



ASP.NET Core MVC and API





Somkiat Puisungnoen

Somkiat Puisungnoen

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Intro

Software Craftsmanship

Software Practitioner at สยามชัมนาณกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Somkiat Puisungnoen 15 mins · Bangkok · ⚙️

Java and Bigdata



somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help ▾

somkiat.cc

@somkiat.cc

Home Posts Videos Photos

Liked Following Share ...

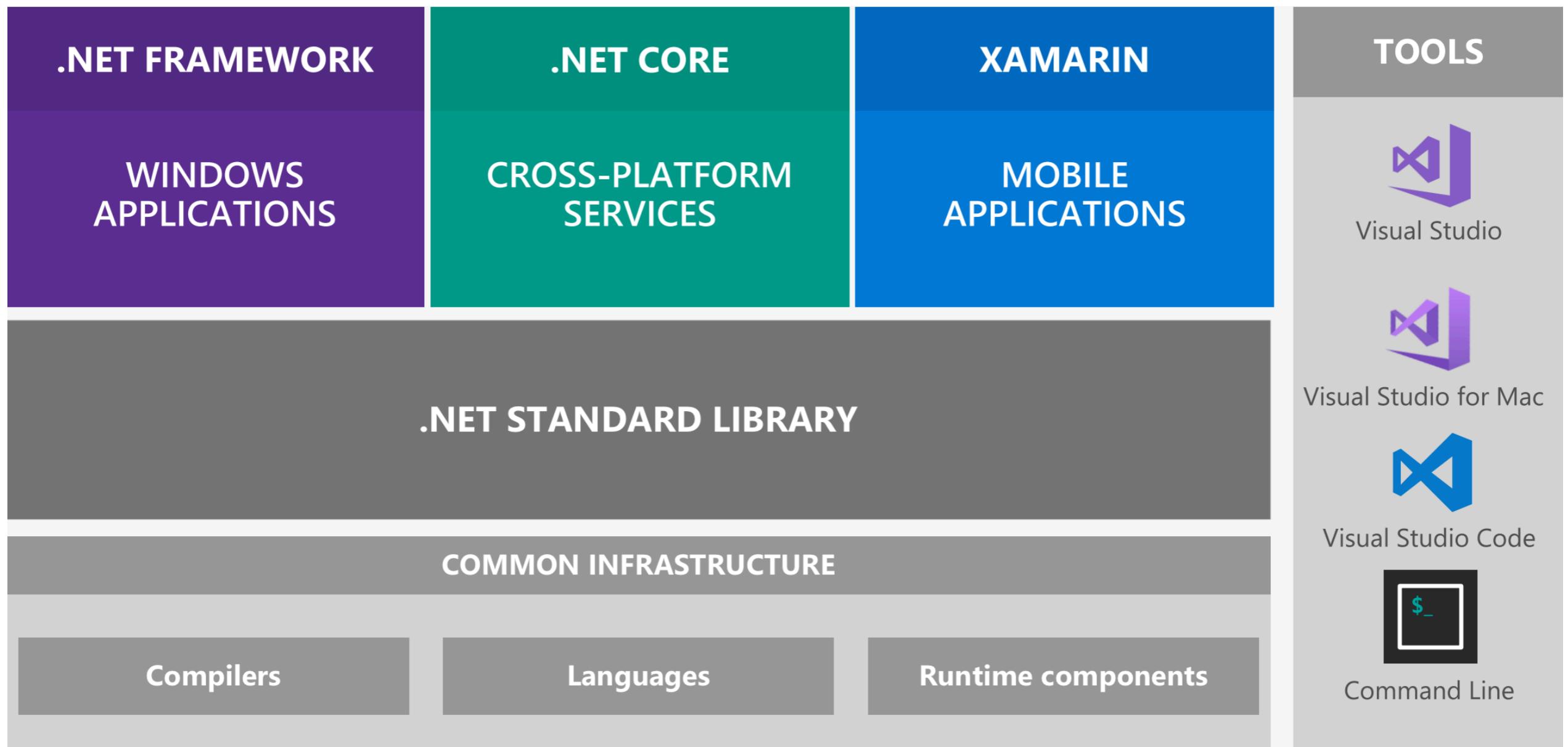
+ Add a Button



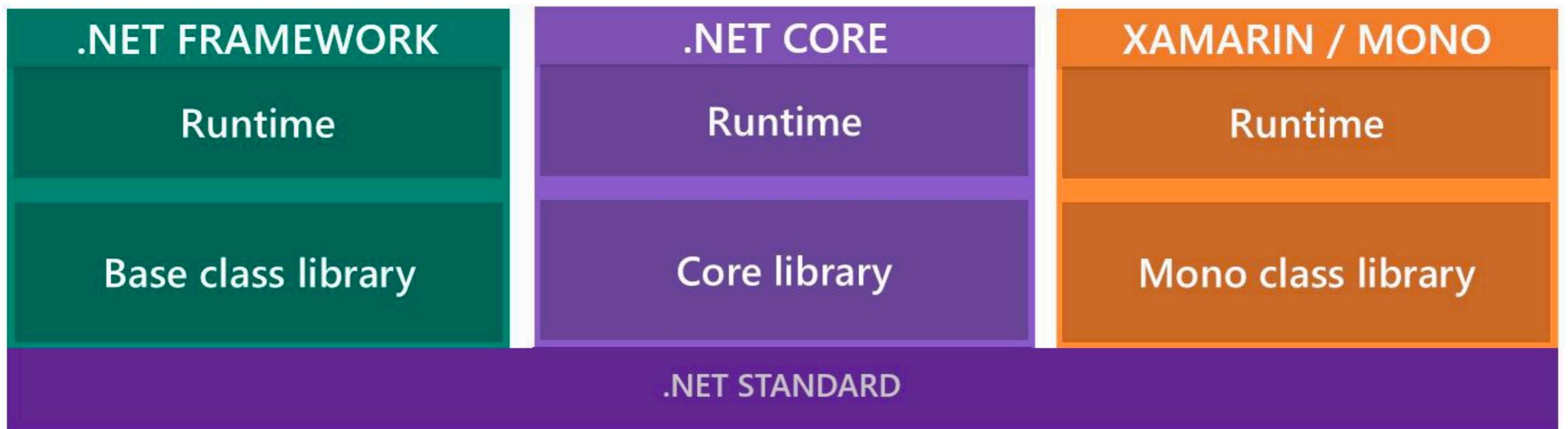
ASP.NET Core



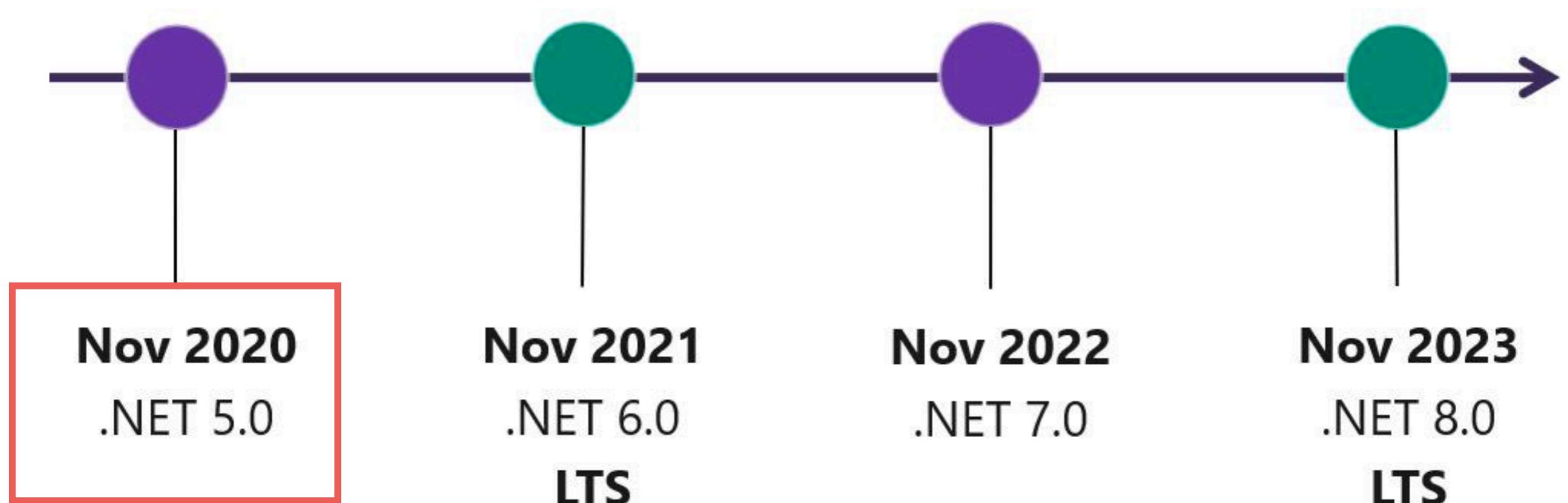
.NET big picture



.NET big picture



.NET Release schedule



.NET 5 = .NET Core vNext



.Net Core

Cross platform

Open source

Modern programming

High performance

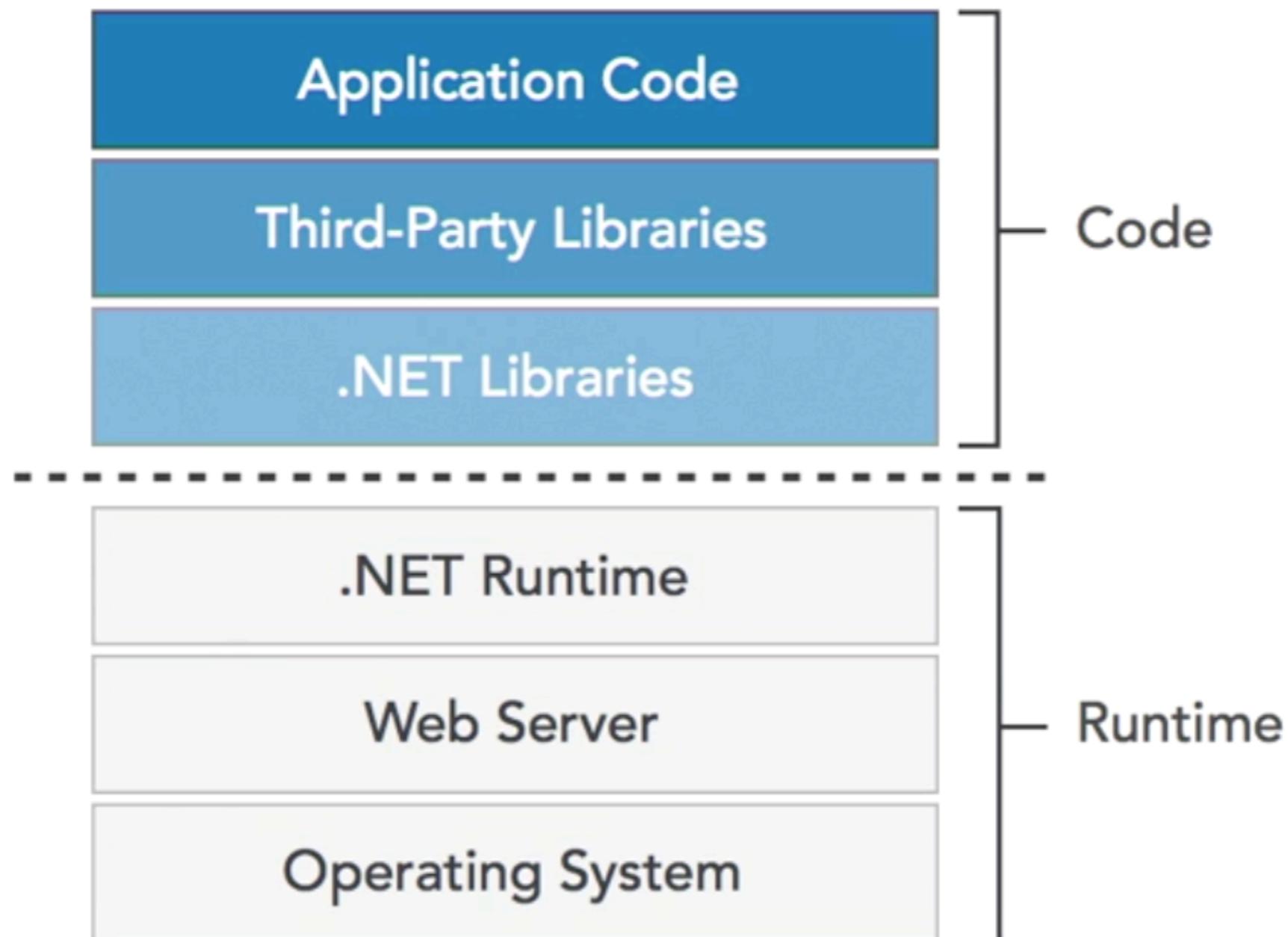
Consistent across environments

Command-line tools

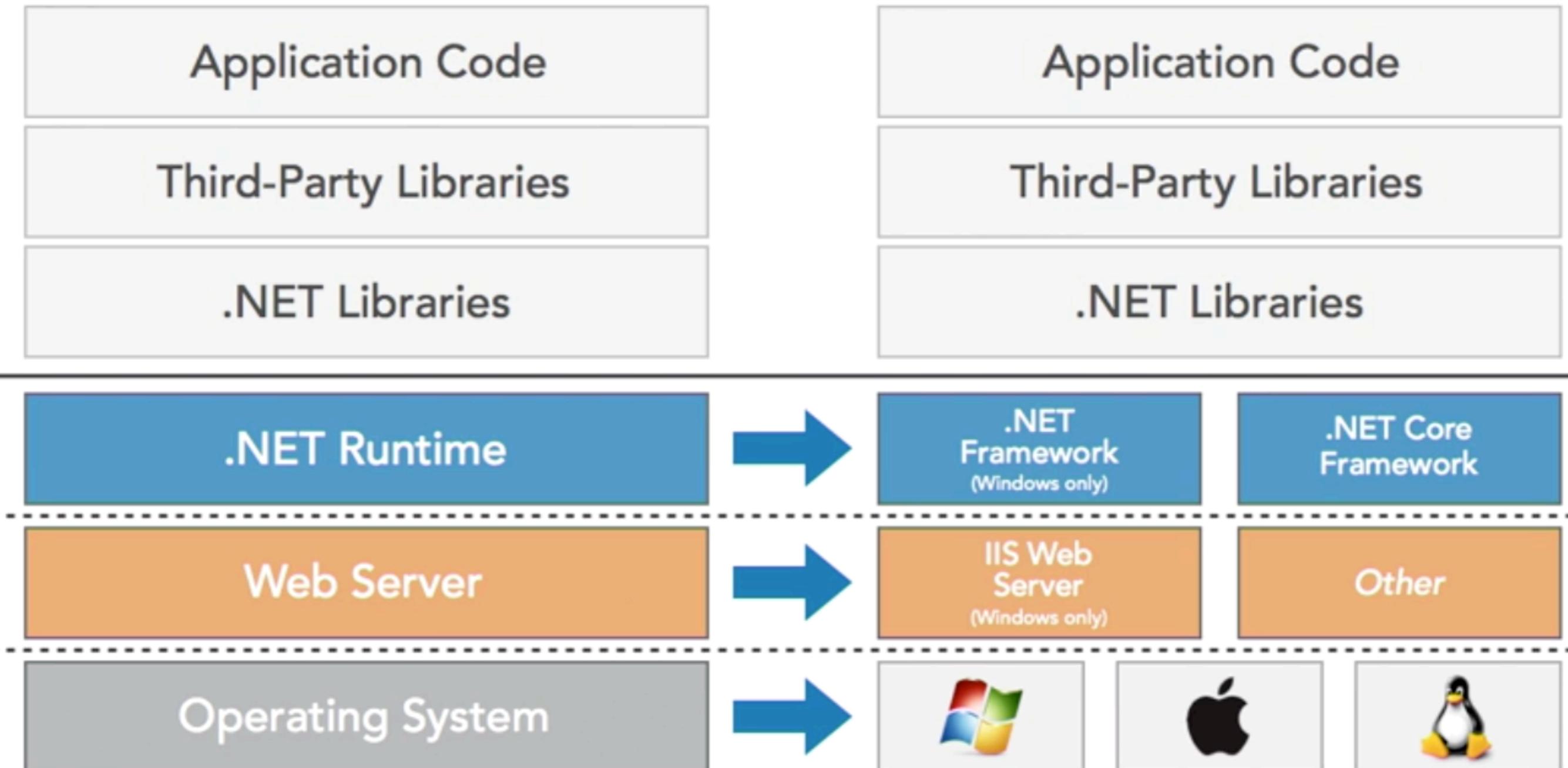
Flexible deployment



ASP.NET Core



ASP.NET Core



Installation

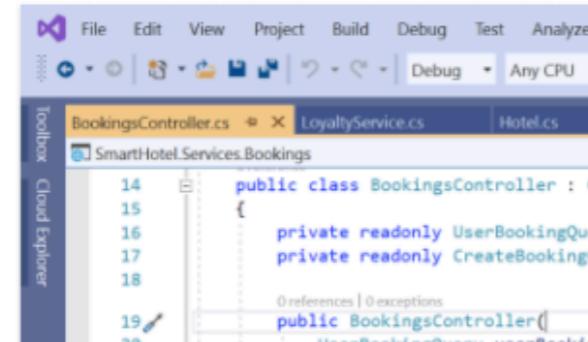


Visual Studio

Visual Studio Best-in-class tools for any developer



Visual Studio



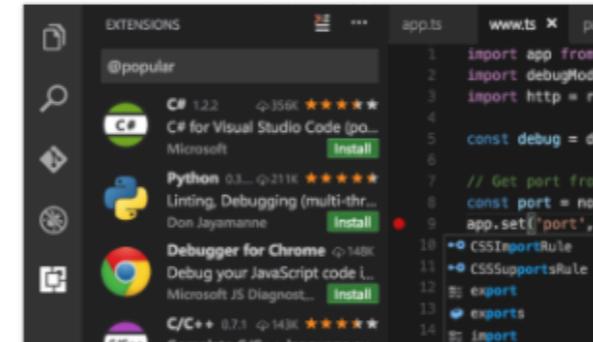
Full-featured IDE to code, debug, test, and deploy to any platform

[Download Visual Studio](#)

[Learn more >](#)



Visual Studio Code



Editing and debugging on any OS

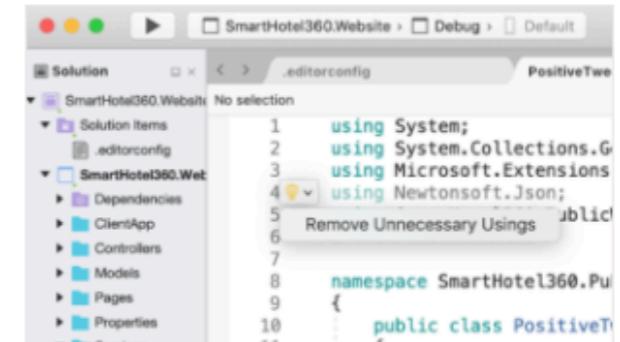
(By using Visual Studio Code you agree to its
[license and privacy statement](#))

[Download Visual Studio Code](#)

[Learn more >](#)



Visual Studio for Mac



Develop apps and games for iOS, Android, and web using .NET

[Read more about activating your license](#)

[Download Visual Studio for Mac](#)

[Learn more >](#)

<https://visualstudio.microsoft.com/>



.Net Core

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

Download .NET

Windows

Linux

macOS

Docker



.NET Core 3.1

.NET Core is a cross-platform version of .NET for building websites, services, and console apps.

Run Apps ⓘ

[Download .NET Core Runtime](#)

Build Apps ⓘ

[Download .NET Core SDK](#)

Advanced ⓘ

[All .NET Core downloads...](#)



.NET Framework 4.8

.NET Framework is a Windows-only version of .NET for building any type of app that runs on Windows.

Run Apps ⓘ

[Download .NET Framework Runtime](#)

Build Apps ⓘ

[Download .NET Framework Dev Pack](#)

Advanced ⓘ

[All .NET Framework downloads...](#)

<https://dotnet.microsoft.com/download>



.Net Core

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

Create first application



Create project

Web application
Model-View-Controller (MVC)
API
Angular/React/Redux





Choose a template for your new project

Recently used

Multiplatform

App
Library
Tests

iOS

App
Library
Tests

Android

App
Library
Tests

.NET Core

App
Library
Tests

Cloud

General

General

- Console Application
- Worker Service

ASP.NET Core

- Empty
- API
- Web Application
- Web Application (Model-View-Controller)
- Blazor Server App
- Razor Class Library
- Angular
- React.js
- React.js and Redux



Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

Support C# and F#

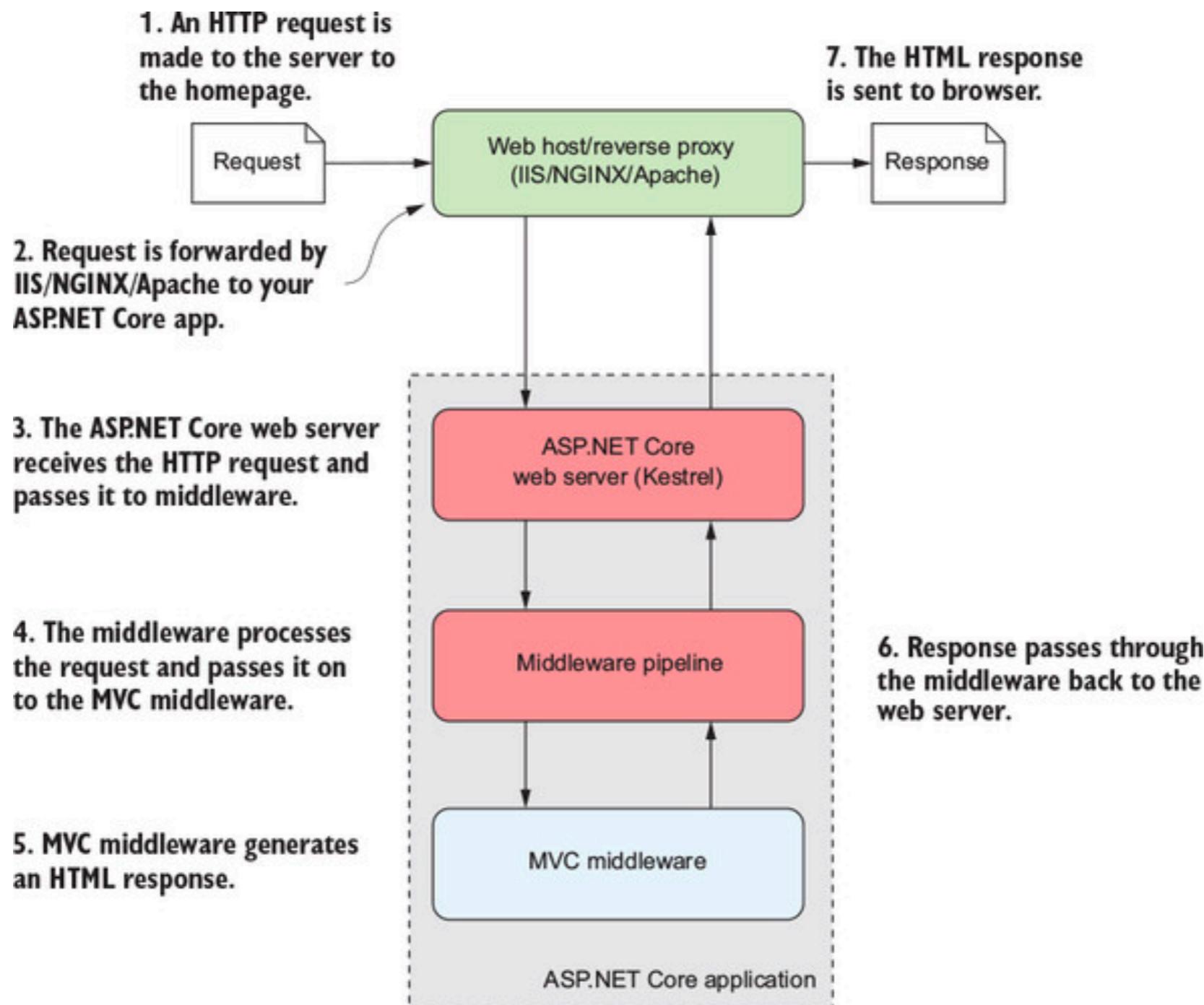
Cancel

Previous

Next



.Net core application



Fundamentals

Startup class

Host

Servers

Configuration

Options

Middleware



Fundamentals

Environments

Logging

Routing

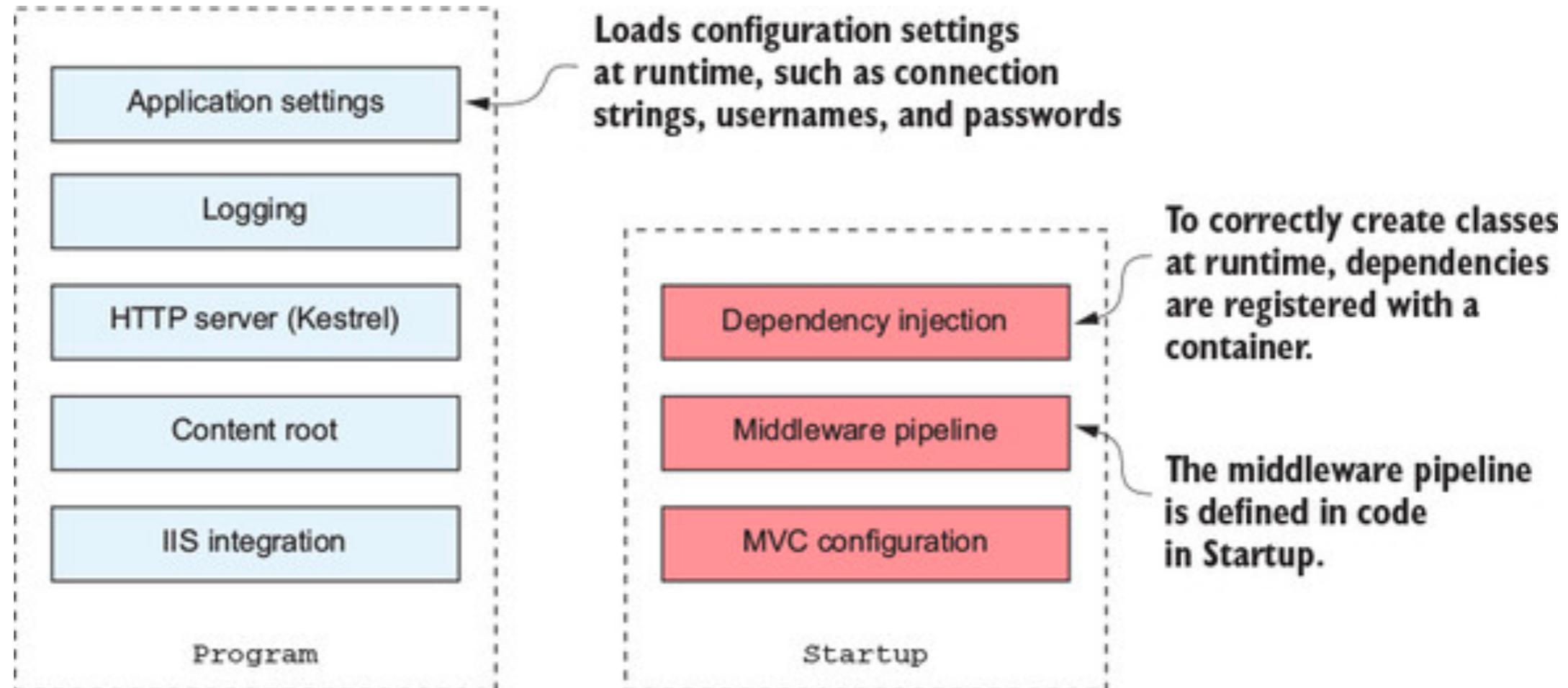
Error handle

HTTP request

Static file



Program.cs vs Startup.cs



Program.cs is used to configure infrastructure that rarely changes over the lifetime of a project.

Startup is used to configure the majority of your application's custom behavior.



Program.cs

Starting your application

Building a web host

Configure infrastructure that rarely changes
over the lifetime of application



Program.cs

```
public class Program
{
    public static void Main(string[] args)
    {
        CreateHostBuilder(args).Build().Run();
    }

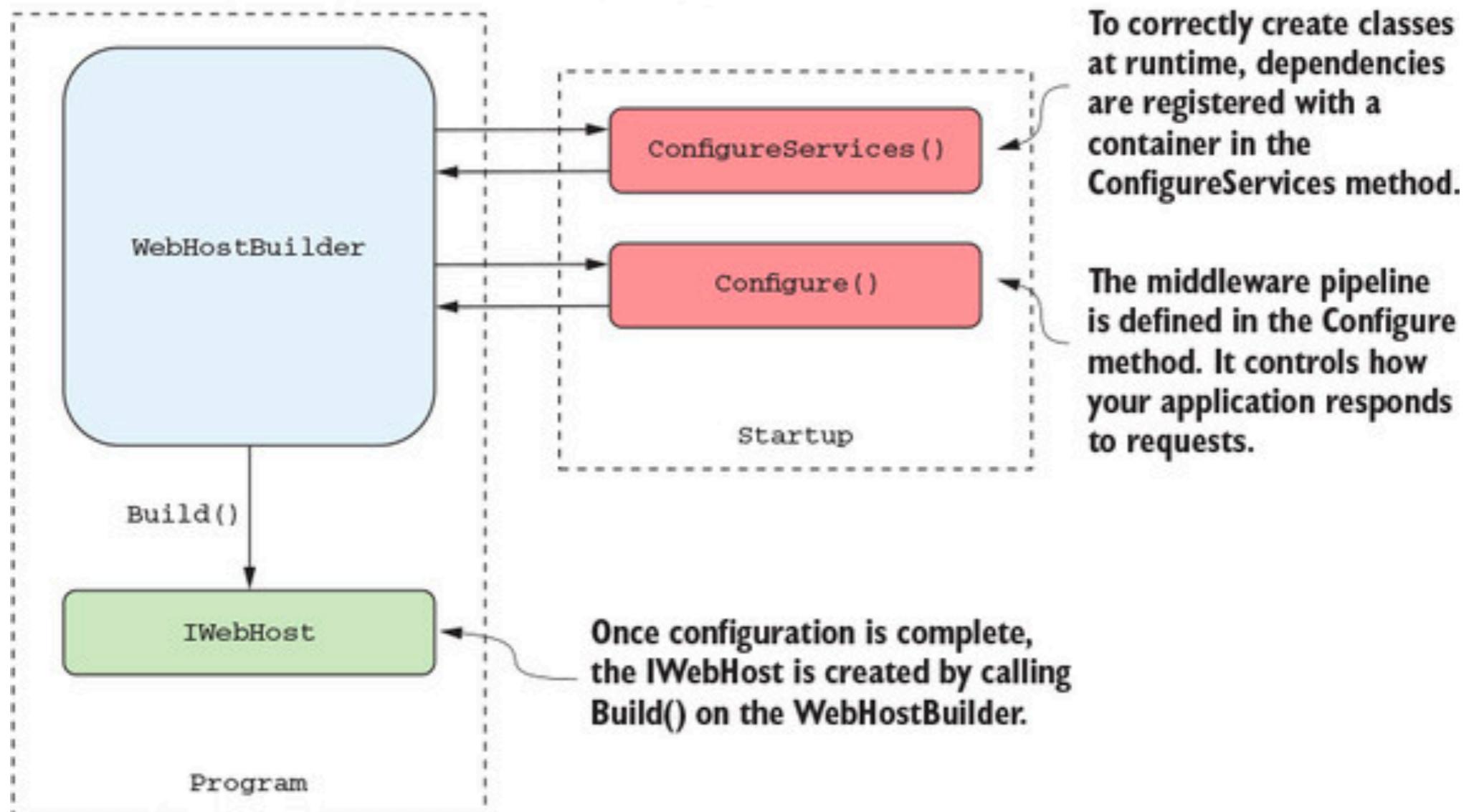
    public static IHostBuilder CreateHostBuilder(string[] args) =>
        Host.CreateDefaultBuilder(args)
            .ConfigureWebHostDefaults(webBuilder =>
            {
                webBuilder.UseStartup<Startup>();
            });
}
```



Program.cs

The `IWebHost` is created in `Program` using the `Builder` pattern, and the `CreateDefaultBuilder` helper method.

