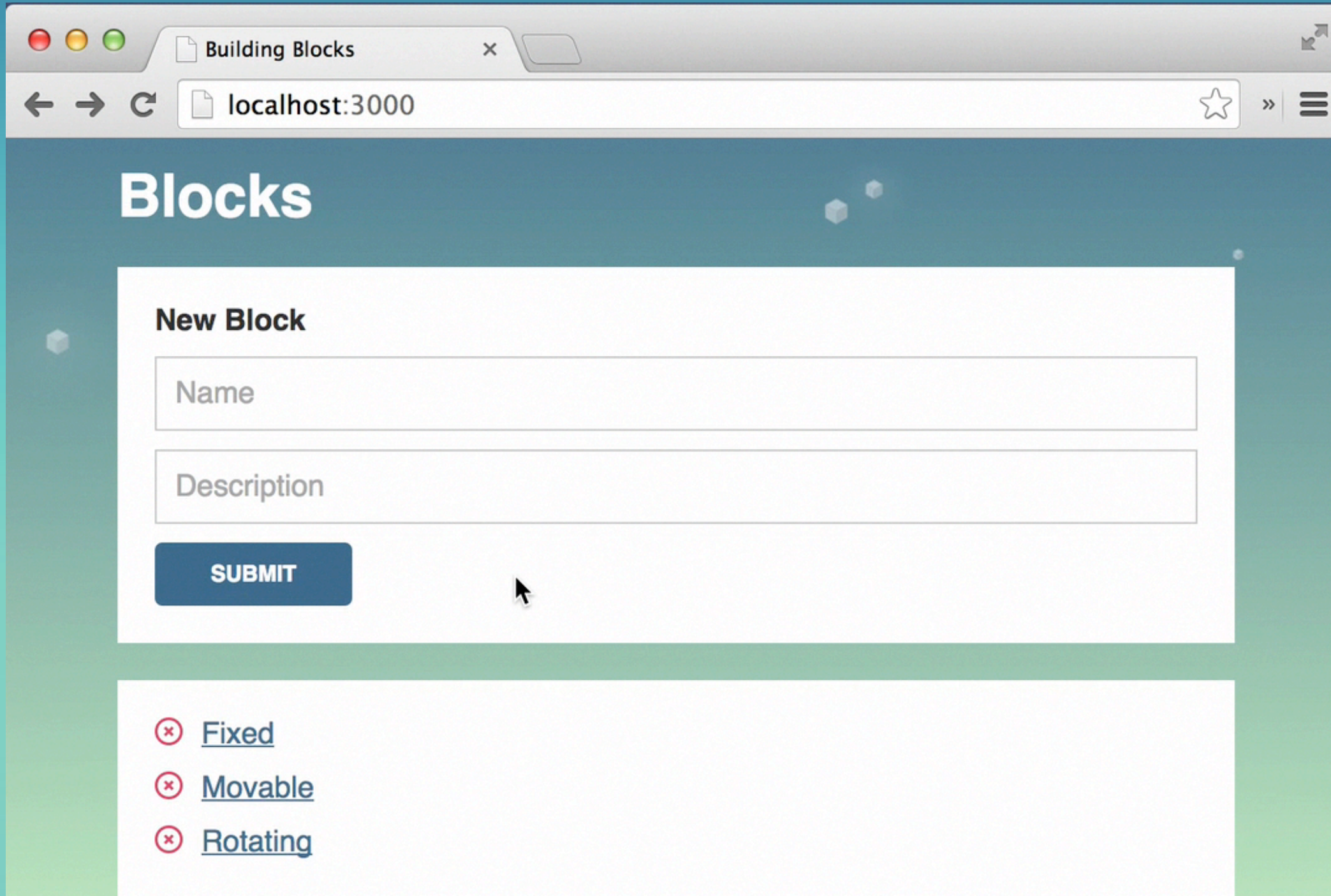
app.delete('/blocks/:name', function(request, response) {
  delete blocks[request.blockName];
  response.sendStatus(200);
});
...
```

also sets response
body to “OK”



Seeing delete in action

Deleting blocks works as expected



The screenshot shows a web browser window with the title 'Building Blocks' and the address bar displaying 'localhost:3000'. The main heading is 'Blocks'. Below it, there is a 'New Block' form with two input fields: 'Name' and 'Description'. A blue 'SUBMIT' button is positioned below the 'Description' field. At the bottom of the form, there are three radio button options: 'Fixed', 'Movable', and 'Rotating', each preceded by a red circle with a white 'x'.

Building Blocks

localhost:3000

Blocks

New Block

Name

Description

SUBMIT

☒ Fixed

☒ Movable

☒ Rotating

