

# **Model-View-Controller (MVC)**

# Model-View-Controller (MVC)

- MVC is a software architecture pattern used for user interfaces (e.g. web applications!)
- Separate code into three parts, with different purposes
  - **Model**: the data
  - **View**: interface for users to view/change the data
  - **Controller**: methods to change the data

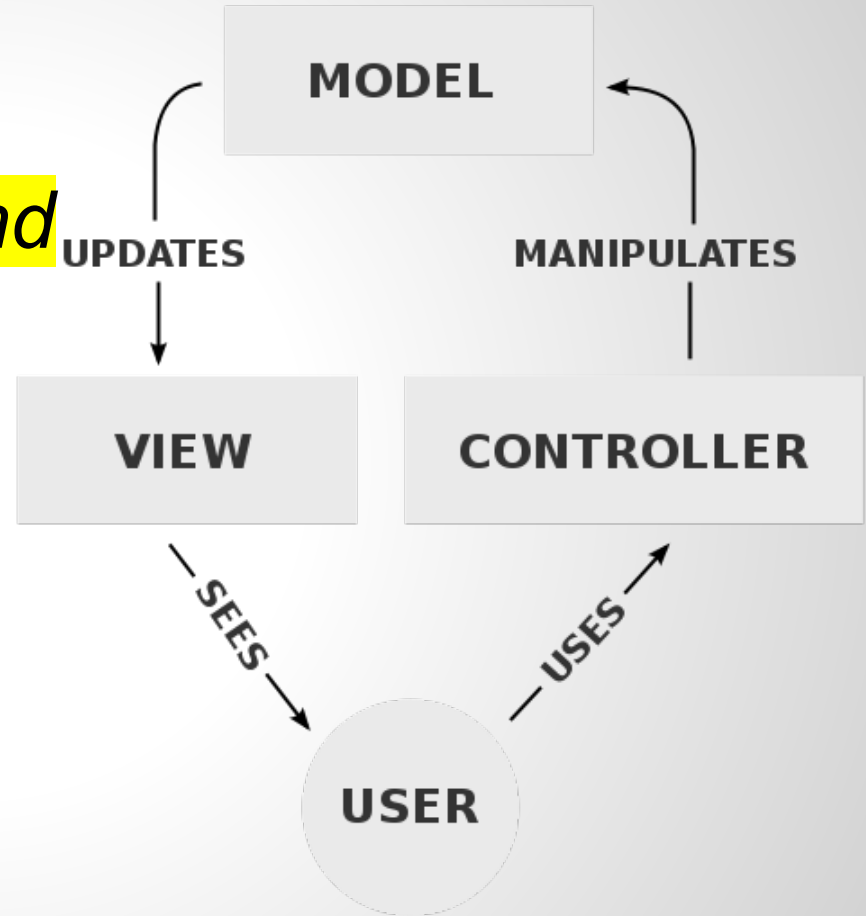
# Why MVC?

- Remember, we software developers love modularity and separation of concerns!
- Same big reason why we:
  - Separate HTML from CSS (no inline styling)
  - Separate servers and clients

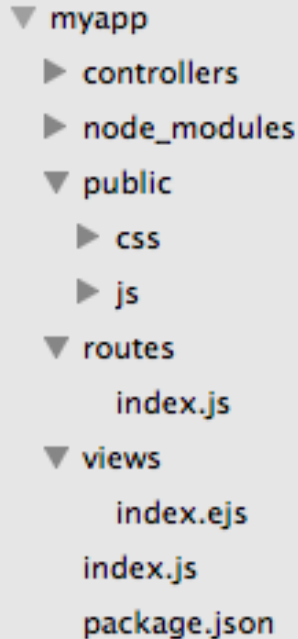
- **Model:** the data. *data model does not depend on views or controllers*

- **View:** interface for users to view/change the data

- **Controller:** methods to change the data



# “MVC” in Express



```
graph TD; myapp[myapp] --> controllers[controllers]; myapp --> node_modules[node_modules]; myapp --> public[public]; myapp --> routes[routes]; myapp --> views[views]; public --> css[css]; public --> js_public[js]; routes --> index_js[index.js]; views --> index_ejs[index.ejs]; views --> index_js_views[index.js]; views --> package_json[package.json];
```

▼ myapp

- ▶ controllers
- ▶ node\_modules
- ▼ public
  - ▶ css
  - ▶ js
- ▼ routes
  - index.js
- ▼ views
  - index.ejs
  - index.js
  - package.json

- *create these directories/files*
- *npm install ejs*
- *app.js*
- *routes/index.js*
- *views/index.ejs*

# Resources

<http://www.tomdalling.com/blog/software-design/model-view-controller-explained/>

1. form teams
2. turn in paper with team member names,  
brainstorm ideas