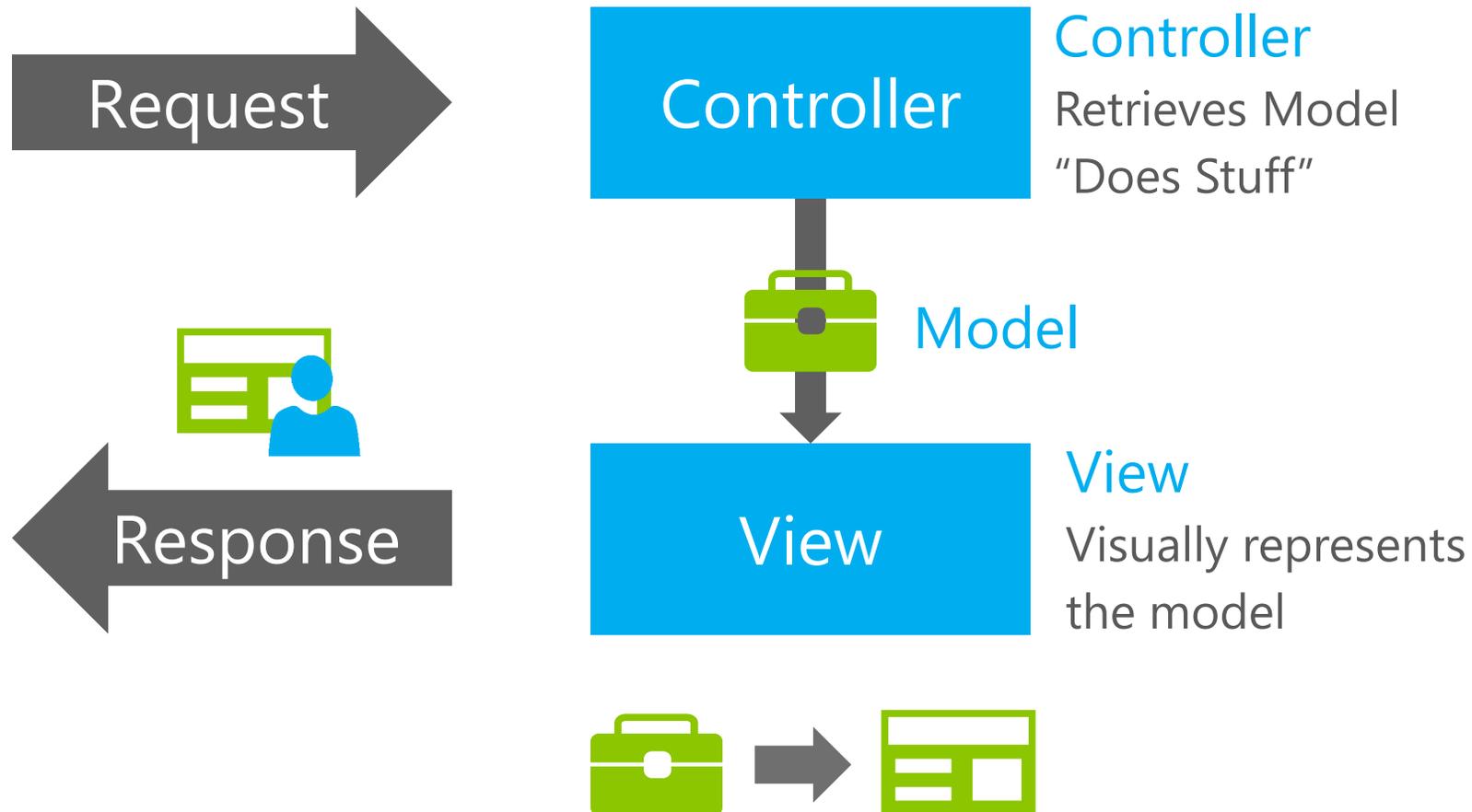


ASP.NET MVC

# Models, Views, and Controllers

What does MVC look like?



