Middleware

How They Work

Level 2 - Part I

Rich JavaScript Applications

They allow for a more **interactive** experience on the web.

Let's build this using **Express!**

Writing index.html

Place HTML files under the **public** folder

app.js
public/
index.html

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Building Blocks</title>
</head>
<body>
  <h1>Blocks</h1>
</body>
</html>
```

Serving files with sendFile

The **index.html** file is served from Express

app.js

public/

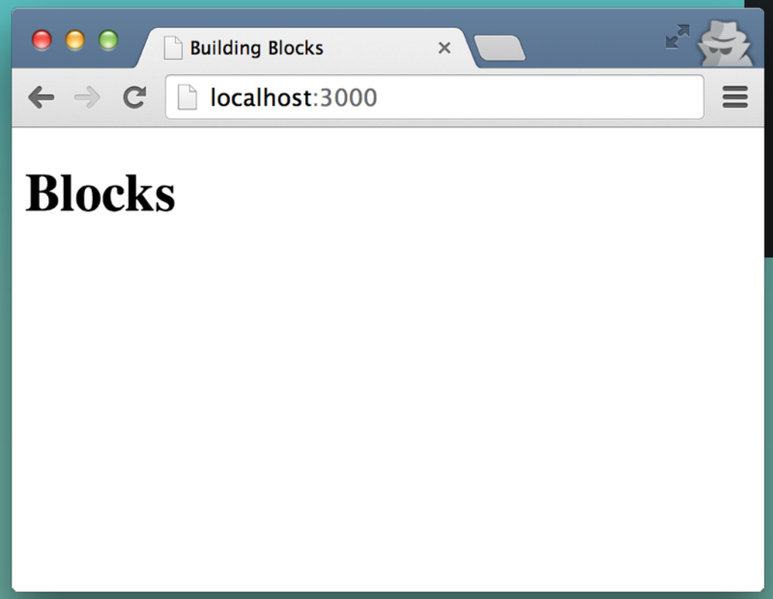
index.html

app.js

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

app.get('/', function(request, response) {
  response.sendFile(__dirname + '/public/index.html');
});

app.listen(3000);
```



name of the directory the currently executing script resides in

Mounting middleware

The **app.use** function adds middleware to the application stack

app.js
public/
index.html

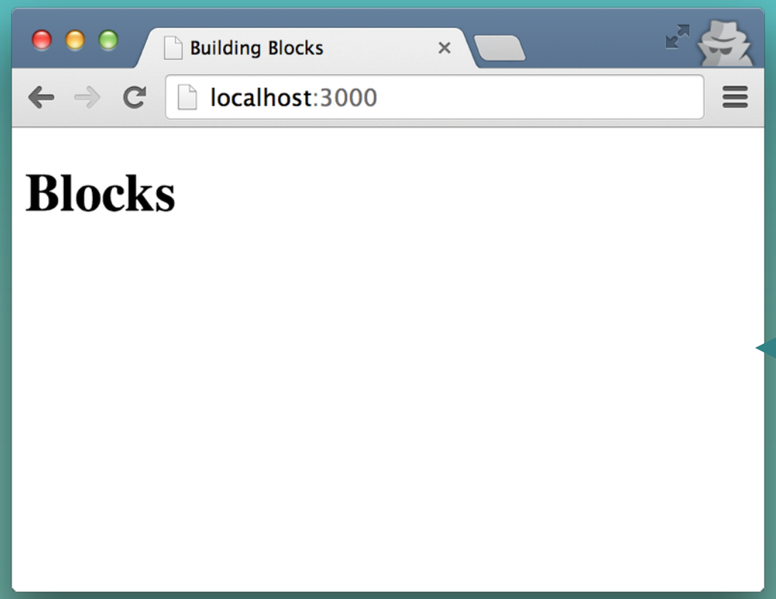
app.js

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

app.use(express.static('public'));

