Web Design and Development

Week 3

More ASP.NET MVC

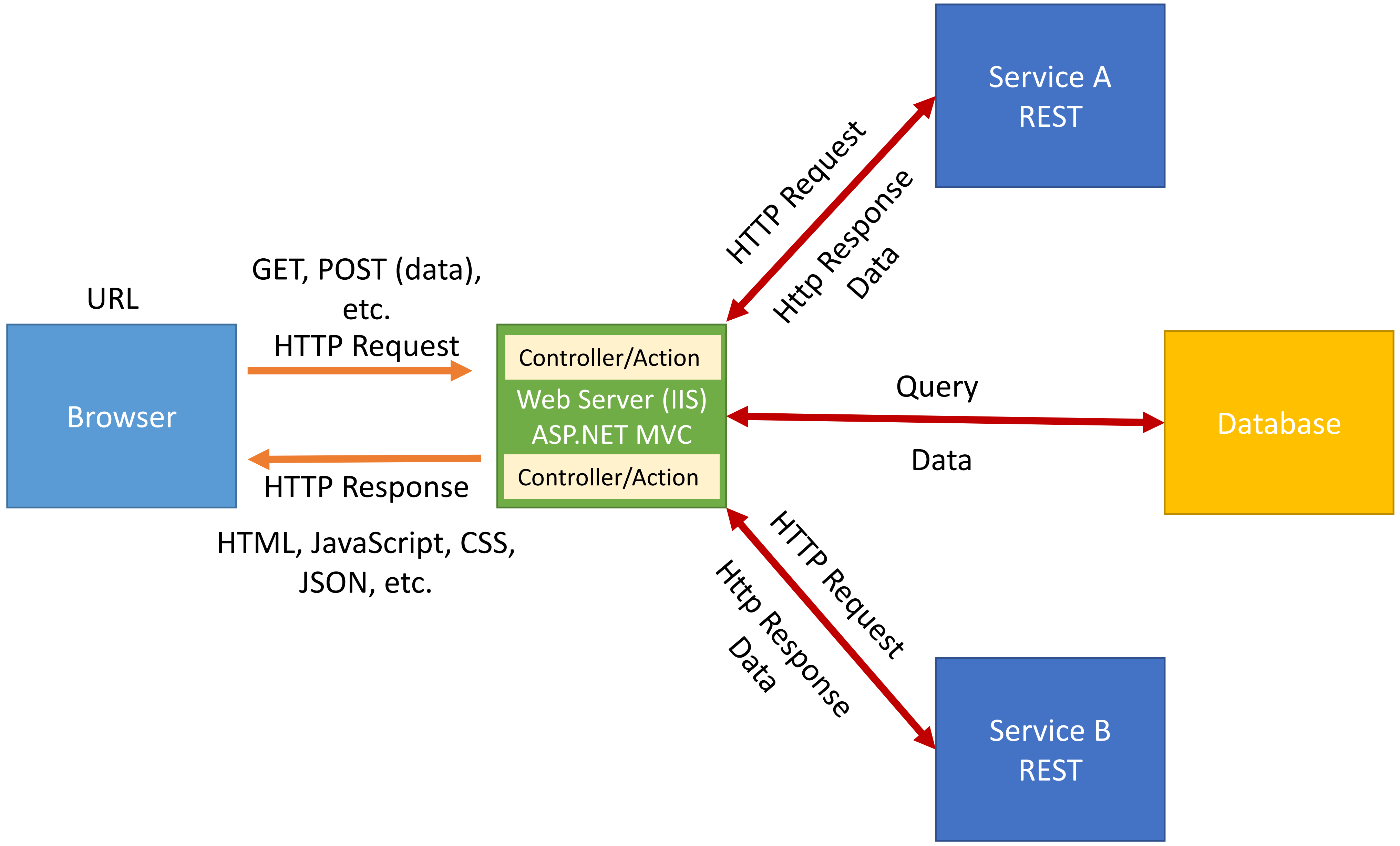
Questions?

