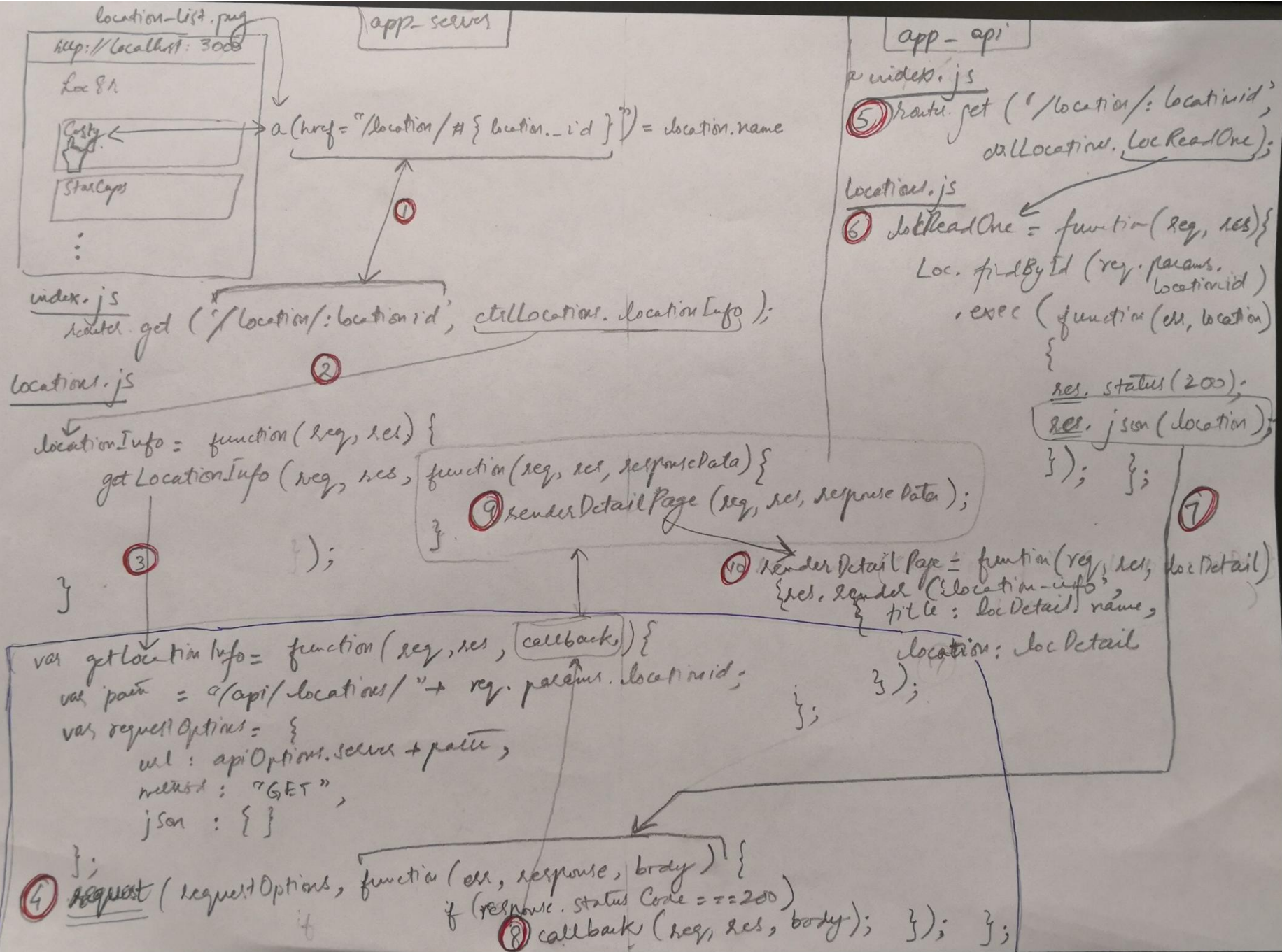


Web Programming

Lecture 16

Consuming a REST API

Consuming an API: The Request Response Cycle



Today's Topics

- Getting single documents from an API
- Adding data to the database via the API

Getting single documents from an API

The Loc8r Details page

- Setting URLs and routes to access specific MongoDB documents
 - The API route for a single location is `/api/locations/:locationid`
 - Update the route for the main Express application to contain the `locationid` parameter

```
1 router.get('/', ctrlLocations.homelist);  
2 router.get('/location/:locationid', ctrlLocations.locationInfo);  
3 router.get('/location/review/new', ctrlLocations.addReview);
```

Update the list view to add the location ID to the relevant links

```
1 block content
2   #banner.page-header
3     .row
4       .col-lg-6
5         h1= pageHeader.title
6         small &nbsp;#{pageHeader.strapline}
7     .row
8       .col-xs-12.col-sm-8
9         .error= message
10      .row.list-group
11        each location in locations
12          .col-xs-12.list-group-item
13            h4
14              a(href="/location/#{location._id}")= location.name
15              small &nbsp;
16                +outputRating(location.rating)
17              span.badge.pull-right.badge-default= location.distance
18              p.address= location.address
19              p
20                each facility in location.facilities
21                  span.label.label-warning= facility
22                  &nbsp;
23      .col-xs-12.col-sm-4
24        p.lead= sidebar
```