The `WebHostBuilder` calls out to `Startup` to configure your application.



Startup.cs

Configuration your application
Service registration
Middleware and MVC



Service registration in Startup

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddRazorPages();
}
```

Add your services

Add MVC

“Called Dependency Injection (DI)”



Dependency Injection in .Net Core

Technique for achieving
Inversion of Control (IoC) between classes
and dependencies

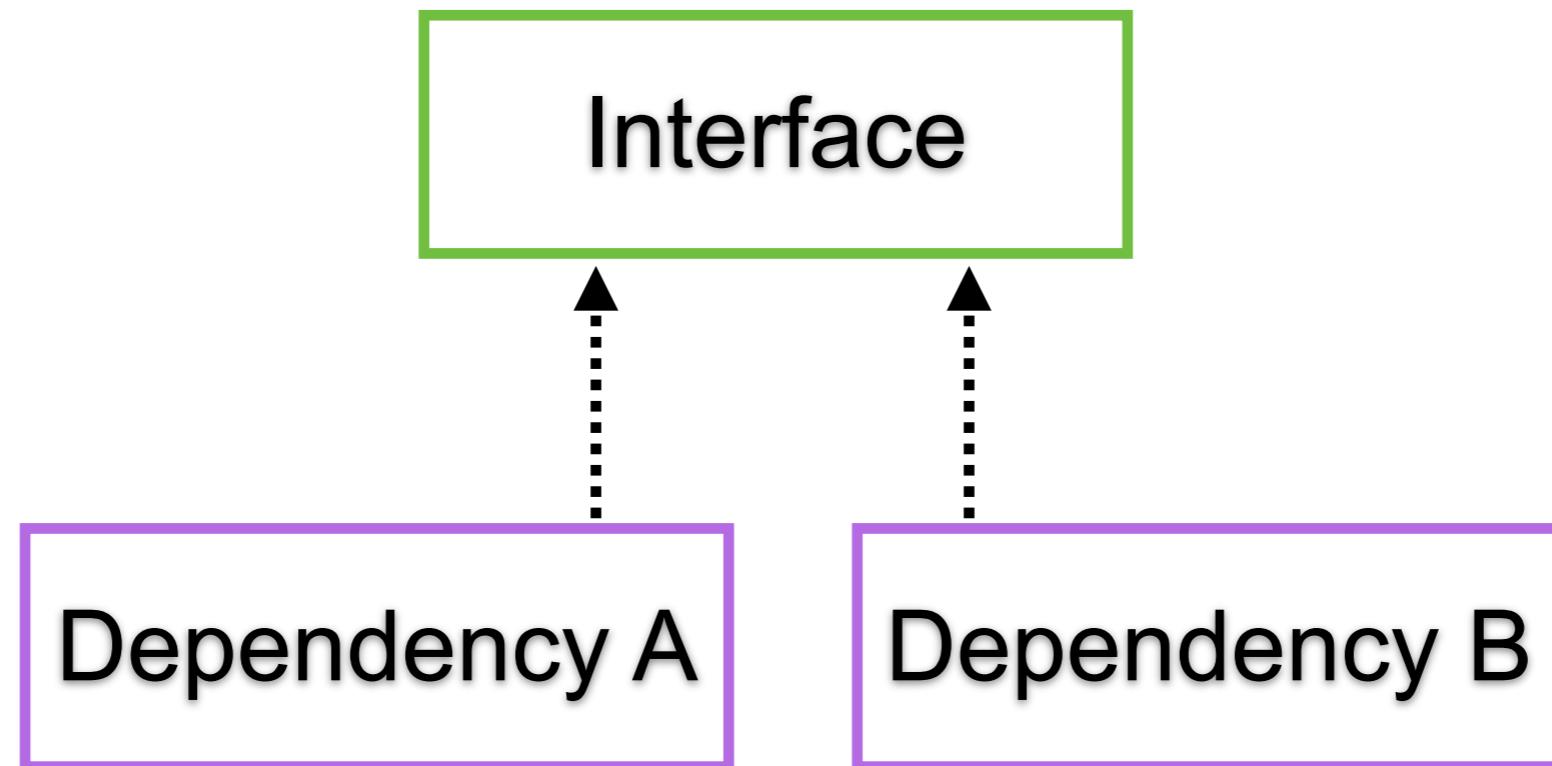


Dependency Injection

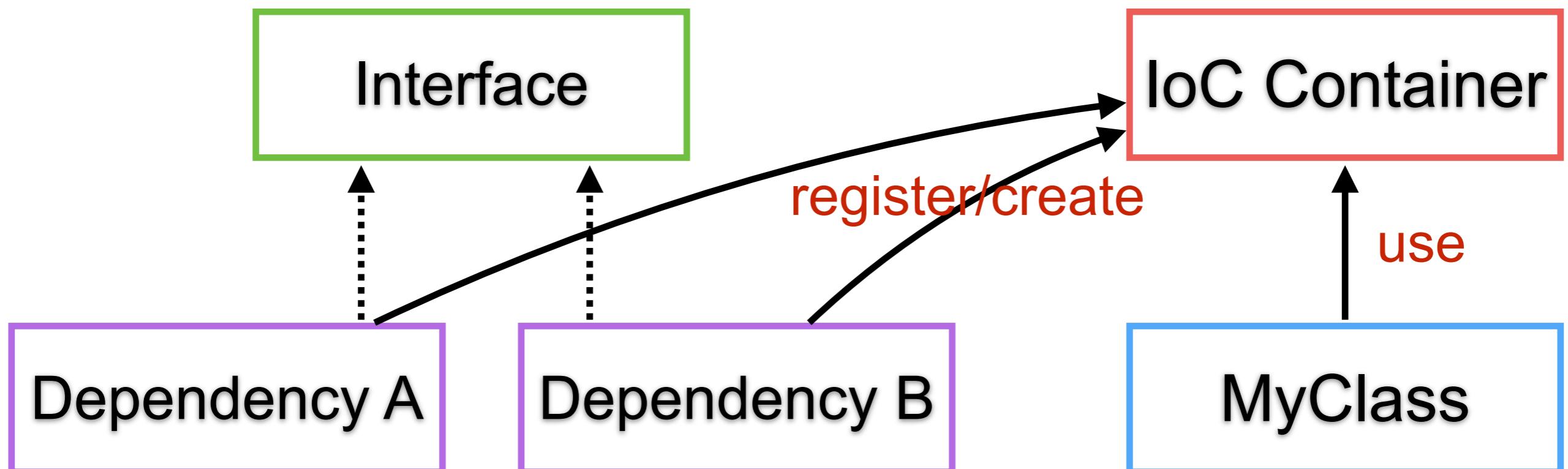
Constructor Injection
Setter Injection
Interface Injection
Method Injection



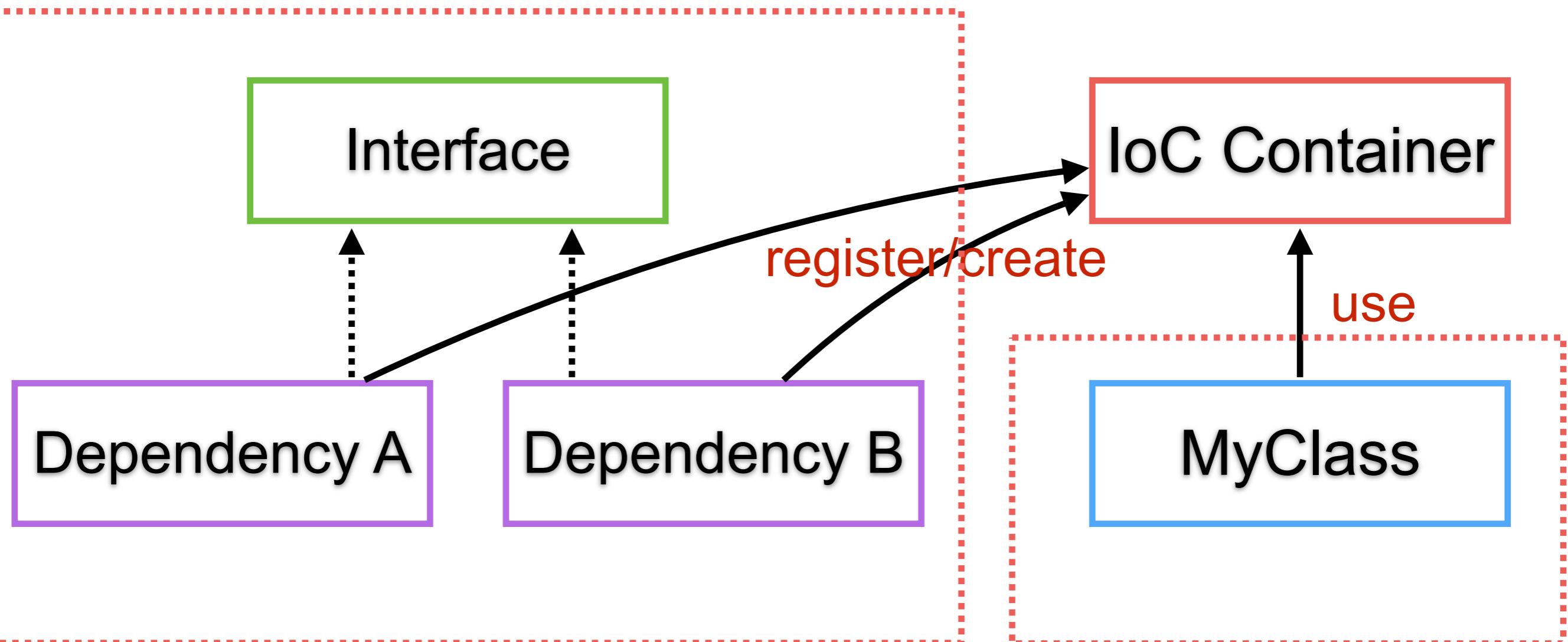
Inversion of Control



Inversion of Control



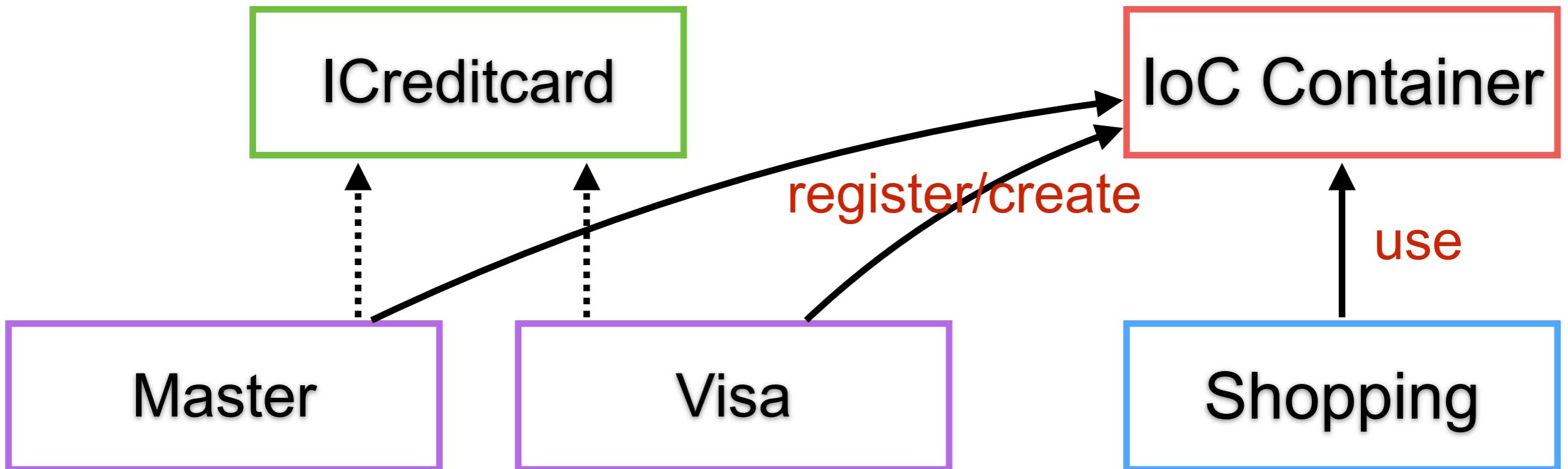
Inversion of Control



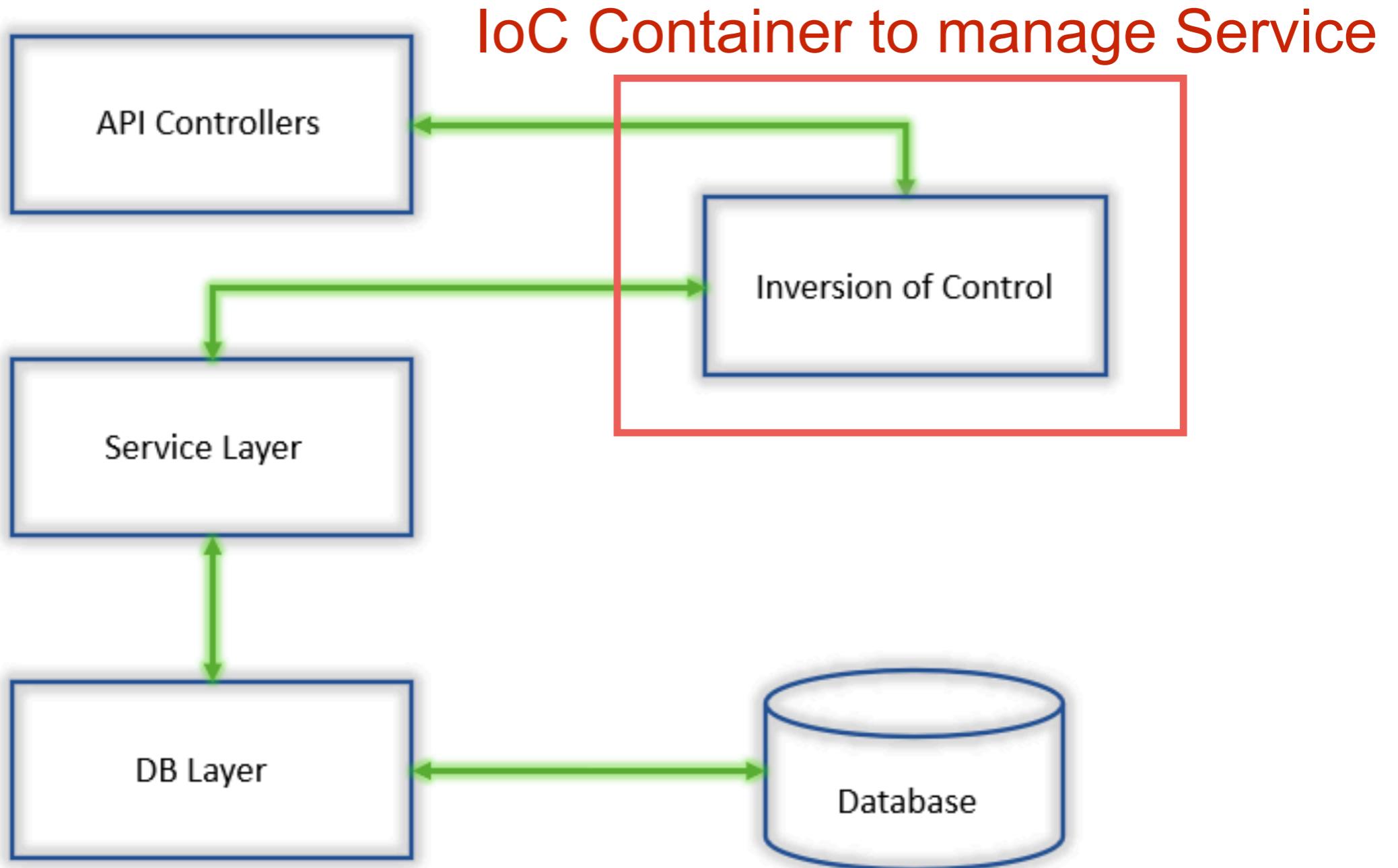
“Loose coupling”



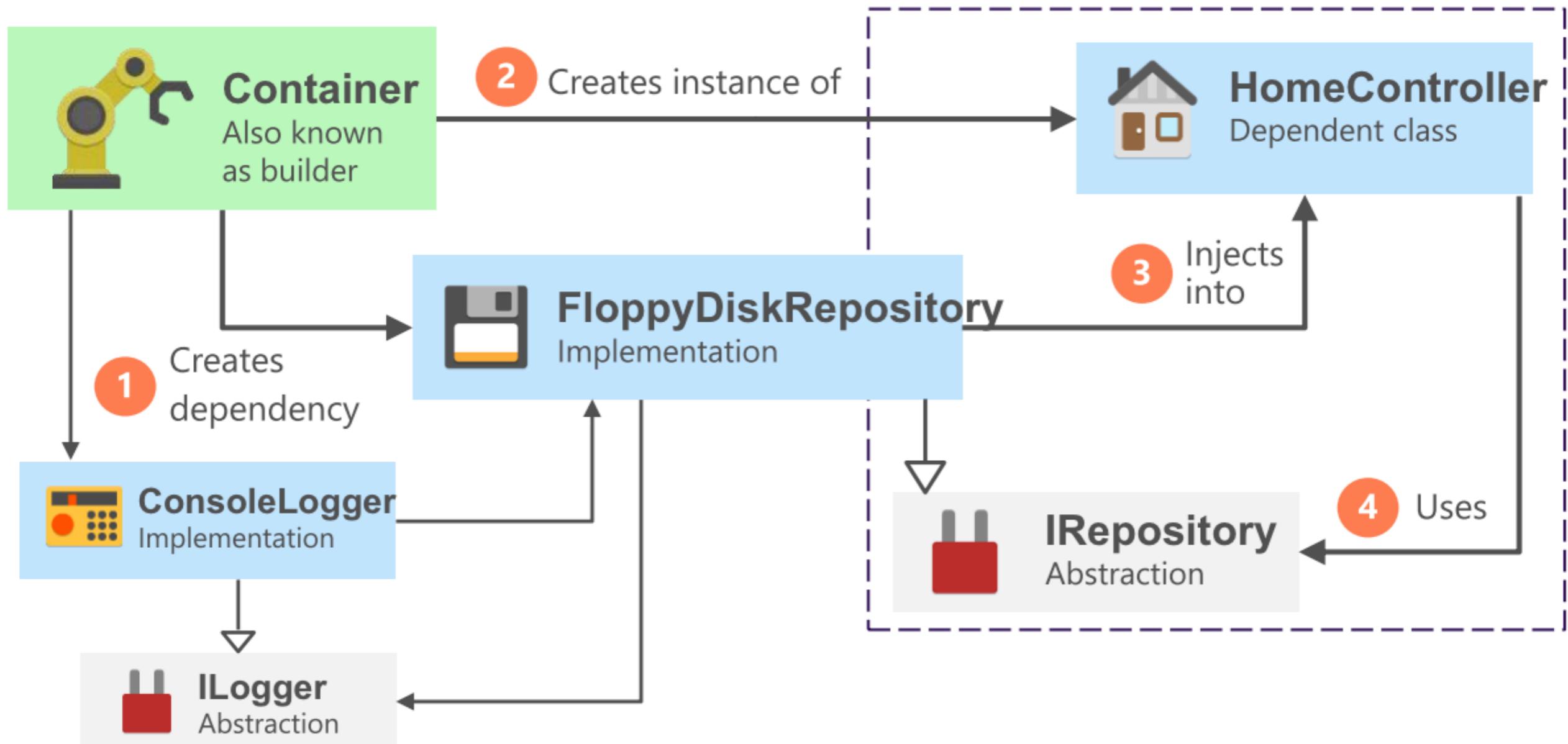
Example



Example



Example



DI Life cycle in .Net Core

1 Register

`IServiceCollection`

- **How**
- Add... and TryAdd... of type mappings
- Add... extension methods
- **Where**
- Application root
- Startup class (ASP.NET)

2 Resolve

`IServiceProvider`

- **Implicit**
- Constructor injection
- ASP.NET Core specifics
- **Explicit**
- `GetService<T>`
- `GetRequiredService<T>`
- Also for enumerables

3 Release

`IDisposable`

- **Automatic**
- Resolved instances and dependencies
- **Scopes**
- End of scope
- Might need to create child scopes



Lifetime of resolved objects

Singleton

- Easy way to implement singleton pattern
- One instance per DI container
- Beware of concurrency and threading issues

Scoped

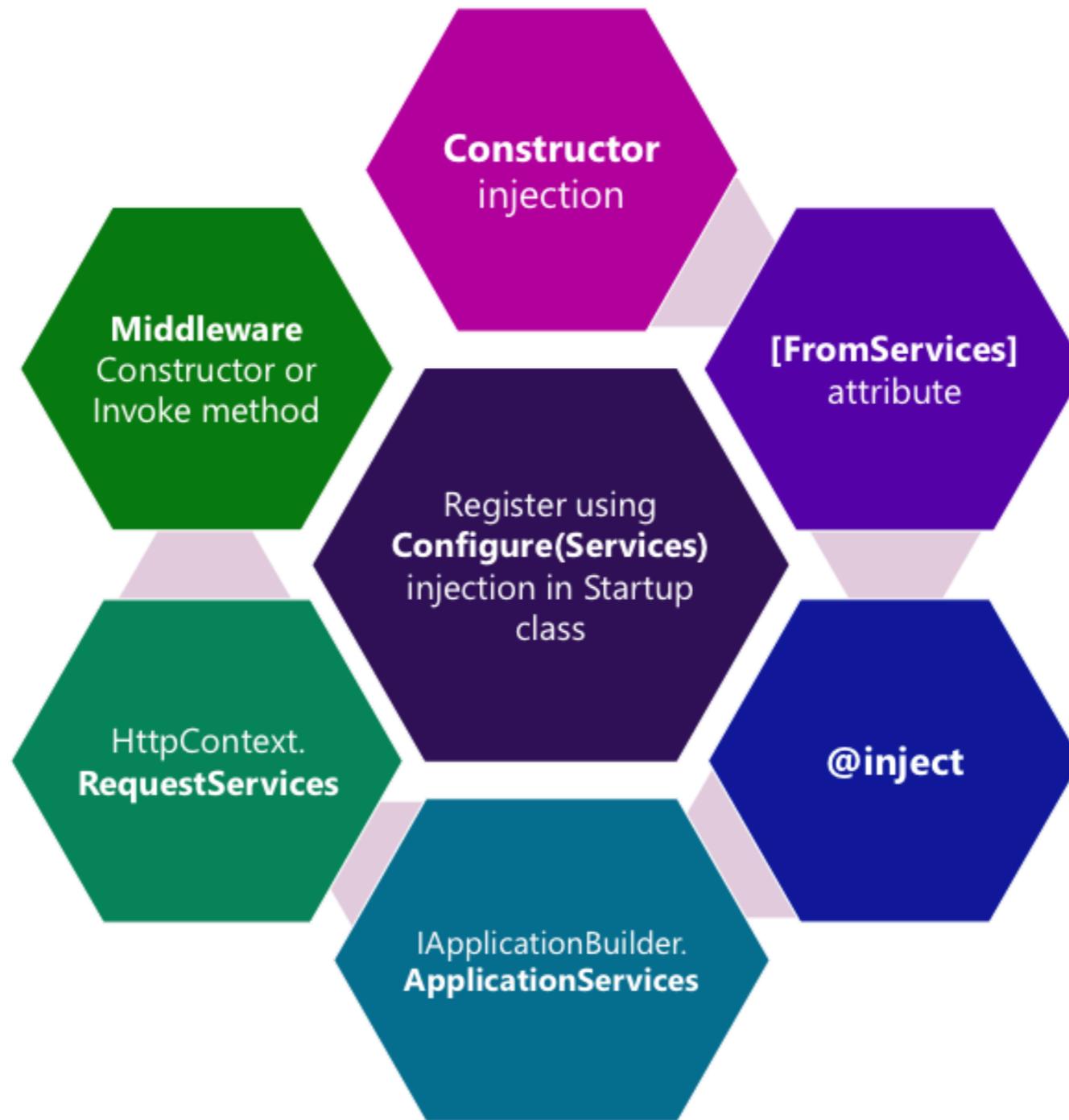
- Duration of scope
- ASP.NET uses web request as scope
- Useful for UnitOfWork objects, e.g. DbContext
- See also ServiceScopeFactory

Transient

- Single use
- Most common and safest



Resolving services



Alternative DI framework

Autofac

Castle Windsor

Unity

Lamar

SimpleInjector

Ninject

LightInject

Spring.NET



Workshop



Middleware in Startup

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Error");
        // The default HSTS value is 30 days. You may want to change this for production
        app.UseHsts();
    }

    app.UseHttpsRedirection();
    app.UseStaticFiles();

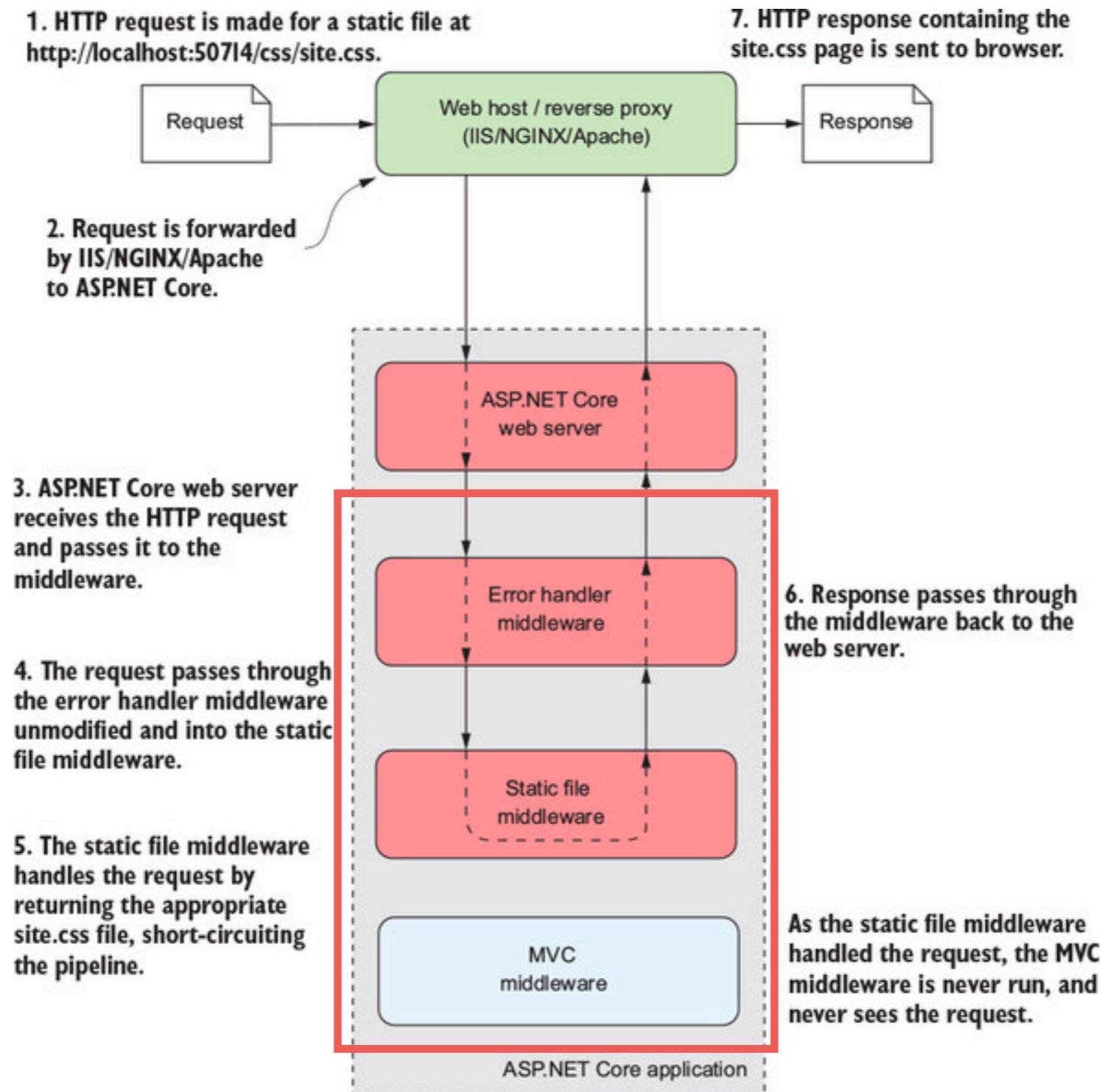
    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapRazorPages();
    });
}
```



Middleware in Startup



Workshop

Response to HTTP request
Serving static files
Error handling
Configuration



Model-View-Controller (MVC)



Model-View-Controller (MVC)

