

A Web Page

http://www.vavatech.pl/szkolenia/wcf.html

Login

Hasło

~~GET /szkolenia/login?login=mateusz&password=12345 HTTP/1.1~~
~~Host: www.vavatech.pl~~
~~Accept: text/html~~
~~Accept-Language: pl-PL, en-US;q=0.7~~
~~{blank-line}~~

GET /szkolenia/wcf.html HTTP/1.1
Host: www.vavatech.pl
Accept: text/html
Accept-Language: pl-PL, en-US;q=0.7
{blank-line}

5000



{method} {resource} {protocol/version}
Host:{host}
Accept: {mime-type}
Accept-Language: {code}
{blank-line}

200 OK
Content-Type: text/html
<html>
<head> <title>WCF</title>
</head>

<body>
<h1>Szkolenie WCF</h1>

</body>

GET wcf.png HTTP/1.1
Host: www.vavatech.pl
Accept: image/png
Accept-Language: pl-PL, en-US;q=0.7
{blank-line}

200 OK
Content-Type: image/png
{obrazek.png}

POST /szkolenia/login HTTP/1.1
Host: www.vavatech.pl
Accept: text/html
Accept-Language: pl-PL, en-US;q=0.7
Content-Type: form-data

login=mateusz&password=12345

```
POST HelloService.svc HTTP/1.1
Host: localhost
Accept: application/xml
Content-Type: application/xml

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope">
  <s:Header>
  </s:Header>
  <s:Body>
    <AddCustomer>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
    </AddCustomer>
  </s:Body>
</s:Envelope>
```

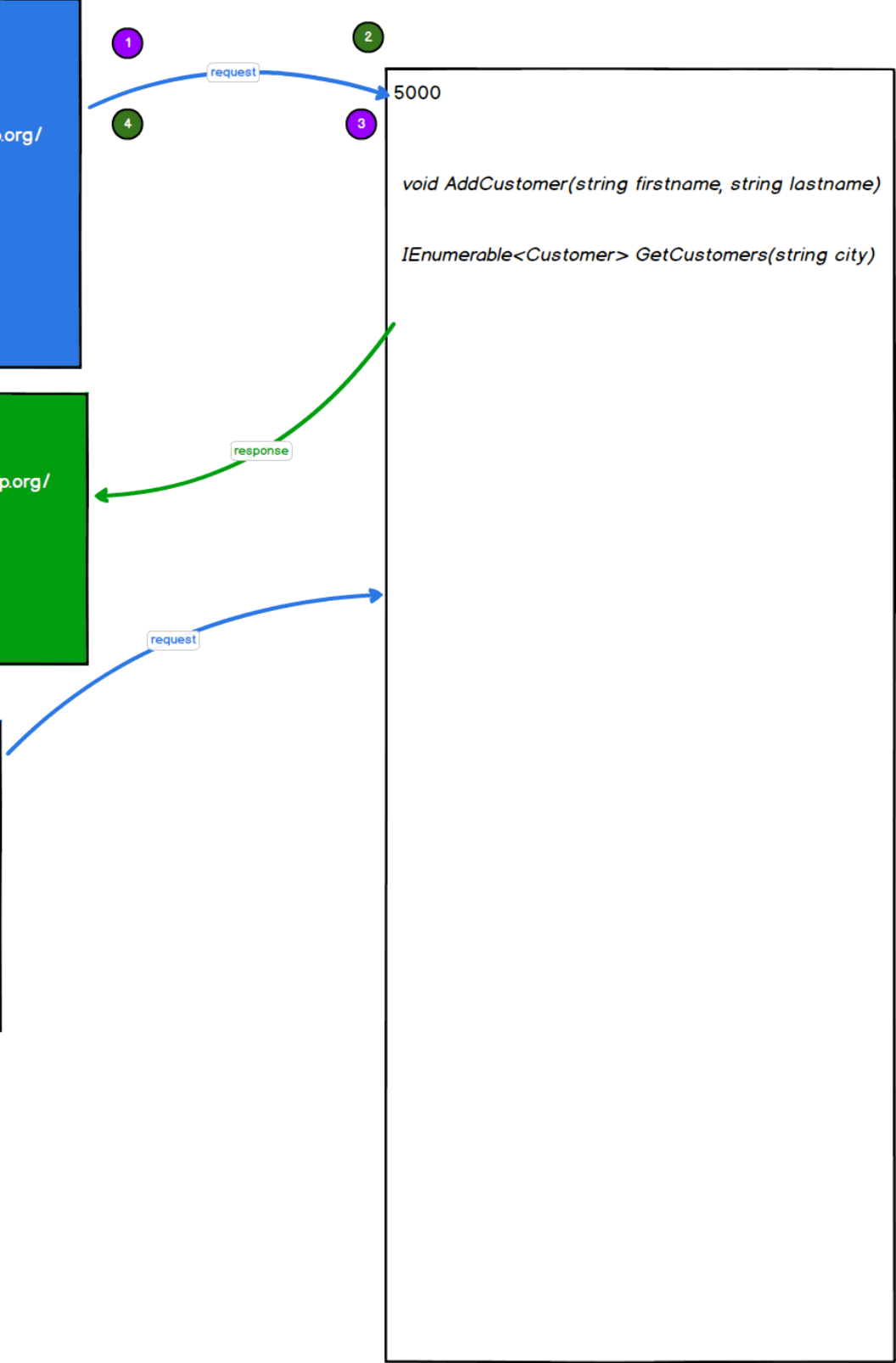
```
200 OK
Content-Type: application/xml

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope">
  <s:Header>
  </s:Header>
  <s:Body>
    <AddCustomerResponse>
    </AddCustomerResponse>
  </s:Body>
</s:Envelope>
```

