

Introduction to Express

Agenda

- What is Express?
- Installing Express
- MVC Pattern
- Template Engines
- Express App Generator

What is Express?

- A minimal and flexible Node.js web application framework
 - Provides robust features for web and mobile apps
 - Helps in creating APIs quickly and easily
 - Lightweight and fast
 - Extensible
 - You can use Express middleware to add support for cookies, sessions, and users, getting POST/GET parameters, etc.
 - Underlying library for a number of other popular Node.js frameworks
 - Sails, MEAN, LoopBack, etc.

Installing Express

- Ensure that Node.js is already installed on your machine
 - Execute 'node -v' command in the Command Prompt or Terminal window

```
E:\>node -v  
v9.9.0  
  
E:\>
```

- If Node.js is present, the command executes successfully and returns the version number of Node.js

Installing Express

- Create a directory to hold your app. Make that your working directory
- Use the 'npm init' command to create a package.json file for your app
- Install Express in my-app directory and save it in the dependencies list

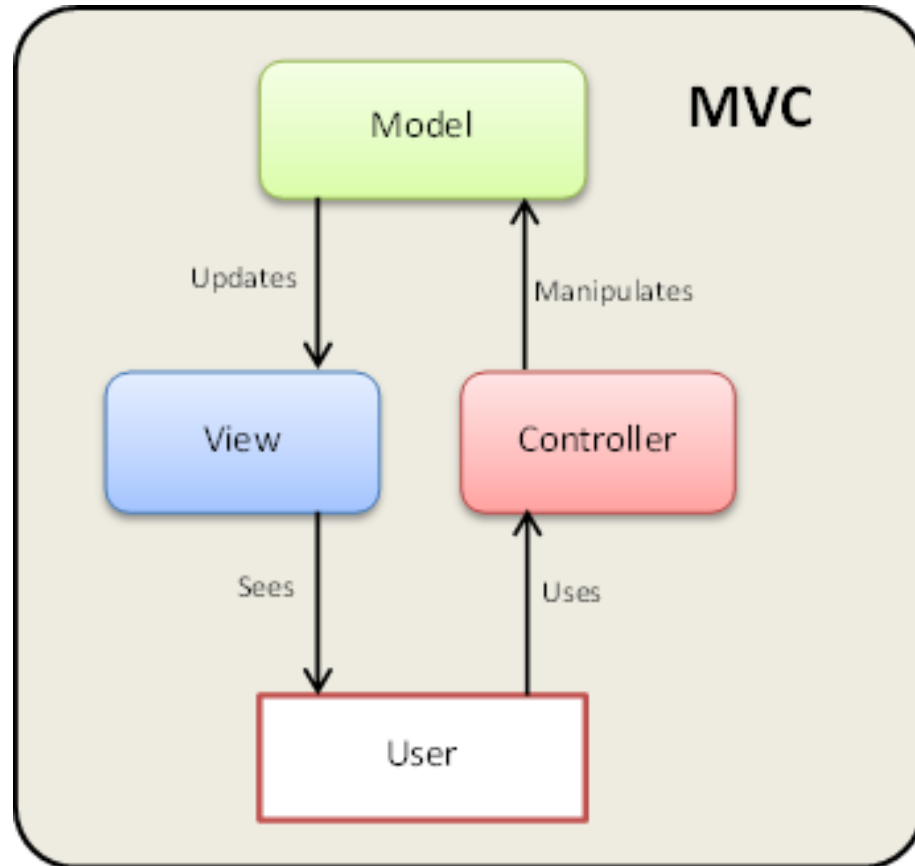
```
> mkdir my-app  
> cd my-app  
> npm init  
> npm install --save express
```

MVC Pattern

- Model-View-Controller (MVC)
 - Is an architectural pattern
 - Enables isolation of domain logic from the user interface
 - Facilitates independent development, testing and maintenance of the different parts of software apps

MVC Pattern

- MVC helps us in dividing our app into 3 parts
 - Model
 - View
 - Controller



MVC Pattern

- Model
 - Manages the behavior and data of the application domain
 - Responds to requests (from the view) for information about its state
 - Carries out the instructions for change in state, issued by the controller

MVC Pattern

- View
 - Concerned with presenting the information (from the model) to users in a UI element
 - Facilitates user interaction
 - Represents one representation of the model state
 - Multiple views can exist for a single model for different purposes

MVC Pattern

- Controller
 - Receives user input
 - Converts it to commands for the model or view
 - Can send commands to the model to update model's state
 - Can send commands to its associated view to change the view's presentation of the model

Template Engines

- A template engine
 - Enables you to use static template files in your application
 - Replaces variables in a template file with actual values at runtime
 - Transforms the template into an HTML file sent to the client
 - Popular template engines: Pug, Mustache, EJS
 - For a list of template engines, visit
 - <https://github.com/expressjs/express/wiki#template-engines>

Template Engines

- To use a template engine, follow these steps
 - The application should set the template engine properties using this code

```
// use the following application setting properties  
app.set('views', './views');  
app.set('view engine', 'pug');
```

- views – specifies a directory or an array of directories where the template files are located
 - view engine – specifies the template engine to be used
- Install the corresponding template engine

```
npm install --save pug
```

Template Engines

- Create a template (index.pug) and place it in 'views' directory
- Create a route to render the 'index.pug' file

```
app.get('/', function (req, res) {  
  res.render('index', { title: 'Hey', message: 'Hello there!' });  
})
```

- When you make a request to the home page, the index.pug file will be rendered as HTML.

Express App Generator

- An application generator tool
- Used to quickly create an Express application skeleton
- Setup Steps
 - Install express-generator
 - Create an express app
 - Install dependencies
 - Run the app

```
> npm install express-generator -g  
> express myapp  
> cd myapp  
> npm install  
> npm start
```



THANK YOU