User interface design pattern

Promotes separation of concerns across
multiple application layers



Web page

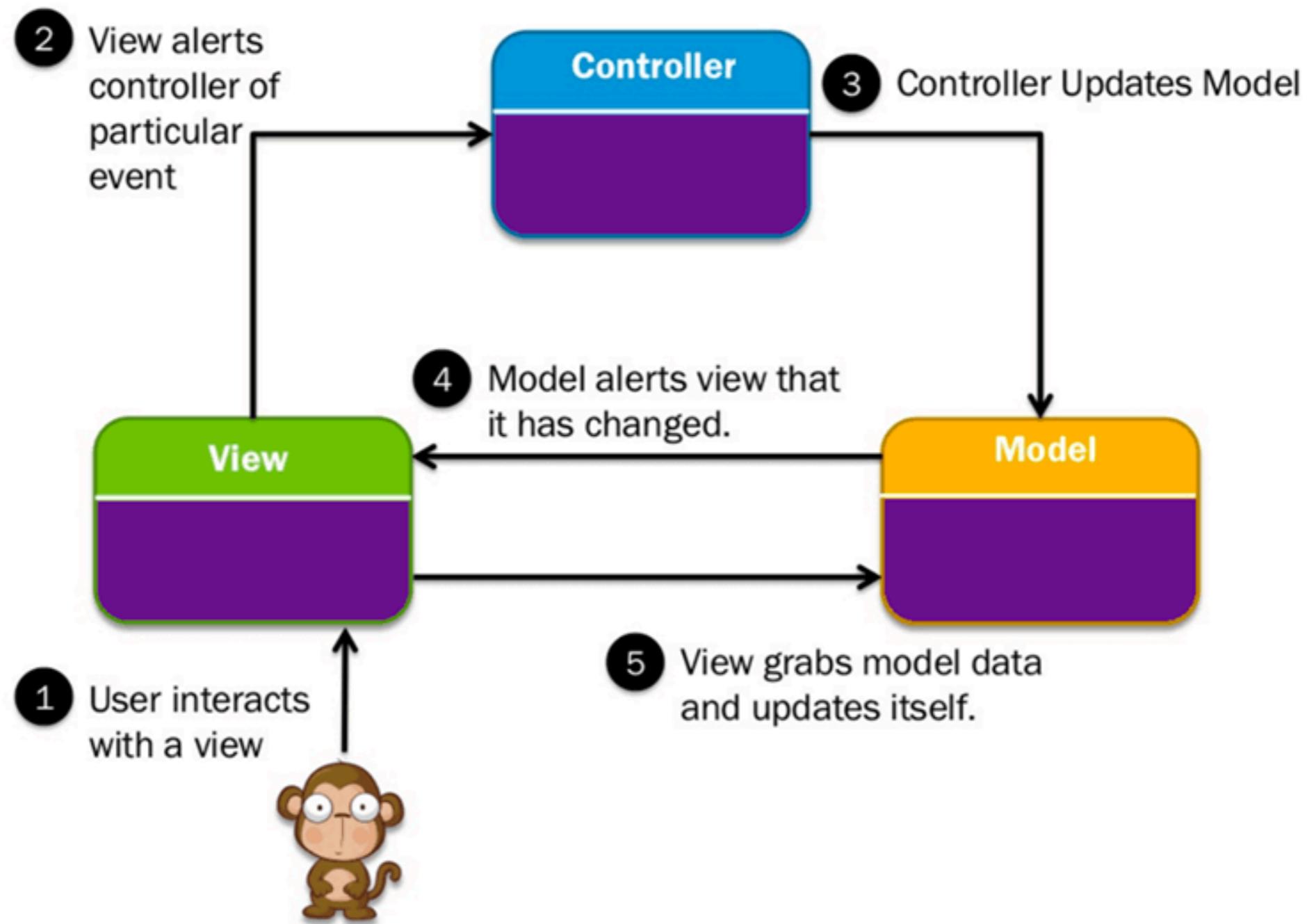
View logic

Application logic

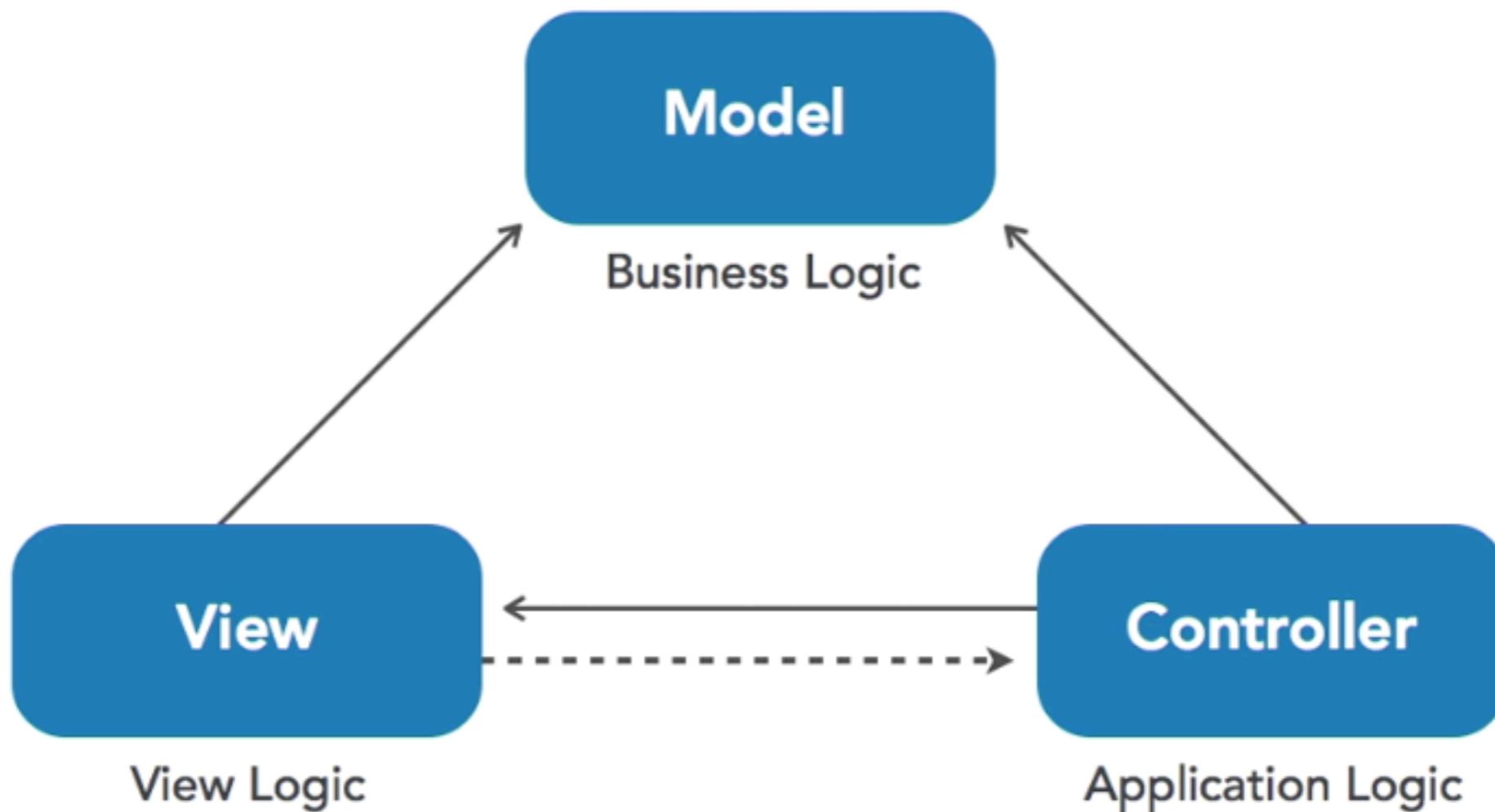
Business logic



Model-View-Controller (MVC)



Model-View-Controller (MVC)



Example

My Blog

Middleware

...

...

[View Comments](#)

Mobile First

...

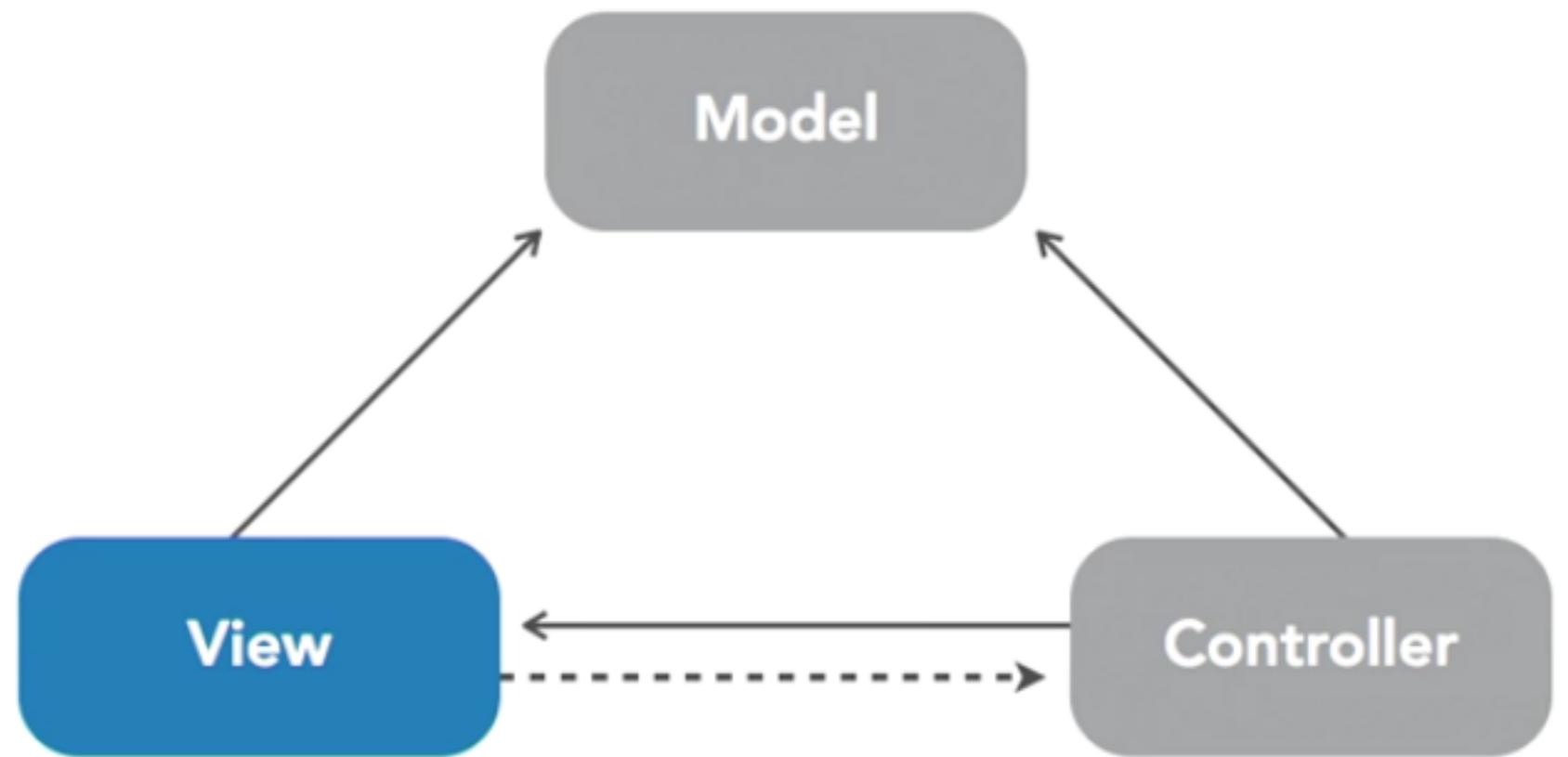
...

[View Comments](#)