# Seems complicated. What's the point?

- Every web application needs some structure
- MVC helps you stay organized, start to finish
- Often end up with less code, not more
- Smoother learning curve as your project grows

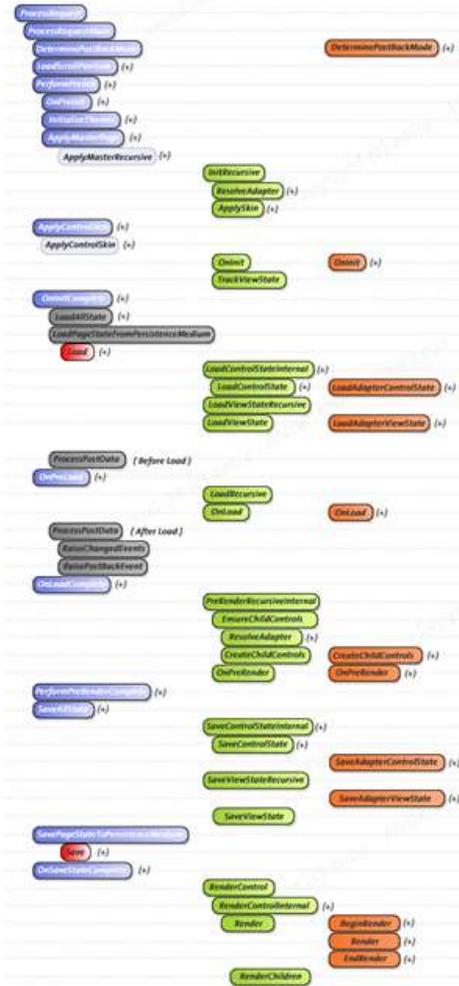
# ASP.NET Web Forms Values

- Productive way to build web applications
- Control and event-based programming model
- Controls that abstract HTML, JS and CSS
- Rich UI controls – datagrids, charts, Ajax
- Browser differences are handled for you

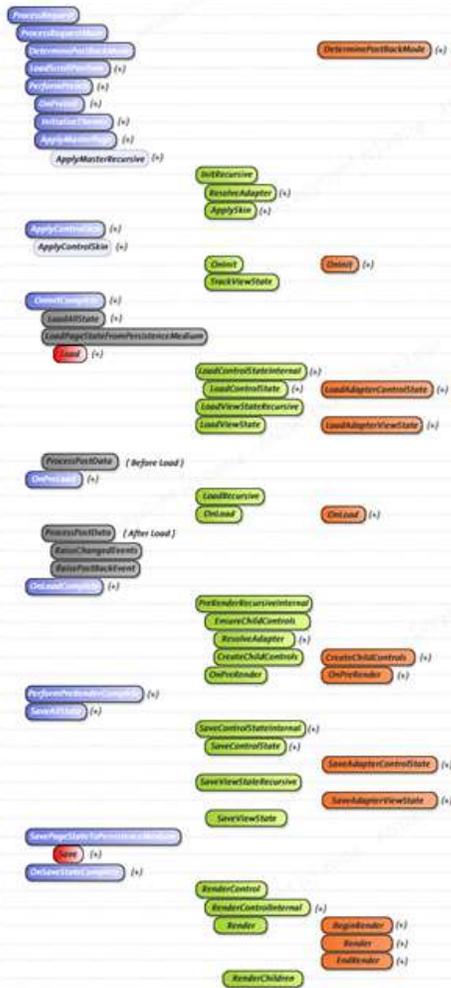
Summary: Web Forms handles a lot of things for you.