app.listen(3000);
```

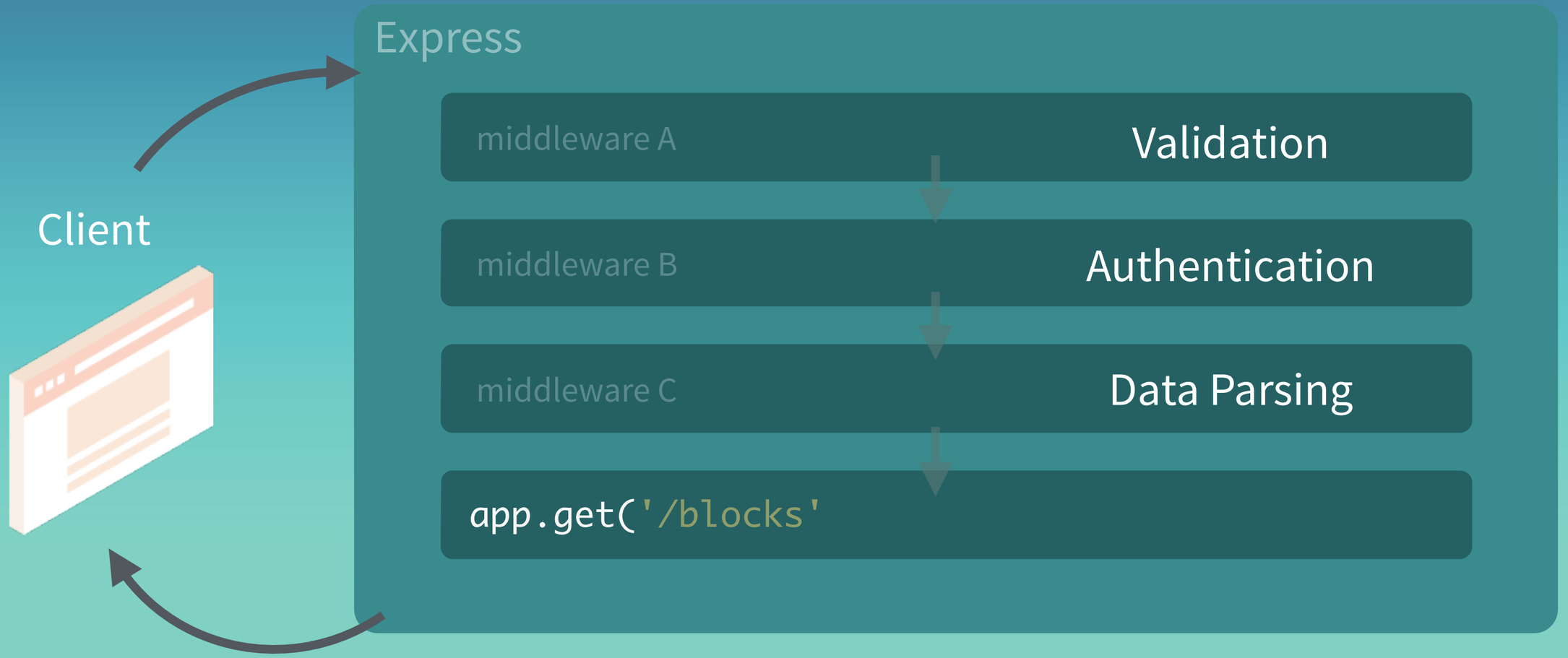
static middleware serving files from the **public** folder



same result!

Understanding Middleware

Functions executed **sequentially** that access **request** and **response**

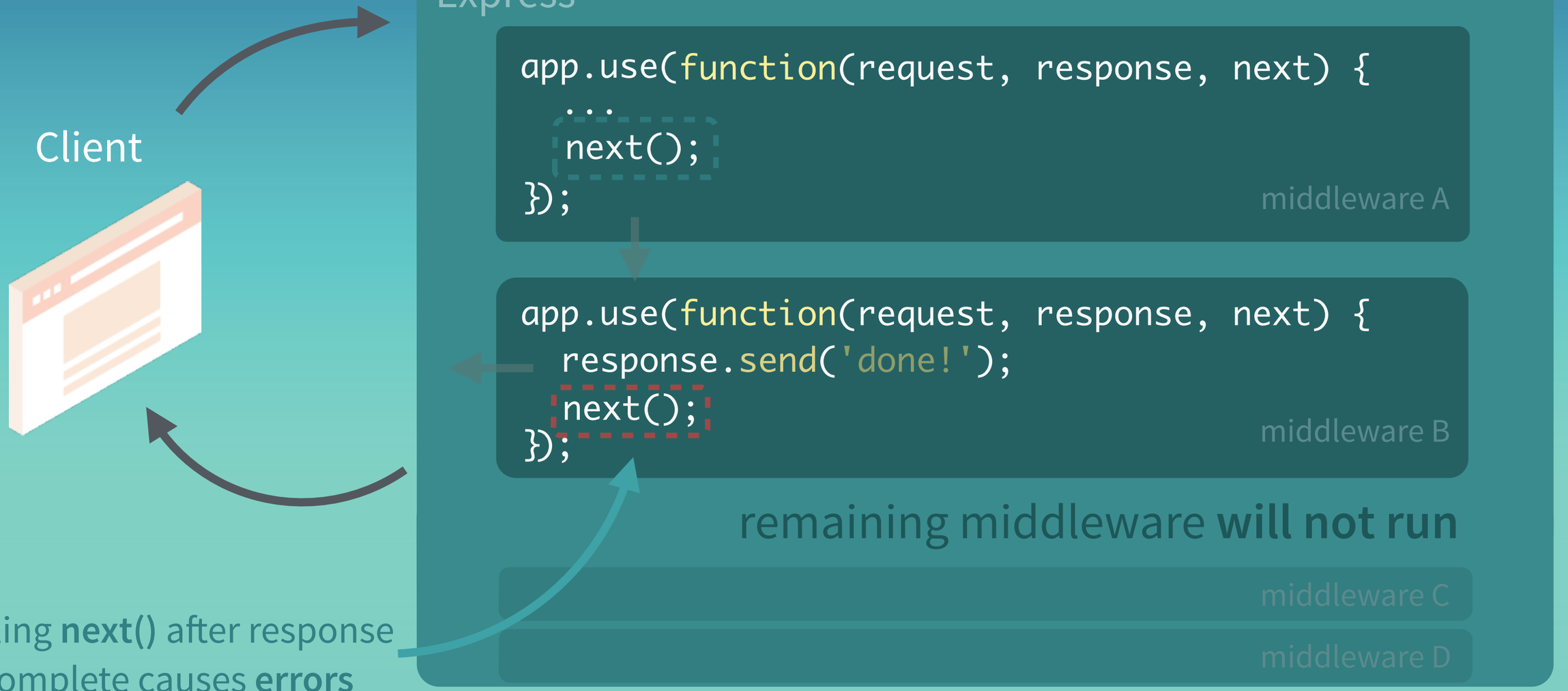


Executing Middleware functions

When **next** is called, processing moves to the next middleware.