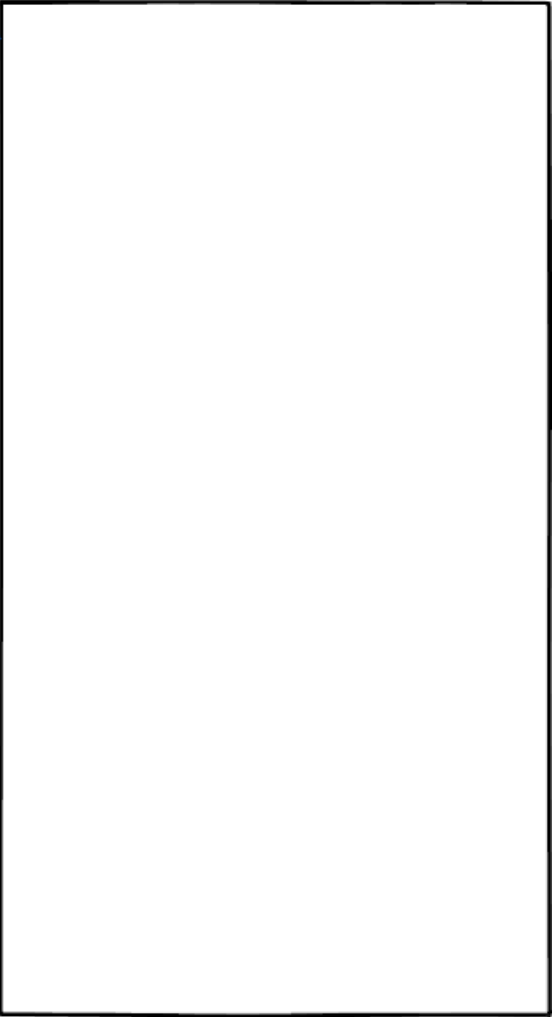
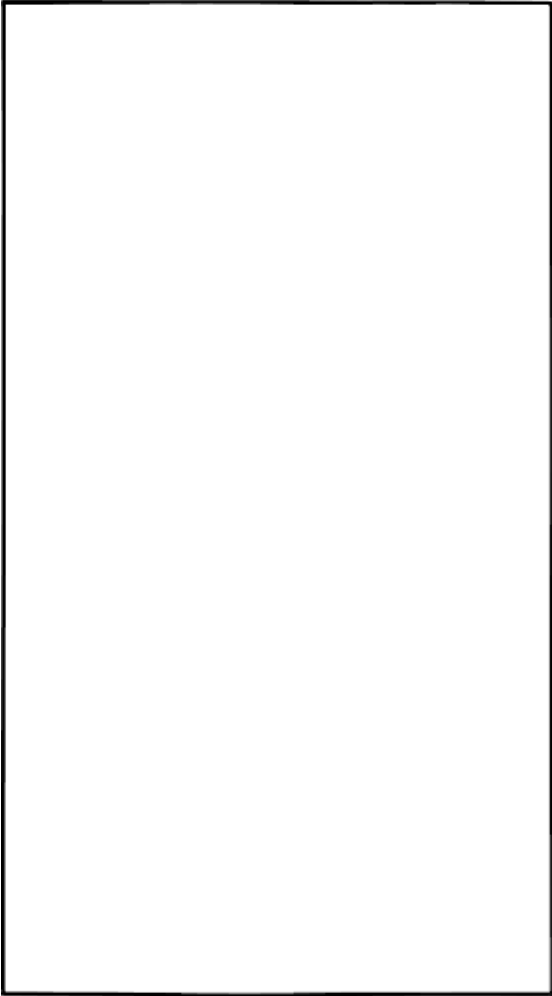
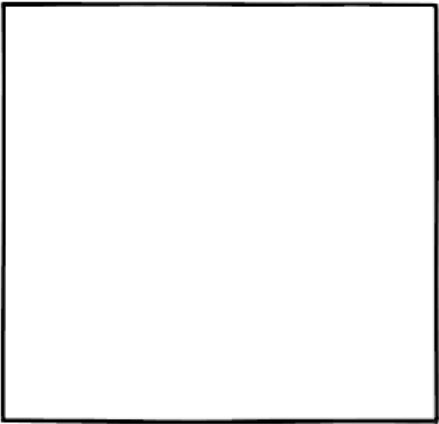
```
POST HelloService.svc HTTP/1.1
Host: localhost
Accept: application/xml
Content-Type: application/xml

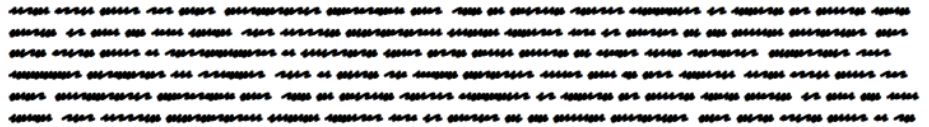
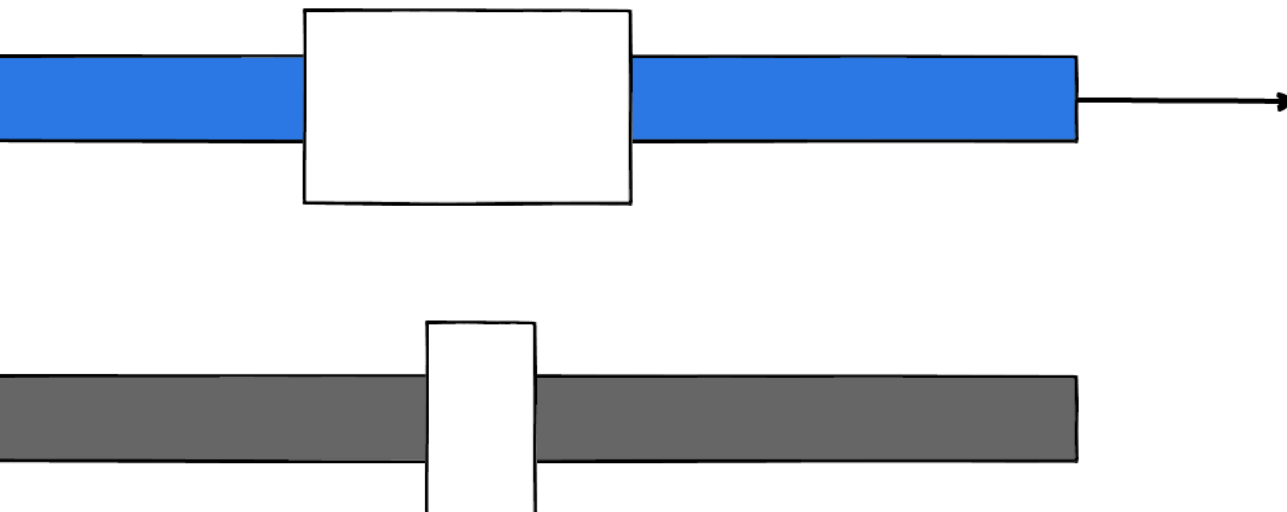
<xml>
  <GetCustomers>
    <City>Warsaw</City>
  </GetCustomers>
</xml>
```