ASP.NET MVC
Controllers/Actions: Review

- Provide an HTTP interface for clients (browsers)
- Requests sent over HTTP from a client are routed to a controller/action
- Actions are class methods that implement an interface to application logic
- Process requests, including data they receive from the client
- Create and populate models using data they receive from databases and/or services (REST)
- Returns response (HTML, JSON, file, etc.) back to client
- Named [Entity]**Controller** and **derives from Controller**

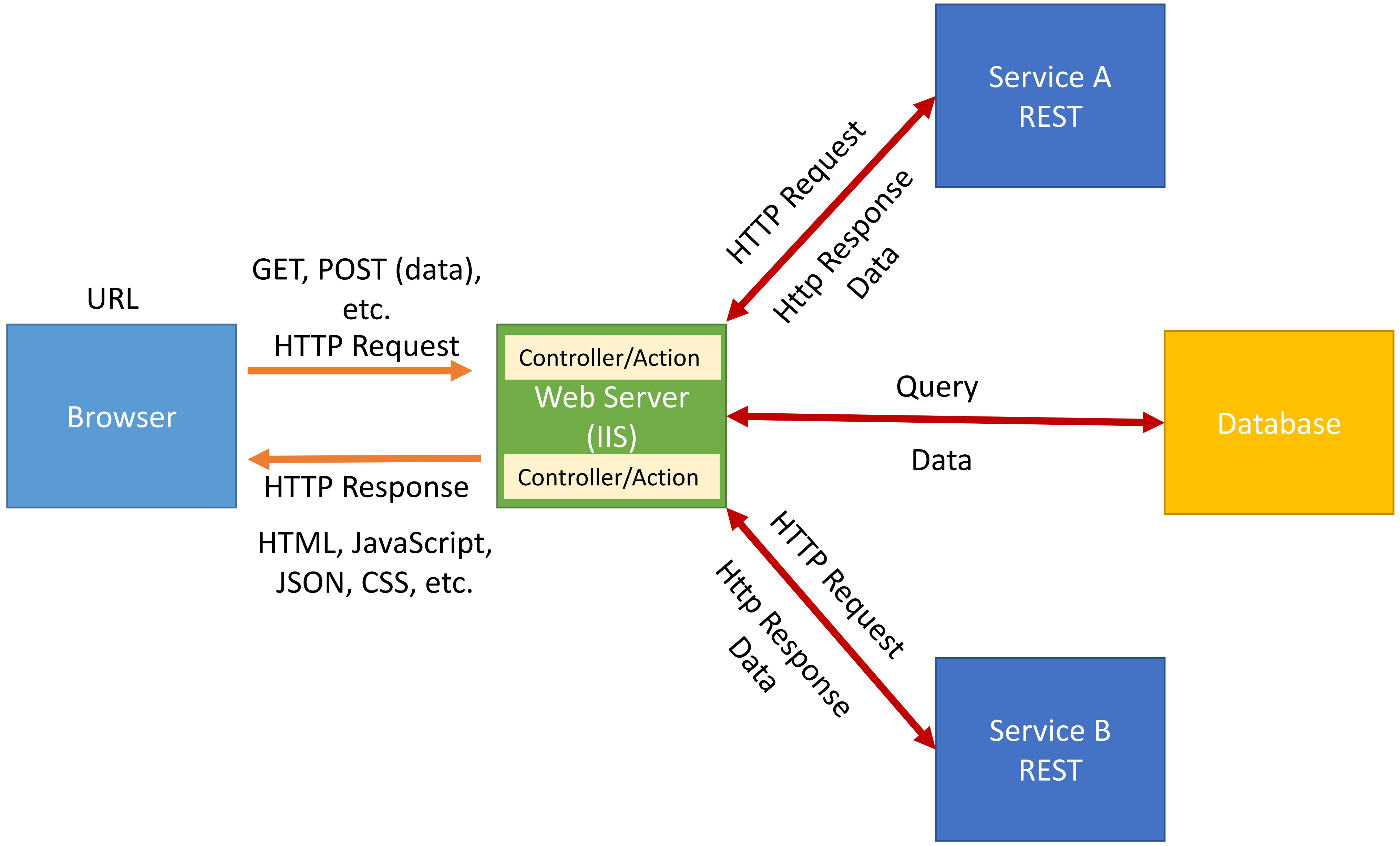
ASP.NET
Controllers: Routes

- Control what controller action a URL is “mapped” to
- Can also specify data mapped from values that are part of the URL
- Routes are configured by the developer either globally or per controller action
- Example URL: http://www.mysite.com/person/details/123


