Telerik Academy

JavaScript Patterns and Single-page Applications

Telerik Software Academy
Learning & Development Team
http://academy.telerik.com

Telerik Academy

Table of Contents

- Single-page applications
  - What is a SPA app?
  - Building SPA apps with Sammy.js and mustache.js
- Architectural Design Patterns
  - MV* patterns - MVC, MVVM and MVP
  - Sample implementation in JavaScript

Telerik Academy

Single-page Applications

What and Why?

Telerik Academy

Single Page Application

- A single-page application (SPA) is a web application that fits on a single web page
  - Page resources are retrieved either at page load or loaded in response to user actions
  - The page does not reload at any point
- SPA apps can contain multiple fake pages
  - Not real pages, but ones that look like a page

Telerik Academy

Architecture of SPA Apps

- SPA applications have a different architecture than regular JavaScript applications
- Commonly SPA apps have:
  - A thick layer of server logic (Web services), located in the cloud
    - Containing both Database and Business logic
  - A thin client layer, implemented with HTML5 and JavaScript
    - Containing communication with the server and UI logic

Telerik Academy

SPA Applications: Pros and Cons

Telerik Academy

SPA Applications: Pros and Cons

- SPA applications have:
  - Better performance
    - There is logic on both the server and the client
  - Lower bandwidth
    - Only JSON/XML data is send over HTTP
    - The JSON data is rendered on the client
  - Higher code reusability
    - The server gives an API, that is fully decoupled with the UI

Telerik Academy

SPA Applications: Pros and Cons (2)

- Yet SPA applications have their challenges:
  - The server must have a solid data validation
    - The client is more easily tampered with
  - Routing (i.e. History)
    - The regular user expects to go to the previous screen, not page, when they hit the back button
    - Easily done with Sammy.js, AngularJS, etc…
  - A combination of lots of frameworks are required
    - jQuery is not enough any more…

Telerik Academy

Design Patterns in SPA Application

Telerik Academy

SPA Applications Architecture

- Most SPA applications relay on solid JavaScript business logic
  - They have a server that contains the database and exposes web services
    - Web services are kind of requirement for SPA
- The JavaScript communicates with the server using HTTP requests and AJAX calls

Telerik Academy

SPA Architecture

- A SPA application is build from some layers to meet the "separation of concerns"
  - UI layer
    - Contains Views (HTML, CSS and UI logic)
  - Data layer
    - A way to communicate with the server (mainly HTTP requests)
  - Business layer
    - The layer that connects the Data with the UI
    - Contains controllers or view-models

Telerik Academy

Architectural Patterns in JavaScript

- Many design patterns exist that solve the separation of concerns
- The most used patterns are the MV* patterns
  - Model-View-*
  - The * contains the business logic of the app
- MV* has three concrete implementations:
  - MVC (Model-View-Controller)
  - MVVM (Model-View-ViewModel)
  - MVP (Model-View-Presenter)