Returning from Middleware functions

The flow **stops** once the response is sent back to the client



Reading the static Middleware source

The code for **static** is a good example of Express Middleware

source code from static

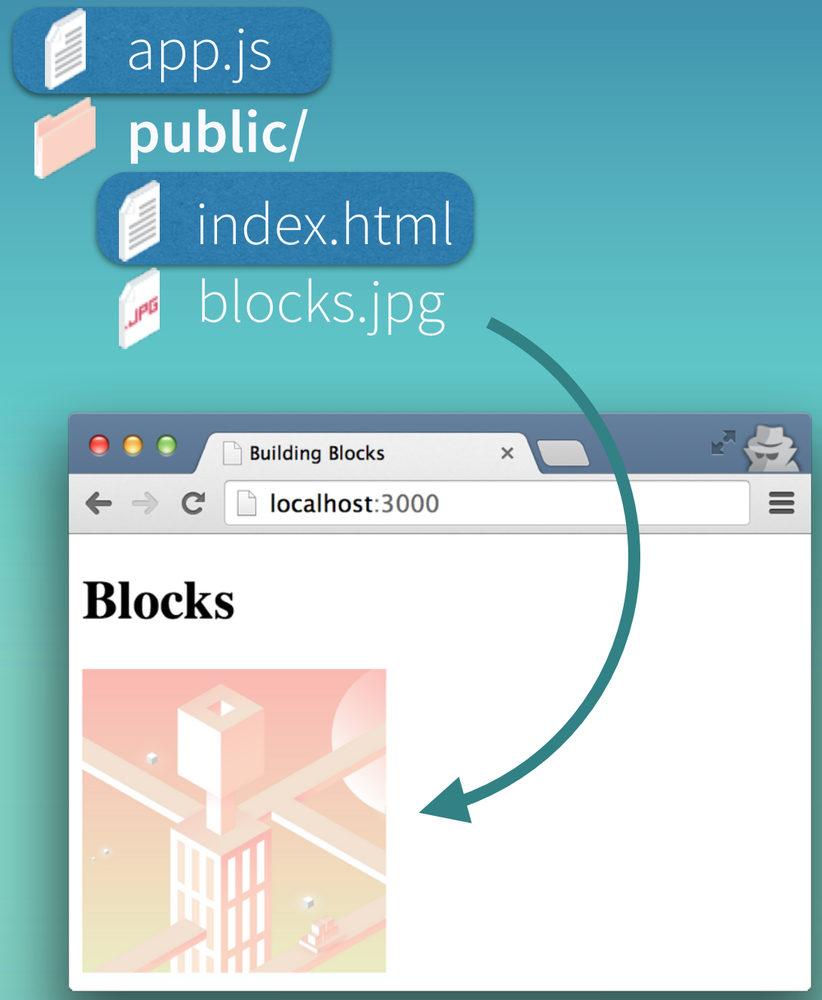
index.js

https://github.com/expressjs/serve-static

