ASP.NET MVC 5

Christen Zarif Foad

Course Prerequisites

- **Before attending this course, students must have**:
  - Experience with Visual Studio 2019.
  - Experience with C# programming, and concepts such as Lambda expressions, LINQ, and anonymous types.
  - Experience in using the .NET Framework.
  - Experience with HTML, CSS and JavaScript.
  - Experience with querying and manipulating data with ADO.NET.
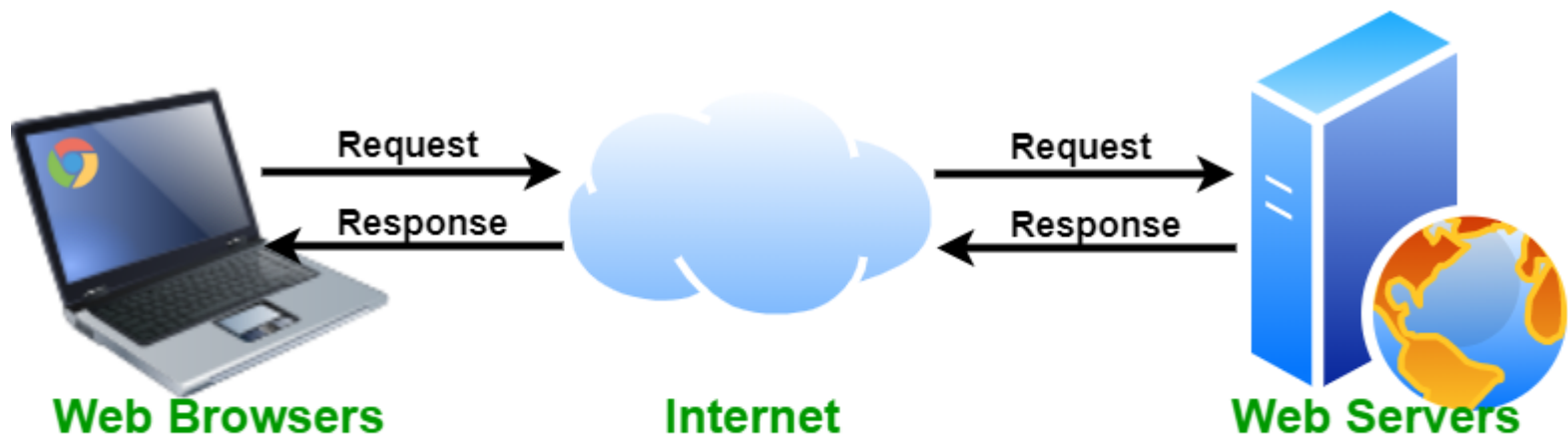 - Knowledge of XML and JSON data structures.

