CSCI 4140 – Tutorial 11

45-minute tour on Backbone.js

Matt YIU, Man Tung (mtyiu@cse)
SHB 118
Office Hour: Wednesday, 3-5 pm

2016.04.14

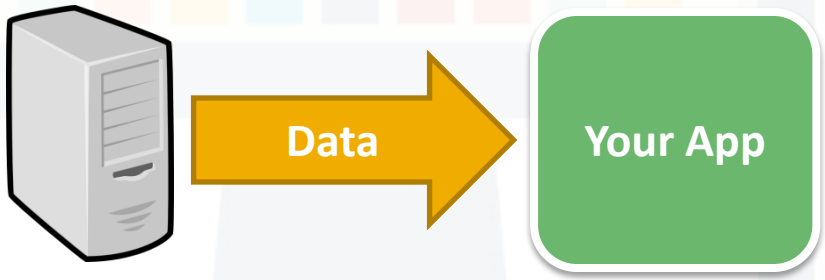
Outline

- Introduction
  - MVC (Model-View-Controller)
  - Backbone.js
- Develop a To-do list in Backbone.js
- Create the back-end – RESTful APIs
- Connect the front-end and back-end

Note: *The content of this tutorial is mostly adapted from* https://addyosmani.com/backbone-fundamentals/

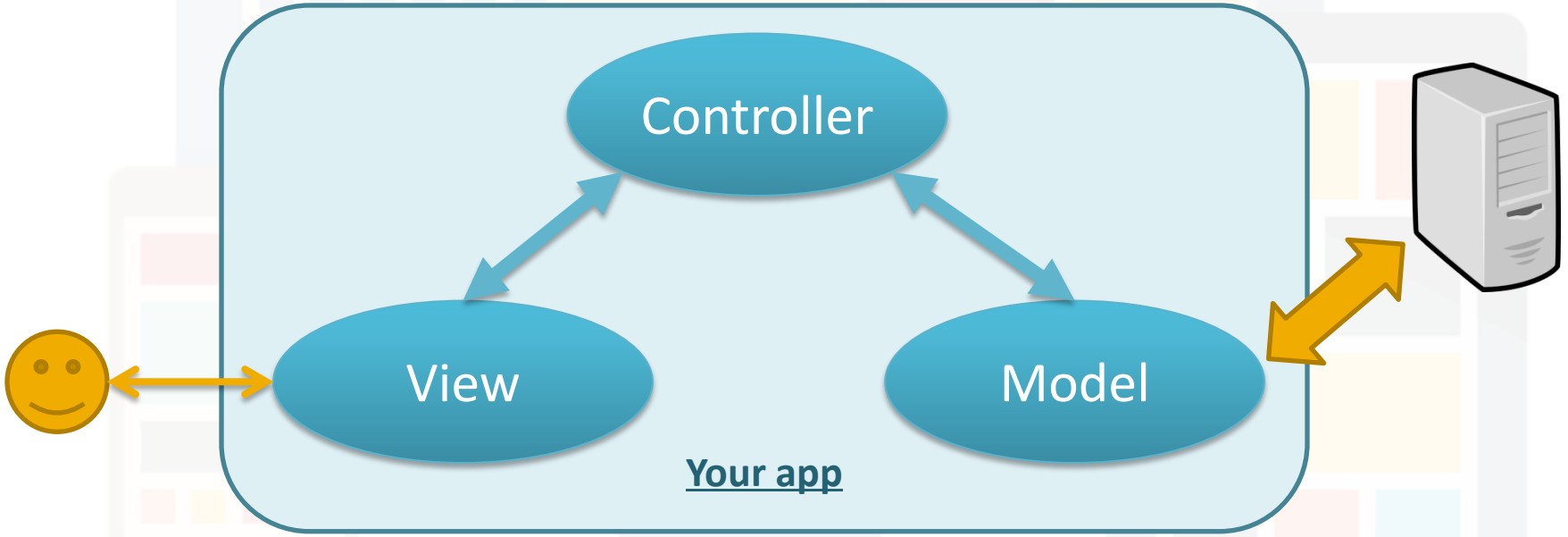
Introduction

- Suppose you are writing a web app which keeps downloading the latest data from server (e.g., by **polling**, using WebSocket)
- How would you **update** the page?
- How to handle **user interactions**?
- How to **structure** and **organize** your code?
- **Answer:** Use a **front-end MVC framework**!