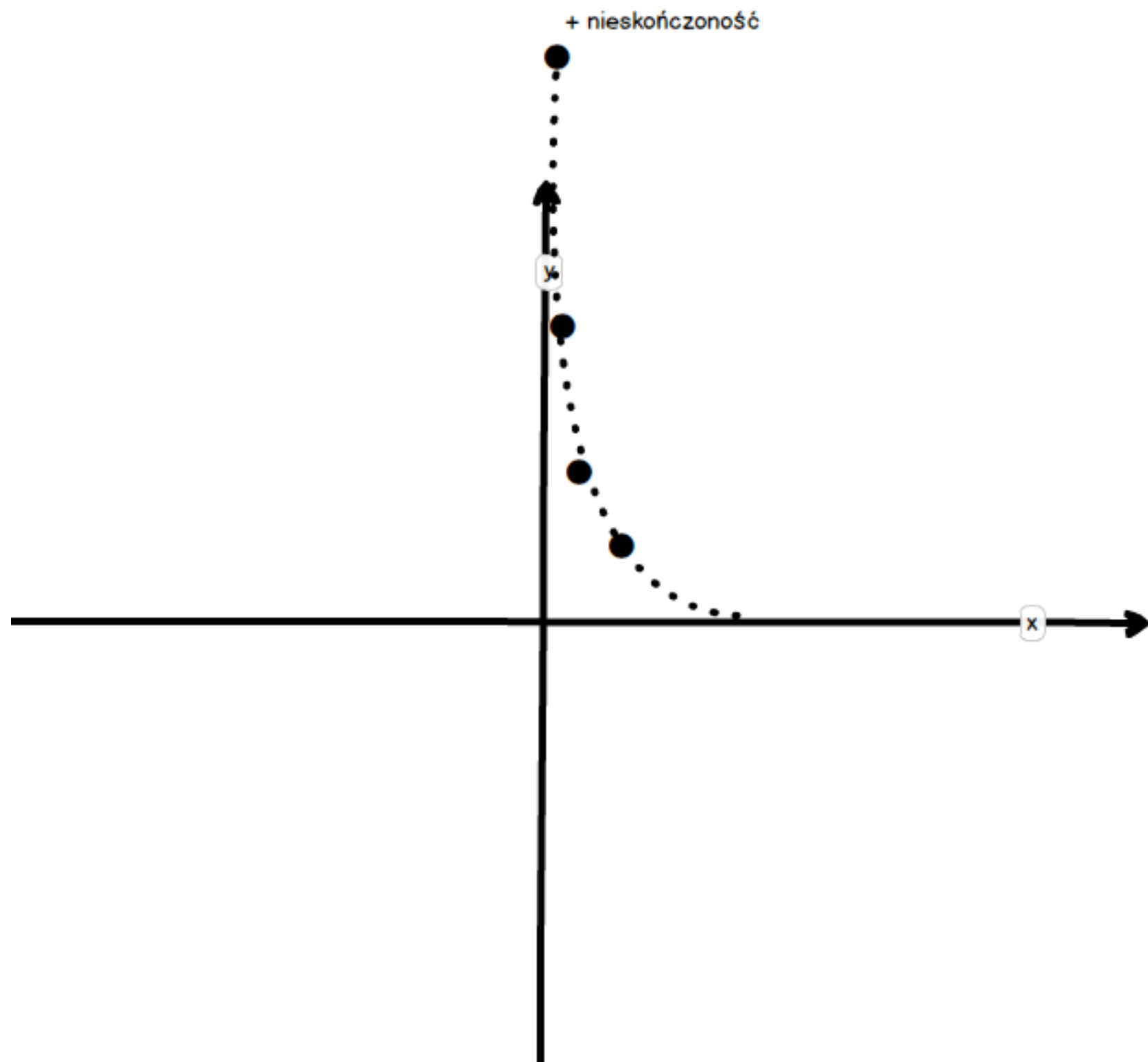
SOAP



Contract



[illegible][illegible][illegible][illegible]




```
{method} {resource} {protocol/version}
Host: {host}
Accept: {mime-type}
Accept-Language: {code}
{blank-line}
```

```
GET /customers/10 HTTP/1.1
Host: localhost
Accept: application/xml
```

```
200 OK
Content-Type: application/xml
Version: v1

<Customer>
  <FirstName>John</FirstName>
  <LastName>Smith</LastName>
</Customer>
```

```
GET /customers?City=Lódź&Street=Piotrkowska HTTP/1.1
Host: localhost
Accept: application/json
```

```
200 OK
Content-Type: application/json
Version: v1

[
  {
    "FirstName": "John",
    "LastName": "Smith",
  },
  {
    "FirstName": "Ann",
    "LastName": "Smith",
  }
]
```

```
POST /customers HTTP/1.1
Host: localhost
Content-Type: application/json

{
  "FirstName": "John",
  "LastName": "Smith",
}
```

```
PUT /customers/10 HTTP/1.1
Host: localhost
Content-Type: application/json

{
  "FirstName": "Ann",
  "LastName": "Smith",
}
```

```
DELETE /customers/10 HTTP/1.1
Host: localhost
```

REST API

GET api/customers

GET api/customers/10

GET api/customers/933304323

GET api/customers/Lódź

GET api/customers?city=Lódź&Street=Piotrkowska

GET api/customers/10/orders

POST api/customers