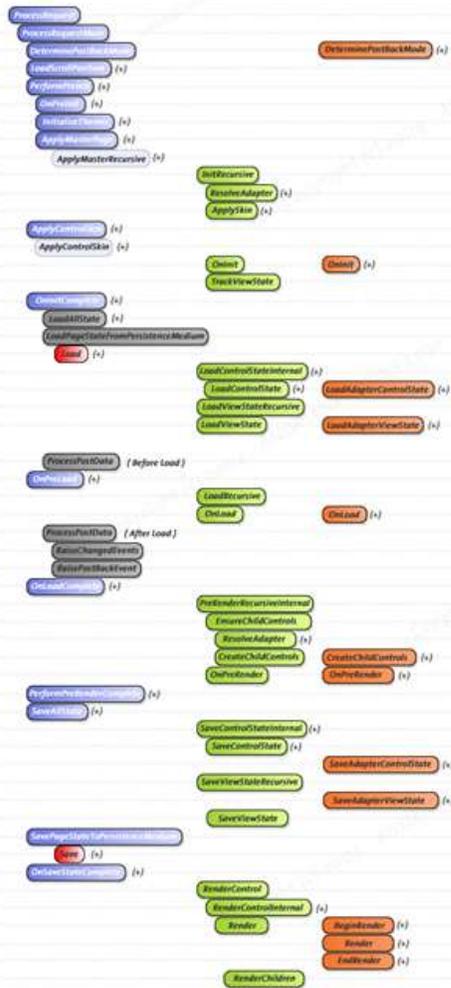
# Default.aspx



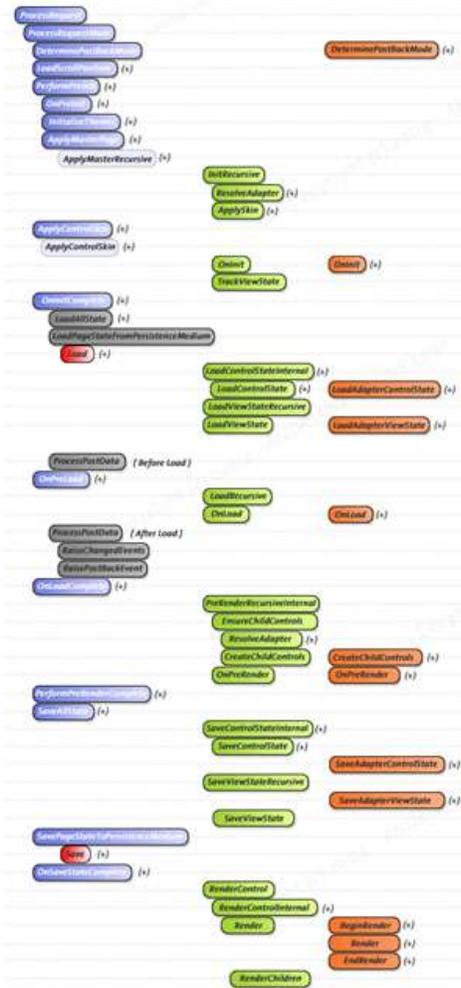
# Site.master



# Products.aspx

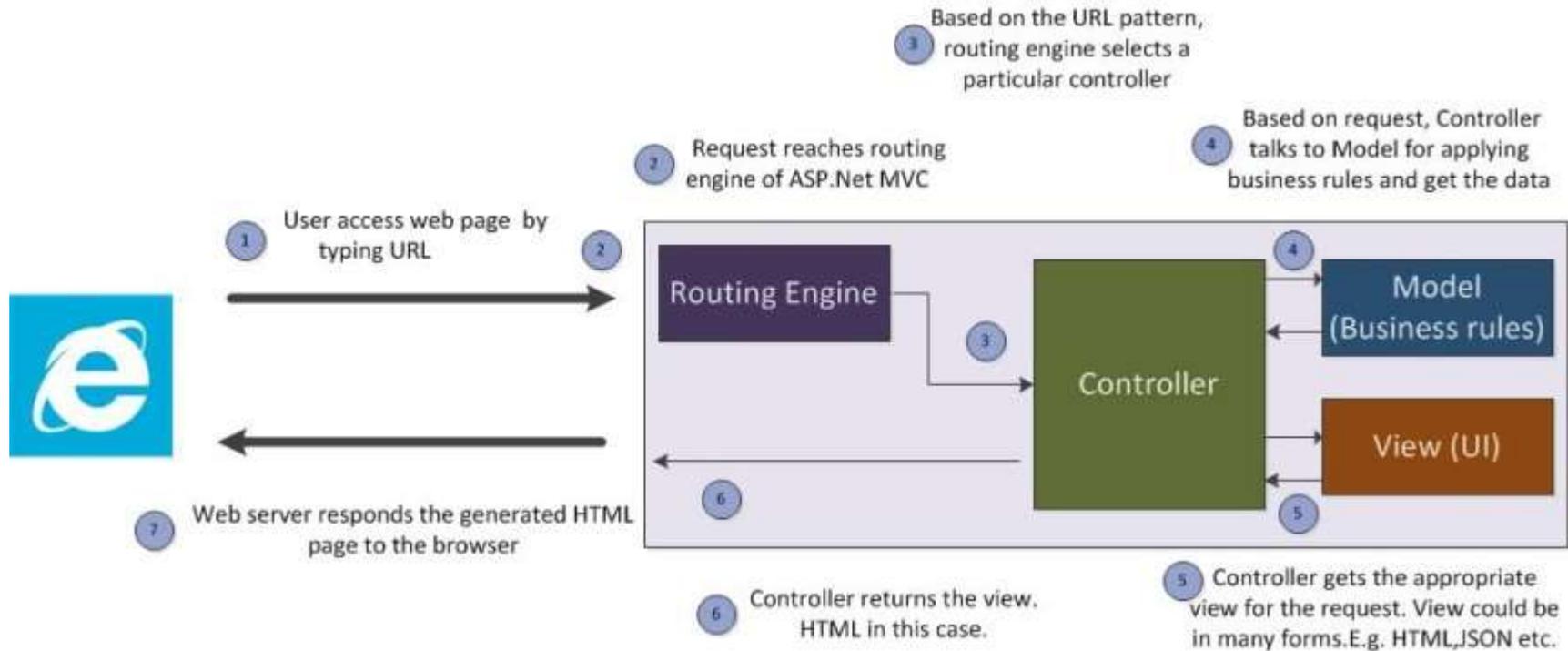


# Cart.ascx





# MVC Architecture



## Model

It is the specific representation of the information with which the system operates. Logic ensures the integrity of data and allows to derive it.

