Route Files

Level 5 - Part II

Single application file is too long

app.js

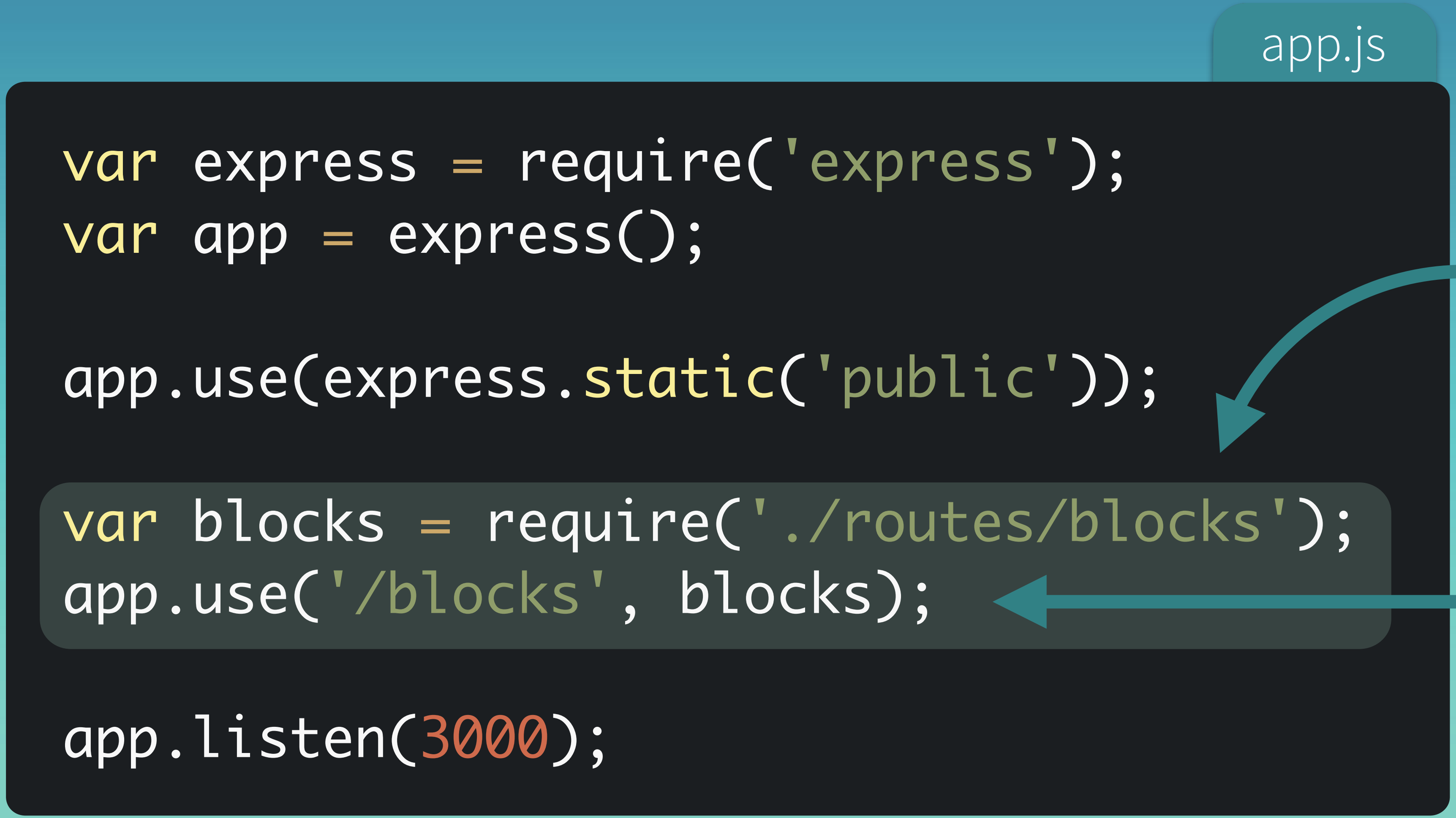
```
var express = require('express');
...
app.route('/blocks')
  .get(function(request, response) {
    ...
  });
  .post(parseUrlencoded, function(request, response) {
    ...
  });
app.route('/blocks/:name')
  .get(function(request, response) {
    ...
  })
  .delete(function(request, response) {
    ...
  });
...
```

Too many lines of code in a text file is a **bad smell**

Our **app.js** file
is growing too long.