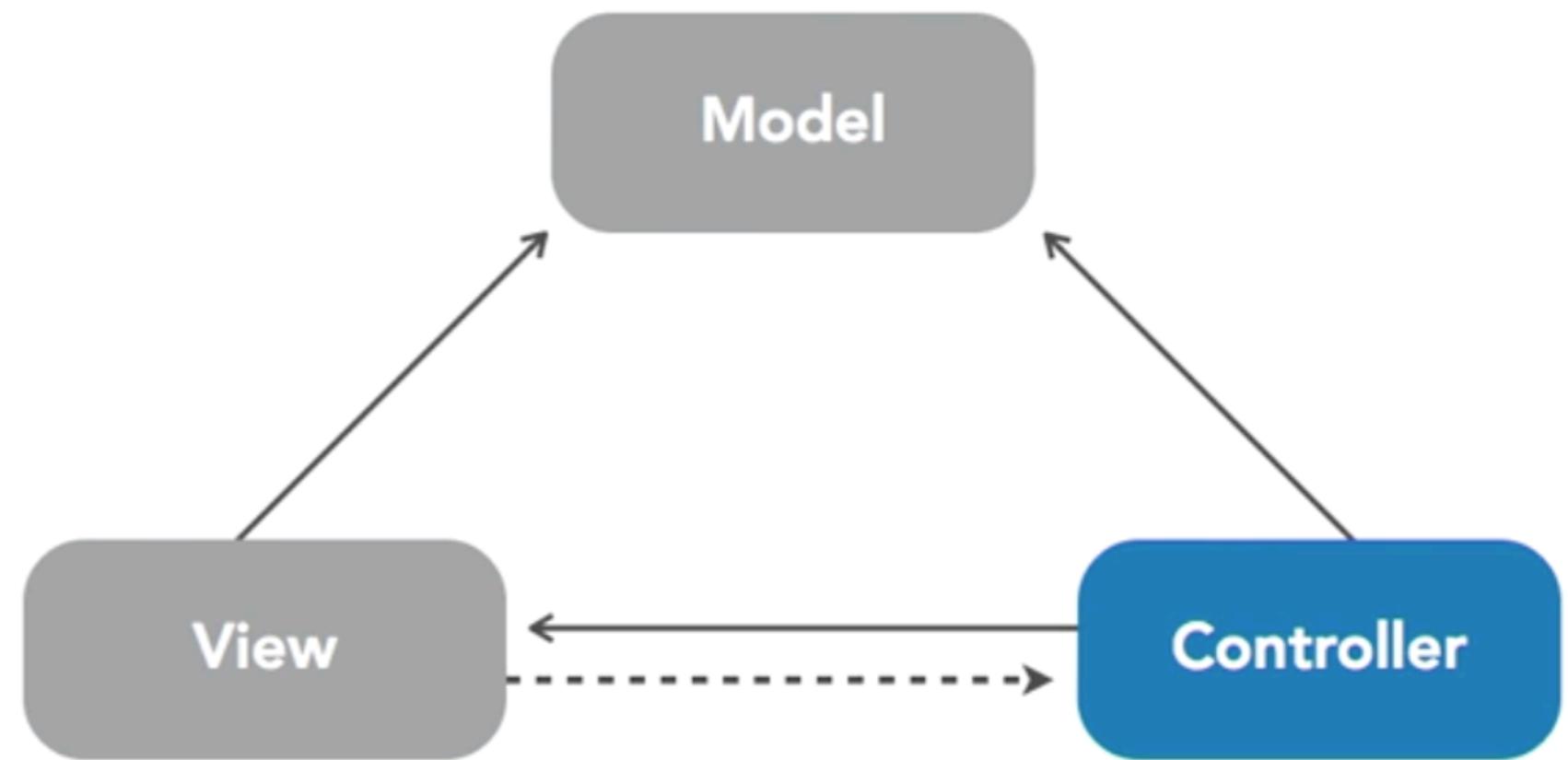
View

Good look and feel
Good user experience
Little logic



Controller

Controls application workflow
Orchestrates interaction between model and view



Model

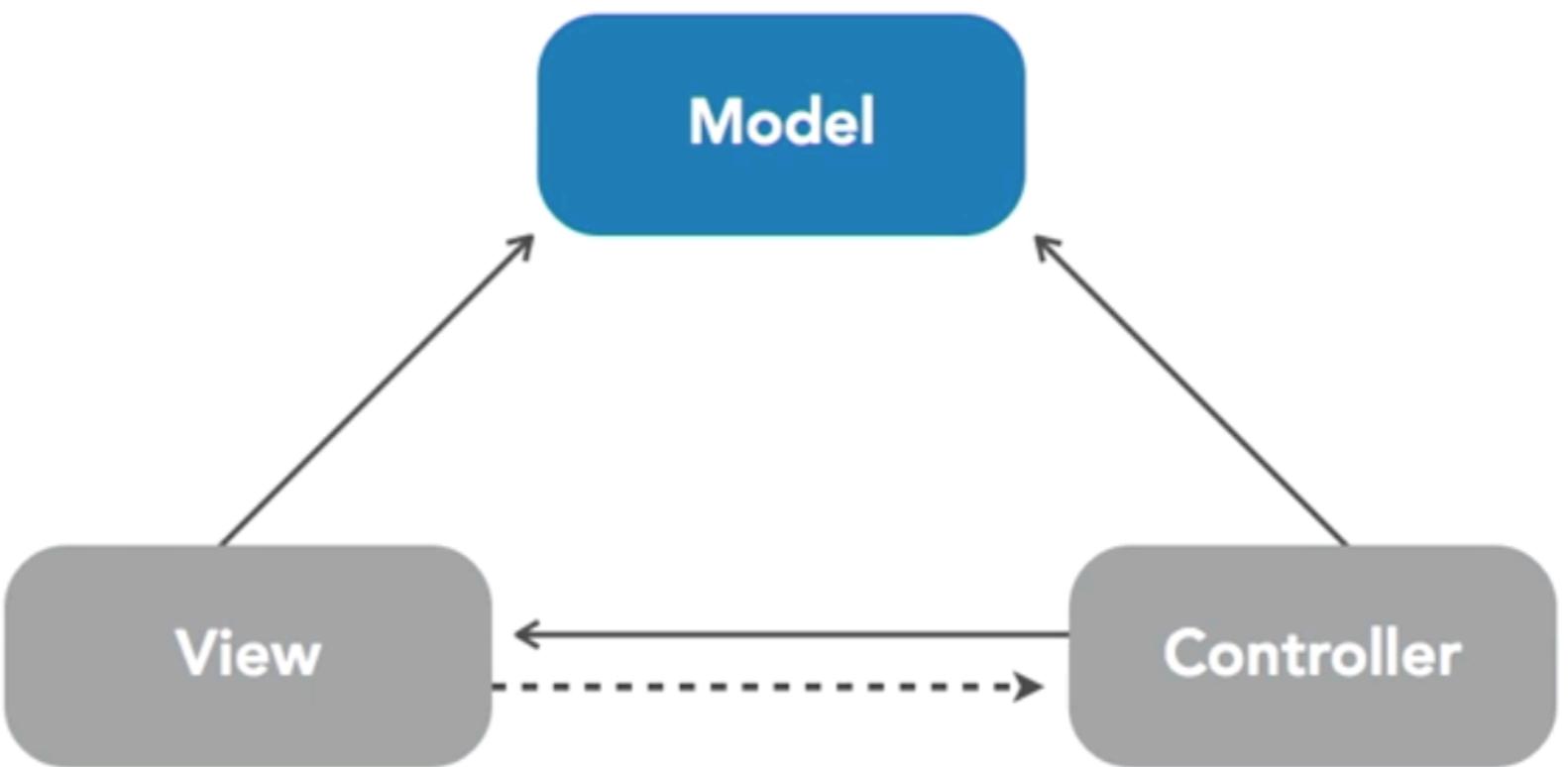
All business logic

Business process

Validation rules

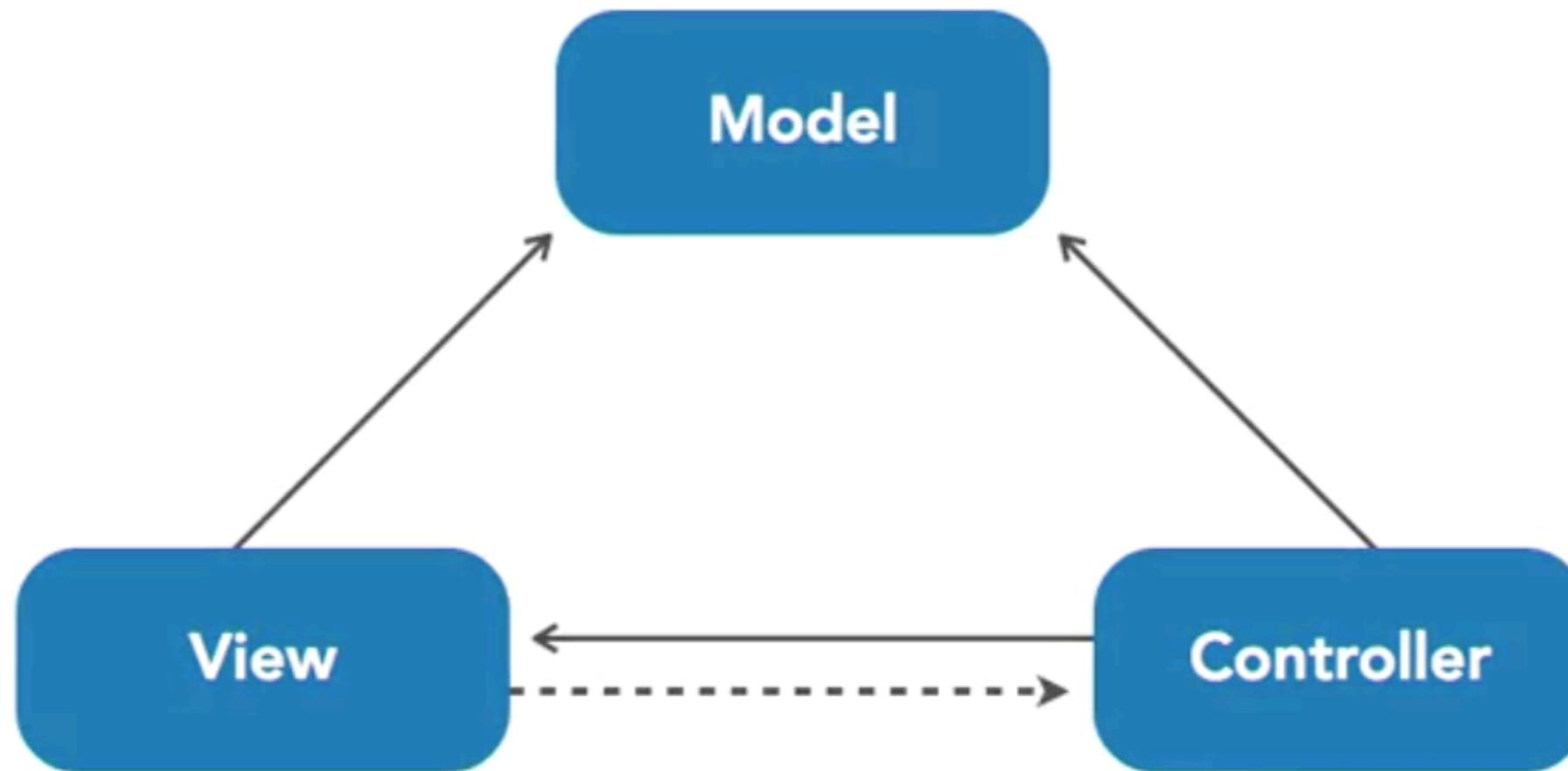
System integration

Not know about view, controller

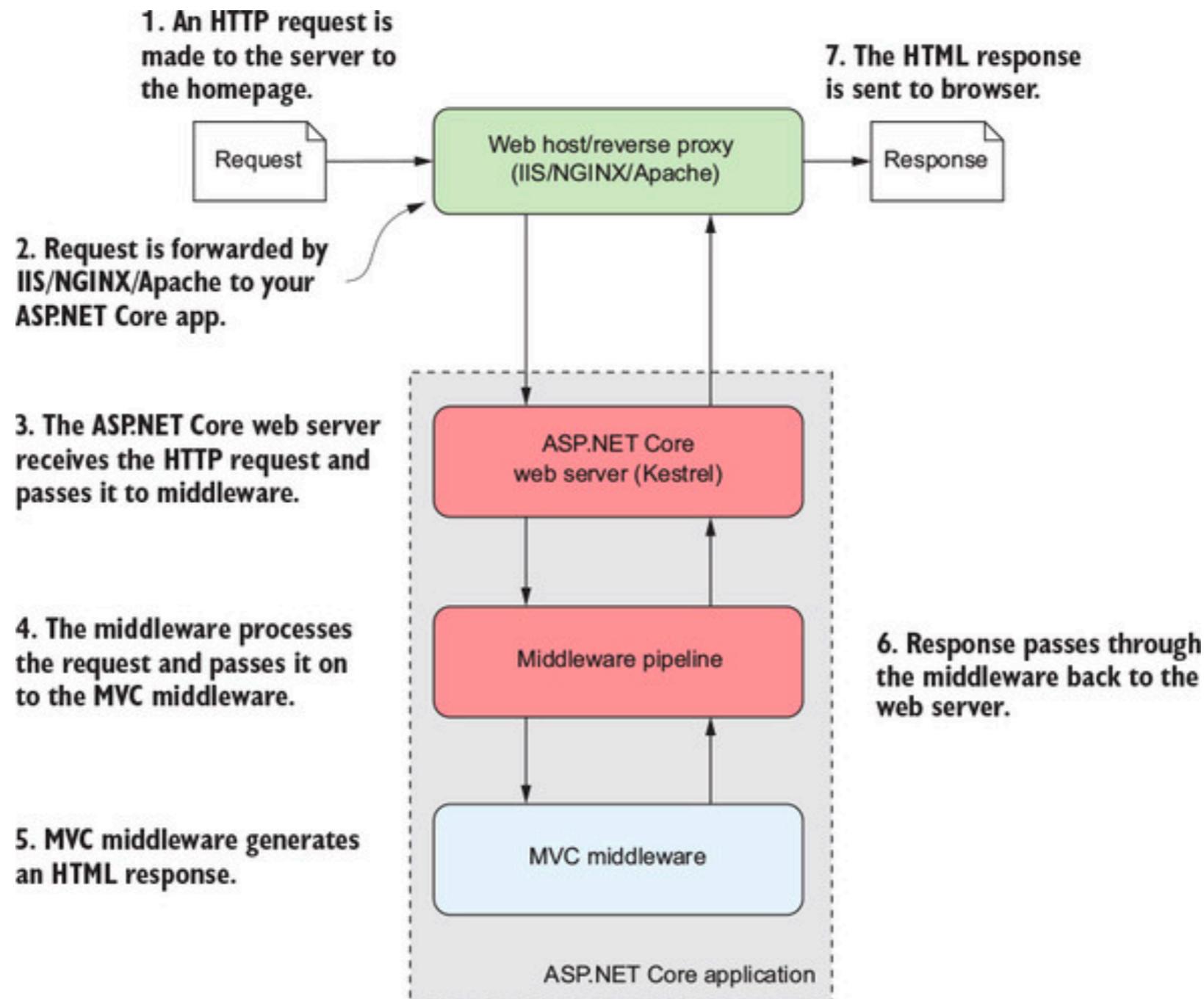




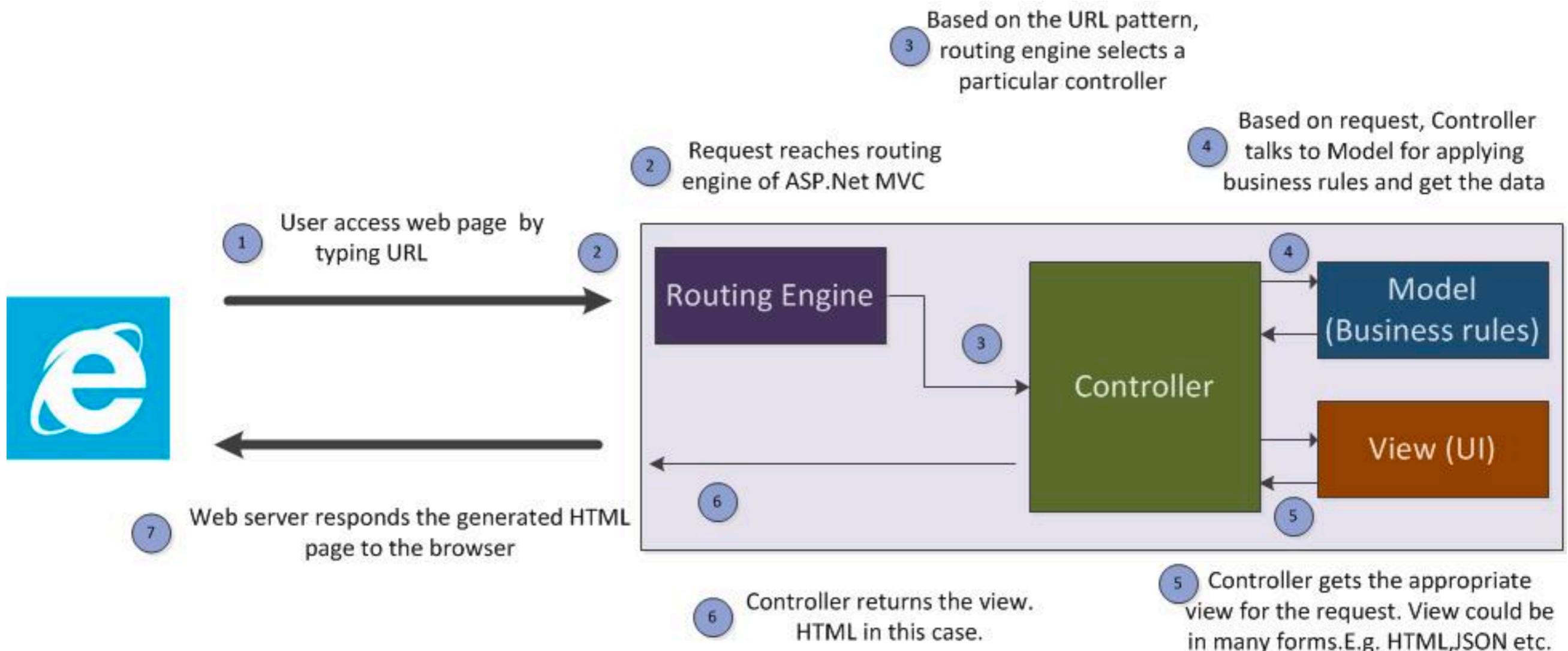
.NET MVC



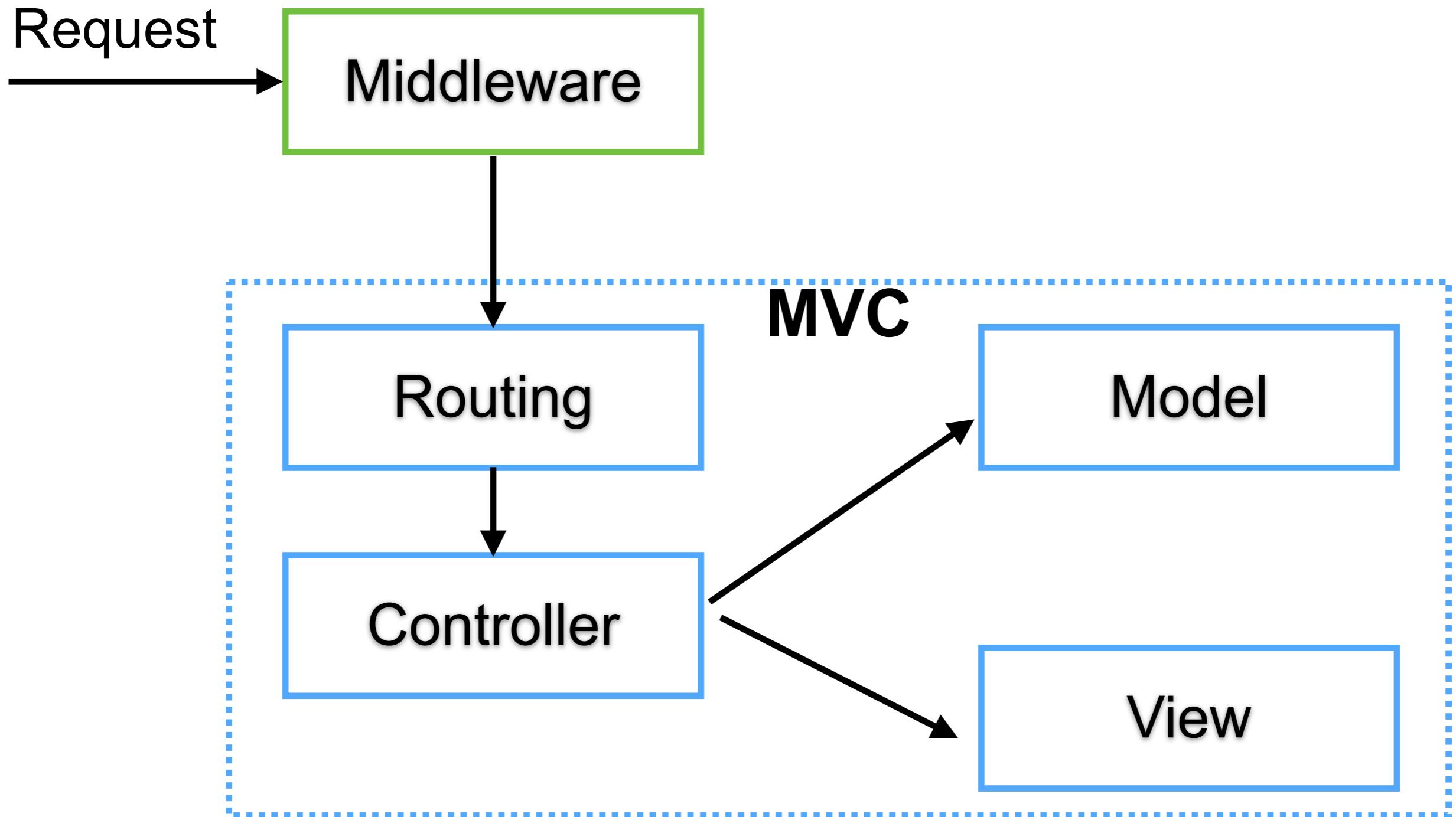
.Net Core MVC



.Net Core MVC



.Net Core MVC

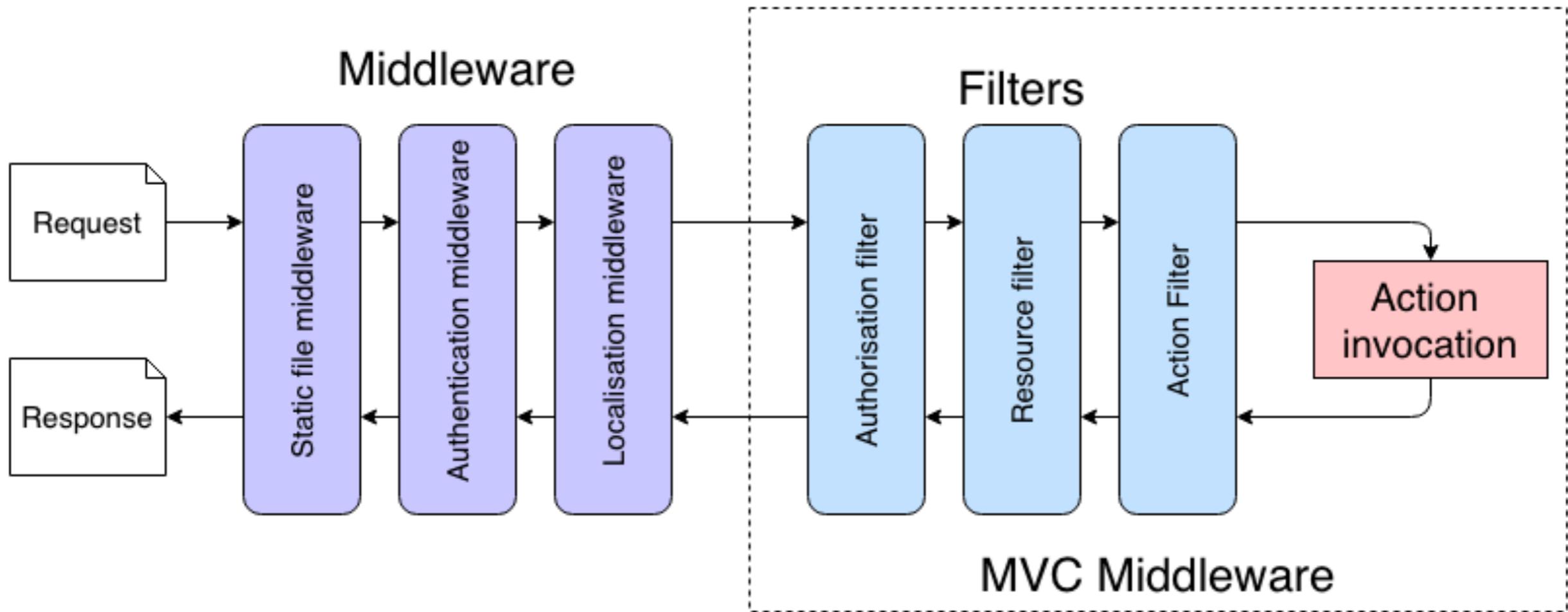


MVC workshop

Project structure
Site configuration

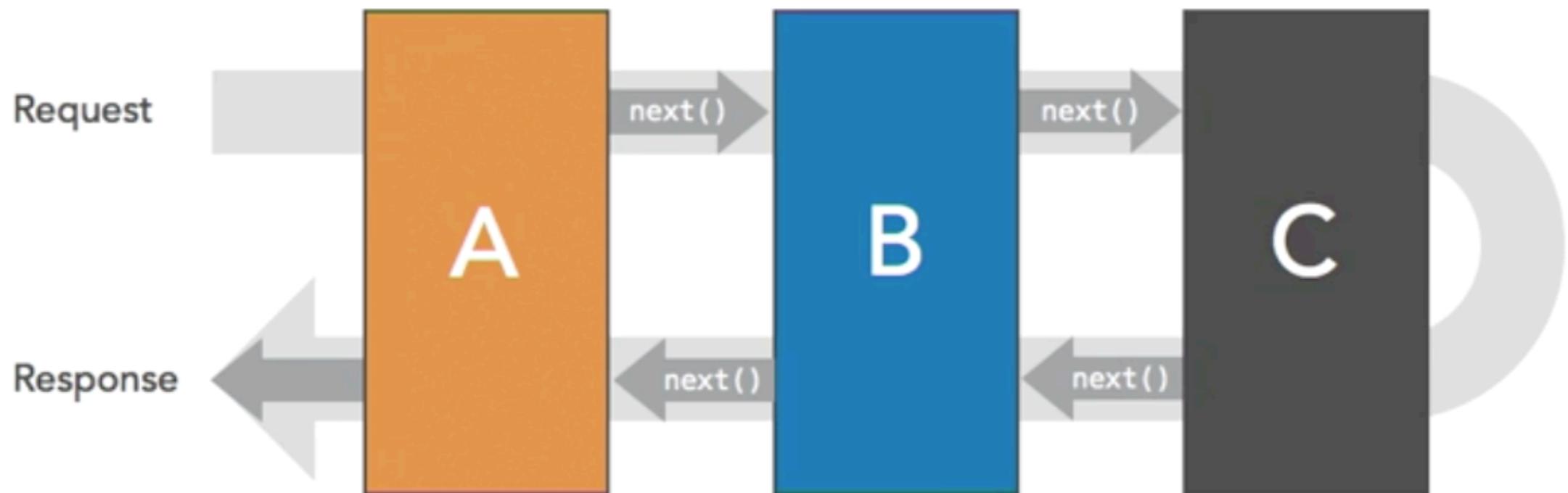


.Net Core MVC



What is Middleware ?

Classes that can handle an HTTP request/response



What is Middleware ?

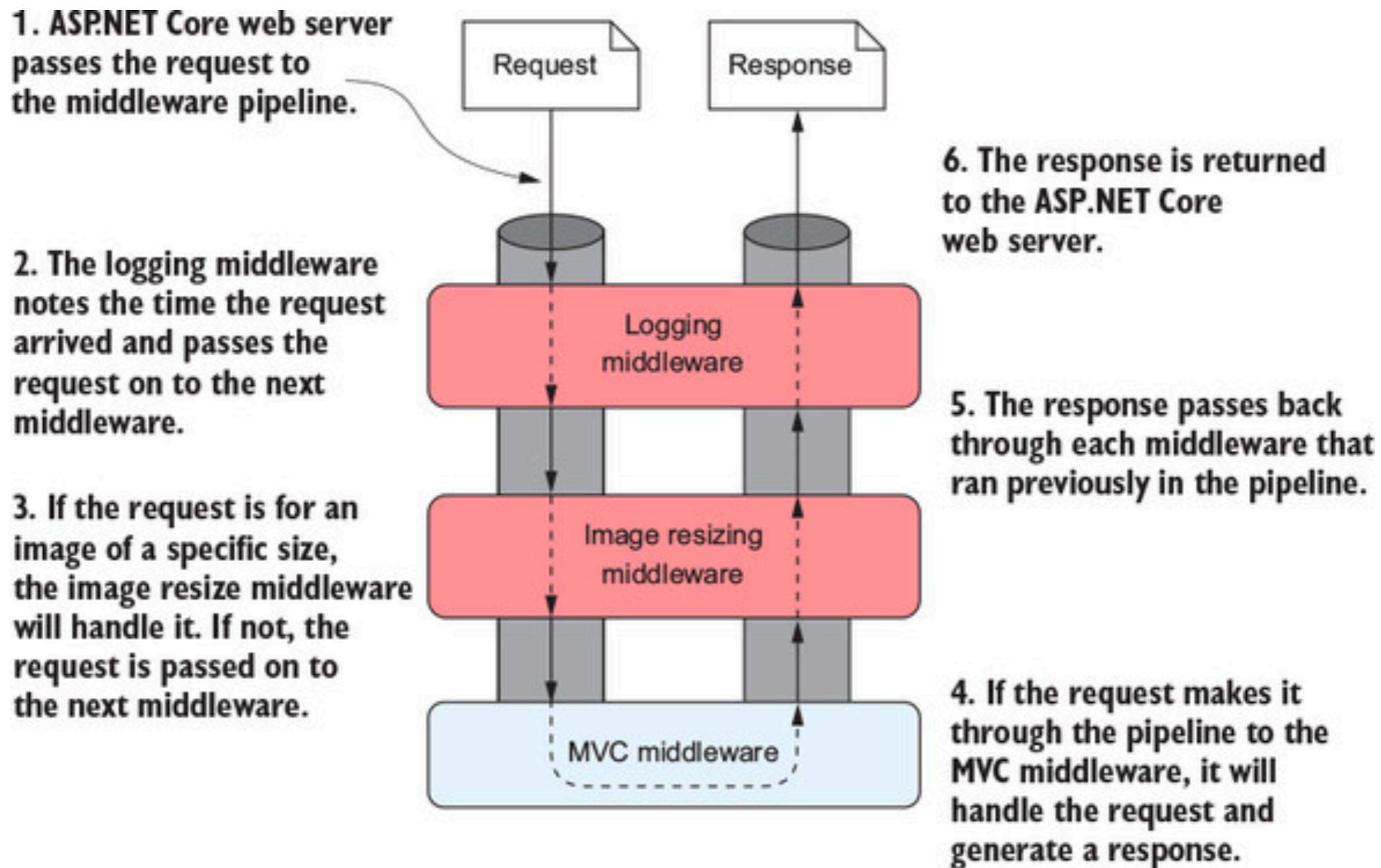
Handle an incoming HTTP request

Process an incoming HTTP request

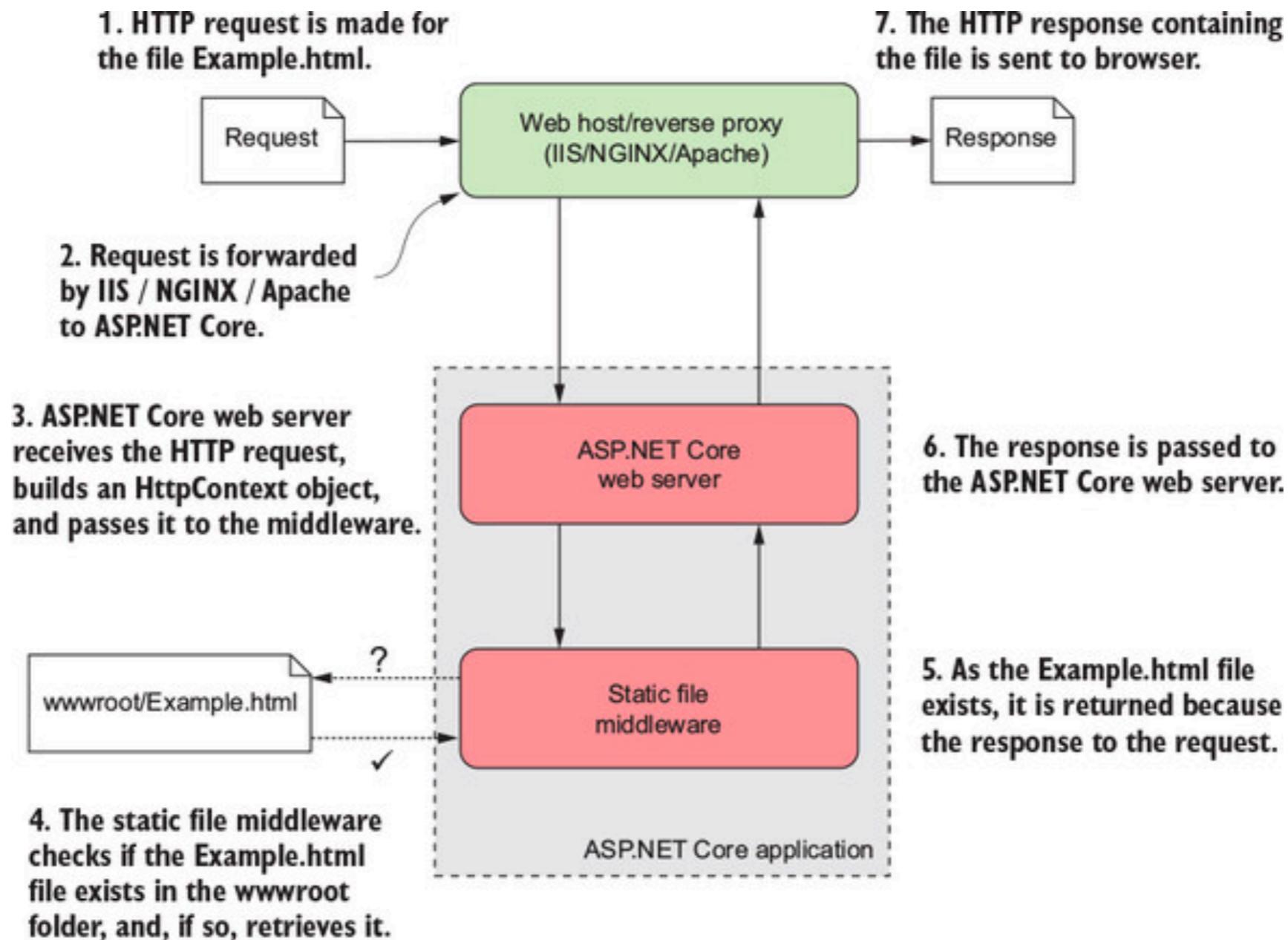
Process an outgoing HTTP response



Middleware pipeline

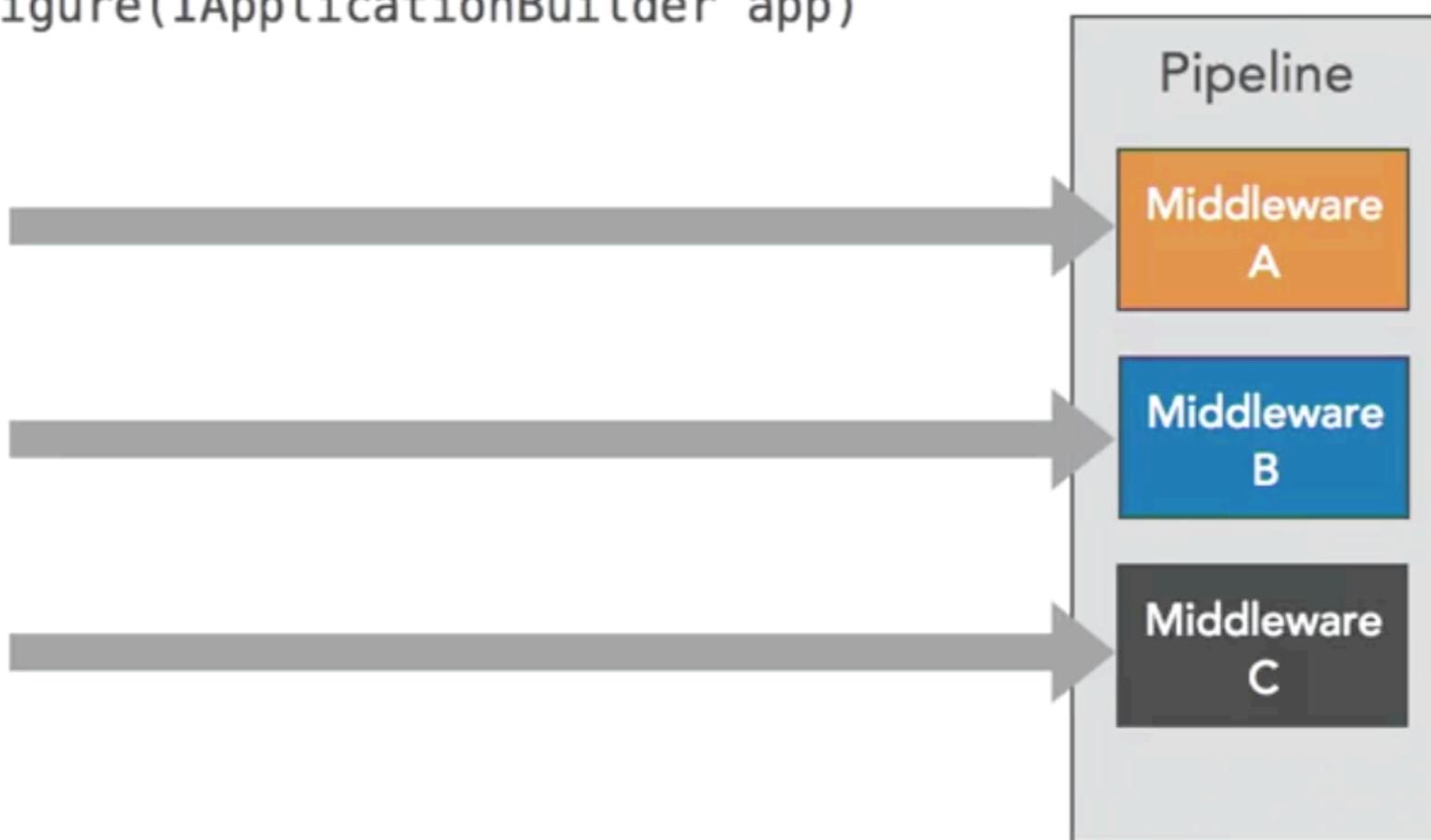


Static file middleware



Middleware

```
public void Configure(IApplicationBuilder app)  
{  
    app.UseA();  
  
    app.UseB();  
  
    app.UseC();  
}
```



Common Middlewares

Logging each request

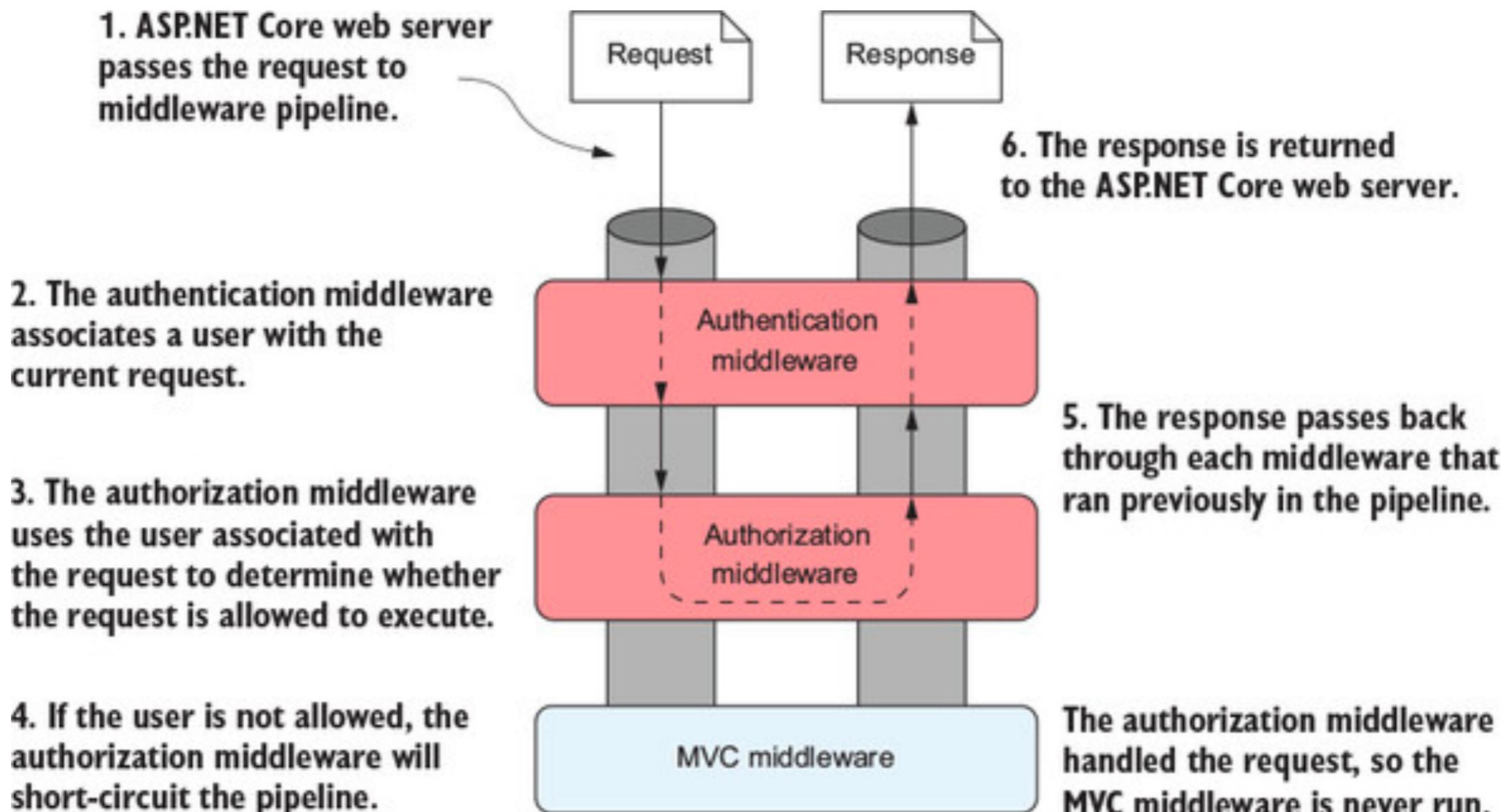
Add standard security headers to response

Associating a request with relevant user

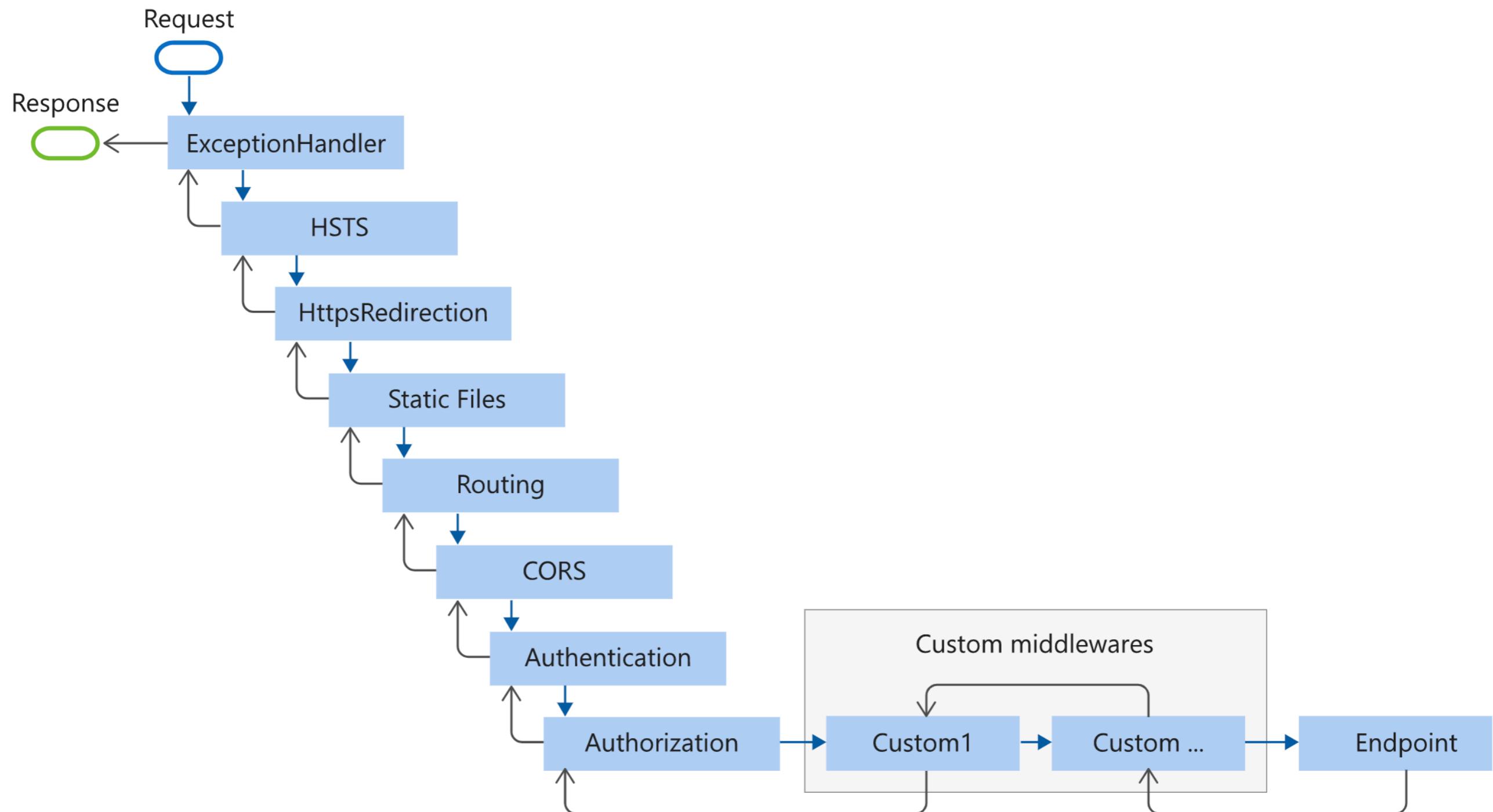
Setting the language for current request



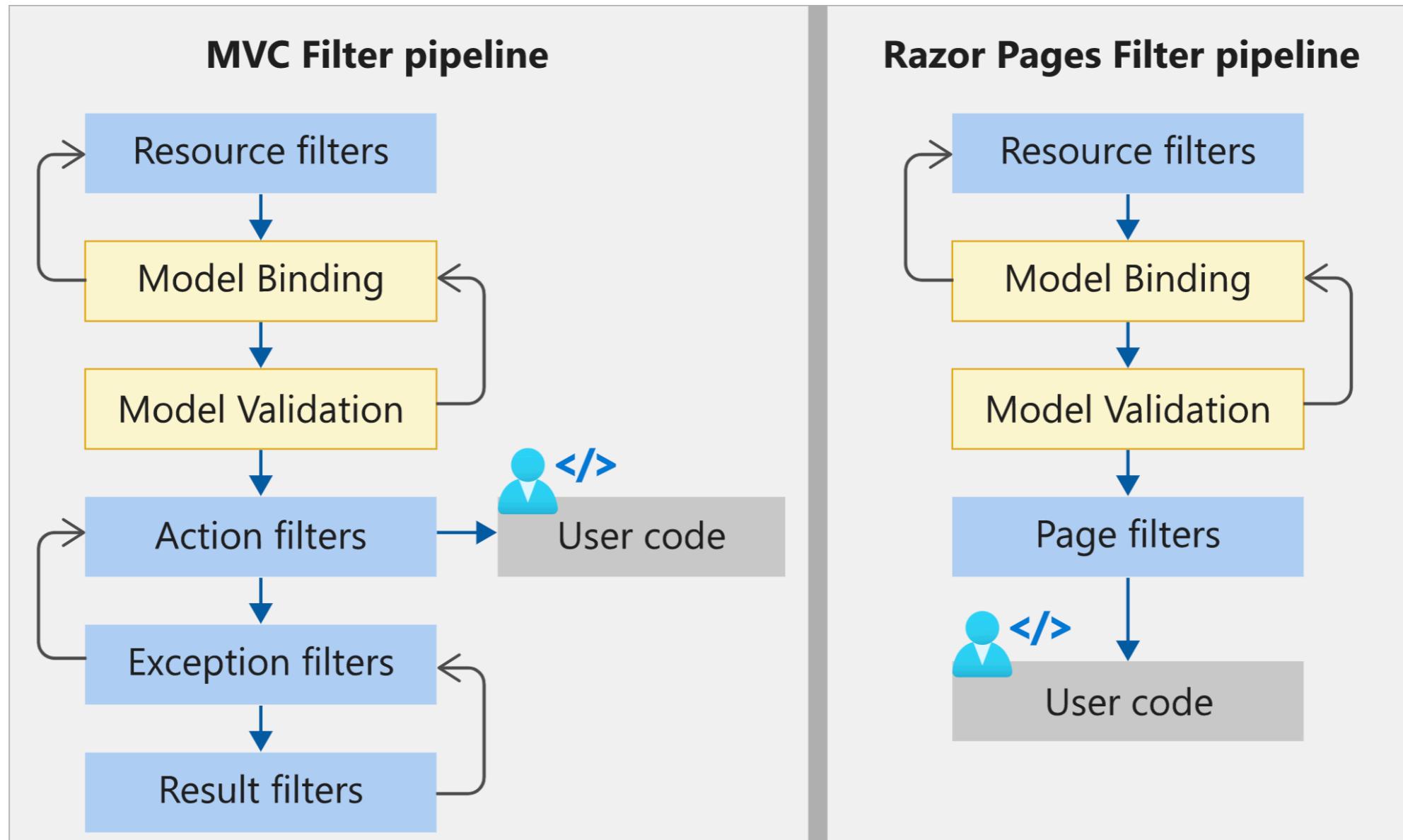
Authenticate and Authorize



Middleware



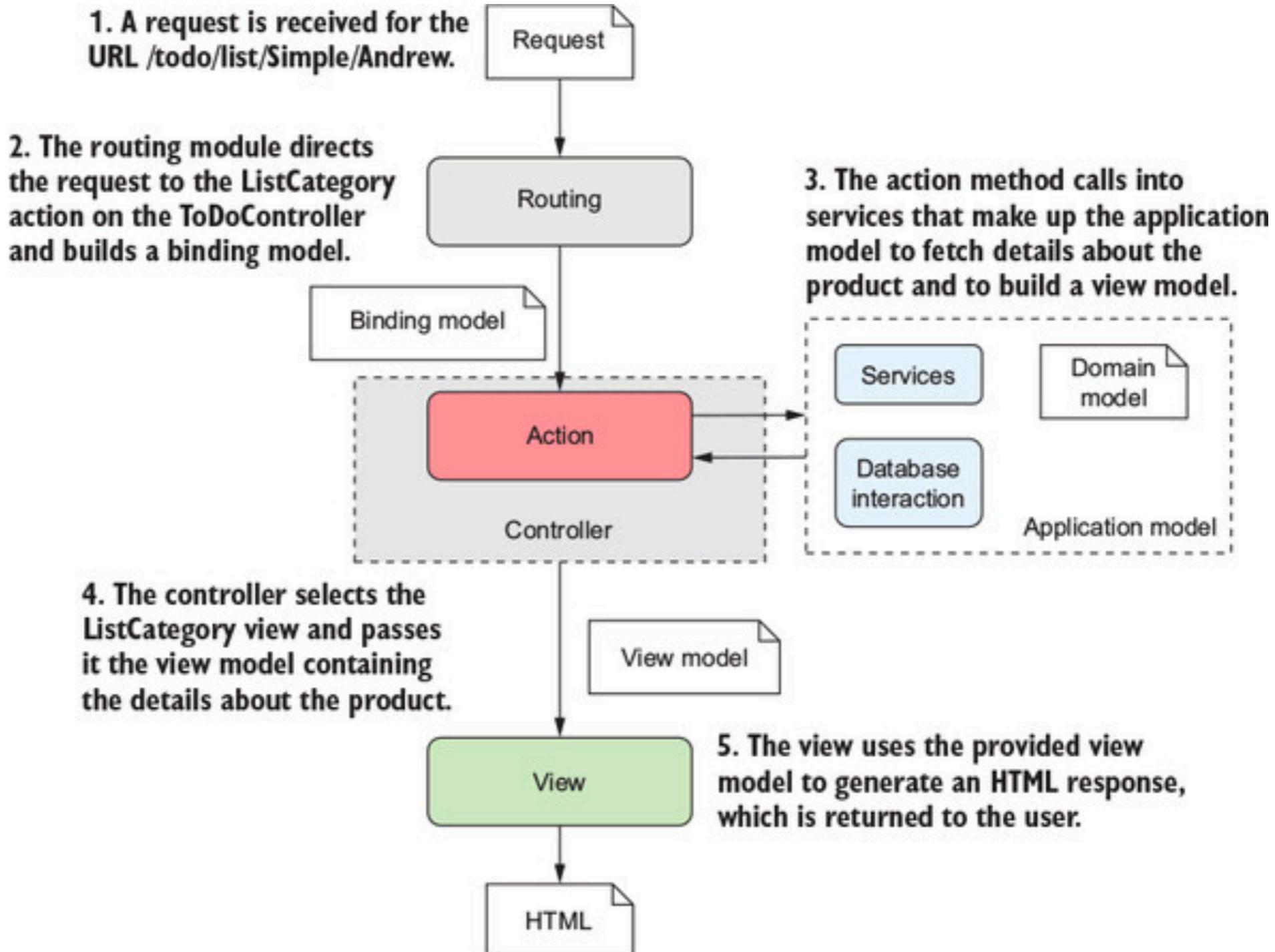
MVC Endpoint



Workshop Middleware

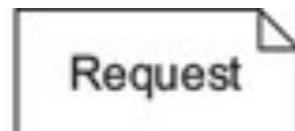


MVC lifecycle



Routing to controller

1. A request is received and handled by the middleware.



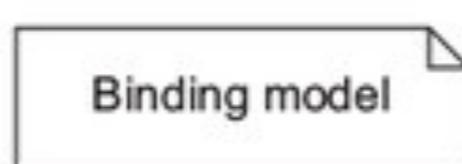
GET URL
/todo/list/Simple/Andrew

2. The routing module directs the request to a specific controller and action.



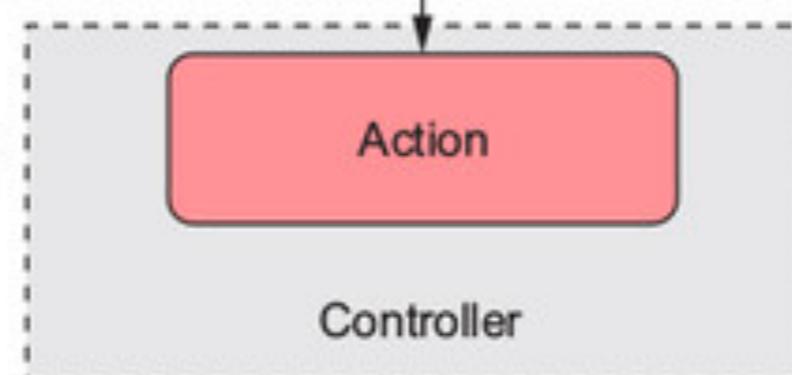
URL mapped to action method
TodoController.ListCategory

