Views Reflection

Question #1

- Assume you are doing a PUT of a new item into a collection of cats, and the full set of cats will be returned to be displayed
- Reorder the following set of steps
- Note whether each step is executed on the client or server
- Indicate where network operations go between two steps.

1. success: function(result)
2. $ajax()
3. res.render()
4. catsModel.push(newcat)
5. <% for (var x = 0 … %>

Question #2

- What does "render" mean?

Question #3

- Does the order of the following in red matter? Why?

```
app.use(morgan('tiny'));

// Load all routes in the routes directory
fs.readdirSync('./routes').forEach(function (file){
  // There might be non-js files in the directory that
  should not be loaded
  if (path.extname(file) == '.js') {
   require('./routes/'+ file).init(app);
   }
});

// Handle static files
app.use(express.static(__dirname + '/public'));
```

Question #4

- Regarding:
  app.use(express.static(__dirname + '/public'));

- Why __dirname+'/public'?
- Why not "."?
- Why not __dirname+'/views'?

__dirname and .

- __dirname is the directory name of the current module. So in this case, your app.js.
- "." is the directory the terminal shell was in from which you ran node

- With require, e.g. require(./routes/ducks.js), the file is always relative to the current file.
  - So you don't need __dirname

REST*ful* style

- An architectural pattern (i.e. a style of design)
- Client-server
- The web app is thought of as being comprised of *resources*
- Resources are addressable
- HTTP methods adhere to the HTTP Standard
  - E.g. get, put, delete, post respect safety and idempotence
- The client-server communication is stateless
  - No client context is kept on the server
  - This is a different approach than keeping sessions
    - For convenience, many otherwise REST*ful* web services keep sessions (e.g. for user logins).

Question 4: Restful URL language

Critique your URL language for today's homework in terms of these four Restful characteristics:

- Addressability
  - Interesting aspects of your service are immediately accessible by a unique URI.
- Granularity
  - How fine-grained are the interesting aspects of your service addressable?
- Transparency
  - The meaning and context of a path are easily apparent to those who did not design the site
- Persistence
  - Addresses of resources on your site do not change over time.

Source: *Restful Web Services by Leonard Richardson and Sam Ruby (O'Reilly Media, 2007).*