Extracting routes to modules

This helps clean up our code and allows our main **app.js** file to easily accommodate additional routes in the future.



app.js

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

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

var blocks = require('./routes/blocks');
app.use('/blocks', blocks);

app.listen(3000);
```

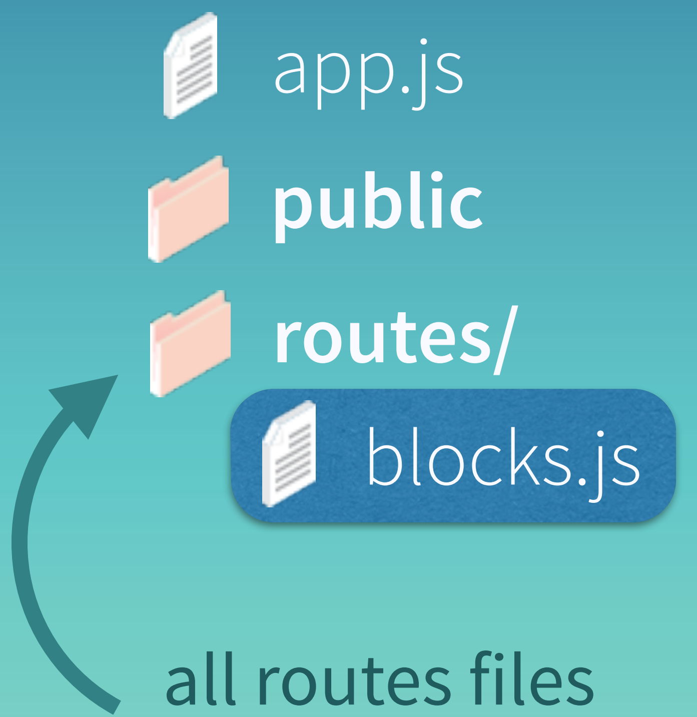
we'll move our routes to this new file

router is mounted in a particular **root url**

Let's see how we can do this by taking advantage of Node's **module** system.

Writing the new routes file

A dedicated folder for **routes** can help organize our code



all routes files
will go in this folder

blocks.js

```
var express = require('express');
var router = express.Router();
```

returns router instance which can
be *mounted* as a middleware

Exporting the router object

We assign the router to **module.exports** to make it accessible from other files

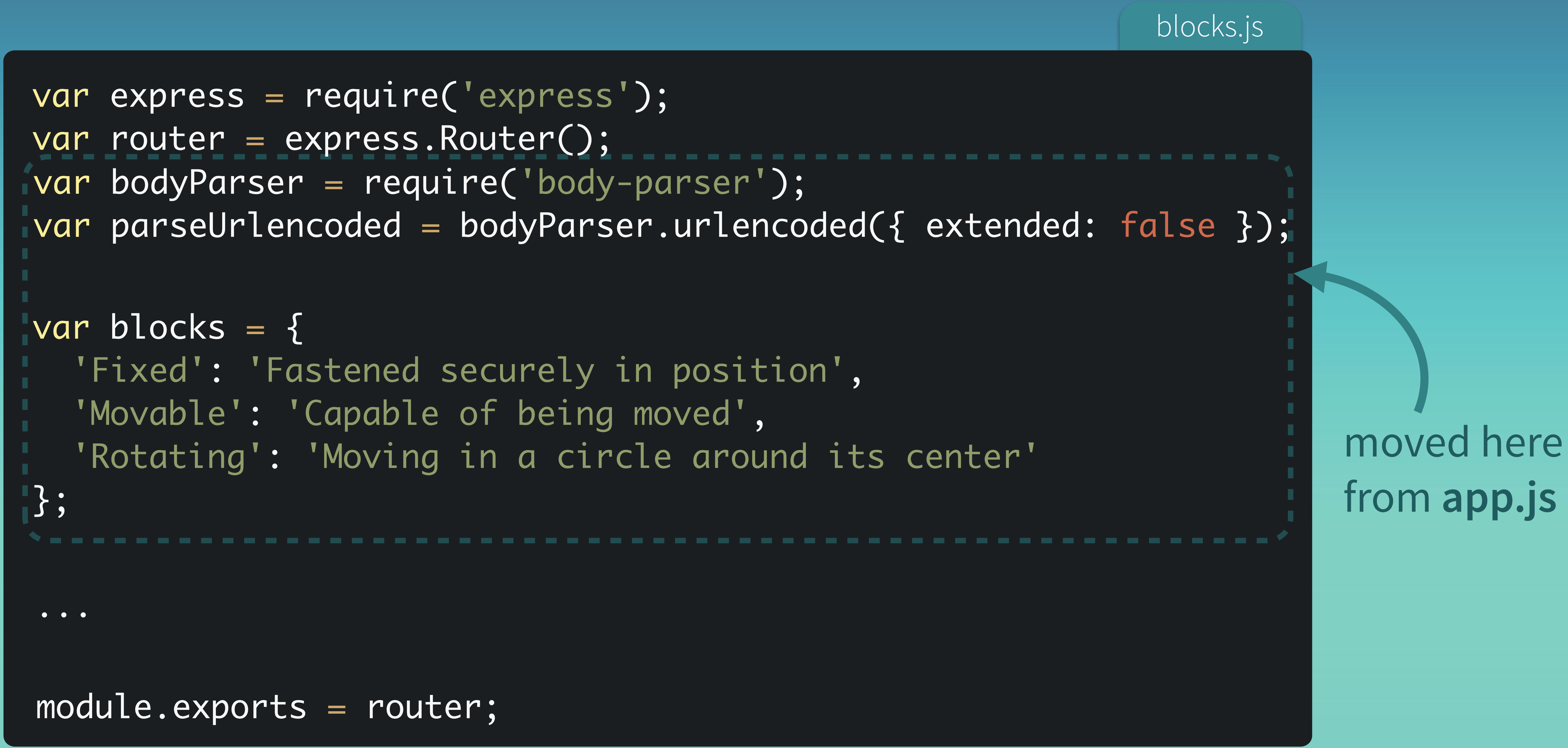
blocks.js