3. A binding model is built from the details provided in the request.



model.Category = "Simple"
model.User = "Andrew"

4. The action is passed to the binding model and a method parameter and is executed.



Action method is executed
ListCategory(model)

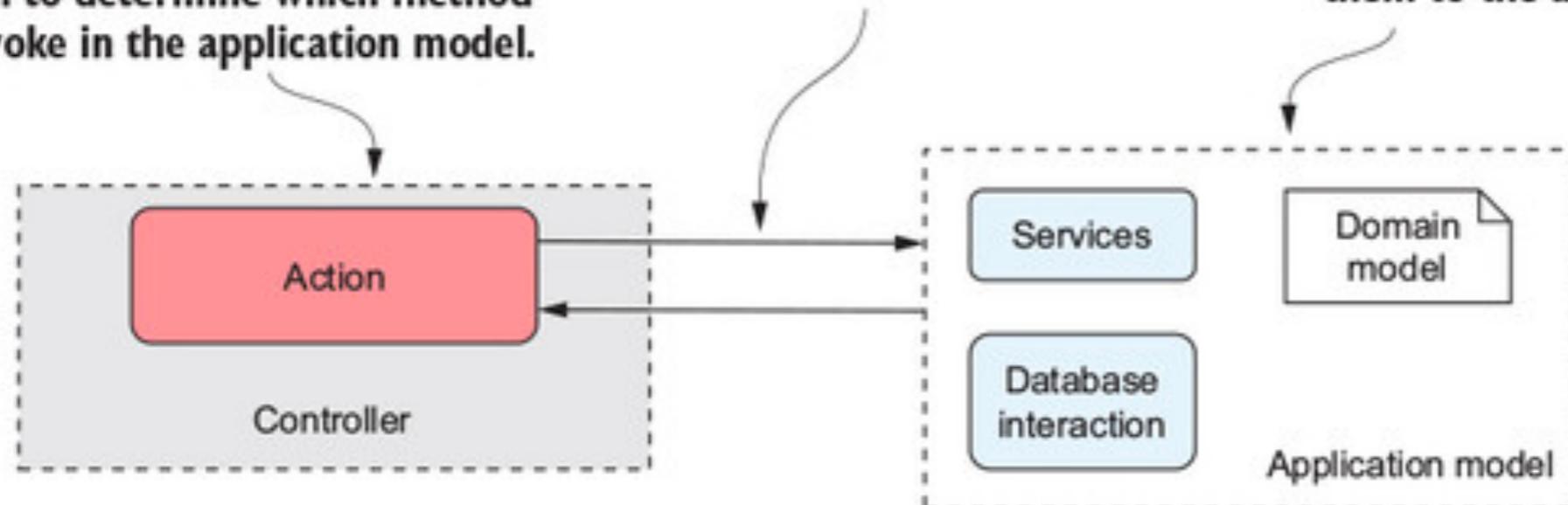


Controller to action

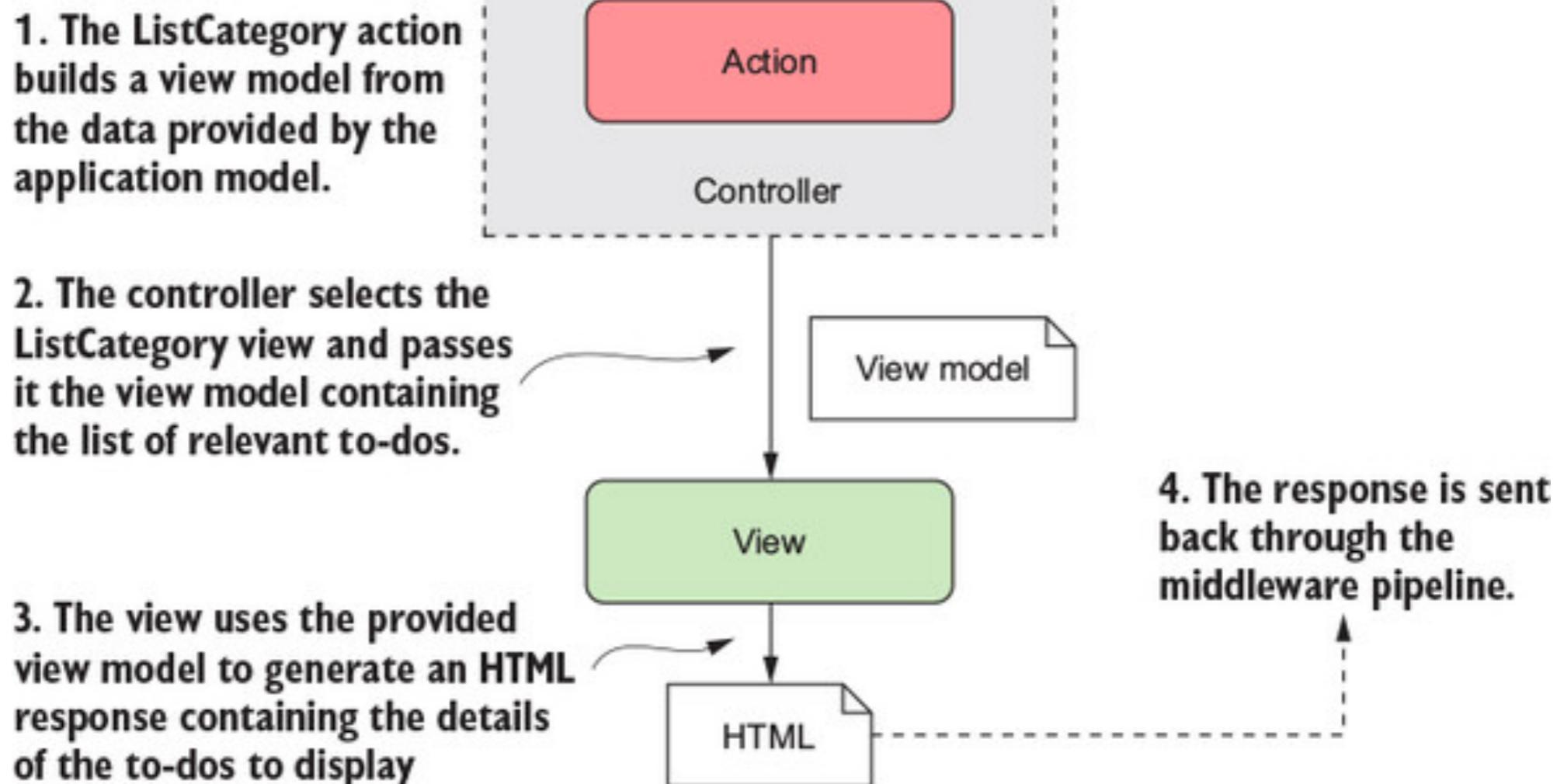
1. The action uses the category and user provided in the binding model to determine which method to invoke in the application model.

2. The action method calls into services that make up the application model. This might use the domain model to calculate the price of the product, for example.

3. The services load the details of the product from the database and return them to the action model.



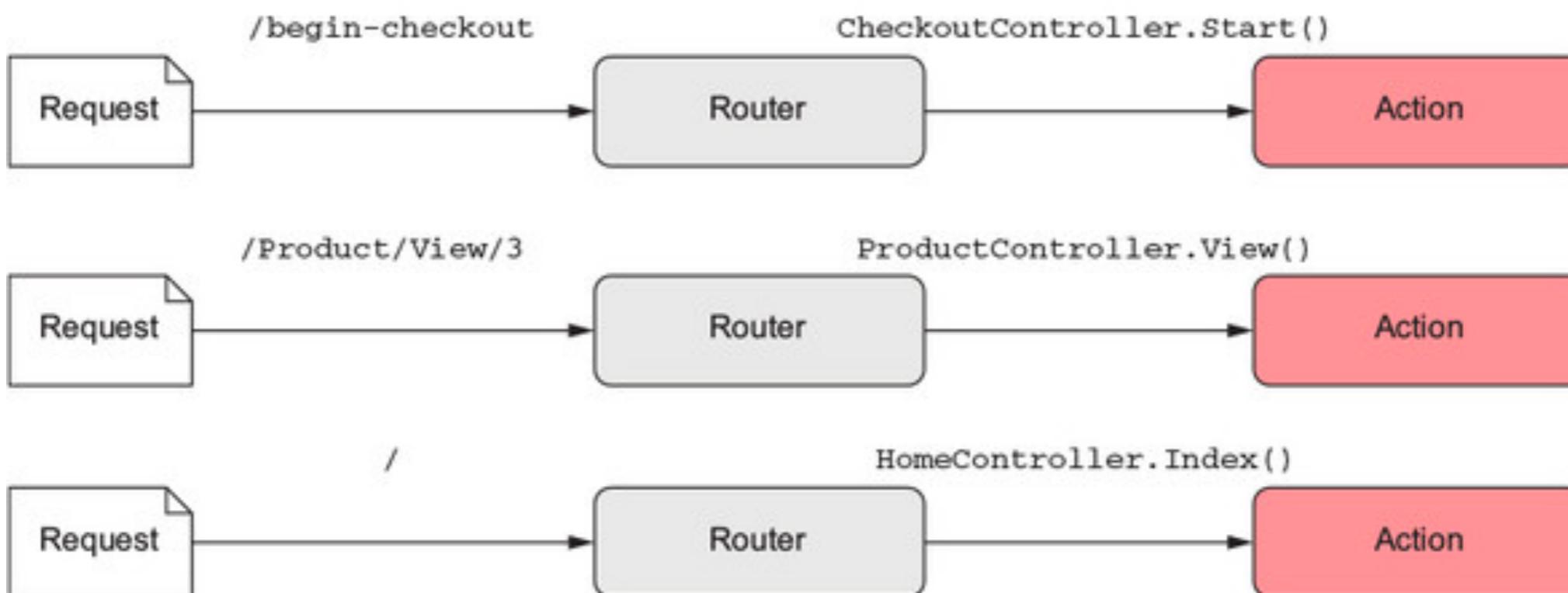
Controller with view



URL routing

Each request is passed to the MVC router, which inspects the request's URL.

The router selects a single action method to execute, based on the incoming URL and the app's configuration.



Workshop

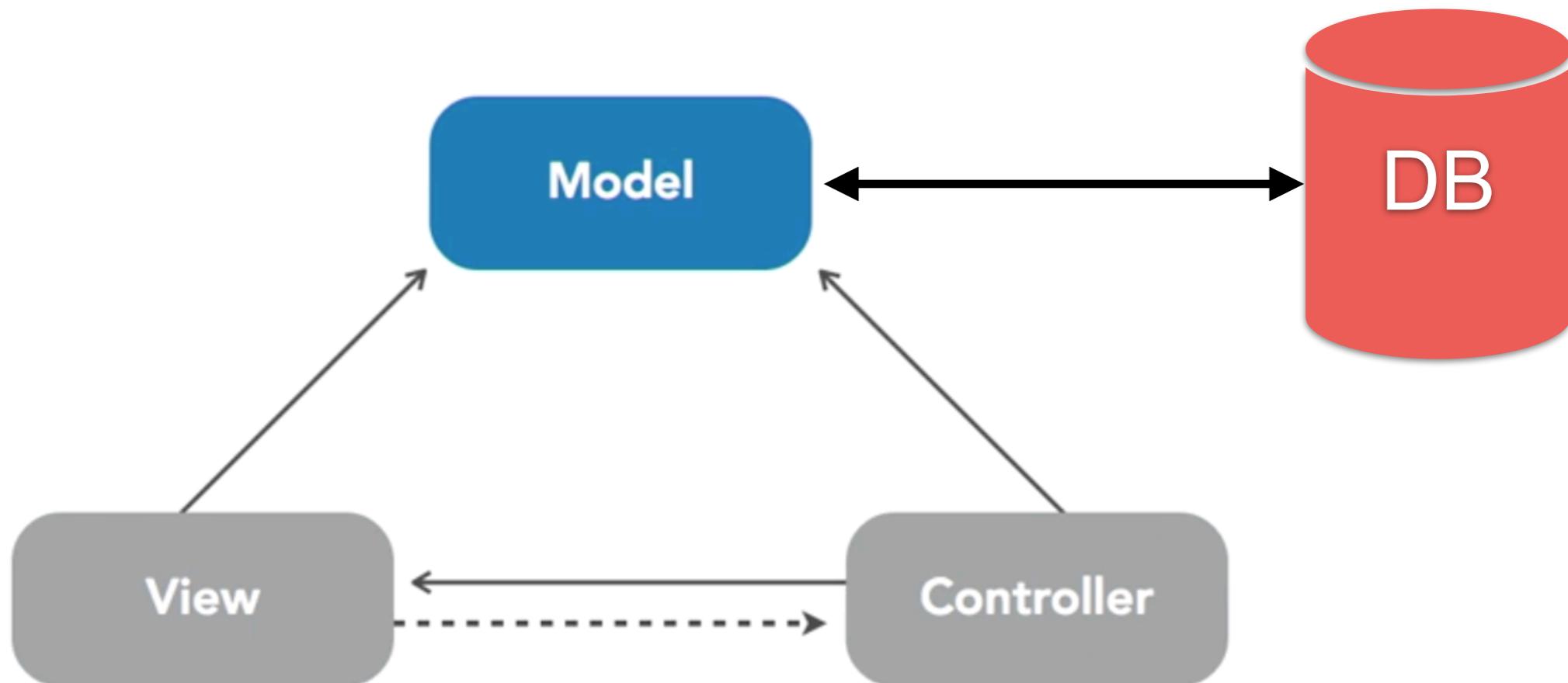
Working with middleware
Handling requests with controllers
Pass parameters to controllers
Customize applications URLs



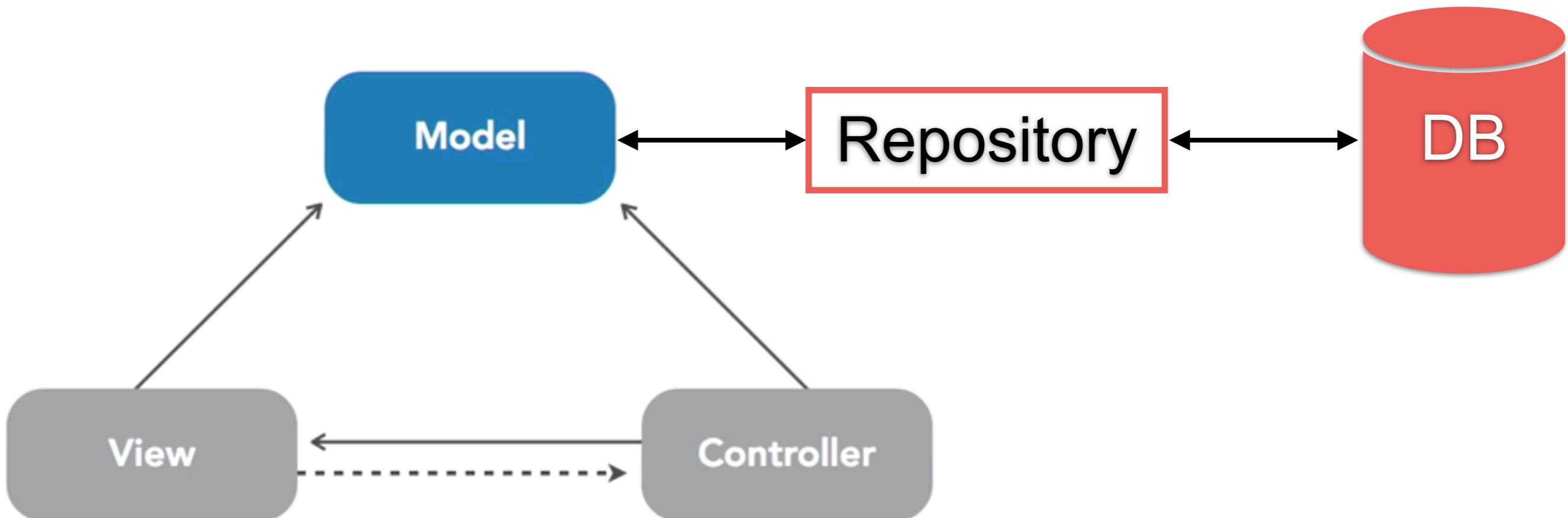
Working with Database



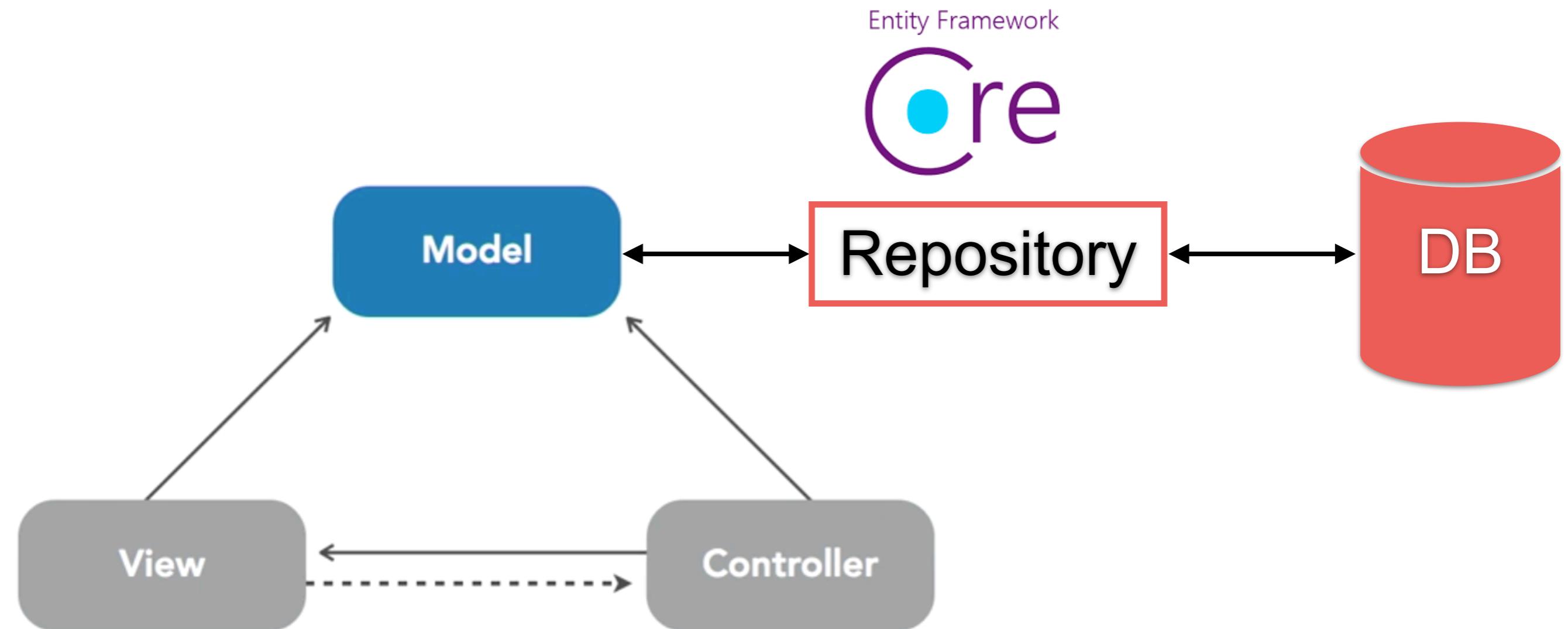
Working with Database



Working with Database

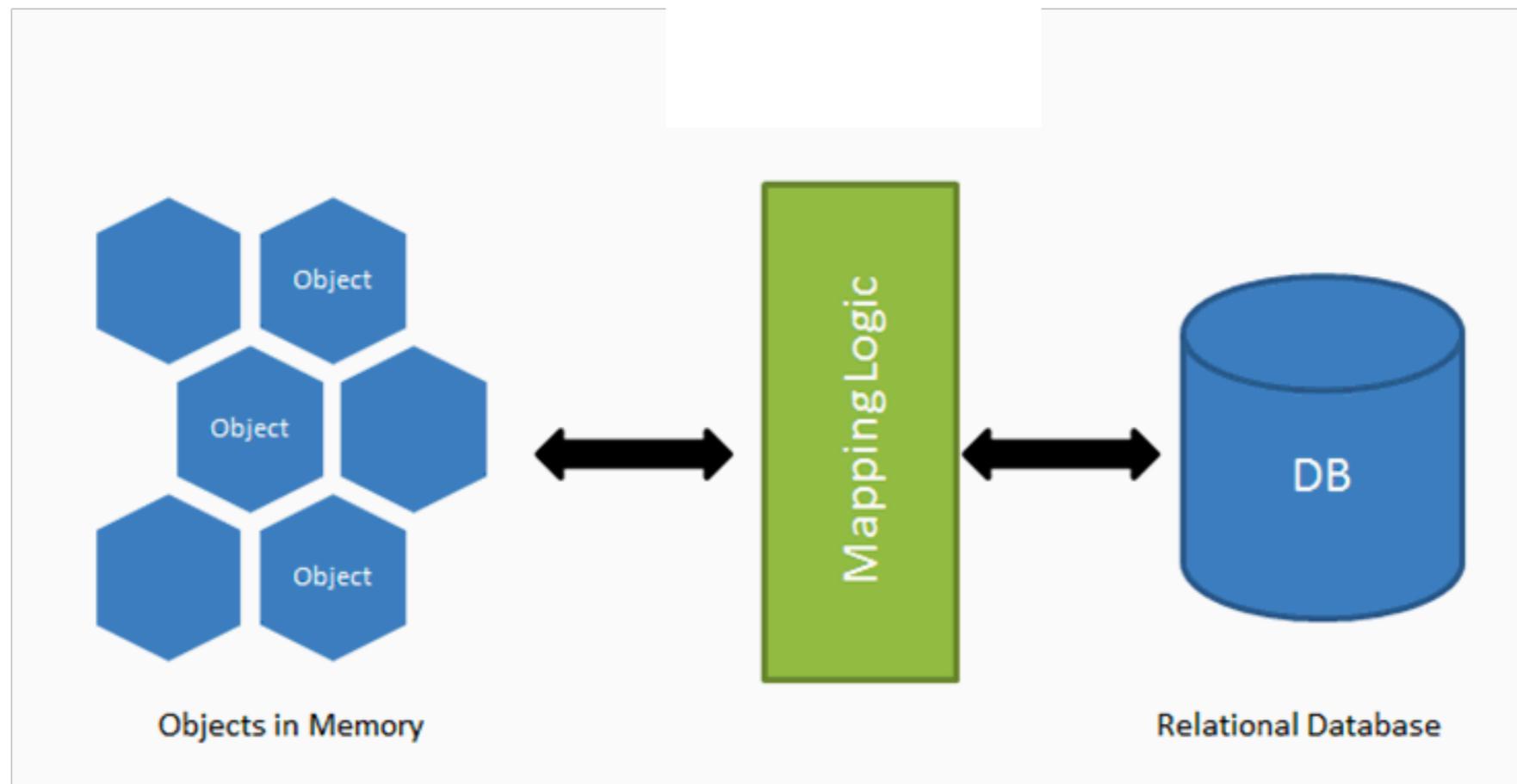


Working with Database



Entity Framework Core (EF Core)

ORM (Object Relational Mapping)



Entity Framework Core (EF Core)

Lightweight and cross-platform
LINQ support
Open-source
Support SQL Server and other
Support non-relational database
Code first



Entity Framework Core (EF Core)



Entity Framework Core (EF Core)

Domain class

Class

```
public class Pie
{
    public int PieId { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
}
```

Table

Field	Type
PieId	Int (PK)
Name	String
Description	string



Add EF Core to application

Domain class

Database context

Application configuration

Package



Table in database

Restaurant
id
name
location



Domain class

```
public class Restaurant
{
    public int Id { get; set; }

    [Required, StringLength(80)]
    public string Name { get; set; }

    [Required, StringLength(255)]
    public string Location { get; set; }
    public CuisineType Cuisine { get; set; }
}
```



Database context

```
public class OdeToFoodDbContext : DbContext
{
    public OdeToFoodDbContext(DbContextOptions<OdeToFoodDbContext> options)
        : base(options)
    {

    }

    public DbSet<Restaurant> Restaurants { get; set; }
}
```



Application configuration

Connection string in AppSettings.json

```
{  
  "Logging": {  
    "LogLevel": {  
      "Default": "Warning"  
    }  
  },  
  "AllowedHosts": "*",  
  "Message": "Hello from appsettings!",  
  "ConnectionStrings": {  
    "OdeToFoodDb": "Data Source=(localdb)\\MSSQLLocalDB;Initial Ca  
  }  
}
```



Application configuration

Add DbContext in Startup.cs

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContextPool<OdeToFoodDbContext>(options =>
    {
        options.UseSqlServer(Configuration.GetConnectionString("OdeToFoodDb"));
    });
}
```



Workshop

CRUD operation

Initial database

Customize applications URLs



Secure your application



Overview

ASP.NET Core Identity
Add authentication
Enable authorization



ASP.NET Core Identity

Membership system
Authentication
Authorization
Support external provider
SQL server support build-in



Adding ASP.NET Core Identity

IdentityDbContext Configuration changes

```
services.Configure<IdentityOptions>(options =>
{
    options.Password.RequireDigit = true;
    options.Password.RequiredLength = 8;
    options.Password.RequireNonAlphanumeric = true;
    options.User.RequireUniqueEmail = true;
});
```



Adding Authentication

Using scaffolding
Support in ASP.NET Core 2.1, 3.0



Step

Using Razor class library
Reuse functionality
(views, pages, controllers view components)
Can be overridden



Allow user to login

Use build-in RCL

Scaffold items

Layout changes



Workshop



Enable Authorization

Add authorize attribute in controller

```
[Authorize]
public class HomeController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
}
```



Enable Authorization

In Action level