PUT api/customers/10

PATCH api/customers/10

DELETE api/customers/10

OPTIONS api/customers

method resource

GET - pobierz

POST - utwórz

PUT - zamień

PATCH - zmień

DELETE - usuń

HEAD - sprawdź czy istnieje

OPTIONS - dostępne operacje

GET api/customers/10?action=remove

GET api/customers/CalculateSalary?id=10

GET api/orders?customerId=10

~~GET posts?id=32103~~

GET posts/wcf/message-constructs.html

GET posts/wcf/data-constructs.html

GET posts/wpf/mvvm/delegate-command.html

```
POST HelloService.svc HTTP/1.1
Host: localhost
Accept: application/xml
Content-Type: application/xml

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope">
  <s:Header>
  </s:Header>
  <s:Body>
    <AddCustomer>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
    </AddCustomer>
  </s:Body>
</s:Envelope>
```

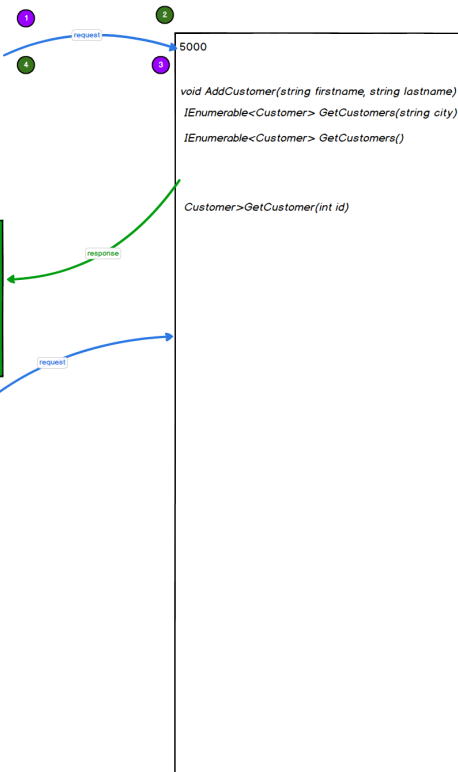
```
200 OK
Content-Type: application/xml

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope">
  <s:Header>
  </s:Header>
  <s:Body>
    <AddCustomerResponse>
    </AddCustomerResponse>
  </s:Body>
</s:Envelope>
```

```
POST HelloService.svc HTTP/1.1
Host: localhost
Accept: application/xml
Content-Type: application/xml

<xml>
  <GetCustomers>
    <City>Warsaw</City>
  </GetCustomers>
</xml>
```