```
var express = require('express');
var router = express.Router();

...

module.exports = router;
```

exports the router
as a Node **module**

Extracting GET /blocks to routes file

All block-related logic is **encapsulated** inside its routes file



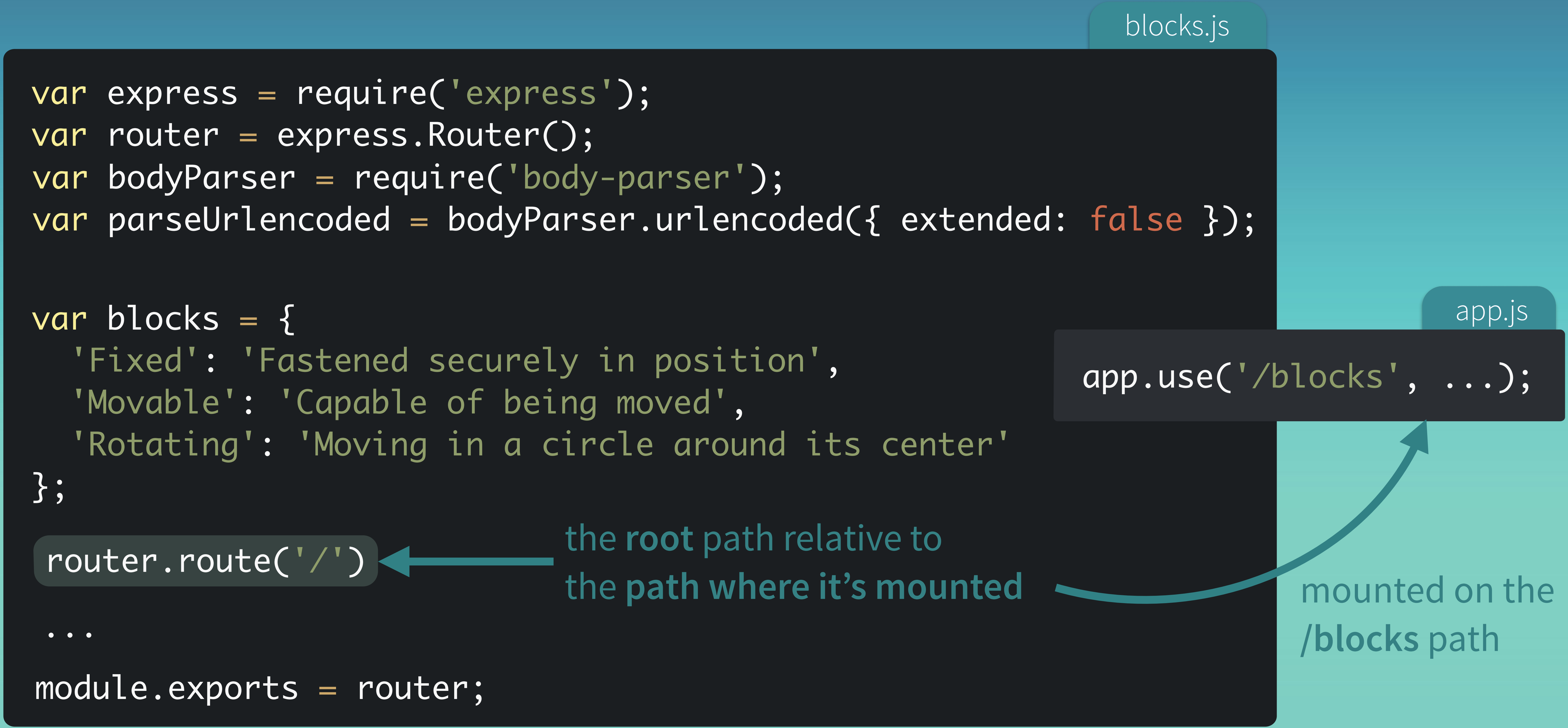
blocks.js