URL
Browser
GET, POST (data), etc.
HTTP Request
HTTP Response
HTML, JavaScript, CSS, JSON, etc.
Controller/Action
Web Server (IIS)
ASP.NET MVC
Controller/Action
HTTP Request
Http Response
Data
Service A
REST
Query
Data
Database
HTTP Request
Http Response
Data
Service B
REST

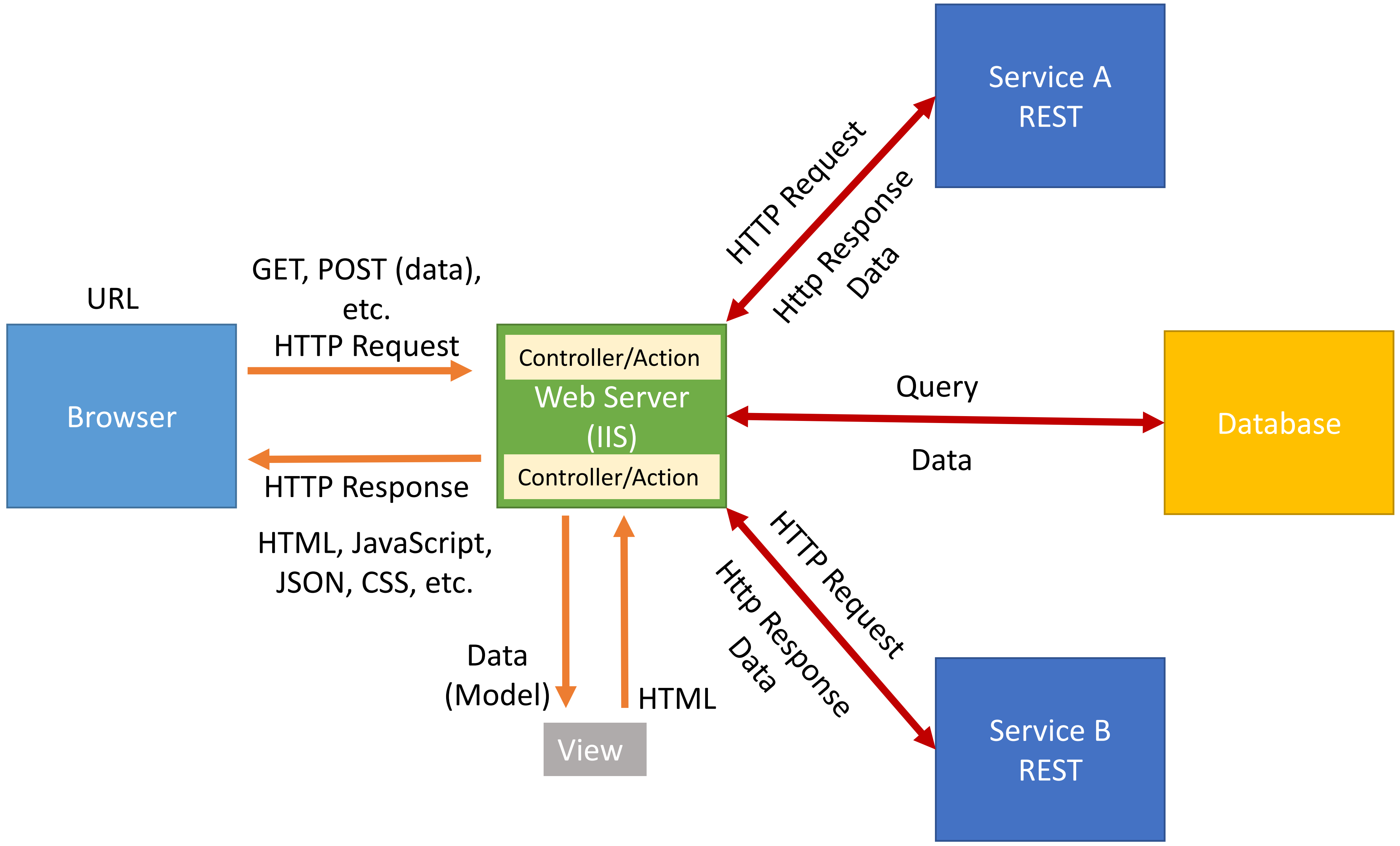
ASP.NET MVC
Models: Review

- In memory representation (model) of application entities:
  - student, patient, friend, pet, etc.
  - login attempt, temperature reading, etc.
- Content of model are in the form of class properties comprised of data required by your application (Name, Date/Time, etc.)
- Generally only contain data (no methods)
- Populated by controllers from data controllers receive from users (via HTTP request), databases and/or services (REST)