Separating concerns: Moving the rendering into a named function

Listing 7.11. Move the contents of the `locationInfo` controller into an external function

```
1 var renderDetailPage = function (req, res) {  
2   res.render('location-info', {  
3     title: 'Starcups',  
4     ...  
5   });  
6 };  
7 module.exports.locationInfo = function(req, res){  
8   renderDetailPage(req, res);  
9 };
```

1

2

Querying the API using a unique ID from a URL parameter

Listing 7.12. Update the `locationInfo` controller to call the API

```
1 module.exports.locationInfo = function(req, res){
2   var requestOptions, path;
3   path = "/api/locations/" + req.params.locationid;
4   requestOptions = {
5     url : apiOptions.server + path,
6     method : "GET",
7     json : {}
8   };
9   request(
10    requestOptions,
11    function(err, response, body) {
12      renderDetailPage(req, res);
13    }
14  );
15 };
```

Passing the data from the API to the view

Listing 7.13. Preprocessing data in the controller

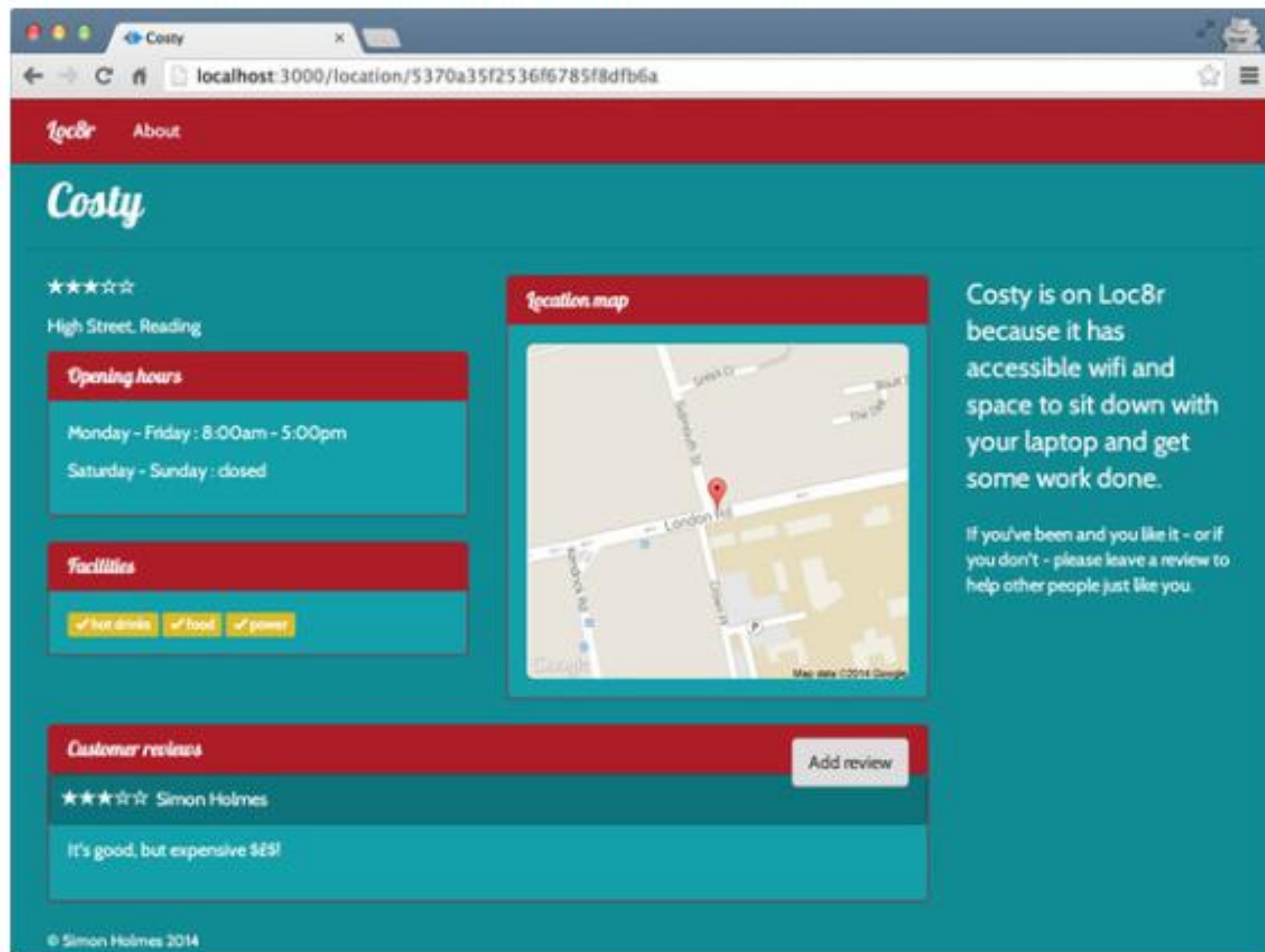
```
1 request(  
2   requestOptions,  
3   function(err, response, body) {  
4     var data = body;  
5     data.coords = {  
6       lng : body.coords[0],  
7       lat : body.coords[1]  
8     };  
9     renderDetailPage(req, res, data);  
10  }  
11 );
```