SOAP



zapis funkcyjny

$f(x) = x + 2$ x należy do C

$f(x,y) = x + y$ x,y należy do C

$g(x) = x + 2$ x należy do C

$h(x) = f(x) + g(x)$ x należy do C

zapis imperatywny

`int F(int x) { return x + 2 }`

`int F(int x, int y) { return x + y }`

`int G(int x) { return x + 2 }`

`delegate (int x) { return x + 2 }`

rachunek lambda

$\lambda x \rightarrow x + 2$ x należy do C

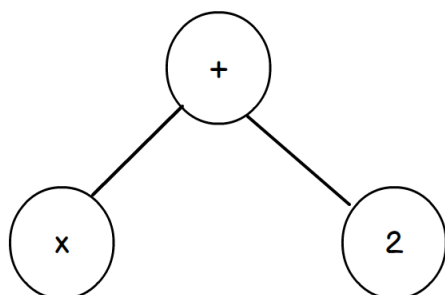
$\lambda x, y \rightarrow x + y$ x,y należy do C

`int x = 10;`

zapis deklaratywny

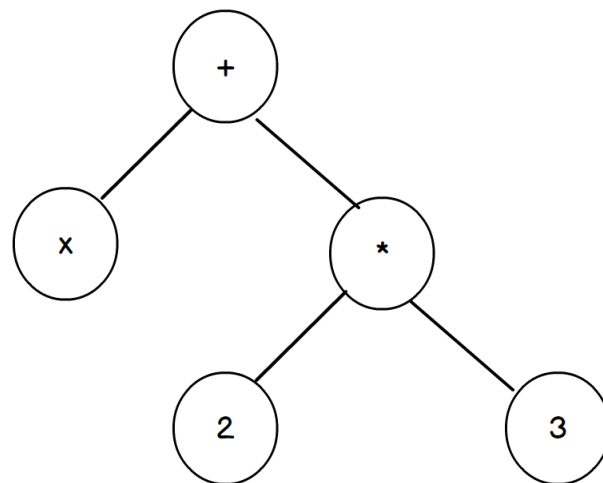
$x \Rightarrow x + 2$

$x + 2$

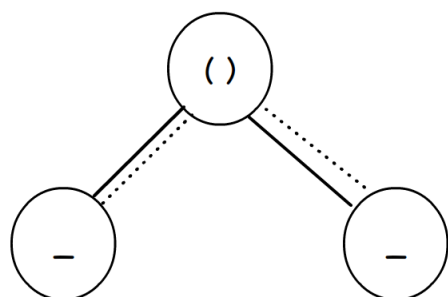


$x + 2 * 3$

Expression



operator



lewy operand

prawy operand

przykłady operatorów z jednym operandem:

$x++$

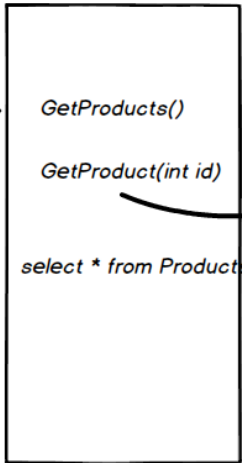
$++x$

$!x$

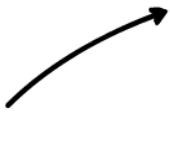
$x!$



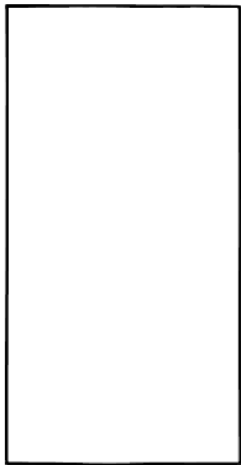
DbProductService



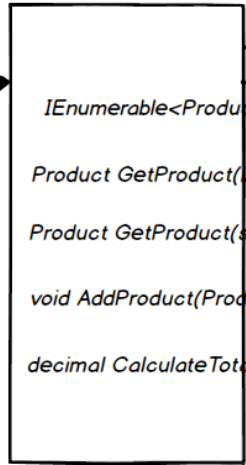
SQL Server



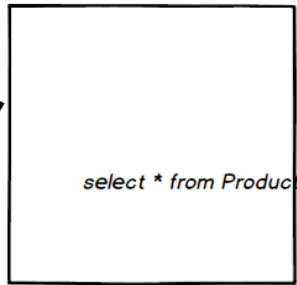
service



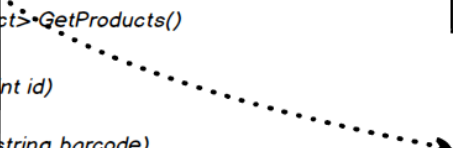
interface IProductService



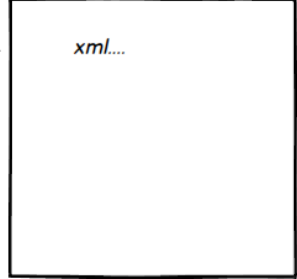
class DbProductService



SQL Server



class XmlFileProductService



products.xml

decimal CalculateUnitPrice(decimal amount, Wheather wheater)

interface IProductService

Get()