URL
Browser
GET, POST (data), etc.
HTTP Request
HTTP Response
HTML, JavaScript, JSON, CSS, etc.
Controller/Action
Web Server (IIS)
Controller/Action
HTTP Request
Http Response Data
Service A REST
Query
Data
Database
HTTP Request
Http Response Data
Service B REST

ASP.NET MVC
Views: Review

- Templates
- Contain mostly HTML
- Contain display (no application) logic and placeholders for data
- Page contents (HTML, CSS, JavaScript) that are sent back to client (rendered by browser)
- Invoked by a controller through MVC convention
- Any data to be displayed is in a defined model (view model) passed to the view
- Data is inserted into placeholders: @Model.Name


URL
Browser
GET, POST (data), etc.
HTTP Request
HTTP Response
HTML, JavaScript, JSON, CSS, etc.
Controller/Action
Web Server (IIS)
Controller/Action
Data (Model)
HTML
View
HTTP Request
Http Response Data
Service A REST
Query
Data
Database
HTTP Request
Http Response Data
Service B REST

ASP.NET MVC
Controllers: Action Parameters

- How the controller receives data from the client (browser)
- Actions called by the MVC framework
- Passes parameters to action methods in the form of either single values or objects
- Object parameters are generally defined as C# classes with public properties

ASP.NET MVC

Controllers: Model Binding

- MVC uses **model binding** to **bind** the data in the HTTP request to parameter values
- HTTP request data
  - URL (http://hostname/app/student/100)
  - Query string (http://hostname/app/student?id=100 (not generally used)
  - Form data

ASP.NET MVC
Models: Validation

- MVC can automatically validate data received
- Requires model class to be decorated with validation attributes

ASP.NET MVC
Views: Layouts

- Contains content common to all or a subset of views
    - Headers
    - Footers
    - Navigation bars (top, side, etc.)
    - JavaScript references
    - CSS references
- Has .cshtml as extension
    - Layout.cshtml
    - AdminLayout.cshtml
- Located in the Views/Shared folder by convention
- Often defined in the file _ViewStart.cshtml located in Views folder

ASP.NET MVC
Views: Razor Syntax

- Server side code (C#)
- Provides a way to incorporate logic into views
- Should only be used for view altering logic and **not** application logic
- Calculated (in controller) values should be encapsulated in the model passed to the view

ASP.NET MVC
Views: Razor Syntax (continued)

- HTML Helpers: MVC functions that generate HTML snippets
- Access properties and methods on the model defined
  - Uses a C# feature called lambda expressions (=>)
- Go through the Razor tutorial (see link under Course Materials/Resources/ASP.NET MVC)

ASP.NET MVC
Views: Validation

- Associate a validation message with individual form fields
- Validations are associated via **DataAnnotations** (attributes on model properties)
- Can have a validation summary section
- Validation messages are generated automatically by the MVC framework using JavaScript
- Validation should be done on both client and server side

ASP.NET MVC
Models: Storing (persisting)

- Relational database is typically used for storage
- Common to use an ORM
  - Object Relational Mapping
  - Maps relational database rows/fields to/from model objects/properties
  - Allows developer to work with objects rather than relational data
  - Popular .NET ORMs are **Microsoft Entity Framework** and NHibernate
- Data storage is typically done in a class separate from a controller action
  - Single responsibility principal (SOLID principals)
  - Separation of concerns
  - Testability

ASP.NET MVC
Demo

- Model creation
- Model binding
- Building a view with an input form using Razor syntax
- Input validation
- HTTP methods (Get and POST)
- Using Fiddler to view HTTP request