1. Different Between Desktop Application , Web Application ?

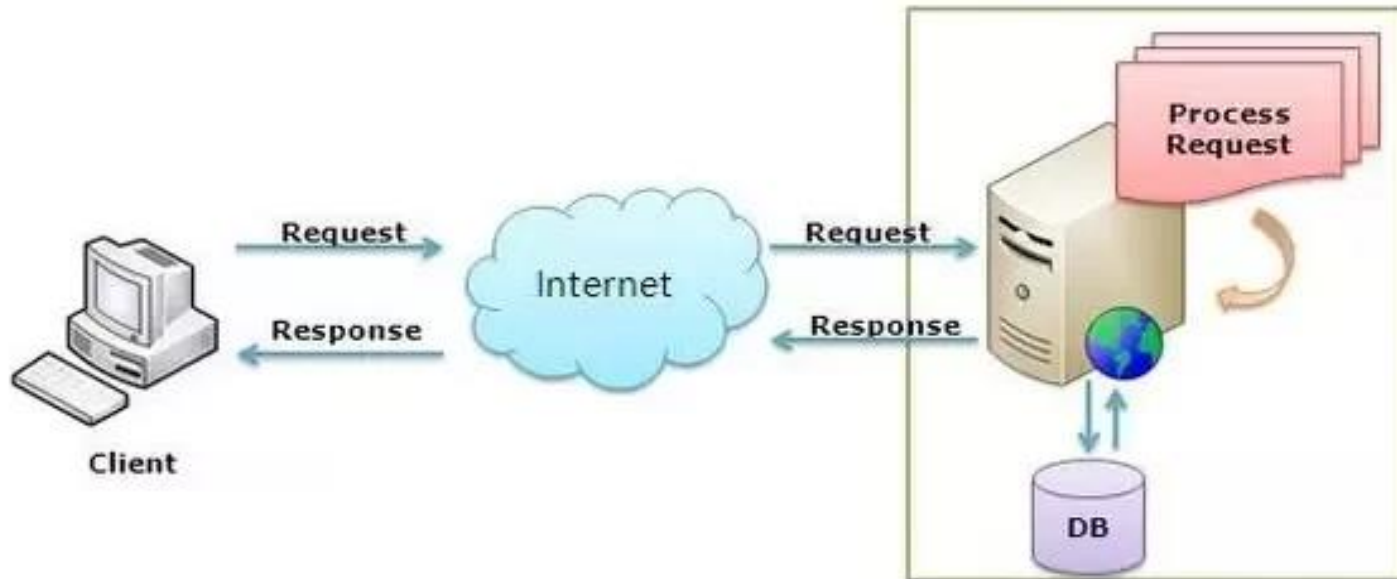2. Different Between Web Server , Web Application ?

Web Server

- A web server is software and hardware that uses **HTTP** (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web.

- The main job of a web server is to display website content through storing, processing and delivering webpages to users.

IIS

- ***IIS (Internet Information Server)*** is one of the most powerful web servers from Microsoft that is used to host your Web application.

- **IIS** has it's own Process Engine to handle the request. So, when a request comes from client to server, IIS takes that request and process it and send response back to clients.
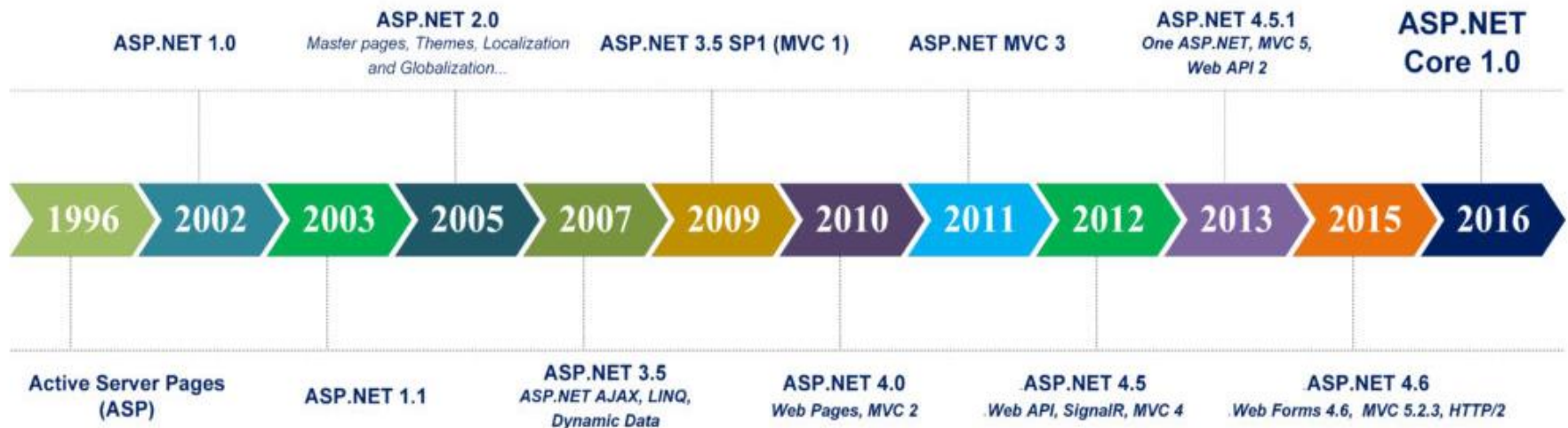
ASP.NET

One ASP.Net

- ASP.NET is an open source **web framework**, created by Microsoft, for building modern web apps and services with .NET

- ASP.NET extends the .NET platform with tools and libraries specifically for building web apps.