```
var express = require('express');
var router = express.Router();
var bodyParser = require('body-parser');
var parseUrlencoded = bodyParser.urlencoded({ extended: false });

var blocks = {
  'Fixed': 'Fastened securely in position',
  'Movable': 'Capable of being moved',
  'Rotating': 'Moving in a circle around its center'
};

...

module.exports = router;
```

moved here
from **app.js**

Building the router for /blocks

Router path is **relative** to the where it's mounted



blocks.js

```
var express = require('express');
var router = express.Router();
var bodyParser = require('body-parser');
var parseUrlencoded = bodyParser.urlencoded({ extended: false });

var blocks = {
  'Fixed': 'Fastened securely in position',
  'Movable': 'Capable of being moved',
  'Rotating': 'Moving in a circle around its center'
};

router.route('/')

 ...

module.exports = router;
```

the **root** path relative to the **path where it's mounted**

app.js

```
app.use('/blocks', ...);
```

mounted on the **/blocks** path

Extracting GET /blocks to routes file

Routes are moved unmodified from the app.js file

blocks.js

```
var express = require('express');
var router = express.Router();
...
router.route('/')
  .get(function (request, response) {
    ...
  });
  .post(parseUrlencoded, function(request, response) {
    ...
  });



...
```

same as before

Building the router for /blocks/:name

The **route** function uses the same format for dynamic routes

blocks.js

```
var express = require('express');
var router = express.Router();
...
router.route('/')
  .get(function (request, response) {
    ...
  });
  .post(parseUrlencoded, function(request, response) {
    ...
  });
router.route('/:name')


...
```

the **/:name** path relative to the **path where it's mounted**

app.js

```
app.use('/blocks', ...);
```

mounted on the **/blocks** path

Extracting GET /blocks to routes file

The **all** route is called for **all** requests for a given URL path

blocks.js

```
var express = require('express');
var router = express.Router();
...
router.route('/')

...
router.route('/:name')
  .all(function (request, response, next) {
    var name = request.params.name;
    var block = name[0].toUpperCase() + name.slice(1).toLowerCase();
    request.blockName = block;
    next();
  })

...
```

the same code that turns first characters to uppercase and remaining characters to lowercase

used to be here

```
app.param('name/'...
```

Extracting GET /blocks to routes file

blocks.js

```
var express = require('express');
var router = express.Router();
...
router.route('/')

...

router.route('/:name')
  .all(function (request, response, next) {
    ...
  })
   .get(function (request, response) {
     ...
  });
   .delete(function (request, response) {
     ...
  });
...
```

same as before

The complete blocks routes file

Our routes file is ready to be mounted in our application

blocks.js

```javascript
var express = require('express');
var router = express.Router();
var bodyParser = require('body-parser');
var parseUrlencoded = bodyParser.urlencoded({ extended: false });

var blocks = { ... }

router.route('/')
  .get(function (request, response) { ...
  .post(parseUrlencoded, function(request, response) { ...

router.route('/:name')

   .all(function (request, response) { ...
   .get(function (request, response) { ...
   .delete(function (request, response) { ...

module.exports = router;
```

Requiring the router in the application

Our router is simply a Node module and can be **required** like so

- app.js
- **public**
- **routes/**
  - blocks.js

app.js

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

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

var blocks = require('./routes/blocks');


app.listen(3000);
```