Telerik Academy

Model-View-Controller

The MVC Pattern

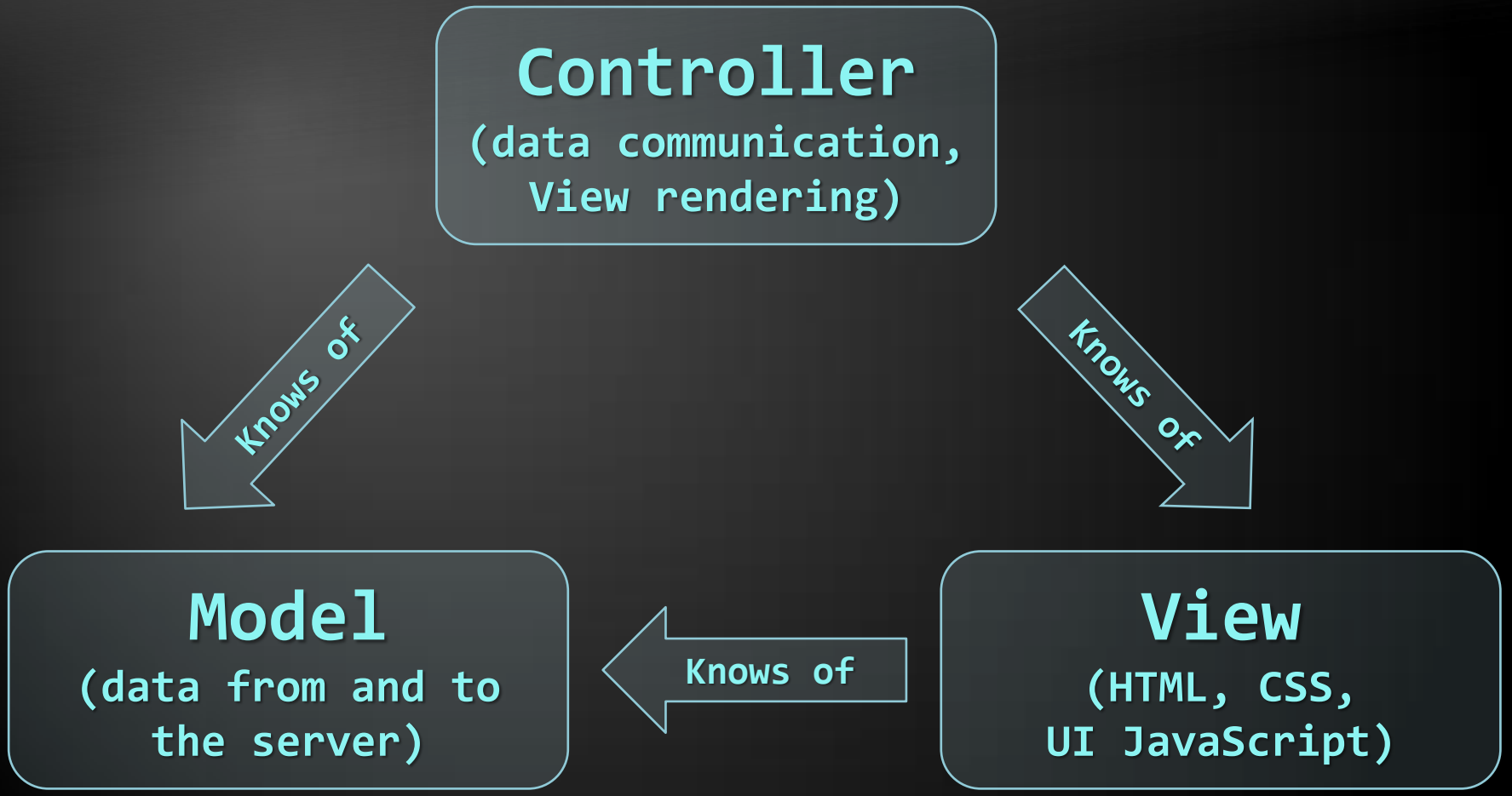
Telerik Academy

Model-View-Controller

- The Model-View-Controller has three sublayers:
  - Model
    - Contains data models (JSON or POCO)
  - View
    - Contains UI logic (HTML,CSS, UI JS)
  - Controller
    - Contains business logic
    - Fills models with data and passes them to a View
    - Plays the role of the middleman

Telerik Academy

MVC Architecture



Telerik Academy

JavaScript Frameworks Implementing MVC

- Most of the JavaScript frameworks implement the MVC pattern:
  - AngularJS
  - Backbone.js
  - Ember.js
  - Sammy.js
  - Google Closure
  - Batman.js