Calculate();

class FakeProductService : IProductService

public Get() { ... }

public Calculate();

private SaveToFile() { }

class DbProductService : IProductService

Get()

IProductService productService = new FakeProductService();

productService.Calculate();

~~*productService.Calculate();*~~

ProductsController(

IProductService productService = new FakeProductService(new ProductFaker());

IProductService productService = new DbProductService(new SqlConnection(string connectionString));

class Factory

T Create<T>()

IProductService productService = Factory.Create<IProductService>()

Container

Key	Value	Cycle Life
IProductService	FakeProductService	Singleton
Faker<Product>	ProductFaker	Singleton
SqlConnection	SqlConnection	Instance

Framework: Unity, AutoFac, Ninject, MEF

```
for(int i=0; i<customers.Count; i++)
```

```
foreach(Customer customer in customers) { .... }
```



3

IList

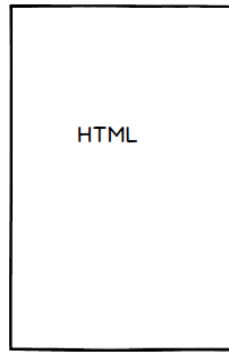
ICollection

IEnumerable

IEnumerator

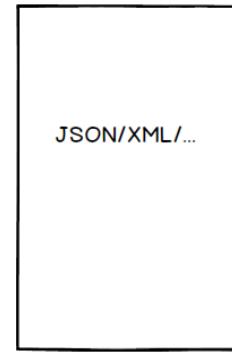
.NET Framework

Controller
ActionResult View()



ASP.NET MVC

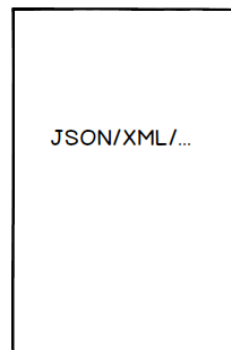
ApiController
IHttpActionResult Ok()