# View

Represents the model in a suitable format to interact and access the data, usually called "User Interface" (HTML, XML).

# Controller

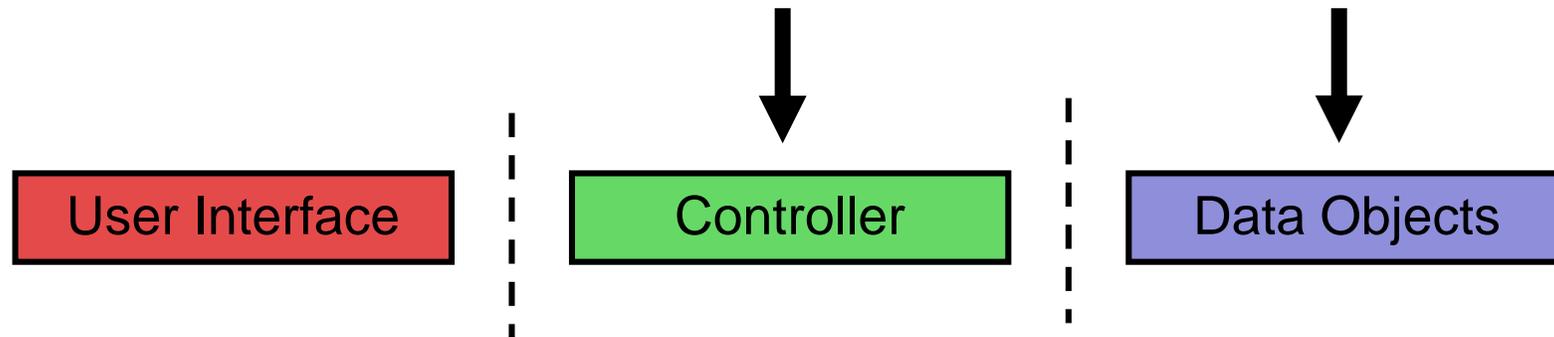
It is the link between the view and the model, is responsible for receiving and responding to events, typically user actions and invokes changes on the model and probably in the view