Mounting the router in the application

All requests to the /**blocks** url are dispatched to the **blocks** router

- app.js
- **public**
- **routes/**
  - blocks.js

app.js

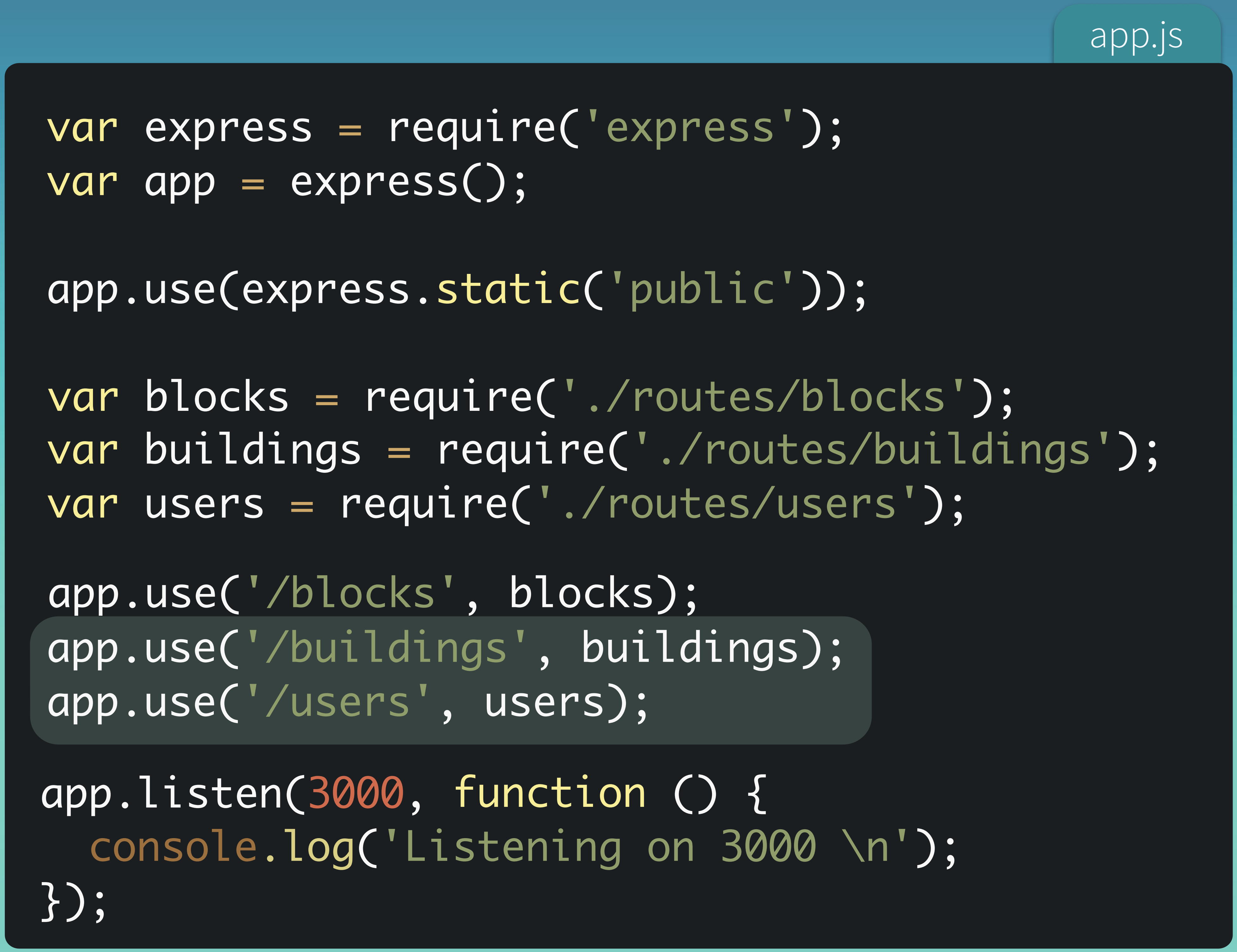
```
var express = require('express');
var app = express();

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

var blocks = require('./routes/blocks');
app.use('/blocks', blocks);

app.listen(3000);
```

router is mounted in a particular **root url**

Additional route mappings

The **app.js** file is now ready to support multiple routes and still look clean

- app.js
- **public**
- **routes/**
  - blocks.js
  - buildings.js
  - users.js



app.js

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

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

var blocks = require('./routes/blocks');
var buildings = require('./routes/buildings');
var users = require('./routes/users');

app.use('/blocks', blocks);
app.use('/buildings', buildings);
app.use('/users', users);

app.listen(3000, function () {
  console.log('Listening on 3000 \n');
});
```