```
exports = module.exports = function serveStatic(root, options) {
  ...
  return function serveStatic(req, res, next) {
    if (req.method !== 'GET' && req.method !== 'HEAD') {
      return next()
    }
    ...
    stream.pipe(res)
  }
}
```

Serving static assets

The static middleware serves **everything** under the specified folder



app.js

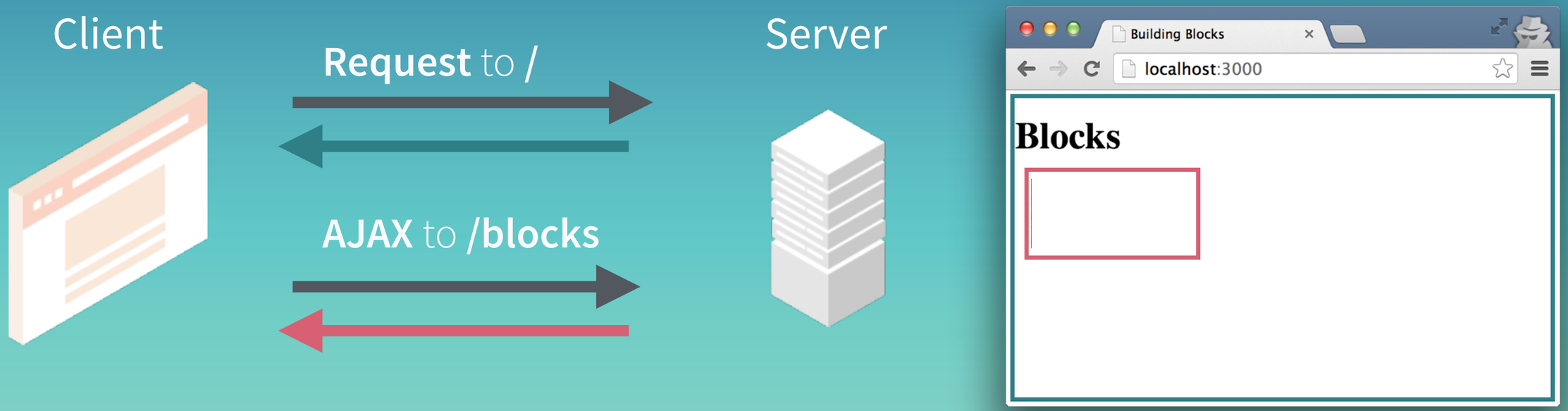
```
app.use(express.static('public'));
```

index.html

```
<!DOCTYPE html>
...
<body>
  <h1>Blocks</h1>
  <p><img src='blocks.png'></p>
</body>
</html>
```

Fetching a List of Blocks

Loading **data** from Express with AJAX calls



Adding client-side JavaScript

Place all files under the **public** folder

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Building Blocks</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h1>Blocks</h1>

  <ul class='block-list'></ul>

  <script src="jquery.js"></script>
  <script src="client.js"></script>
</body>
</html>
```

Making AJAX calls

Request to **/blocks,** then append results to **block-list**

returns blocks in **JSON** format

client.js

```
$(function(){

  $.get('/blocks', appendToList);

  function appendToList(blocks) {
    var list = [];
    for(var i in blocks){
      list.push($('<li>', { text: blocks[i] }));
    }
    $('.block-list').append(list);
  }
});
```

Responding with JSON

app.js

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

app.use(express.static('public'));

app.get('/blocks', function(request, response) {
  var blocks = ['Fixed', 'Movable', 'Rotating'];
  response.json(blocks);
});

app.listen(3000);
```