- https://dotnet.microsoft.com/learn/aspnet

Asp.net History

ASP.NET 1.0

ASP.NET 2.0
Master pages, Themes, Localization
and Globalization...

ASP.NET 3.5 SP1 (MVC 1)

ASP.NET MVC 3

ASP.NET 4.5.1
One ASP.NET, MVC 5,
Web API 2

ASP.NET
Core 1.0

1996 2002 2003 2005 2007 2009 2010 2011 2012 2013 2015 2016

Active Server Pages
(ASP)

ASP.NET 1.1

ASP.NET 3.5
ASP.NET AJAX, LINQ,
Dynamic Data

ASP.NET 4.0
Web Pages, MVC 2

ASP.NET 4.5
Web API, SignalR, MVC 4

ASP.NET 4.6
Web Forms 4.6, MVC 5.2.3, HTTP/2

Introduction to MVC Design Patterns?

- A <u>design pattern</u> is a formal way of documenting a solution to a design problem in a particular field of expertise. Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to it.

- The Model-View-Controller design pattern solves the interdependencies between <u>data access code, business logic code, and presentation code</u> by decoupling them. High coupling can cause strong ripple effects whenever a change is made anywhere
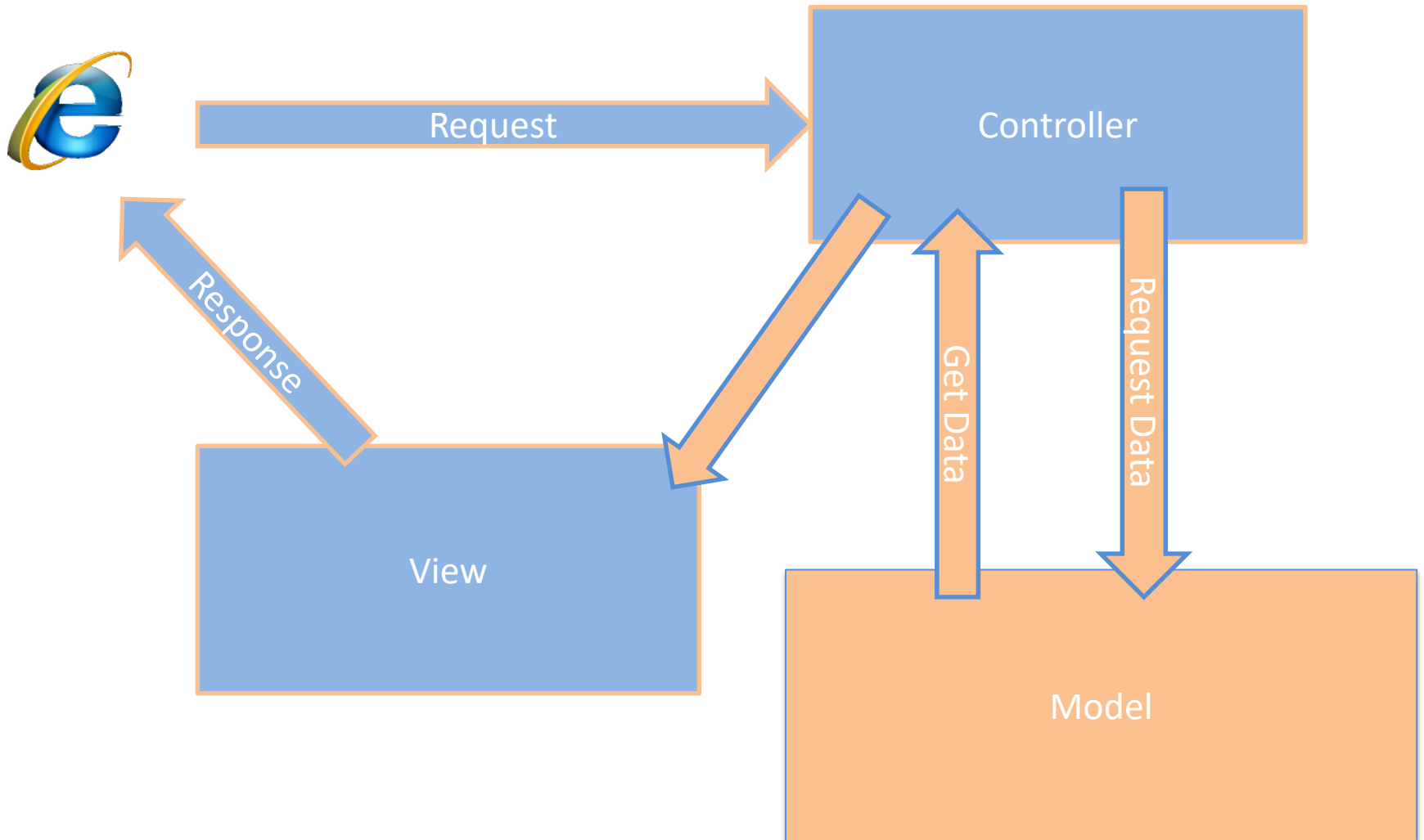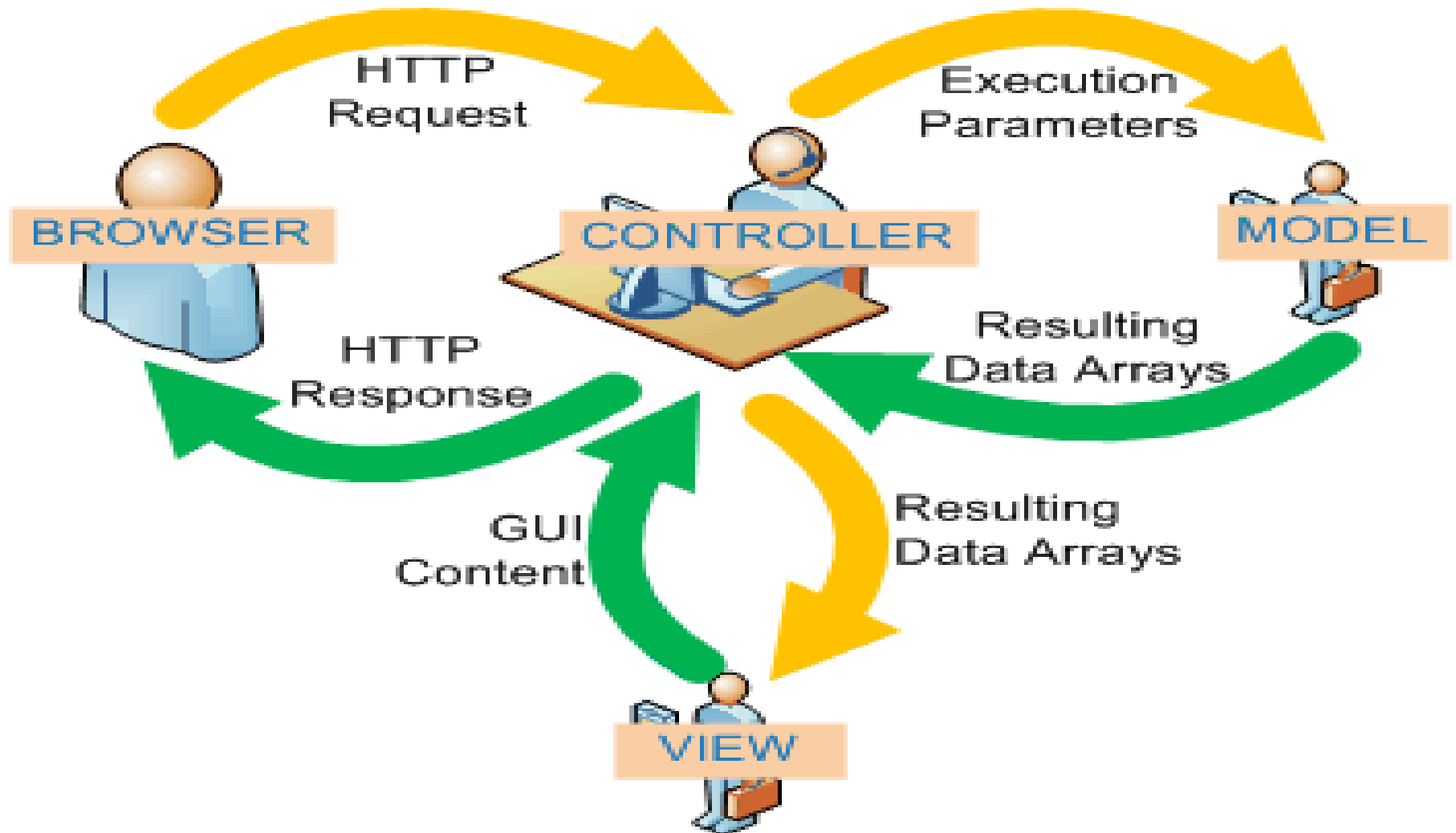
Asp.net *MVC* Framework