Telerik Academy

Model-View-ViewModel

The MVVM Architectural Design Pattern

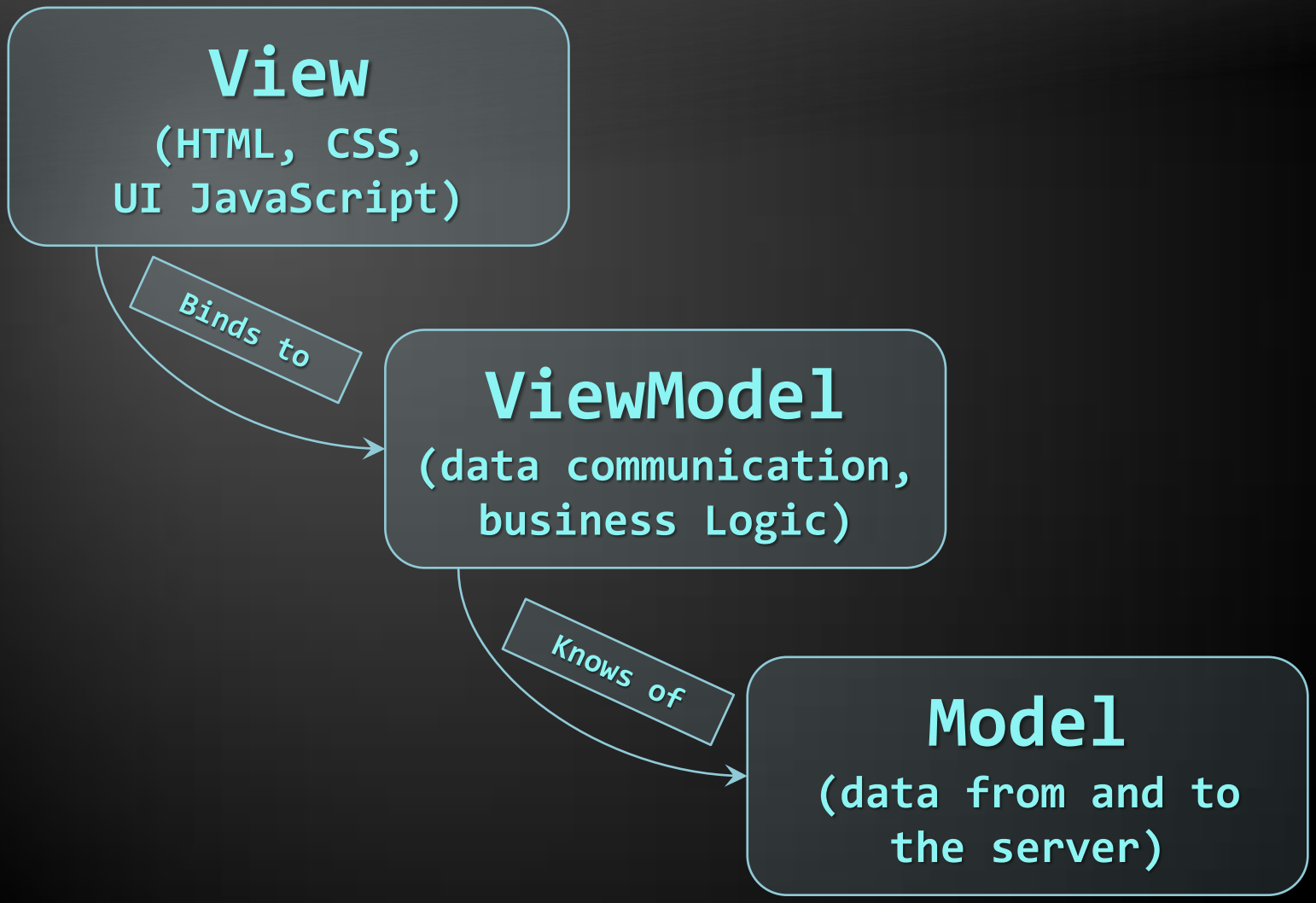
Telerik Academy

Model-View-ViewModel

- Model-View-ViewModel has three sublayers:
  - Model
    - Contains data models (JSON or POCO)
  - View
    - Contains UI logic (HTML,CSS, UI JS)
  - ViewModel
    - Model of the View - The View binds to the VM
    - Contains business logic
    - Keeps models of data, Views get what they need
    - Plays the role of the middleman

Telerik Academy

MVVM Architecture



Telerik Academy

JavaScript Frameworks Implementing MVVM

- MVVM was designed mostly for use in WPF/Silverlight, but its usable in JavaScript as well
  - Kendo.UI
  - Kendo UI Mobile
  - Knockout.js
  - Knockback.js

Telerik Academy

JavaScript Patterns and SPA

http://academy.telerik.com

Homework

- Create a SPA application "Crowd Chat"
  - The application must use the following web services:
    - GET http://crowd-chat.herokuapp.com/posts
      - Return all posts
    - POST http://crowd-chat.herokuapp.com/posts
      - Body: { "user": "USER_NAME",
        "text": "MESSAGE_TEXT"}
      - Sends a new post