Passing the data from the API to the view

Listing 7.14. Update `renderDetailPage` to accept and use data from the API

```
1 var renderDetailPage = function (req, res, locDetail) {  
2   res.render('location-info', {  
3     title: locDetail.name,  
4     pageHeader: {title: locDetail.name},  
5     sidebar: {  
6       context: 'is on Loc8r because it has accessible wifi and space t  
7 down with your laptop and get some work done.',  
8       callToAction: 'If you\'ve been and you like it - or if you don\  
9 please leave a review to help other people just like you.'  
10    },  
11    location: locDetail  
12  });  
13 };
```

Figure 7.6. Details page pulling in data from MongoDB via the API



Adding data to the database via the API: Add Loc8r reviews

- Steps:
 - Make the review form aware of which location the review is for
 - Create a route for the form to POST to
 - Send the new review data to the API
 - Show the new review in place on the Details page

Setting up the routing and views

- Defining the two review routes
- In the index.js file in the /routes folder
- ```
router.get('/', ctrlLocations.homelist);
router.get('/location/:locationid', ctrlLocations.locationInfo);
router.get('/location/:locationid/reviews/new',
ctrlLocations.addReview);
router.post('/location/:locationid/reviews/new',
ctrlLocations.doAddReview);
```

- Fixing the location detail view

```
1 a.btn.btn-default.pull-right(href="/location/#{location._id}/reviews/new")
2 | Add review
```

# Setting up the routing and views

- Updating the review form view

```
form.form-horizontal(action="/location", method="get", role="form")
```

---

/location/:locationid/reviews/new

```
form.form-horizontal(action="", method="post", role="form")
```

## Creating a named function for rendering the Add Review page

```
1 var renderReviewForm = function (req, res) {
2 res.render('location-review-form', {
3 title: 'Review Starcups on Loc8r',
4 pageHeader: { title: 'Review Starcups' }
5 });
6 };
7 /* GET 'Add review' page */
8 module.exports.addReview = function(req, res){
9 renderReviewForm(req, res);
10 };
```

# Getting the location detail

Listing 7.19. Create a new reusable function to get location information

```
1 var getLocationInfo = function (req, res, callback) {
2 var requestOptions, path;
3 path = "/api/locations/" + req.params.locationid;
4 requestOptions = {
5 url : apiOptions.server + path,
6 method : "GET",
7 json : {}
8 };
9 request(
10 requestOptions,
11 function(err, response, body) {
12 var data = body;
13 if (response.statusCode === 200) {
14 data.coords = {
15 lng : body.coords[0],
16 lat : body.coords[1]
17 };
18 callback(req, res, data);
19 } else {
20 _showError(req, res, response.statusCode);
21 }
22 }
23);
24 };
```

## Getting the location detail

```
25 module.exports.locationInfo = function(req, res){
26 getLocationInfo(req, res, function(req, res, responseData) {
27 renderDetailPage(req, res, responseData);
28 });
29 };
30 module.exports.addReview = function(req, res){
31 getLocationInfo(req, res, function(req, res, responseData) {
32 renderReviewForm(req, res, responseData);
33 });
34 };
```



# Displaying the location detail

Listing 7.20. Removing hard-coded data from the `renderReviewForm` function

```
1 var renderReviewForm = function (req, res, locDetail) {
2 res.render('location-review-form', {
3 title: 'Review ' + locDetail.name + ' on Loc8r',
4 pageHeader: { title: 'Review ' + locDetail.name }
5 });
6 };
```

Figure 7.10. Add Review page pulling in the location name via the API, based on the ID contained in the URL

Review Costy on Loc8r

localhost:3000/location/5370a35f2536f6785f8dfb6a/reviews/new

*Loc8r* About

# *Review Costy*

Name

Rating

Review

Add my review

© Simon Holmes 2014

# POSTing the review data to the API

- Steps
  - Get the locationid from the URL to construct the API request URL
  - Get the data posted in the form and package it up for the API
  - Make the API call
  - Show the new review in place if successful
  - Display an error page if not successful

## Listing 7.21. `doAddReview` controller used to post review data to the API

```
1 module.exports.doAddReview = function(req, res){
2 var requestOptions, path, locationid, postdata;
3 locationid = req.params.locationid;
4 path = "/api/locations/" + locationid + '/reviews';
5 postdata = {
6 author: req.body.name,
7 rating: parseInt(req.body.rating, 10),
8 reviewText: req.body.review
9 };
10 requestOptions = {
11 url : apiOptions.server + path,
12 method : "POST",
13 json : postdata
14 };
15 request(
16 requestOptions,
17 function(err, response, body) {
18 if (response.statusCode === 201) {
19 res.redirect('/location/' + locationid);
20 } else {
21 _showError(req, res, response.statusCode);
22 }
23 }
24);
25 };
```

# QUIZ 3