

## Routes / Express

# Shortcomings of using the core http

Writing your own code to:

- Handle static pages (html, js, css...)
- Parse and use conditional statements to figure out the route.

```
if (req.pathname == "/doCalc") {  
    doCalcPath(req.query.queryStringNumber, response);  
} else {  
    // Pass doFilePath the filename minus the initial slash  
    doFilePath(req.pathname.slice(1), response);  
}
```
- Select appropriate HTTP method
- In short: it would be nice to have some additional support for the basic and common needs of a web application.

# Express

- A "Fast, unopinionated, minimalist web framework"
  - For building single / multi-page / hybrid web apps
  - Builds on:
    - HTTP (core node)
    - And a lot of other modules
- Home: <http://expressjs.com>
- Provides
  - Easy routing
  - Simple serving static pages
  - A framework for lots of 3<sup>rd</sup> party middleware for help with:
    - Sessions
    - Cookies
    - And lots more...
  - Connection to templating (similar to erb ...)

# Express – defining paths

---

- Express allows for using regular expression pattern matching in paths
- Path options include
  - fixed string paths
  - parameters
  - optional parameters
  - wildcard parameters
  - any regular expressions

# Fixed

---

- `/user/list/all`
  - matches the path: `/user/list/all`

# Optional

---

- `/user(s)?/list`
  - matches `/user/list` and `/users/list`

# Wildcard

---

- `/users/*`
  - matches anything starting with `/users/`
  - `req.params[0]` would be the value of `*`
- E.g.
  - `/users/editors/Fred`
  - `req.params[0]` would be `"editors/Fred"`

# Parameters

---

- `/user/:username/:property`
  - matches `/user/anything/anything`
  - `req.params.username` will have the value of the 2nd part of the path
  - `req.params.property` will have the value of the 3rd part of the path
- E.g.
  - `/user/joemertz/firstname`
    - `req.params.username` will be `joemertz`
    - `req.params.property` will be `firstname`



# Optional parameters

- `/user/:username/:property?`
  - The `?` makes property optional
- E.g.
  - `/user/joemertz`
    - `req.params.username == "joemertz"`
  - `/user/joemertz/firstname`
    - `req.params.username == "joemertz"`
    - `req.params.property == "firstname"`
- Question: If you leave off the `?`, how would these paths be handled differently?

# Regular expressions

---

- Paths can be described as full regular expressions
- Each matched component `n` will be addressable by `req.params[n]`
- You probably won't need to get this in your project.