```
public class HomeController : Controller
{
    [Authorize]
    public IActionResult Index()
    {
        return View();
    }
}
```



Enable Authorization

Working with roles

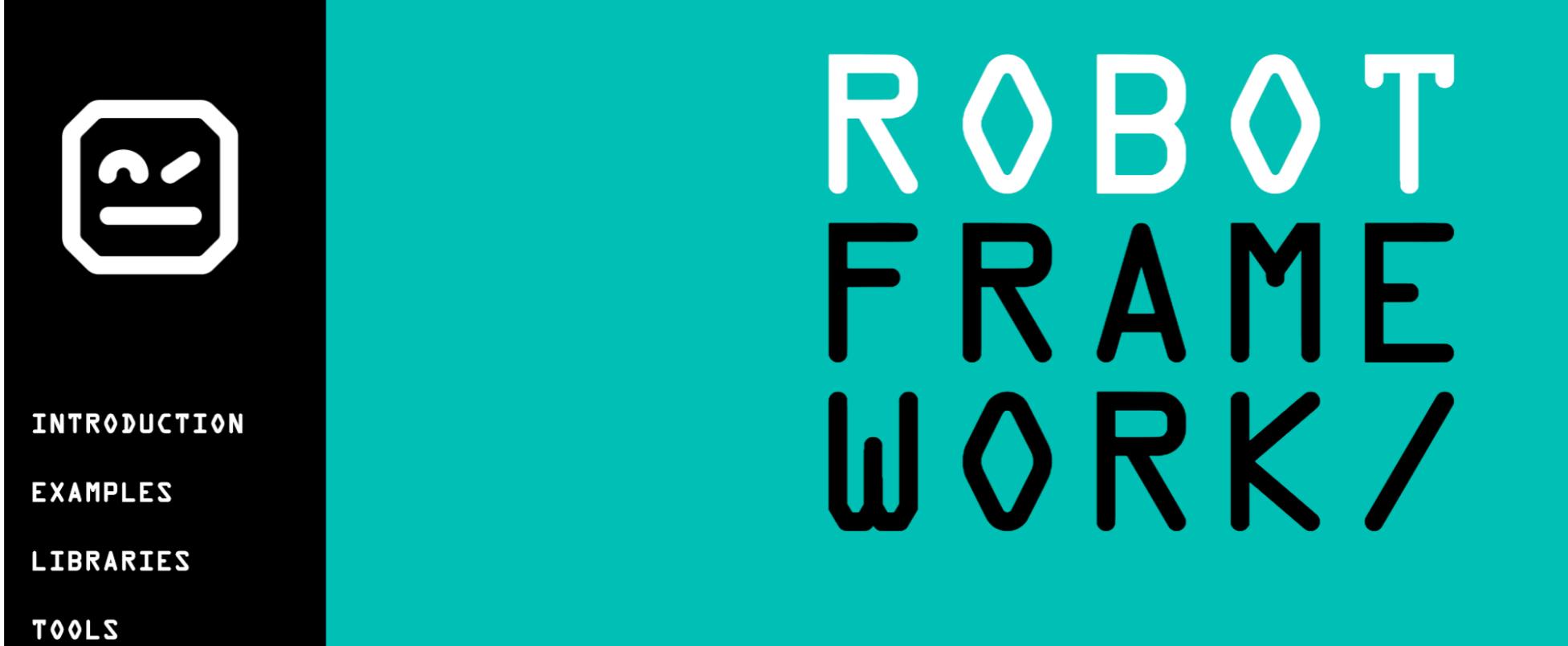
```
[Authorize(Roles ="Administrator")]
public class HomeController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
}
```



Web/UI Testing



UI testing with Robot framework



The screenshot shows the Robot Framework website. On the left, there's a sidebar with a logo of a robot head and links: INTRODUCTION, EXAMPLES, LIBRARIES, TOOLS, DOCUMENTATION, SUPPORT, FOUNDATION, SHOP, and ROBOCON. The main content area has a teal background with the text "ROBOT FRAMEWORK/" in large, bold, white and black letters. Below this, a horizontal line separates the header from the "INTRODUCTION" section, which is also in teal. To the right of the introduction, a block of text describes the framework.

INTRODUCTION

Robot Framework is a generic test automation framework for acceptance testing and acceptance test-driven development (ATDD). It has easy-to-use tabular test data

<http://robotframework.org/>



Software installation

Python 3

Pip (package manager)

Robot framework and library

Text Editor with VS Code



Python 3

The screenshot shows the Python.org homepage. At the top, there's a navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. Below this is a search bar with a magnifying glass icon and buttons for 'GO', 'Socialize', and 'Sign In'. A sidebar on the left contains a snippet of Python code demonstrating basic operations like printing and user input. The main content area is titled 'Download for Mac OS X' and offers links for Python 3.6.0 and Python 2.7.13. It also mentions that Python can be used on many operating systems and provides a link to view the full list of downloads.

Python 3: Si
>>> print("Hello, I'm Pyt
Hello, I'm Pyt

Input, assig
>>> name = inp
>>> print('Hi,
What is your n
Python
Hi, Python.

All releases
Source code
Windows
Mac OS X
Other Platforms
License
Alternative Implementations

Download for Mac OS X

Python 3.6.0 Python 2.7.13

Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. [»» Learn More](#)

<https://www.python.org/>



.Net Core

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

Install robot framework

pip install robotframework

pip install robotframework-seleniumlibrary



Testing robot framework

\$robot



Text Editor with VS Code

Visual Studio Code Docs Updates Blog API Extensions FAQ Search Docs Download

Version 1.45 is now available! Read about the new features and fixes from April.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

[Download for Mac](#) Stable Build

[Other platforms and Insiders Edition](#)

By using VS Code, you agree to its [license](#) and [privacy statement](#).

The screenshot shows the Visual Studio Code interface. On the left, there's a sidebar with icons for file operations like Open, Save, Find, and Share. The main area displays a code editor with three tabs: 'blog-post.js', 'index.js', and 'utils.js'. The 'blog-post.js' tab is active, showing code related to a blog post component. Below the code editor is a 'PROBLEMS' panel showing compilation logs. At the bottom, there's a status bar with information like 'master', 'Gatsby Develop (gatsby-graphql-app)', and file statistics ('Ln 6, Col 21 Spaces: 2 UTF-8 LF JavaScript'). On the right side of the interface, there's a large window titled 'EXTENSIONS: MARKETPLACE' showing a list of installed extensions, including Python, GitLens, C/C++, ESLint, Debugger for Chrome, Language Support, vscode-icons, and Vetur. Each extension entry includes a star rating, file size, and an 'Install' button if it's not yet installed.

<https://code.visualstudio.com/>



Install extension

Visual Studio Code > Programming Languages > Robot Framework Intellisense

The screenshot shows the extension details page for 'Robot Framework Intellisense'. It features a large circular icon with a stylized blue 'R' inside a black square. The title 'Robot Framework Intellisense' is displayed in bold. Below it, the author 'Tomi Turtiainen' is listed, along with the download count '64,252 installs' and a five-star rating '(7)'. A green 'Install' button is prominent, followed by a link 'Trouble Installing?'. At the bottom, there are tabs for 'Overview' (which is underlined), 'Q & A', and 'Rating & Review'.

Robot Framework Intellisense

Tomi Turtiainen | 64,252 installs | ★★★★★ (7) | Free

Robot Framework Intellisense

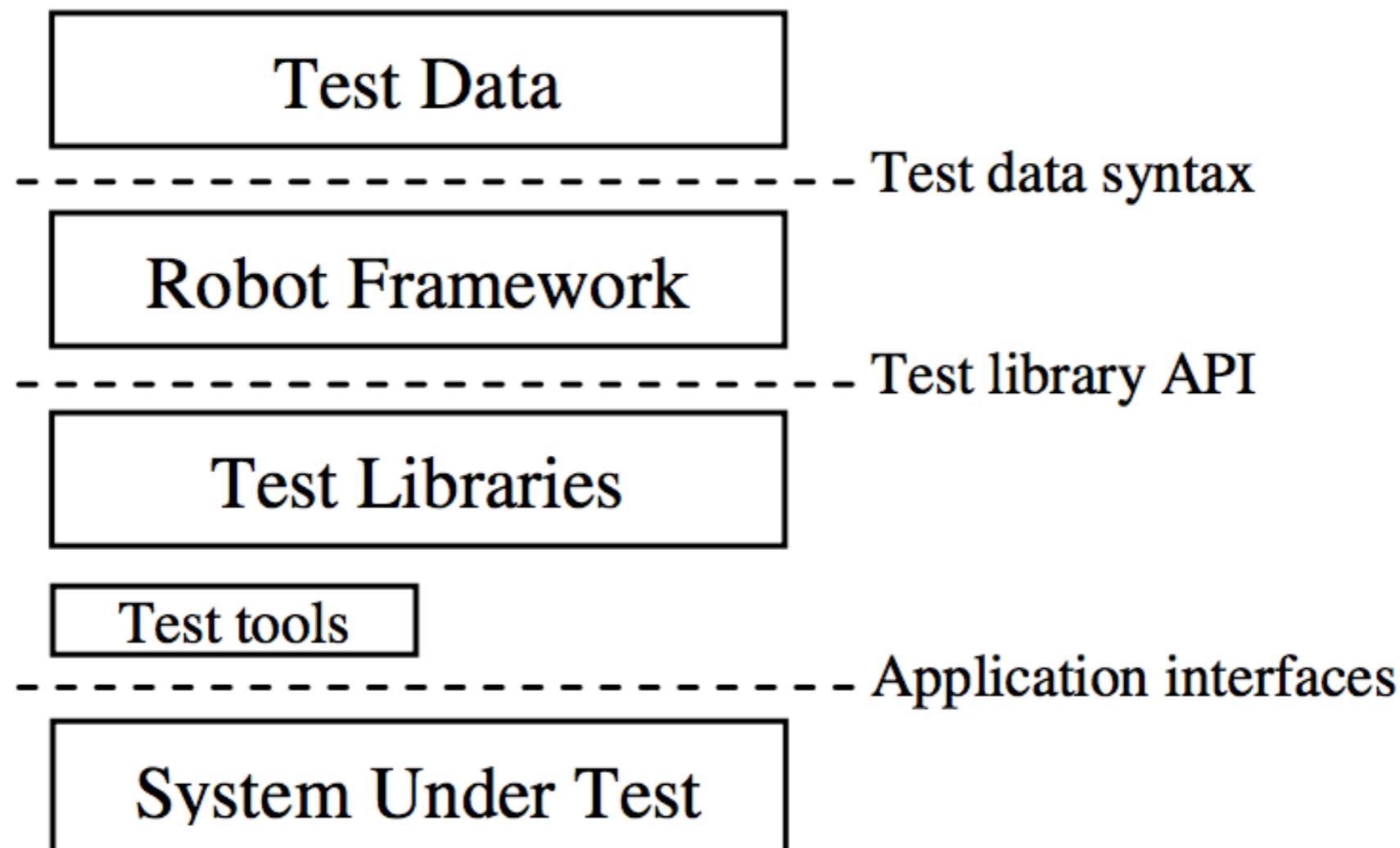
Install Trouble Installing?

Overview Q & A Rating & Review

<https://marketplace.visualstudio.com/items?itemName=TomiTurtiainen.rf-intellisense>

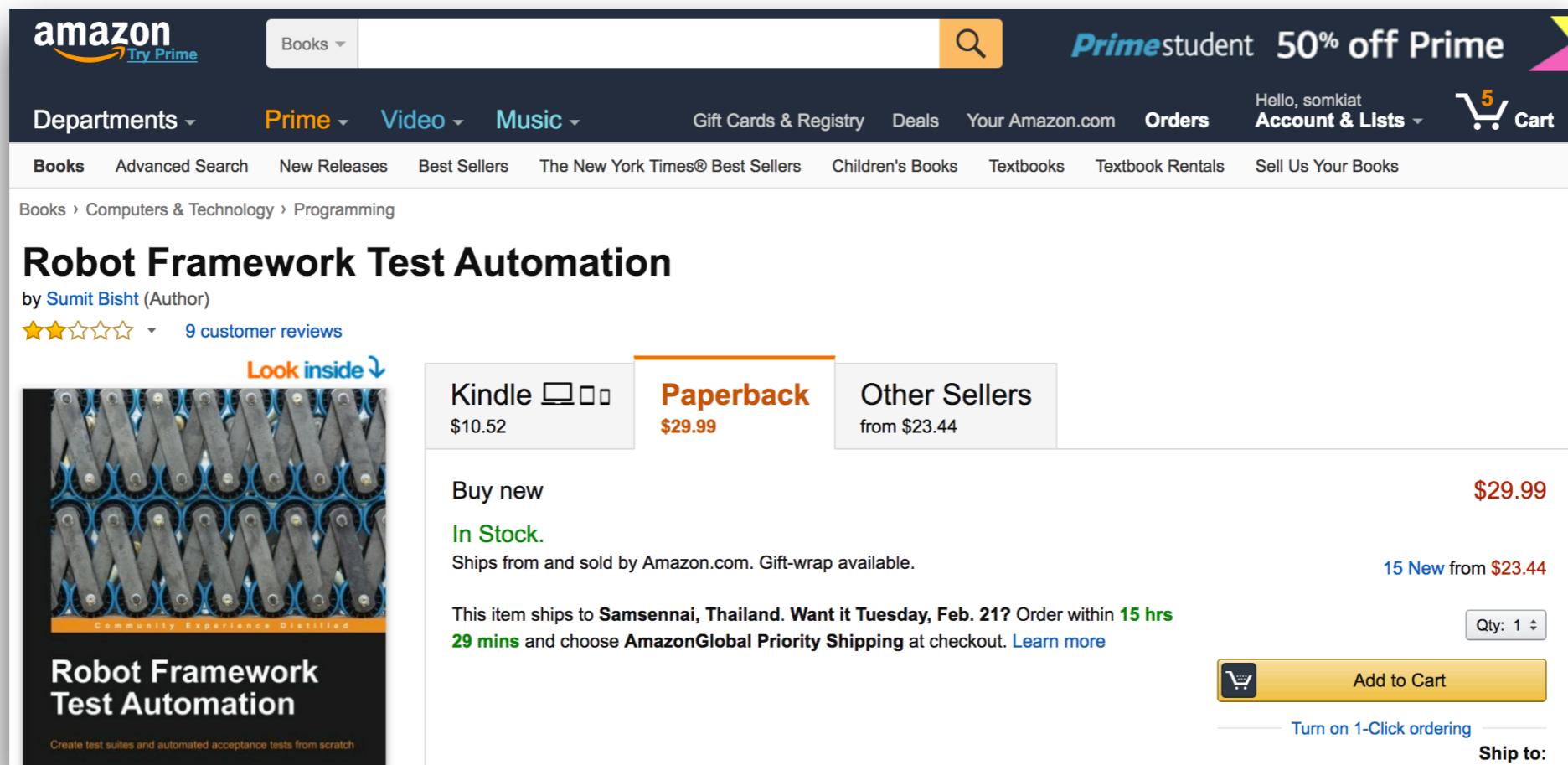


Robot Framework architecture

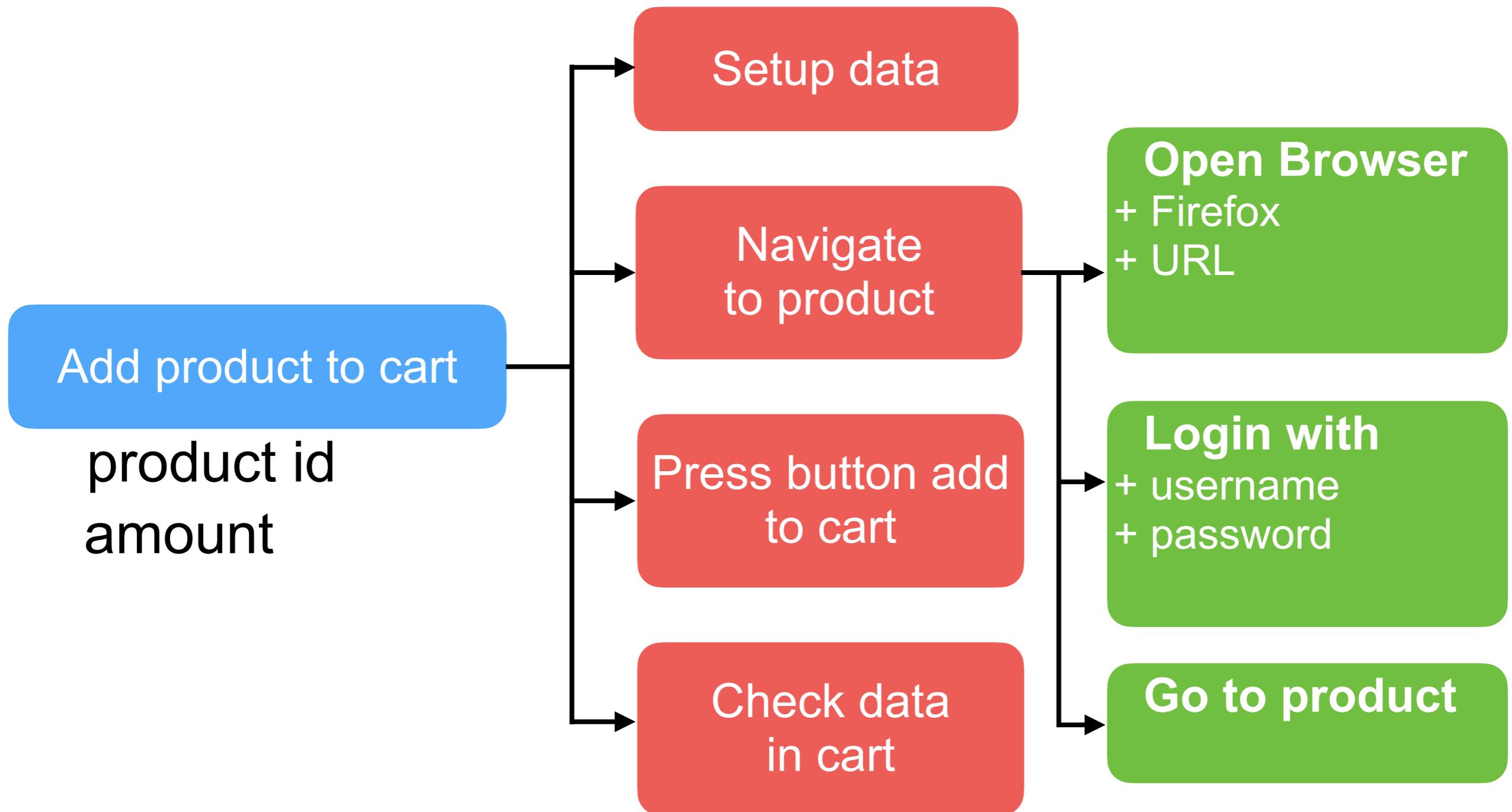


Keyword-driven development

“Add product to cart”



Add product to cart



Working with web browser



Working with Web Browser

Robot Framework IDE/Editor



Robot Framework

Build-in
library

Selenium
library

Faker
library

Python

PIP



Web Driver (ChromeDriver, GeckoDriver, IEDriver)



Web Browser (Google Chrome, Firefox, IE)



SeleniumLibrary keywords

Checkbox Should Be Selected · **C**heckbox Should Not Be Selected · **C**hoose File · **C**lear Element Text · **C**lick Button · **C**lick Element · **C**lick Element At Coordinates · **C**lick Image · **C**lick Link · **C**lose All Browsers · **C**lose Browser · **C**lose Window · **C**over Element · **C**reate Webdriver · **C**urrent Frame Should Contain · **C**urrent Frame Should Not Contain · **D**elete All Cookies · **D**elete Cookie · **D**ouble Click Element · **D**rag And Drop · **D**rag And Drop By Offset · **E**lement Attribute Value Should Be · **E**lement Should Be Disabled · **E**lement Should Be Enabled · **E**lement Should Be Focused · **E**lement Should Be Visible · **E**lement Should Contain · **E**lement Should Not Be Visible · **E**lement Should Not Contain · **E**lement Text Should Be · **E**lement Text Should Not Be · **E**xecute Async Javascript · **E**xecute Javascript · **F**rame Should Contain · **G**et All Links · **G**et Browser Aliases · **G**et Browser Ids · **G**et Cookie · **G**et Cookies · **G**et Element Attribute · **G**et Element Count · **G**et Element Size · **G**et Horizontal Position · **G**et List Items · **G**et Location · **G**et Locations · **G**et Selected List Label · **G**et Selected List Labels · **G**et Selected List Value · **G**et Selected List Values · **G**et Selenium Implicit Wait · **G**et Selenium Speed · **G**et Selenium Timeout · **G**et Session Id · **G**et Source · **G**et Table Cell · **G**et Text · **G**et Title · **G**et Value · **G**et Vertical Position · **G**et WebElement · **G**et WebElements · **G**et Window Handles · **G**et Window Identifiers · **G**et Window Names · **G**et Window Position · **G**et Window Size · **G**et Window Titles · **G**o Back · **G**o To · **H**andle Alert · **I**nput Password · **I**nput Text · **I**nput Text Into Alert · **L**ist Selection Should Be · **L**ist Should Have No Selections · **L**ocation Should Be · **L**ocation Should Contain · **L**ocator Should Match X Times · **L**og Location · **L**og Source · **L**og Title · **M**aximize Browser Window · **M**ouse Down · **M**ouse Down On Image · **M**ouse Down On Link · **M**ouse Out · **M**ouse Over · **M**ouse Up · **O**pen Browser · **O**pen Context Menu · **P**age Should Contain · **P**age Should Contain Button · **P**age Should Contain Checkbox · **P**age Should Contain Element · **P**age Should Contain Image · **P**age Should Contain Link · **P**age Should Contain List · **P**age Should Contain Radio Button · **P**age Should Contain Textfield · **P**age Should Not Contain · **P**age Should Not Contain Button · **P**age Should Not Contain Checkbox · **P**age Should Not Contain Element · **P**age Should Not Contain Image · **P**age Should Not Contain Link · **P**age Should Not Contain List · **P**age Should Not Contain Radio Button · **P**age Should Not Contain Textfield · **P**ress Key · **P**ress Keys · **R**adio Button Should Be Set To · **R**adio Button Should Not Be Selected · **R**egister Keyword To Run On Failure · **R**eload Page · **R**emove Location Strategy · **S**croll Element Into View · **S**elect All From List · **S**elect Checkbox · **S**elect Frame · **S**elect From List By Index · **S**elect From List By Label · **S**elect From List By Value · **S**elect Radio Button · **S**elect Window · **S**et Browser Implicit Wait · **S**et Focus To Element · **S**et Screenshot Directory · **S**et Selenium Implicit Wait · **S**et Selenium Speed · **S**et Selenium Timeout · **S**et Window Position · **S**et Window Size · **S**imulate Event · **S**ubmit Form · **S**witch Browser · **S**witch Window · **T**able Cell Should Contain · **T**able Column Should Contain · **T**able Footer Should Contain · **T**able Header Should Contain · **T**able Row Should Contain · **T**able Should Contain · **T**extarea Should Contain · **T**extarea Value Should Be · **T**extfield Should Contain · **T**extfield Value Should Be · **T**itle Should Be · **U**nselect All From List · **U**nselect Checkbox · **U**nselect Frame · **U**nselect From List By Index · **U**nselect From List By Label · **U**nselect From List By Value · **W**ait For Condition · **W**ait Until Element Contains · **W**ait Until Element Does Not Contain · **W**ait Until Element Is Enabled · **W**ait Until Element Is Not Visible · **W**ait Until Element Is Visible · **W**ait Until Location Contains · **W**ait Until Location Is · **W**ait Until Page Contains · **W**ait Until Page Contains Element · **W**ait Until Page Does Not Contain · **W**ait Until Page Does Not Contain Element

<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>



.Net Core

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

Open Browser

Browser	Name(s)
Firefox	firefox, ff
Google Chrome	googlechrome, chrome, gc
Headless Firefox	headlessfirefox
Headless Chrome	headlesschrome
Internet Explorer	internetexplorer, ie
Edge	edge
Safari	safari
Opera	opera
Android	android
Iphone	iphone
PhantomJS	phantomjs
HTMLUnit	htmlunit
HTMLUnit with Javascript	htmlunitwithjs

<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>



Locator strategies

Strategy	Match based on	Example
id	Element <code>id</code> .	<code>id:example</code>
name	<code>name</code> attribute.	<code>name:example</code>
identifier	Either <code>id</code> or <code>name</code> .	<code>identifier:example</code>
class	Element <code>class</code> .	<code>class:example</code>
tag	Tag name.	<code>tag:div</code>
xpath	XPath expression.	<code>xpath://div[@id="example"]</code>
css	CSS selector.	<code>css:div#example</code>
dom	DOM expression.	<code>dom:document.images[5]</code>
link	Exact text a link has.	<code>link:The example</code>
partial link	Partial link text.	<code>partial link:he ex</code>
sizzle	Sizzle selector deprecated.	<code>sizzle:div.example</code>
jquery	jQuery expression.	<code>jquery:div.example</code>
default	Keyword specific default behavior.	<code>default:example</code>

<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>



Verify elements in browser

Using keyword that have prefix “Wait Until”

Wait Until Element Contains	<code>locator, text, timeout=None, error=None</code>	Waits until the element <code>locator</code> contains <code>text</code> . Fails if <code>timeout</code> expires before the text appears. See the Timeouts section for details about the locator syntax. <code>error</code> can be used to override the default error message.
Wait Until Element Does Not Contain	<code>locator, text, timeout=None, error=None</code>	Waits until the element <code>locator</code> does not contain <code>text</code> . Fails if <code>timeout</code> expires before the text disappears. See the Timeouts section for details about the locator syntax. <code>error</code> can be used to override the default error message.
Wait Until Element Is Enabled	<code>locator, timeout=None, error=None</code>	Waits until the element <code>locator</code> is enabled. Element is considered enabled if it is not disabled nor read-only. Fails if <code>timeout</code> expires before the element is enabled. See the Timeouts section for details about the locator syntax. <code>error</code> can be used to override the default error message. Considering read-only elements to be disabled is a new feature in Selenium 4.
Wait Until Element Is Not Visible	<code>locator, timeout=None, error=None</code>	Waits until the element <code>locator</code> is not visible. Fails if <code>timeout</code> expires before the element is not visible. See the Timeouts section for details about the locator syntax.

<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>



First test case

Create file => **google_success.robot**

***** Settings *****

***** Variables *****

***** Test Cases *****

***** Keywords *****



Run test and see result

```
$robot google_success.robot
```



Working with Selenium

*** Settings ***

Library SeleniumLibrary

*** Variables ***

\${URL} http://www.google.com

*** Test Cases ***

ค้นหาคำว่า เหล็กไฟล

[Tags] done

เข้าไปยังหน้าค้นหาของ google

ค้นหาคำว่า "เหล็กไฟล"

จะต้องเจอ "เหล็กไฟล" นะ

*** Keywords ***

ค้นหาคำว่า "\${keyword}"

Input Text

name:q

\${keyword}

Press Keys

name:q

RETURN



Setup web browser driver

Web Browser	Driver
Google Chrome	ChromeDriver
FireFox	GeckoDriver
Internet Explorer	InternetExplorerDriver
Microsoft Edge	EdgeDriver



eg. ChromeDriver