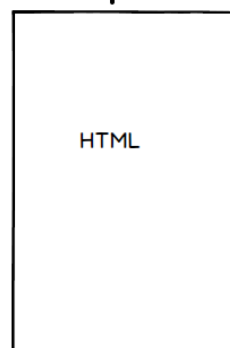
ASP.NET Web API 2

.NET Core

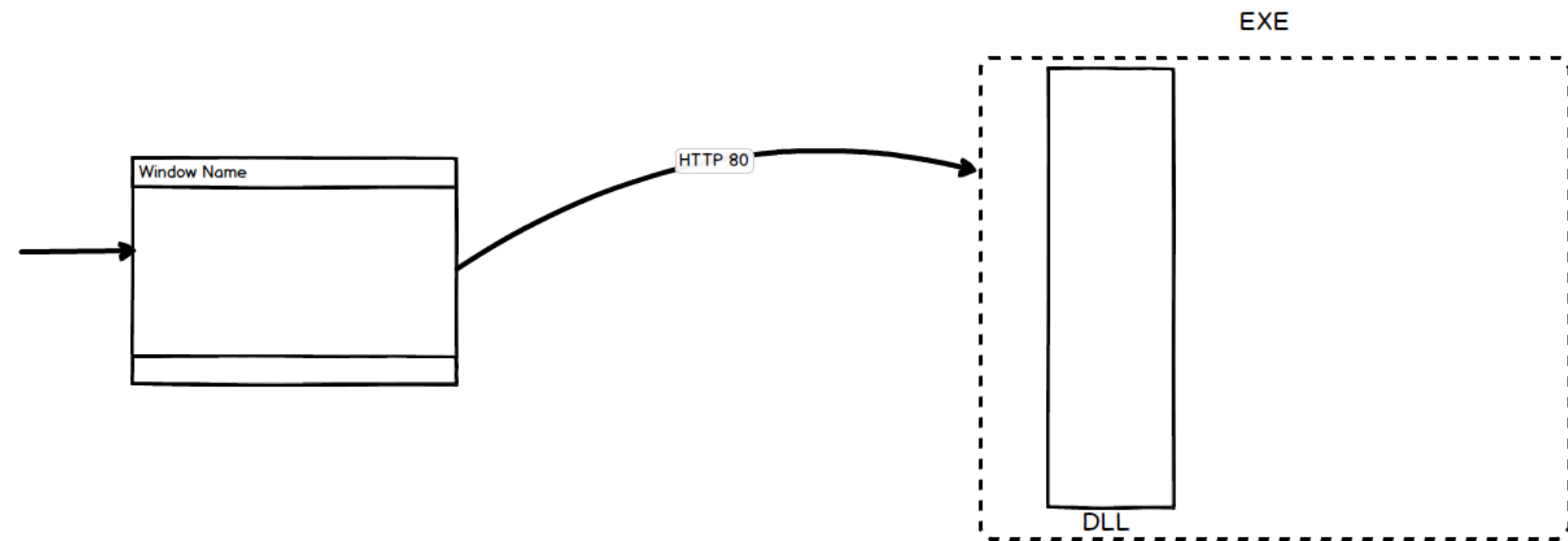
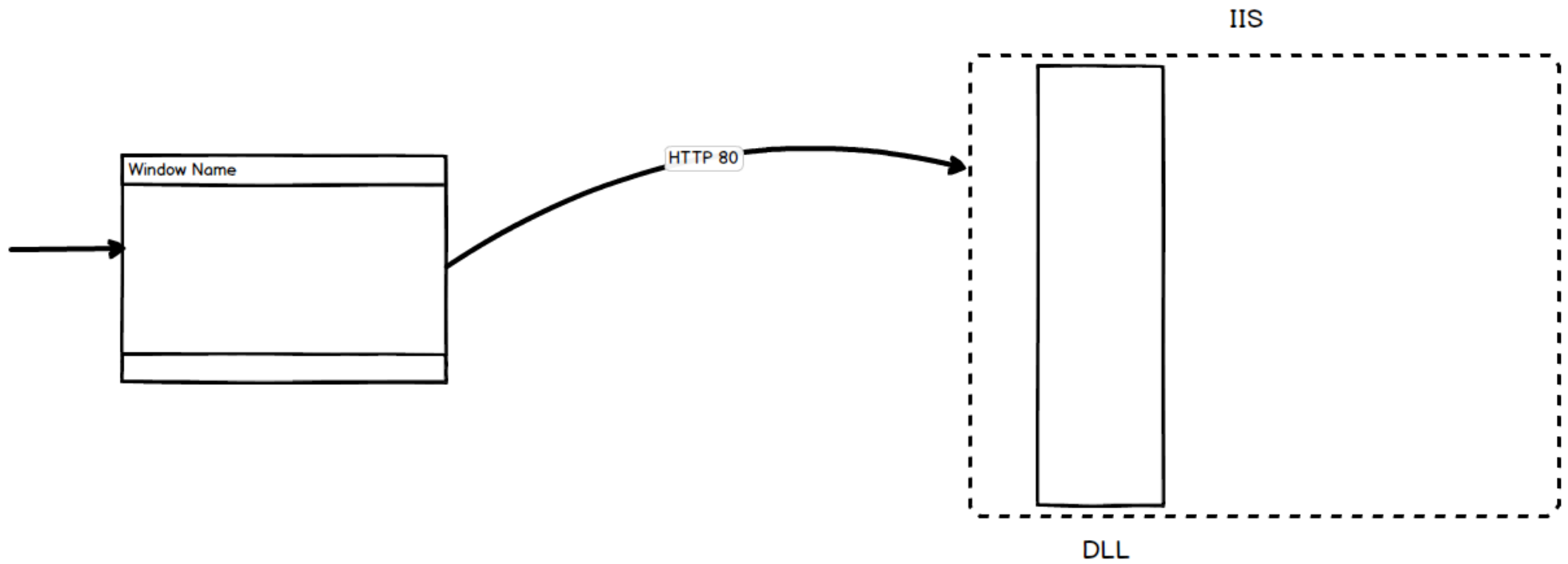
ControllerBase
IActionResult Ok()