# What new does ASP.NET MVC bring in terms of web architecture?

1. Clear separation of logic: Model, View, Controller
2. Test-Driven Development
3. Full control over HTML and JavaScript
4. Friendly URLs

## Test-Driven Development

Since the UI is completely separated from the business logic, it's now easy to write Unit Tests for the ASP.NET MVC application.



## Full control over HTML & JS

In ASP.NET MVC, designed HTML & JS pages are not more messed up with “postbacks” and “ViewStates”, which is specific for “Web-forms” pattern.

```
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPaA8FDzhjYjBhZjA0ODYyMTM2NBgGBR5fX0NvbnRyb2xzUmVxdWlyZVBvc3RCYWNRs2V5X18WAQUhY3RsMDAkaGVhZGVyJHVjTWFP
bk1lbnUkc2VhcmNoQnRuBTdjGwwMCRNYWluQ29udGVudCRycHRWaWRlbyRjdGwwMyRwcmVzZW50YXRpb25DYXJkIG11bHRp" />
```

## Friendly URLs

MVC binds web paths to the logical URLs rather than to the physical files.

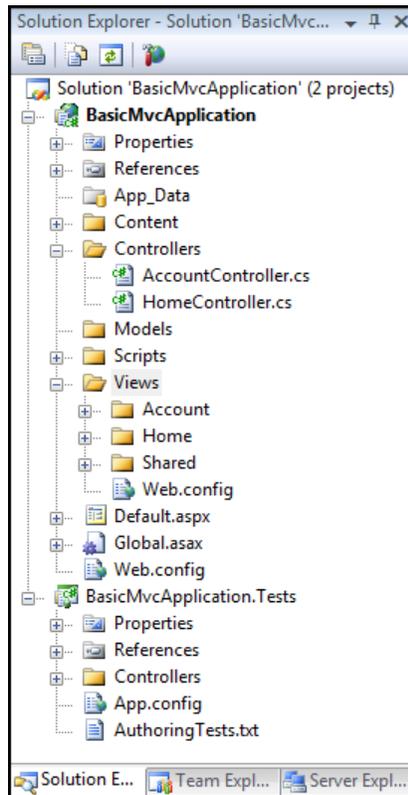
C:\inetpub\wwwroot\WebSite\Products.aspx

<http://www.website.com/Products.aspx?name=Meat>

<http://www.website.com/Products/Meat>

## How does ASP.NET MVC looks like?

A newly created project has an ASP.NET MVC specific directory structure.



**Content** folder keeps files such as scripts, CSS, images, and so on.

**Controllers** folder is the location for controllers. The MVC framework requires the names of all controllers to end with "Controller"—for example, HomeController, LoginController, or ProductController.

**Models** stores classes that handle application business logic.

**Scripts** folder is for script files that support the application. By default, this folder contains AJAX script files and the JQuery library.

**Views** is the recommended location for views..

## Conclusion

ASP.NET MVC supports pure **MVC** pattern, the same development pattern **Rails** are based on. The advantages of ASP.NET MVC applications:

- 1.It makes it easier to manage complexity by dividing an application into the model, the view, and the controller.
- 2.It does not use view state or server-based forms.
- 3.It uses a Front Controller pattern that processes Web application requests through a single controller. This enables you to design an application that supports a rich routing infrastructure with friendly URLs.
- 4.It provides better support for test-driven development (TDD).
- 5.It works well for Web applications that are supported by large teams of developers and Web designers who need a high degree of control over the application behavior.