Introduction – MVC

- MVC = Model-View-Controller
  - **Model:** Represent the domain-specific knowledge and data
  - **View:** Constitute the user interface
  - **Controller:** Handle inputs and update models



Introduction – MVC

- MVC = Model-View-Controller
  - **Model:** Represent the domain-specific knowledge and data
  - **View:** Constitute the user interface
  - **Controller:** Handle inputs and update models
- User input is acted upon by Controllers; controllers update models
- Views observe Models and update the user interface when changes occur
- Yet, many frameworks do not strictly follow the above pattern
  - E.g., Backbone.js combines the Controller with View

Introduction – Backbone.js

- Backbone.js is a lightweight JavaScript library that structures your client-side code in MVC architecture
- Useful for creating **single-page applications** (SPAs)
  - **Reasons of SPAs:** Reduce number of page loads → reduce latencies → improve user experience!
- Provide a minimal set of **data structuring** (Models, Collections) and **user interface** (Views, URLs) primitives
- **Dependency:** Underscore.js

Introduction – Backbone.js

- Read the following sections (as I cannot write better than this book…):
  - https://addyosmani.com/backbone-fundamentals/#client-side-mvc---backbone-style
  - https://addyosmani.com/backbone-fundamentals/#implementation-specifics

Develop a To-do list in Backbone.js

- I developed a To-do list application in Backbone.js
  - Let's learn the basics of Backbone.js by reading the code
- Guided code reading (client side): https://github.com/mtyiu/backbone-todo-list/tree/local
  - `views/index.html`
  - `public/js/*`
- How to send/receive data with the server and update the UI?
  - Here comes the power of Backbone.js
  - Backbone.js works seamlessly with **RESTful APIs**!

Create the back-end – RESTful APIs

- REST = **RE**presentational **S**tate **T**ransfer
  - More details: https://en.wikipedia.org/wiki/Representational_state_transfer and http://www.restapitutorial.com/index.html
- RESTful systems (systems that conform to the constraints of REST) typically communicate over HTTP with the same **HTTP verbs** (GET, POST, PUT, DELETE, etc.)
  - **POST**: **Create** a new record to the server
  - **GET**: **Read** one or more records from the server
  - **PUT**: **Update** an existing record in the server
  - **DELETE**: **Delete** an existing record in the server

Create the back-end – RESTful APIs

- Now let's implement the following API for our To-do list:

URL	HTTP Method	Operation
`/api/items`	`GET`	Get an array of all items
`/api/items/:id`	`GET`	Get the item with ID of `:id`
`/api/items`	`POST`	Add a new item and return the item with an `id` attribute added
`/api/items/:id`	`PUT`	Update the item with ID of `:id`
`/api/items/:id`	`DELETE`	Delete the item with ID of `:id`

- Guided code reading (server side)
  - https://github.com/mtyiu/backbone-todo-list/blob/master/routes/api.js

Connect the front-end and back-end

- Now we are ready to talk to the server using the RESTful API
- Guided code reading (client side):
  - `public/js/app.js`
  - `public/js/collections/list.js`
  - `public/js/models/item.js`
  - `public/js/views/item.js`
  - `public/js/views/list.js`
- Oops…where are the AJAX calls?!
  - Magic behind: **Backbone's Sync API**
  - https://addyosmani.com/backbone-fundamentals/#backbones-sync-api

Example code

- Both the client-side and server-side implementation are available at: https://github.com/mtyiu/backbone-todo-list
  - The `local` branch is the version without RESTful API support
- Live version available at: http://backbone-todo-4140.herokuapp.com/

- Read the book to know more details about Backbone.js (it is impossible to cover everything in one tutorial)

– End –

Hope you enjoyed the tutorials!

Good luck for your project and final examination!

All tutorial recordings are available on the tutorial page