```
$set WEB_DRIVER=<path of driver>
$set PATH=.;%WEB_DRIVER%;%PATH%
```

<https://chromedriver.chromium.org/downloads>



Reporting

Open file report.html

Comments Api Report

Generated
20190925 16:13:40 UTC+07:00
13 days 20 hours ago

Summary Information

Status: All tests passed
Start Time: 20190925 16:13:40.553
End Time: 20190925 16:13:40.961
Elapsed Time: 00:00:00.408
Log File: log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:00	P
All Tests	1	1	0	00:00:00	P

Statistics by Tag

Statistics by Tag	Total
No Tags	

Statistics by Suite

Statistics by Suite	Total
Comments Api	1

Test Details

Totals Tags Suites Search

Type: Critical Tests All Tests

Comments Api Log

Generated
20190925 16:13:40 UTC+07:00
13 days 20 hours ago

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:00	P
All Tests	1	1	0	00:00:00	P

Statistics by Tag

Statistics by Tag	Total
No Tags	

Statistics by Suite

Statistics by Suite	Total
Comments Api	1

Test Execution Log

- **SUITE** Comments Api

Full Name: Comments Api
Source: /Users/somkiat/data/slide/robotframework/aycap-20190924/comments_api.robot
Start / End / Elapsed: 20190925 16:13:40.553 / 20190925 16:13:40.961 / 00:00:00.408
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

- **TEST** Flow 1

Full Name: Comments Api.Flow 1
Start / End / Elapsed: 20190925 16:13:40.750 / 20190925 16:13:40.960 / 00:00:00.210
Status: PASS (critical)

+ KEYWORD \${email} = Get Comments
+ KEYWORD BuiltIn.Log To Console \${email}

.Net Core
© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

123

.Net Core API



Topics

Use ASP.NET core to build reliable APIs

Understand REST

Use models and validation

Versioning your APIs

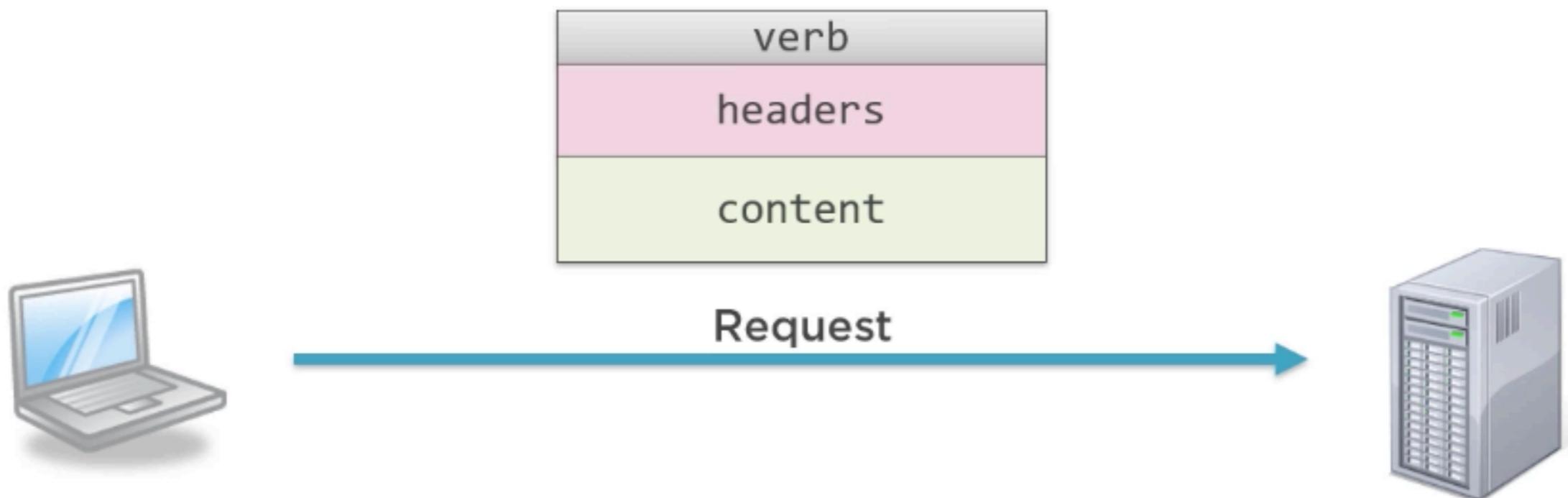
Secure your APIs



REST



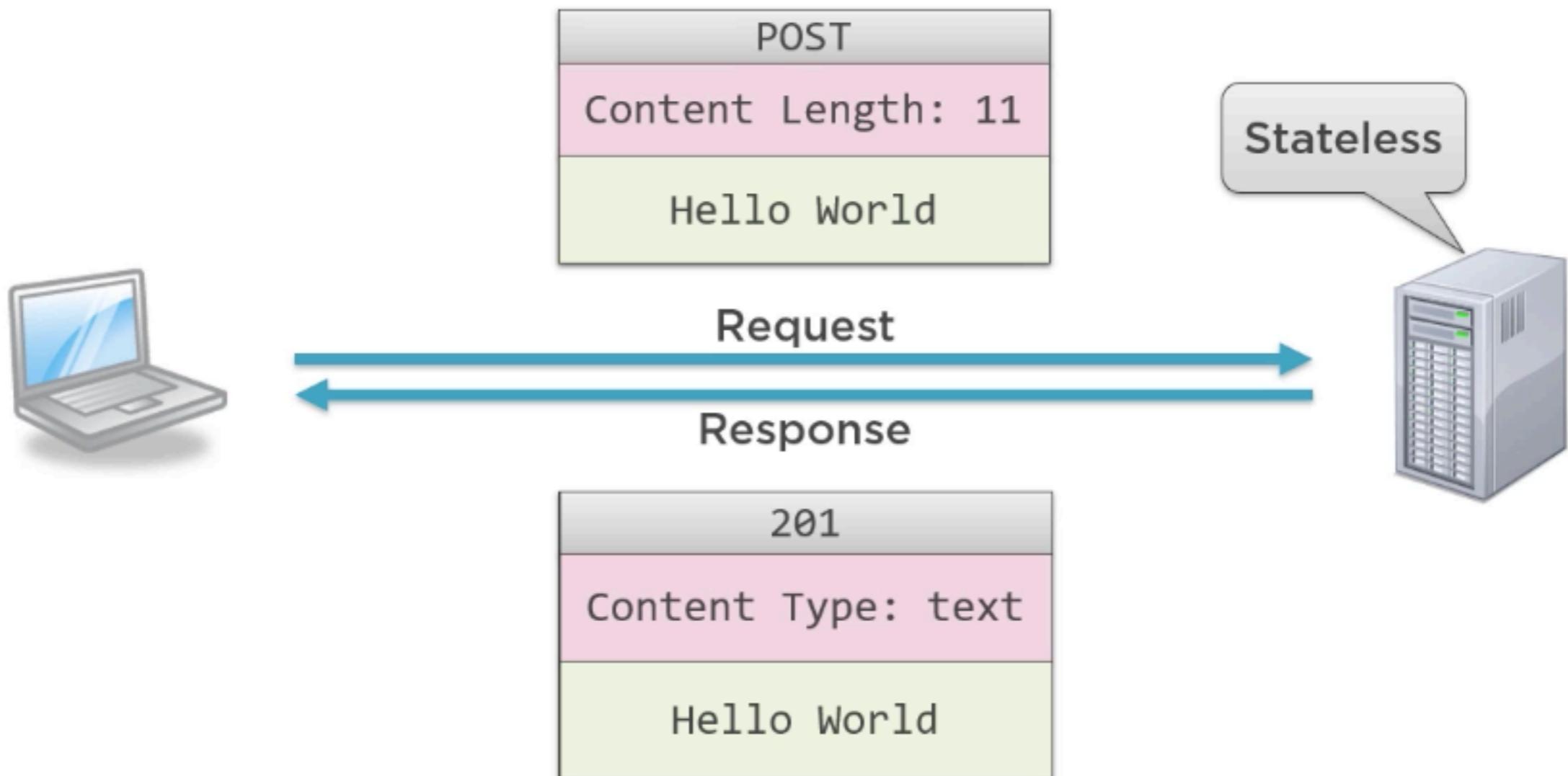
How HTTP work ?



How HTTP work ?



Stateless by default



Verb or HTTP methods

GET : retrieve a resource

POST : add a new resource

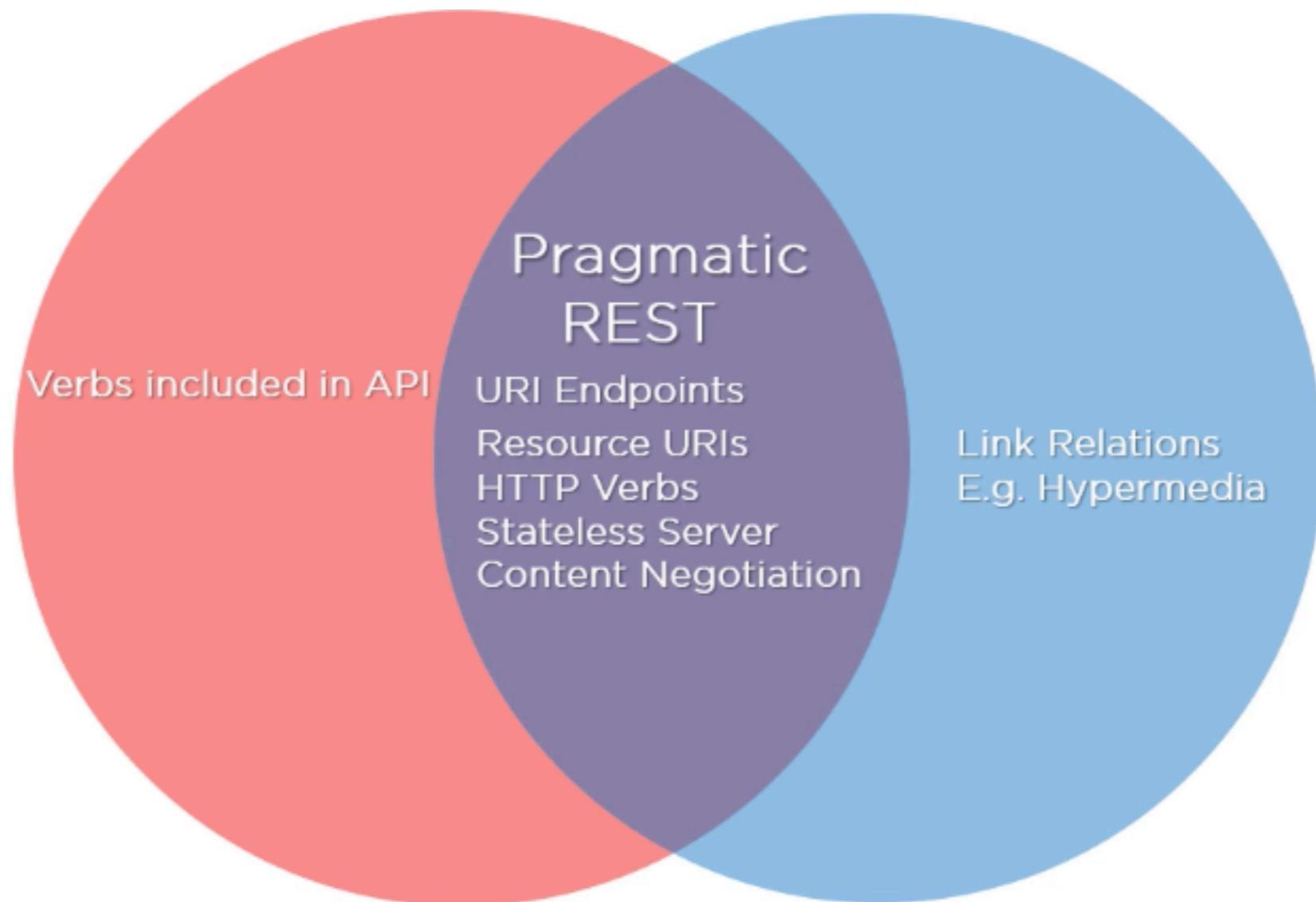
PUT : update an existing resource

PATCH : update an existing resource with set
of changes

DELETE : remove an existing resource



APIs on the web



REST ?

REpresentational State Transfer

Concepts

Separation of client and server

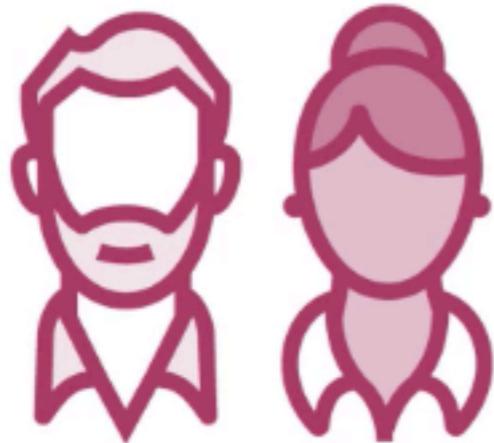
Server requests are **stateless**

Cacheable requests

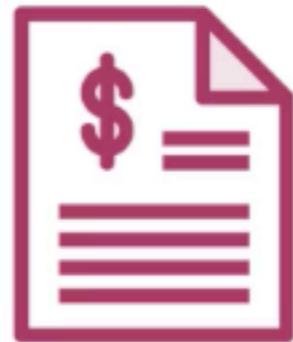
Uniform interface



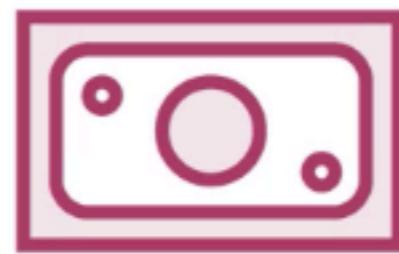
Resources ?



People



Invoices



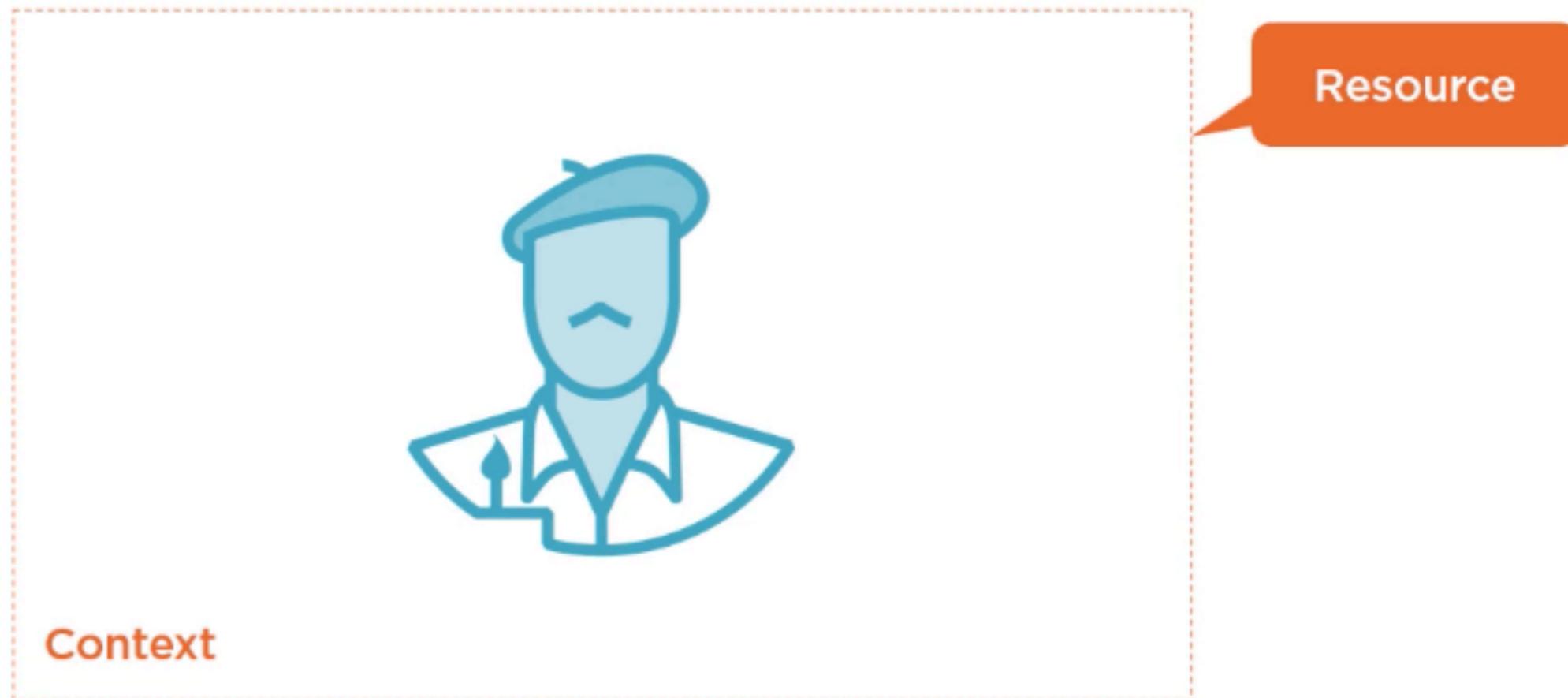
Payments



Products



Resources vs Entities ?



Resources vs Entities ?



What are URIs ?

URIs are just paths to resources
e.g. api.com/person

Query strings for non-data elements
e.g. sorting, searching, formatting



Design API

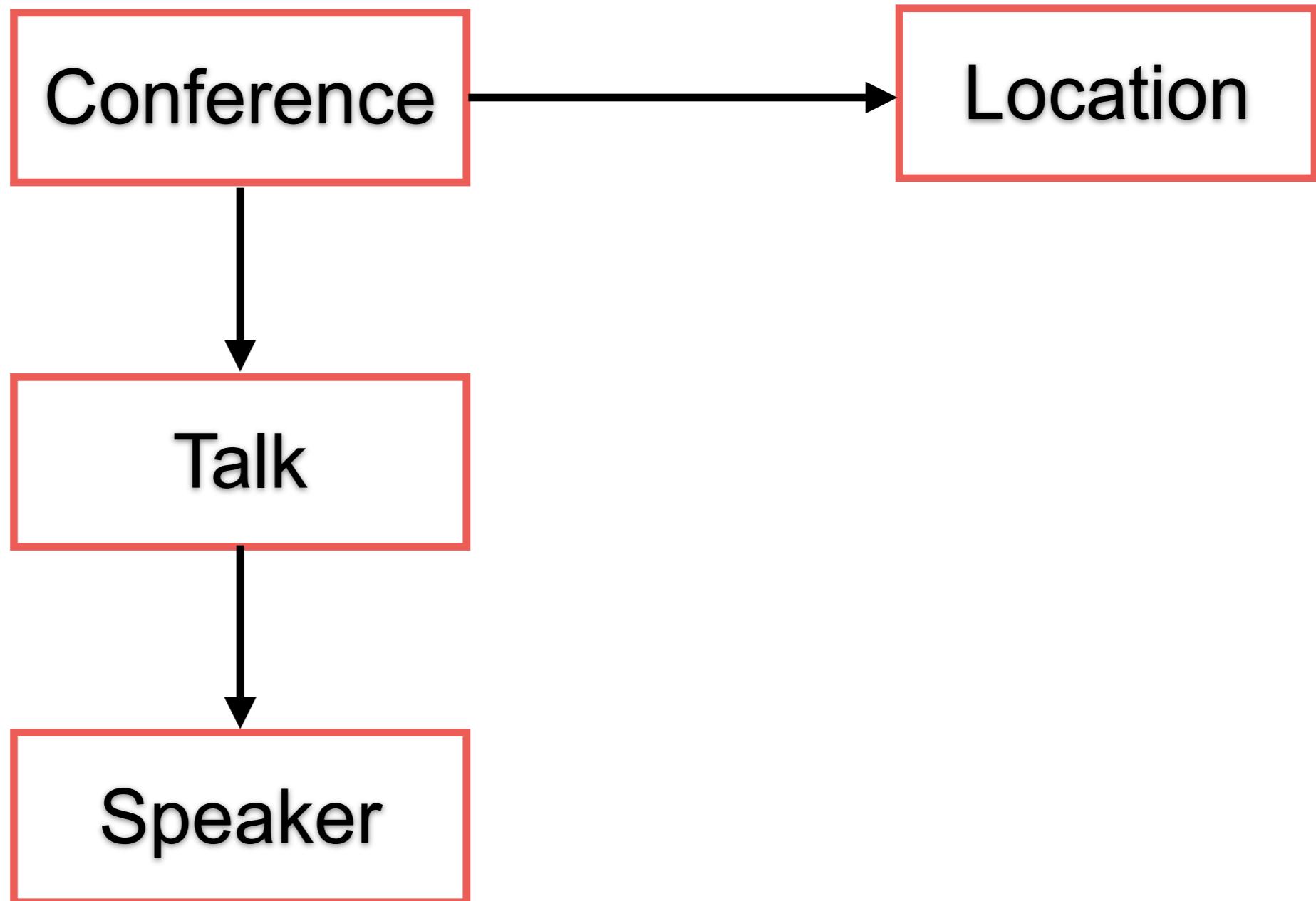


Design APIs

Not to design database
Not to design your website



Example



Design APIs

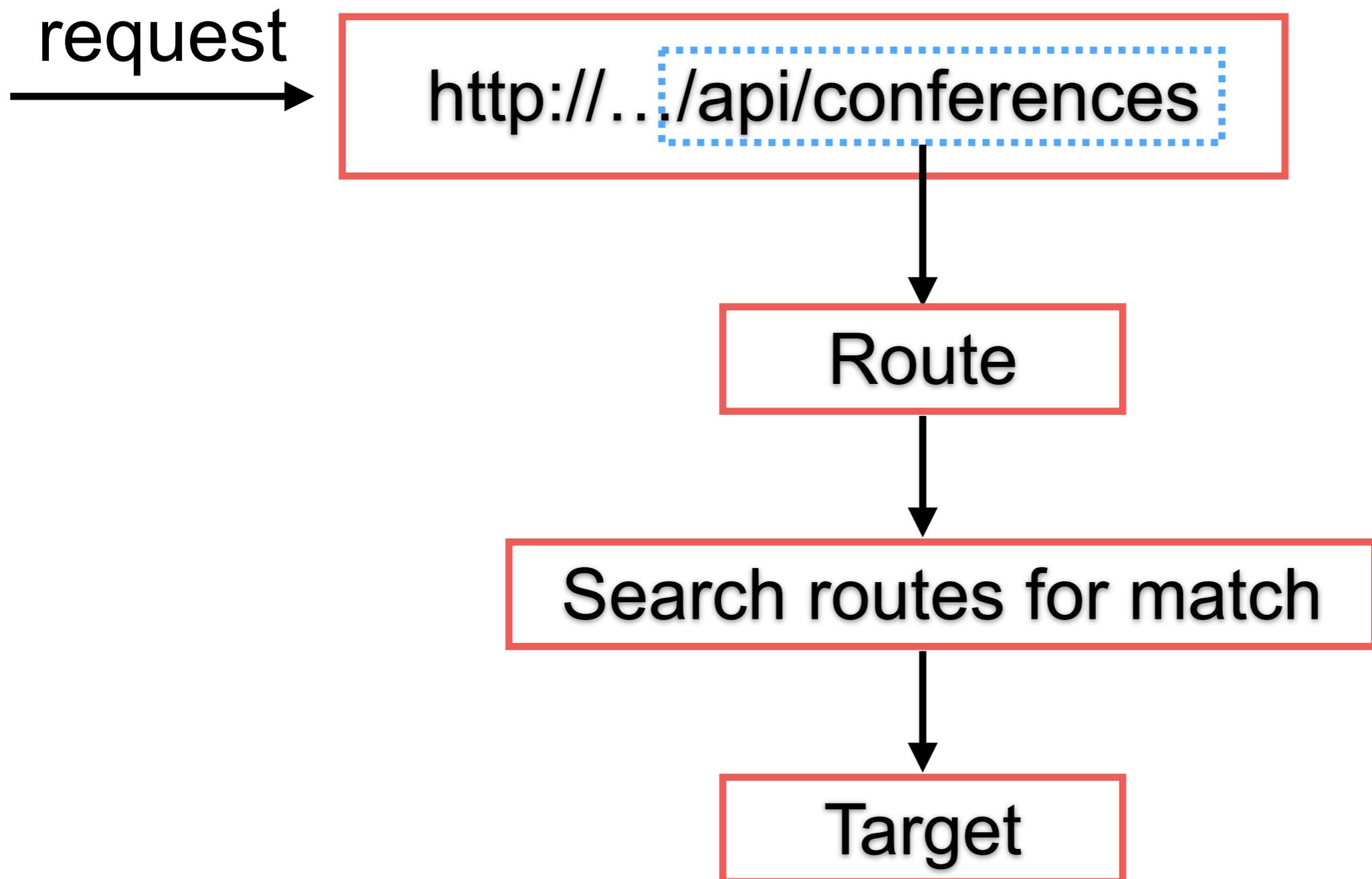
API	Description
/api/conferences	
/api/conferences/THAI2020	
/api/conferences/THAI2020/location	
/api/conferences/THAI2020/talks	
/api/conferences/THAI2020/talks?topic=api	
/api/conferences/THAI2020/talks/1/speaker	



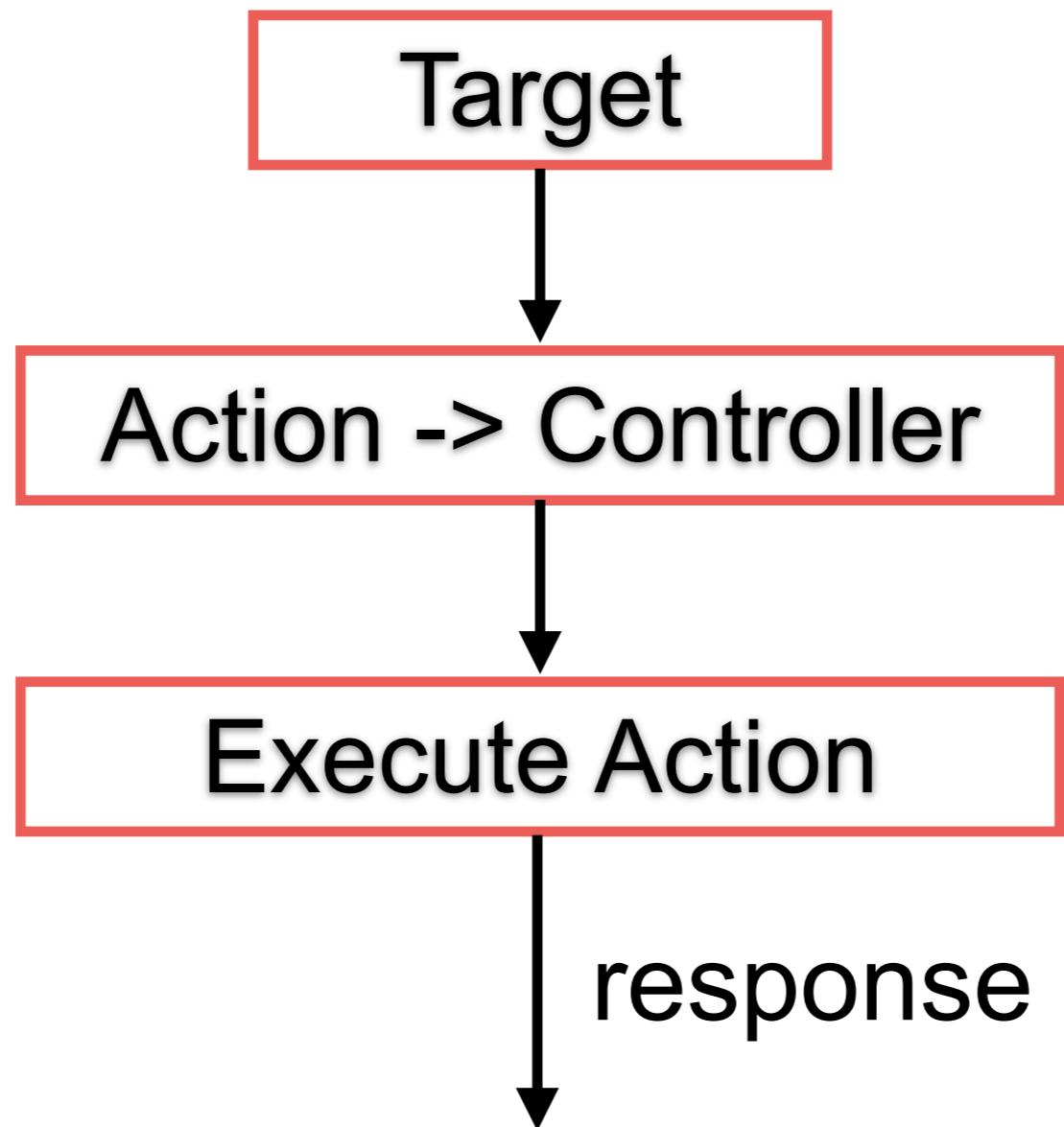
Develop RESTFul API



ASP.NET Core API



ASP.NET Core API



Status code

Code	Description	Code	Description
200	OK	400	Bad request
201	Created	401	Not authorize
202	Accepted	403	Forbidden
302	Found	404	Not found
304	Not modified	405	Method not allow
307	Temporary redirect	409	Conflict
308	Permanent redirect	500	Internal error



Workshop

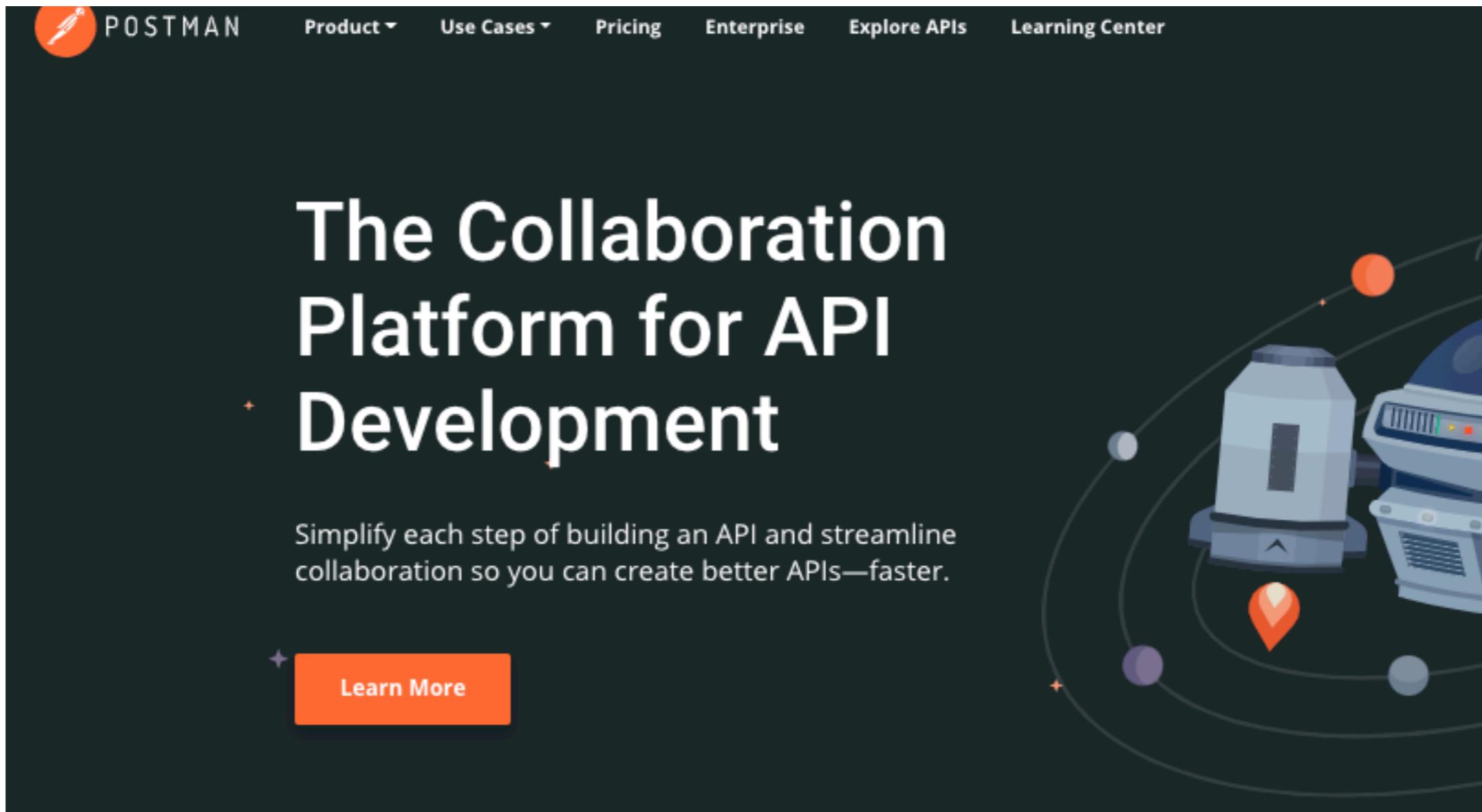
Building first API project
Create Actions
Status code
CRUD operation



API testing



Using Postman



The screenshot shows the official Postman website homepage. At the top, there's a navigation bar with the Postman logo, followed by links for Product, Use Cases, Pricing, Enterprise, Explore APIs, and Learning Center. The main title "The Collaboration Platform for API Development" is prominently displayed in large white text. Below it, a subtitle reads: "Simplify each step of building an API and streamline collaboration so you can create better APIs—faster." A "Learn More" button is located at the bottom left of this section. To the right, there's a stylized illustration of two robots or drones flying over concentric circular orbits, with colored dots (orange, purple, blue) representing data points.

<https://www.postman.com/>



What is Postman ?

User Interface tool for API testing

Easy to use

Support response parsing (XML, JSON)

Define the concept of collection (group of requests)

Free !!



Start Postman

Create New

Templates

API Network

X

BUILDING BLOCKS

Request

GET

Create a basic request



Collection

Save your requests in a collection for reuse and sharing



Environment

Save values you frequently use in an environment

ADVANCED

API Documentation



Create and publish beautiful documentation for your APIs



Mock Server

Create a mock server for your in-development APIs



Monitor

Schedule automated tests and check performance of your APIs

API



Manage all aspects of API design, development, and testing

Not sure where to start? Use a [template](#) to see how Postman can help you in your work.

[Learn more on Postman Docs](#)



.Net Core

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

149

Using Postman

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Postman' in the center, followed by 'My Workspace' and 'Invite' buttons. Below the navigation bar are buttons for 'New', 'Import', 'Runner', and a plus sign icon. To the right of these are icons for refresh, undo, redo, and notifications, along with an 'Upgrade' button.

The main workspace is titled 'Untitled Request'. It includes a 'Test' dropdown, a search bar, and a 'Comments' section. On the left, there's a sidebar with tabs for 'History' (which is selected), 'Collections', and 'APIs'. A 'Save Responses' toggle is also present. A message says 'You haven't sent any requests' with a note: 'Any request you send in this workspace will appear here.' Below this is a 'Show me how' button with a hand cursor icon.

The central area contains a 'GET Untitled Request' card with a close button, a plus sign, and three dots. Below it is a 'Query Params' table:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Below the table is a 'Response' section with a placeholder image of a rocket launch and the text 'Hit Send to get a response'.



Basic of Postman

Request
Collection
Environment



Create request (1)

Create New

Templates

API Network

BUILDING BLOCKS

GET

Request

Create a basic request



Collection

Save your requests in a collection for reuse and sharing

ADVANCED



API Documentation

Create and publish beautiful documentation for your APIs



Mock Server

Create a mock server for your in-development APIs



Create request (2)

Enter request name

SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).

[Learn more about creating collections](#)

Request name

First request

Request description (Optional)

Make things easier for your teammates with a complete request description.



Create request (3)

Create new collection and save

Select a collection or folder to save to:

Search for a collection or folder

All Collections + Create Collection

first_group X ✓

Cancel

Save



Create request (4)

Screenshot of the Postman application interface showing the creation of a new request.

The top navigation bar includes: New, Import, Runner, My Workspace, Invite, Refresh, and Upgrade.

The left sidebar shows: History, Collections (selected), APIs, + New Collection, first_group (1 request).

The main workspace displays a collection named "First request" with one item: "First request".

The request details are as follows:

- Type: GET
- URL: Enter request URL
- Send button
- Save button

The Params tab is selected, showing a table for Query Params:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

The Response section is currently empty.



Detail of request

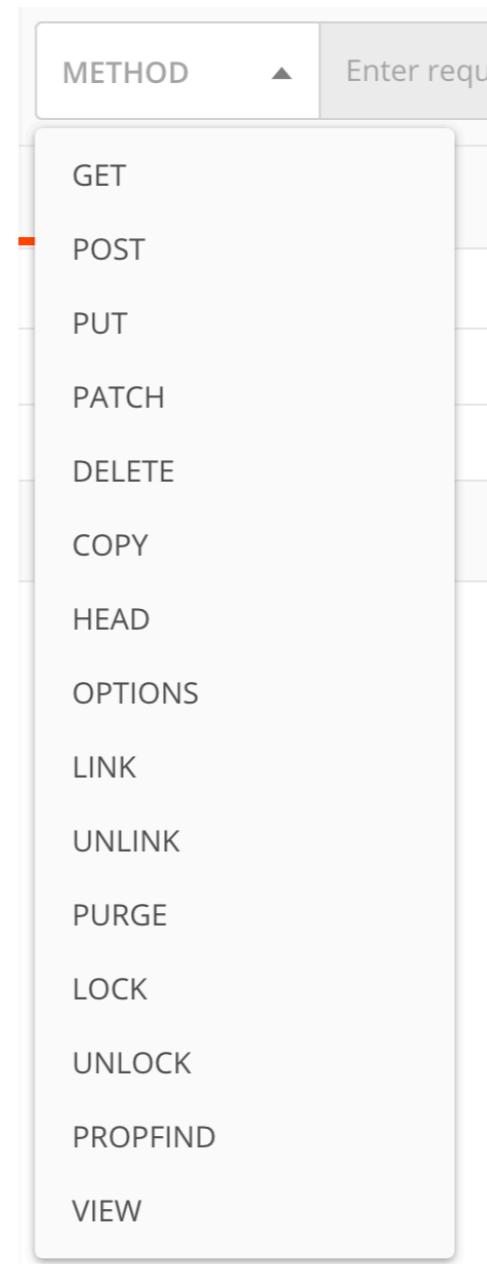
HTTP method and URL

The screenshot shows the Postman application interface. At the top, there is a header bar with a red border. On the left, it says "GET First request" with a close button ("X") and a plus sign ("+"). On the right, it says "No Environment" with a dropdown arrow. Below the header, there is a section titled "First request" with a red border around it. This section contains a "Method" dropdown set to "GET", an "Enter request URL" input field, and a "Send" button. Below this, there are tabs for "Params", "Authorization", "Headers", "Body", "Pre-request Script", "Tests", and "Settings". The "Params" tab is currently selected. Under the "Params" tab, there is a table titled "Query Params" with columns "KEY", "VALUE", "DESCRIPTION", and a three-dot menu icon. One row is present in the table with "Key" in the KEY column and "Value" in the VALUE column. At the bottom, there is a large "Response" section.



HTTP methods

GET, POST, PUT, DELETE ... etc



Example of request

GET https://www.google.com

The screenshot shows the Postman application interface. At the top, there is a header bar with 'GET' selected, a URL field containing 'Https://www.google.com', and buttons for 'Send' and 'Save'. Below the header, a navigation bar includes tabs for 'Params' (which is active), 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', 'Settings', 'Cookies', and 'Code'. The main content area is titled 'Query Params' and contains a table with one row: 'Key' (Value) and 'Value' (Description). In the 'Body' section, tabs for 'Body', 'Cookies (2)', 'Headers (13)', and 'Test Results' are visible, along with status information: 'Status: 200 OK', 'Time: 235ms', and 'Size: 18.53 KB'. A 'Save Response' button is also present. The 'Body' tab is active, displaying the response content in 'Pretty' format. The response is an HTML document starting with the doctype declaration and various meta tags.

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies (2) Headers (13) Test Results Status: 200 OK Time: 235ms Size: 18.53 KB Save Response ▾

Pretty Raw Preview Visualize HTML ▾

```
1 <!doctype html>
2 <html itemscope="" itemtype="http://schema.org/WebPage" lang="th">
3
4 <head>
5   <meta content="text/html; charset=UTF-8" http-equiv="Content-Type">
6   <meta content="/logos/doodles/2020/recognizing-ignaz-semmelweis-and-handwashing-6753651837108746.2-l.png"
7     itemprop="image">
8   <meta
9     content="ระลึกถึงอิกนา"
```



Detail of request

Parameters, Headers, Body and Testing

The screenshot shows the Postman application interface. At the top, there's a header bar with a red border. On the left, it says "GET First request" with a close button ("X") and a plus sign ("+"). On the right, it says "No Environment" with a dropdown arrow. Below the header, there's a section titled "First request" with a "Comments 0" button and an "Example" button. Underneath this, there's a search bar with "GET" and a dropdown arrow, followed by a text input field "Enter request URL". To the right of the URL input is a blue "Send" button with a dropdown arrow. Below these controls, there are several tabs: "Params" (which is selected and highlighted in orange), "Authorization", "Headers", "Body", "Pre-request Script", "Tests", and "Settings". Under the "Params" tab, there's a section titled "Query Params" with a table. The table has columns for "KEY", "VALUE", "DESCRIPTION", and a "..." button. One row is visible with "Key" in the KEY column, "Value" in the VALUE column, and "Description" in the DESCRIPTION column. At the bottom of the interface, there's a large section labeled "Response".



Header of request

▼ Headers (1)

	KEY
<input checked="" type="checkbox"/>	
	Accept
► 1	Accept-Charset
	Accept-Encoding
Body	Accept-Language
Pre	Access-Control-Request-Headers
	Access-Control-Request-Method
	Authorization



Body/payload of request

Working with POST,PUT,DELETE method

The screenshot shows the Postman interface for a GET request to `Https://www.google.com`. The 'Body' tab is highlighted with a red box. Below it, the 'raw' option is selected under the data type dropdown, which is set to 'JSON'. Other options include 'none', 'form-data', 'x-www-form-urlencoded', 'binary', and 'GraphQL'.

Submit form
Upload files
Send XML/JSON data



HTTP response

Target server send the response back to requester/client

Body Cookies (2) Headers (13) Test Results Status: 200 OK Time: 235ms Size: 18.53 KB Save Response ▾

Pretty Raw Preview Visualize HTML ▾

