
iscte
INSTITUTO
UNIVERSITÁRIO
DE LISBOA
emprego
digital
UPskill

# Contents

## What will this module be about?

- What is the internet? How does it work?
  - HTTP
  - WWW
- A brief history of Microsoft's Web Development

# A brief history of Microsoft's Web Development

# A brief history of Web Development

| Period | Technology | Strengths | Weaknesses |
|---|---|---|---|
| Jurassic | Common Gateway Interface | Simple<br>Flexible | Runs outside the web server, so is resource-intensive |
| 1996 | Active Server Pages (ASP) | General Purpose | Interpreted at runtime<br>Encourages “spaghetti code” |
| 2002/2003 | ASP.NET Web Forms 1.0 | Compiled<br>Stateful UI<br>Encourages OOP | Heavy on bandwidth<br>Ugly HTML<br>Untestable |
| 2007 | ASP.NET AJAX | | |
| 2009 | ASP.NET MVC 1.0 | | |

# What is wrong with ASP.NET Web Forms

- Over the years, Microsoft's web development platforms have demonstrated increasing power and, unfortunately, increasing complexity

- Microsoft attempted to hide both HTTP (with its intrinsic statelessness) and HTML (which was unfamiliar to many developers)

- The idea was to make web development feel just the same as Windows Forms development

# What is wrong with ASP.NET Web Forms

- View State weight
  - To maintain state across requests, hundreds of Kb go back and forth each request
- Page life cycle
  - The mechanism for connecting client-side events with server-side event handler code can be complicated and delicate. Few developers manage to do it successfully
- False sense of separation of concerns
  - Developers are encouraged to mix presentation code with their application logic in code-behind classes. The result can be fragile and unintelligible

# What is wrong with ASP.NET Web Forms

- Limited control over HTML
  - Server controls render themselves as HTML, but not necessarily the HTML you want
- Leaky abstraction
- Low testability
  - THe tightly coupled architecture is unsuitable for unit testing. Integration testing can be a challenge too.

# Web development outside Microsoft

- The drive for web standards compliance has increased in recent years
- HTML5 entered mainstream use and provides Web developers with rich capabilities that allow the client to perform work that was previously server's responsibility
- REST has become dominant, overshadowing SOAP
- Software development shifted towards *agile*
- Test-driven development (TDD) and behaviour-driven development (BDD) promoted and assisted this transition

# Web development outside Microsoft

- Ruby on Rails
  - Not revolutionary technology, but the concept was interesting
  - Embraces MVC architecture, works in tune with HTTP by promoting conventions instead of configurations
  - Shows that web standards compliance and RESTfulness don’t need to be hard

- Node.js
  - Javascript as primary programming language
  - Doesn’t apply MVC pattern
  - Low-level way of connecting HTTP requests to your code

# What is a ASP.NET Core?

- Open-source and cross-platform .NET framework for building modern cloud-based web applications on Windows, Mac, or Linux.
- With ASP.NET Core, you can:
  - Build web apps and services, IoT apps, and mobile backends.
  - Use your favorite development tools on Windows, macOS, and Linux.
  - Deploy to the cloud or on-premises.
  - Run on .NET Core or .NET Framework.

# Key Benefits of ASP.NET MVC

- Implements MVC design pattern
  - User interaction follows a natural cycle: the user takes an action, and in response the application changes its data model and delivers an updated view to the user
  - The patterns of combining several web technologies, usually split in tiers or layers, map naturally in MVC
- Extensibility
  - Use default behavior, extend default with a subclass implementation or completely replace component
- Tight control over HTML and HTTP

# Key Benefits of ASP.NET MVC

- Testability
- Powerful Routing System
- Modern API
- Open Source
  - https://github.com/dotnet/aspnetcore

# Who should use ASP.NET MVC?

- Depends on:
  - Team Skills
  - Effort to port any existing projects
  - Confidence in technology source

- [ASP.NET Web Forms] Writing applications for internet or larger intranet applications - better bandwidth efficiencies, browser compatibility and support for automated testing

# Who should use ASP.NET MVC?

- [Ruby on Rails] Focuses on handling Web requests in an MVC-pattern with controllers and actions. Does not have built-in ORM tool, automated testing tool or system for managing database migrations

# What is a Web API?

- *Programmatic* interface to a system that is accessed via standard HTTP methods and headers
- Can be accessed from a variety of HTTP Clients - including browsers and mobile devices

# SOAP Web Services

- Simple Object Access Protocol
- Lightweight protocol for exchange of information in a decentralized, distributed environment
- XML based protocol that consists of three parts:
  - an envelope that defines a framework for describing what is in a message and how to process it,
  - a set of encoding rules for expressing instances of application-defined datatypes, and
  - a convention for representing remote procedure calls and responses.

# SOAP Web Services

- SOAP services are not web-friendly
  - Not easily consumable from HTTP clients such as browsers or tools like cURL
- The client has to
  - Have access to a WSDL (Web Service Description Language)
    - Describes actions available
  - Know how to construct messages

# Web APIs

- In 2000, SalesForce launched a new API
- eBay then launched another
- These APIs were targeting third-party consumers and designed in an HTTP-friendly way

# REST

- Representational State Transfer
  - Roy Fielding’s dissertation
  - Architectural style for distributed hypermedia systems
  - It is a *Style*
    - Not a design pattern, a framework or a technology

# REST Constraints

- Client-server
- Stateless
- Cache
- Uniform interface
- Identification of resources
- Self-descriptive messages
- Manipulation of resources through representations
- Hypermedia as the engine of application state
- Layered System
- Code on demand

# Microsoft ASP.NET Web API

- ASP.NET Web API is a **framework** that makes it easy to build **HTTP services that reach a broad range of clients**, including browsers and mobile devices

- ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework

https://dotnet.microsoft.com/apps/aspnet/apis