- Is a web application framework that **implements the model-view-controller pattern.**

- An ***alternative*** to ASP .NET Web Forms
- Highly testable
- Integrated with the existing ASP .NET features:
    - Master pages
    - Membership-Based
    - Authentication

- defined in the ***Microsoft.AspNetCore.Mvc*** namespace.

Features of ASP.NET MVC Framework

- *Separation* of application logic which makes testing easier

- *Powerful URL-mapping / Clean URL :*component that lets you build applications that have comprehensible and searchable URLs. **URLs do not have to include file-name extensions** MVC provides different routing techniques that do not depend on file systems but Controllers. These URLs are easy to perform SEO..

- Using *Razor* as a view Engine

MVC Architecture

Request → Controller

Response

View

Get Data

Request Data

Model

MVC App Execution 'Stages'

- Receive first request for the application
  - Populating RouteTable
- Perform routing
- Create MVC Request handler
- Create controller
- Execute controller
- Invoke action
- Execute result
  - ViewResult, RedirectToRouteResult, ContentResult, FileResult, JsonResult, RedirectResult

URL Mapping

- http://host:port/Student/find/100

```
0 references
public class StudentController : Controller
{
    0 references
    public IActionResult Find(int id)
    {
        return View();
    }
}
```

Recursos ASP .NET MVC

- Demo

http://nerddinner.codeplex.com/

- Video introductorio

http://channel9.msdn.com/posts/matthijs/ASPNET-MVC-2-Basics-Introduction-by-Scott-Hanselman/

Different developers and different projects have different needs. If ASP.NET MVC meets your needs, use it. If it doesn't, don't use it. Either way, don't be afraid

Thank You

- Microsoft Tutorial:
  - https://dotnet.microsoft.com/apps/aspnet
  - https://docs.microsoft.com/en-us/aspnet/mvc/
  - https://www.asp.net/mvc
- Introduction into mvc
  - https://app.pluralsight.com/player?author=scott-allen&name=mvc3-building-intro&mode=live&clip=0&course=aspdotnet-mvc3-intro
- Advanced Topics
  - https://app.pluralsight.com/player?author=scott-allen&name=aspdotnet-mvc5-fundamentals-m1-introduction&mode=live&clip=0&course=aspdotnet-mvc5-fundamentals