```
1 <!doctype html>
2 <html itemscope="" itemtype="http://schema.org/WebPage" lang="th">
3
4 <head>
5   <meta content="text/html; charset=UTF-8" http-equiv="Content-Type">
6   <meta content="/logos/doodles/2020/recognizing-ignaz-semmelweis-and-handwashing-6753651837108746.2-l.png"
7     itemprop="image">
8   <meta
9     content="ระลึกถึงอิกนา&
#3607;ซ์ &#3648;ซมเมิลไว&
#3626;์ &#3649;ละการล้างม&
#3639;อที่ถูกต้อง">
10  property="twitter:title">
```



Response code

Body Cookies (2) Headers (13) Test Results Status: 200 OK Time: 235ms Size: 18.53 KB Save Response ▾

Pretty Raw Preview Visualize HTML ▾

```
1 <!doctype html>
2 <html itemscope="" itemtype="http://schema.org/WebPage" lang="th">
3
4 <head>
5   <meta content="text/html; charset=UTF-8" http-equiv="Content-Type">
6   <meta content="/logos/doodles/2020/recognizing-ignaz-semmelweis-and-handwashing-6753651837108746.2-l.png"
7     itemprop="image">
8   <meta
9     content="ระลึกถึงอิกนา&
#3607;ซ์ &#3648;ซ็มเมิลไว&
#3626;์ &#3649;ละการล้างม&
#3639;อที่ถูกต้อง">
10  property="twitter:title">
```



Testing with request



Testing your APIs

Validation response of APIs

Automation testing > Manual testing

Postman testing (with less coding)



Testing in Postman

The screenshot shows the Postman interface for testing a GET request to `Https://www.google.com`. The 'Tests' tab is highlighted with a red box and a tooltip. The tooltip provides information about test scripts and lists various snippets.

Test scripts are written in JavaScript, and are run after the response is received.
[Learn more about tests scripts](#)

SNIPPETS

- [Get an environment variable](#)
- [Get a global variable](#)
- [Get a variable](#)
- [Set an environment variable](#)
- [Set a global variable](#)
- [Clear an environment variable](#)
- [Clear a global variable](#)
- ...

Snippet/example code



Useful snippet code

Status code: Code is 200

Response body: JSON value check

Response time is less than 200ms

Use tiny validator for JSON data



Status code: Code is 200

GET ▾ <Https://www.google.com> Send ▾ Save

Params Authorization Headers (8) Body Pre-request Script Tests ● Settings Cookies

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });
```

2

Test scripts are written in JavaScript, and are run after the response is received.
[Learn more about tests scripts](#)

SNIPPETS
[Send a request](#)

Status code: Code is 200

Response body: Contains string

Response body: JSON value check

Response body: Is equal to a string

Response headers: Content-Type header contains

Response time is less than 200ms

Status code: Successful POST request



Response body: JSON value check

GET ▾ <Https://www.google.com> Send ▾ Save

Params Authorization Headers (8) Body Pre-request Script Tests ● Settings Cookies

```
1 pm.test("Your test name", function () {  
2     var jsonData = pm.response.json();  
3     pm.expect(jsonData.value).to.eql(100);  
4 });
```

2

Test scripts are written in JavaScript, and are run after the response is received.
[Learn more about tests scripts](#)

SNIPPETS
[Send a request](#)

Status code: Code is 200

Response body: Contains string

Response body: JSON value check

Response body: Is equal to a string

Response headers: Content-Type header contains

Response time is less than 200ms

Status code: Successful POST request



Try to testing and see result

The screenshot shows the Postman interface for a GET request to `Https://www.google.com`. A red circle labeled '1' highlights the `Send` button. Below it, a red box highlights the `Tests` tab, which contains a snippet of JavaScript code:

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3});
```

To the right of the tests section, there is explanatory text: "Test scripts are written in JavaScript and run after the response is received." and a link "Learn more about tests scripts".

The screenshot shows the test results for the previous request. A red box highlights the `Test Results (1/1)` tab, which displays a single result:

PASS	Status code is 200
------	--------------------

A red circle labeled '2' highlights the status code `200 OK`.



Working with collection



Collection

Group of requests

Share and export with other

Run sequence of requests



Working with collection

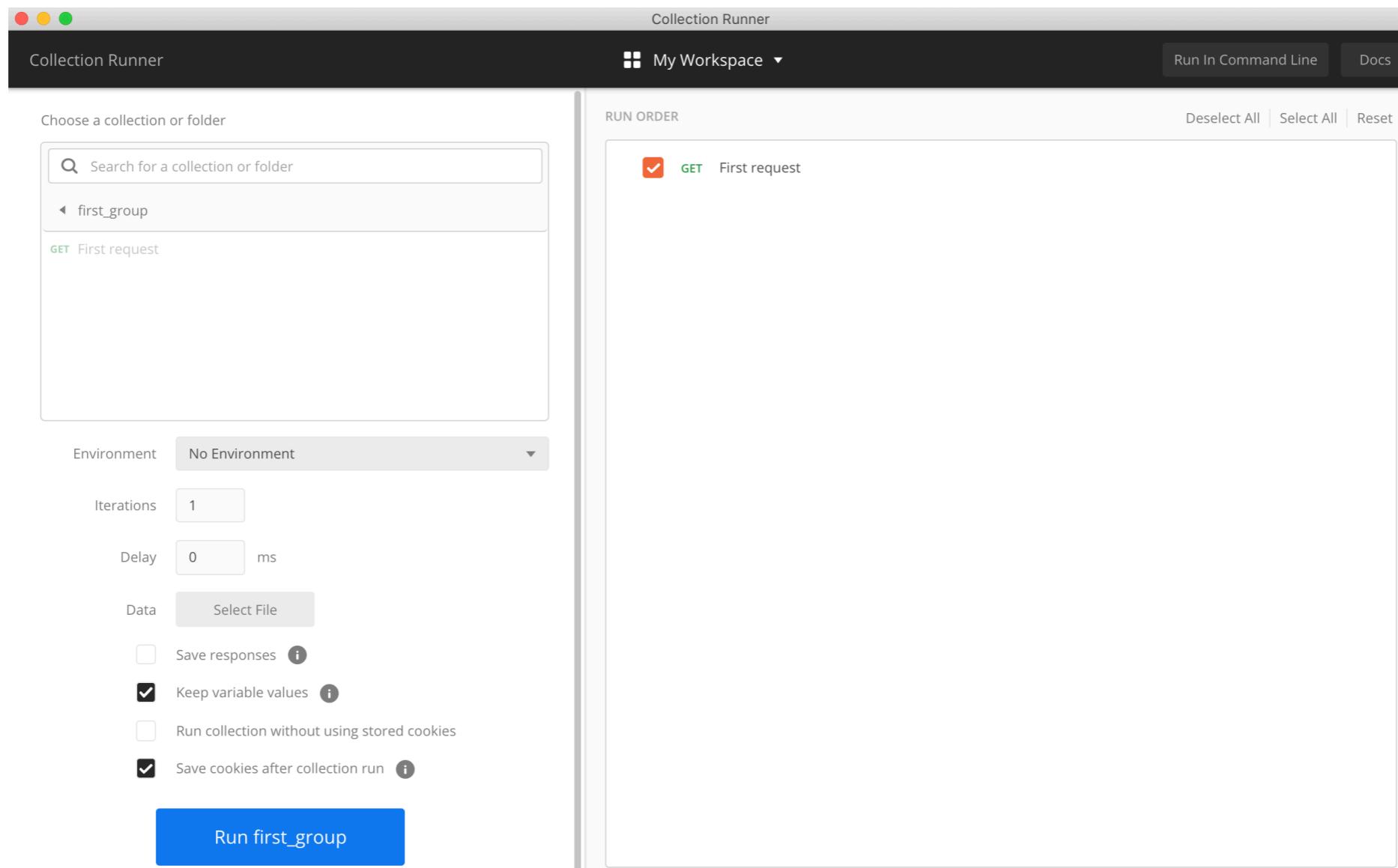
Run with selected collection

The screenshot shows a user interface for managing API collections. On the left, there's a navigation bar with tabs: History, Collections (which is active, indicated by an orange underline), and APIs. Below the tabs are buttons for '+ New Collection' and 'Trash'. A list of collections is shown, with one named 'first_group' highlighted by a red circle labeled '1'. This collection has a star icon and '1 request'. To the right of the list is a vertical toolbar with a back arrow and a three-dot menu. On the right side of the screen, a detailed view of the 'first_group' collection is displayed. It shows the collection name 'first_group' and a message stating 'This collection is not linked to any API'. Below this are buttons for 'Share' (orange), 'Run' (blue and highlighted with a red circle labeled '2'), 'View in web' (grey), and a three-dot menu. Underneath these buttons are tabs for 'Documentation' (which is active, indicated by an orange underline), 'Monitors', 'Mocks', and 'Changelog'. At the bottom of the collection view, there's a section for adding a description with a 'Learn how to document your requests' link, and a 'GET First request' entry.



Collection runner

Config environment, iteration, delay time and data from file



Collection runner

Search for a collection or folder

◀ first_group

GET First request

Environment No Environment ▾

Iterations 1

Delay 0 ms

Data Select File

Save responses ⓘ

Keep variable values ⓘ

Run collection without using stored cookies

Save cookies after collection run ⓘ

Run first_group



Working with environment



Environment

Dynamic data in each environment
E.g. Dev, Test, Staging, Production

The screenshot shows a REST client interface. At the top left, there is a search bar with the text "POST /api/enquiry" and a red "X" button. To the right of the search bar are three buttons: a plus sign (+), a minus sign (-), and three dots (...). On the far right, there is a dropdown menu labeled "No Environment" with a red border around it. Below the search bar, there is a row of buttons: "POST" (with a dropdown arrow), the URL "https://xis.moe.go.th:8080/api/enquiry", a blue "Send" button, and other buttons with dropdown arrows. The main body of the interface is currently empty.



Manage Environment

Dev, Test, Staging, Production

MANAGE ENVIRONMENTS X

An environment is a set of variables that allow you to switch the context of your requests. Environments can be shared between multiple workspaces. [Learn more about environments](#)

DEV	Share		...
PRODUCTION	Share		...
STAGING	Share		...
TEST	Share		...

[Globals](#) [Import](#) [Add](#)



Manage Environment

Create variable name in each environment

Environment Name

DEV

VARIABLE	INITIAL VALUE <small>i</small>	CURRENT VALUE <small>i</small>
<input checked="" type="checkbox"/> SERVER_NAME	DEV	DEV
Add a new variable		

... | Persist All | Reset All

Variable name = **SERVER_NAME**

Environment Name

PRODUCTION

VARIABLE	INITIAL VALUE <small>i</small>	CURRENT VALUE <small>i</small>
<input checked="" type="checkbox"/> SERVER_NAME	https://iapi.bot.or.th	https://iapi.bot.or.th
Add a new variable		

... | Persist All | Reset All



Manage Environment

Use variable from environment in request

{{SERVER_NAME}}

The screenshot shows the Postman application interface. At the top, there is a header bar with a 'POST /api/enquiry' button, a '+' icon, a three-dot menu icon, and a dropdown menu set to 'No Environment'. Below this is a search bar with a dropdown arrow. The main area contains a request card with a 'POST' method, a dropdown arrow, and a URL field containing ' **{{SERVER_NAME}}/api/enquiry** '. To the right of the URL field is a blue 'Send' button with a dropdown arrow. A red rectangular box highlights the URL field.



Manage Environment

Choose environment in each request

The screenshot shows the Postman application interface. At the top, there is a header bar with the URL `POST /api/enquiry`, a close button (`X`), a plus button (`+`), and a three-dot menu button (`...`). Below the header, the method is set to `POST` and the URL is `https://xis.moe.go.th:8080/api/enquiry`. On the left side, there are tabs for `Body`, `Cookies`, `Headers (5)`, and `Test Results`. The `Headers (5)` tab is currently selected. On the right side, the status is shown as `Status: 200 OK`. A red box highlights a dropdown menu titled "Type to filter" containing environment options: `No Environment`, `DEV`, `PRODUCTION`, `STAGING`, and `TEST`.



Using in Collection Runner

The screenshot shows the Postman Collection Runner interface. At the top, there's a navigation bar with "Collection Runner", "My Workspace" (with a dropdown arrow), "Run In Command Line", and "Docs". Below the navigation, a search bar says "Choose a collection or folder" and contains a placeholder "Search for a collection or folder". A sidebar on the left shows a tree structure under "workshop" with two POST requests: "/api/enquiry" and "/api/send". On the right, the "RUN ORDER" section lists two items: "POST /api/enquiry" and "POST /api/send", both checked. Below the run order, there are buttons for "Deselect All", "Select All", and "Reset". At the bottom, there are configuration options: "Environment" set to "No Environment" (highlighted with a red box), "Iterations" (dropdown menu showing "No Environment", "DEV", "PRODUCTION", "STAGING", and "TEST"), "Delay", "Data" (checkbox), "Keep variable values" (checkbox checked), and "Run collection without using stored cookies" (checkbox). The "Iterations" dropdown is the primary focus of the red box.

Collection Runner

My Workspace ▾

Run In Command Line Docs

Choose a collection or folder

Search for a collection or folder

◀ workshop

POST /api/enquiry

POST /api/send

RUN ORDER

Deselect All | Select All | Reset

Environment No Environment

Iterations No Environment

DEV

PRODUCTION

STAGING

TEST

Delay

Data

Keep variable values i

Run collection without using stored cookies



API versioning



API versioning ?

User/customer rely on the APIs

Requirement will changes

Need a way to evolve the API

Don't break clients

API versioning != Product versioning



Types of versioning

Versioning in URI

Versioning in HTTP Header



Versioning convention

Semantic Versioning 2.0.0

Summary

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards compatible manner, and
3. PATCH version when you make backwards compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

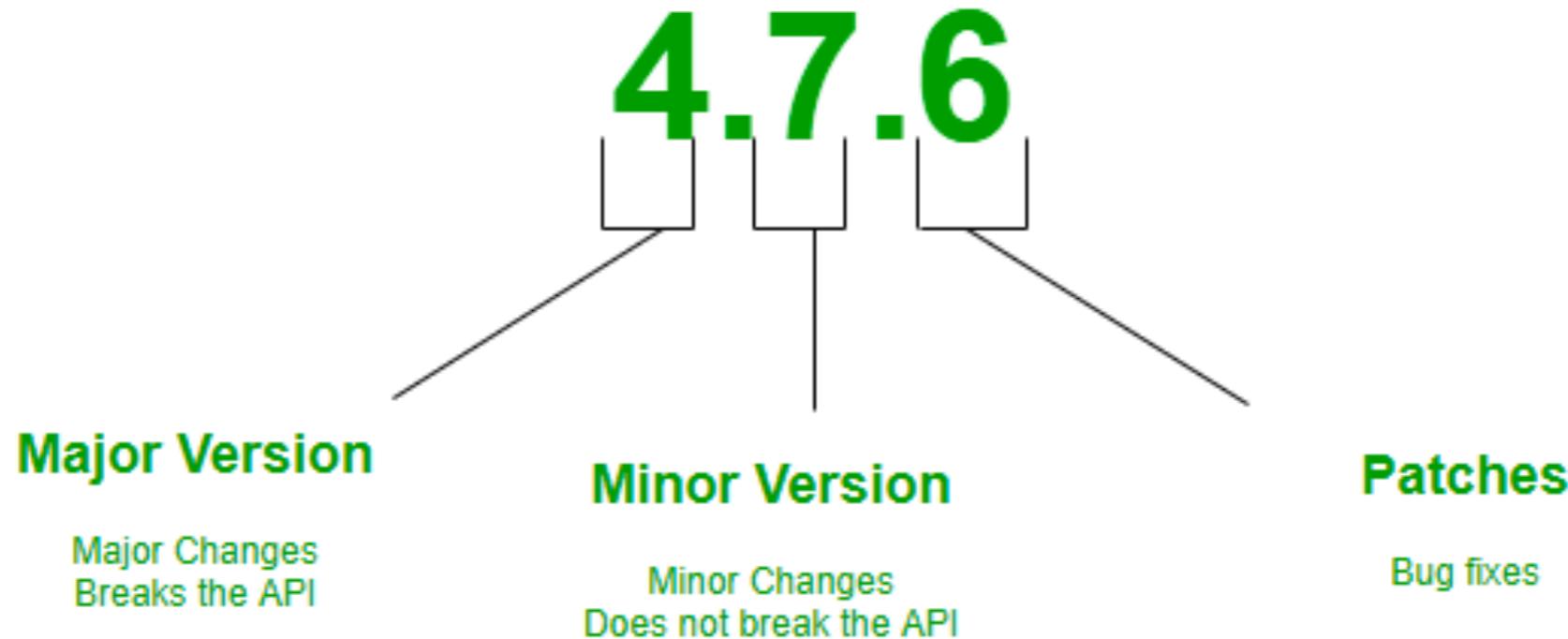
<https://semver.org/>



.Net Core

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

Semantic versioning



<https://semver.org/>



Workshop

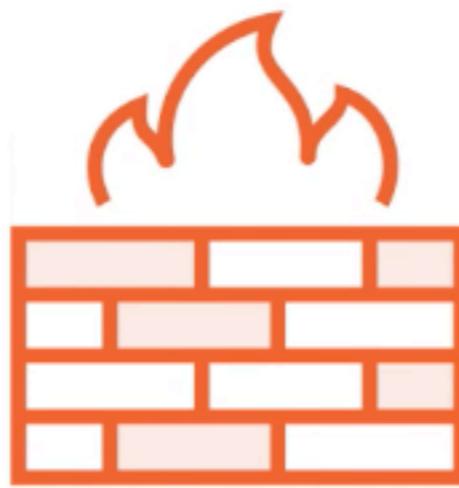
Versioning actions
Versioning controllers
Versioning with Headers
URL versioning



Secure your APIs



Path to secure your API



Understanding the basics



Put your knowledge to practice



Assume everything and everyone to be a threat !!



Authentication vs Authorization



Authorization (Access)



Authentication (Identity)



OAuth ?

Open standard for Authorization
Token based (Access token, JWT)



Finding a viable identity solution



Roll-your Own
NO!!



Framework
IdentityServer



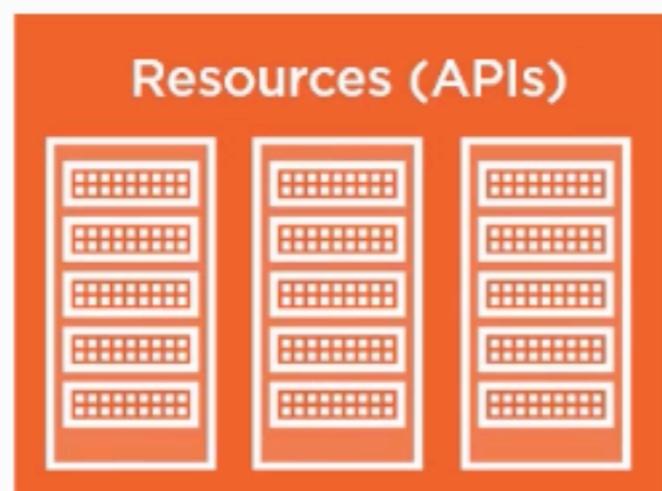
SaaS/PaaS
Amazon Cognito
Auth0
Stormpath



Develop secure API



Create Authorization server



Authorization Server
(OAuth Server)



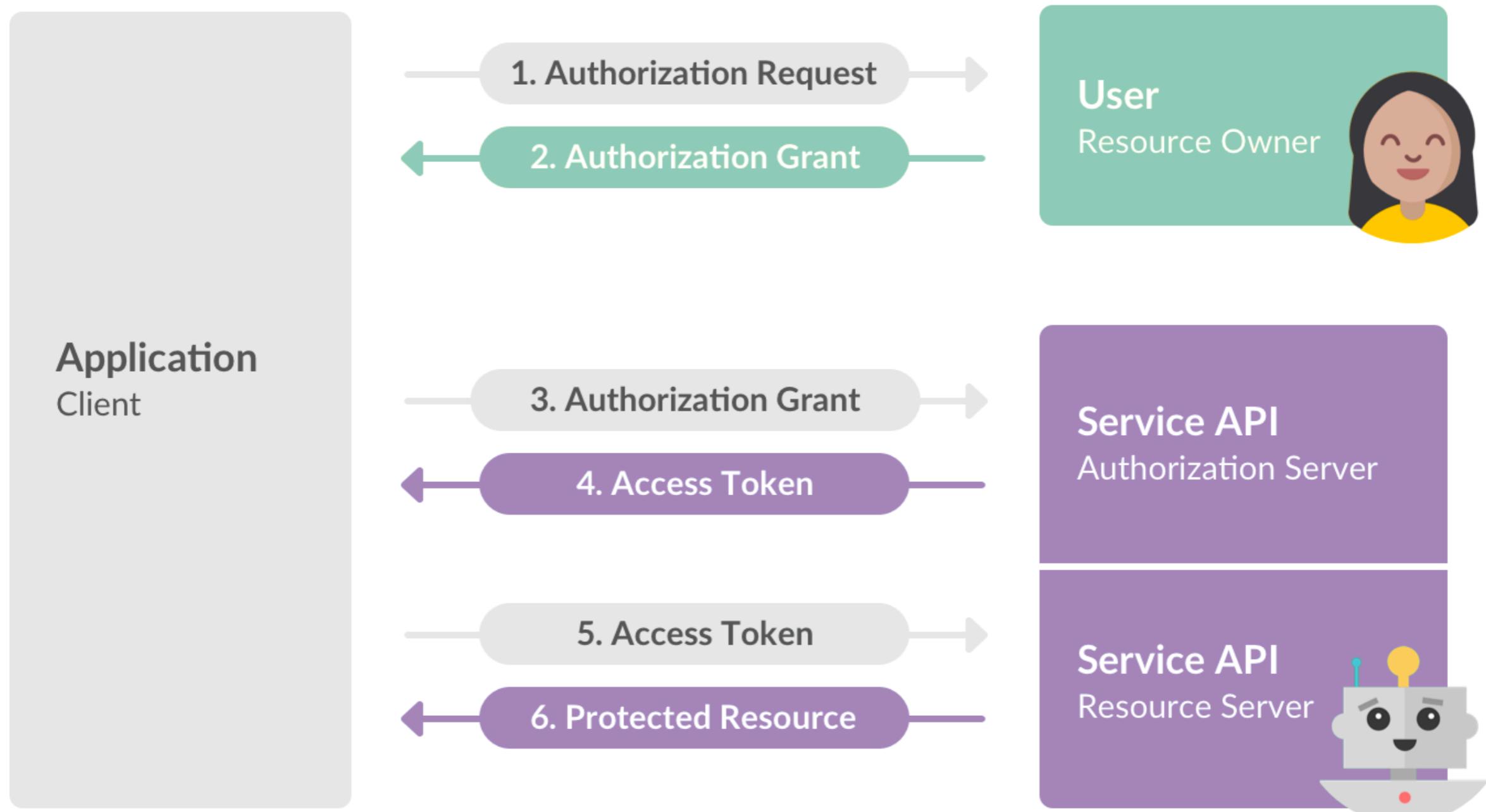
Signature difference in API

```
[HttpGet]  
public async Task<IHttpActionResult> GetAsync(string username, string password)
```

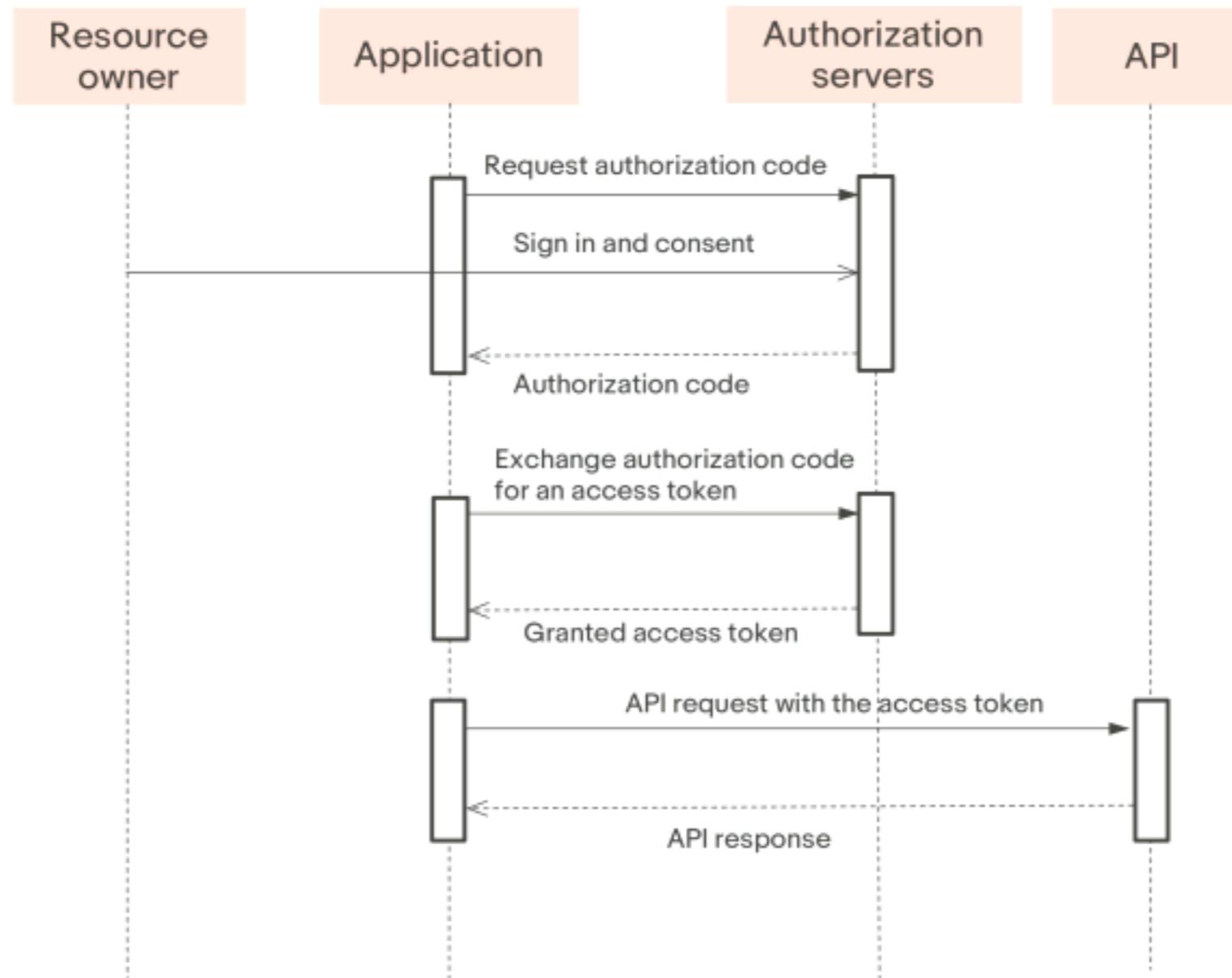
```
[HttpGet]  
[Authorize]  
public async Task<IHttpActionResult> GetAsync()
```



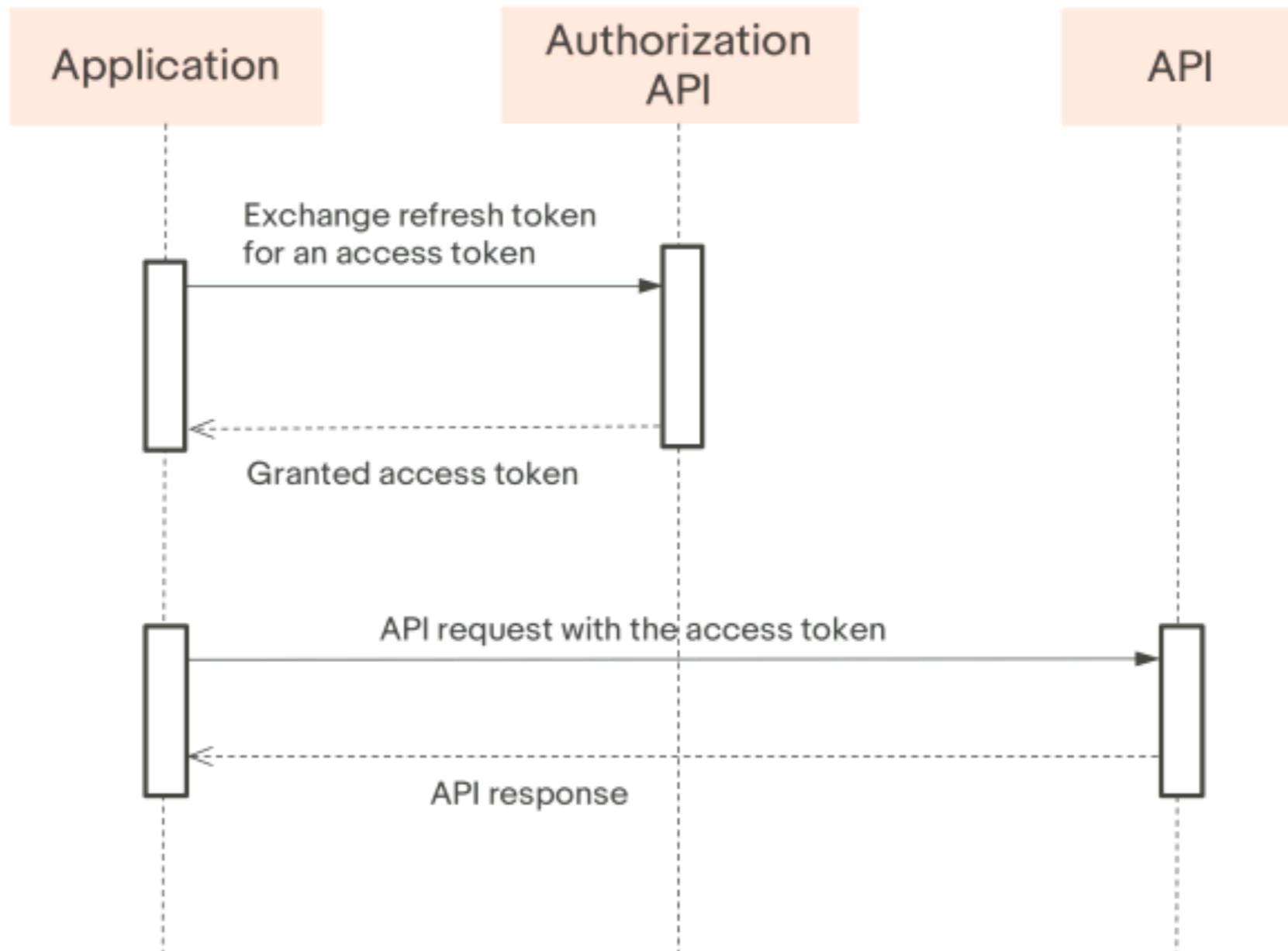
Flow OAuth 2.0



User get access token



Using refresh token



Workshop

