Express #3: Advanced Routing & Error Handling

Routes As We Know Them

- Routes in Express are functions that run when the user requests a page that has the exact same path as the route

- Routes render something to the user (res.render), but can also do things that the user doesn't see (console.log)

- Routes only listen for the HTTP method type we specify the route with

Advanced Routing: Route Parameters

- In addition to query parameters, we can specify parameters in the route

- This is pretty common when there's one "main" parameter, like a username, or some entity's unique ID

- We specify this using : and then naming it, like :user or :id

```
app.get("/books/:bookId", function(req, res) {
    console.log(req.params.bookId); // Path /books/123 would output "123"
});
```

Advanced Routing: Route Parameters (cont.)

- You are allowed to have multiple route parameters in one route

- They must have different names, and be spaced apart by a non-alphanumeric character

```javascript
app.get("/user/:userId/art/:artId", function(req, res) {
  // Path /user/123/art/456
  console.log(req.params.userId); // Logs "123"
  console.log(req.params.artId); // Logs "456"
});

app.get("/flights/:from-:to", function(req, res) {
  // Path flights/ewr-sfo
  console.log(req.params.from); // Logs "ewr"
  console.log(req.params.to); // Logs "sfo"
});
```

Error Handling

- Now that our routes are getting more complicated, we'll want to handle bad requests

- If you recall from the HTTP lesson, requests return a status code

- All our requests have returned 200 so far by default, because anything in the 2XX range is a success

- But if the user does something wrong, we want to send a 4XX code

- Or if we experience an error on our end, we'll want a 5XX code

Error Handling: Code

- Using `res`, we can set the response code using the `status()` method

```
app.get("/admin", function(req, res) {
    // If they're not an admin, error out
    if (!isAdmin) {
        res.status(403); // 401 - Unauthorized
        return res.send("You don't have permission!");
    }

    // Otherwise, render the admin...
    res.render("admin");
});
```

Error Handling: Code (cont.)

- You can also chain status, since it returns `res`

```
// This bit of code...
res.status(500);
res.send("Error!");

// Is exactly the same as this...
res.status(500).send("Error!");
```

Exercise: Country Info App

- Using everything we just learned, we're going to make a page that displays some info about any country

- Grab the JSON file from <u>here</u>. It's a big object that contains country info, keyed by 3 letter code.

- Given a path that contains that country's 3 char code, i.e. /country/usa, display that country.

- Render the English language name, native name, population, region, and show a picture of the flag by passing the country to res.render.

- You should also return an error if they provide an incorrect country, 400 is a good code for this.

- Bonus: If you <u>set the query</u> to ?json=1, make it <u>output JSON</u> instead of a page

You should reuse your template, and just make a new page for this.
Try styling up the country display as well.

Additional Resources

- Express' Routing Guide