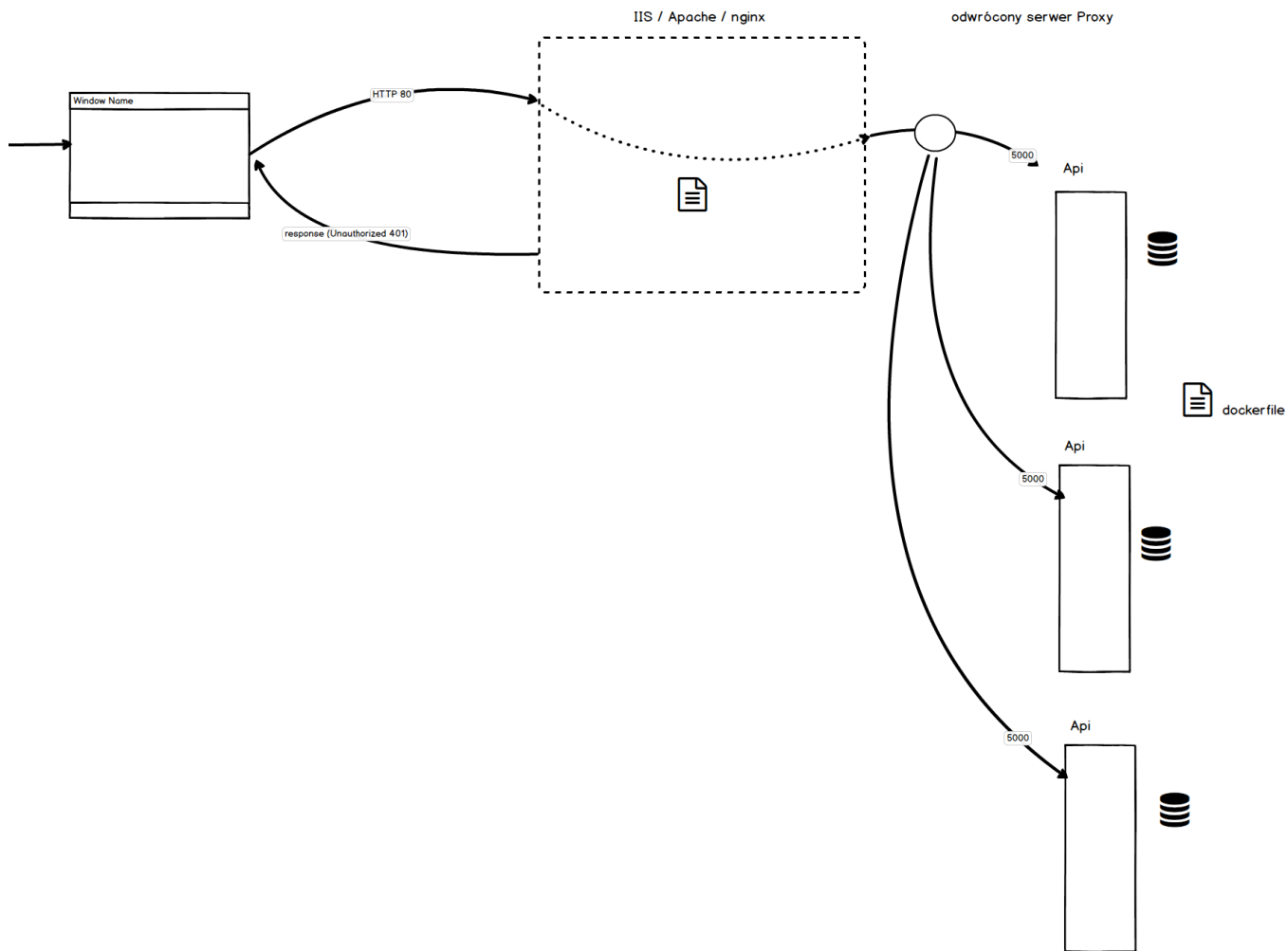
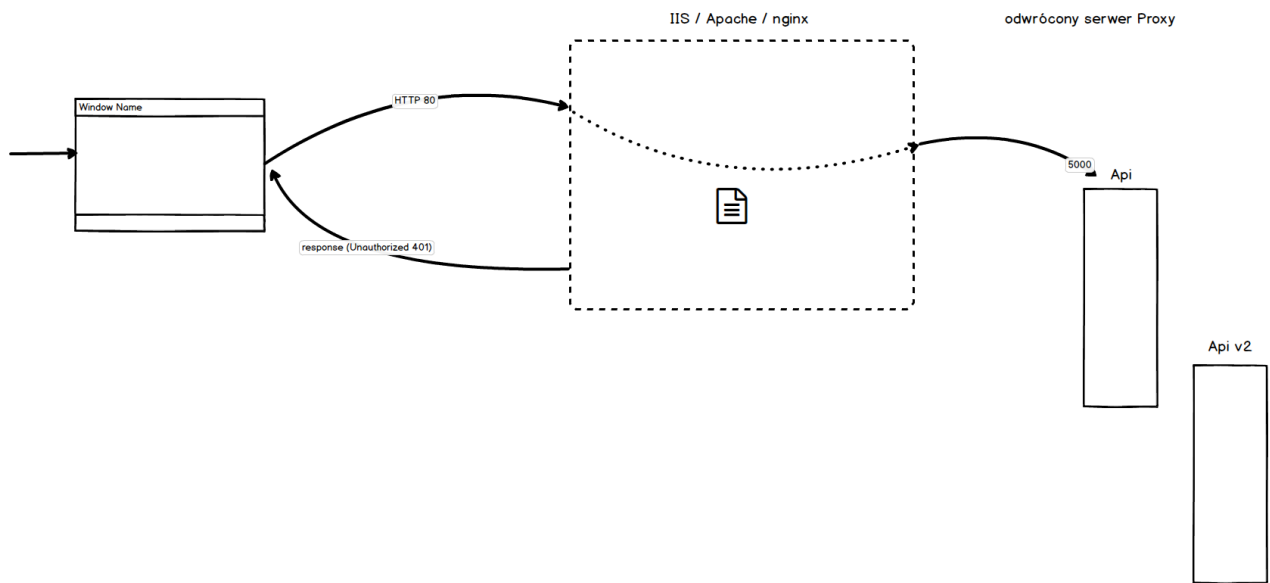
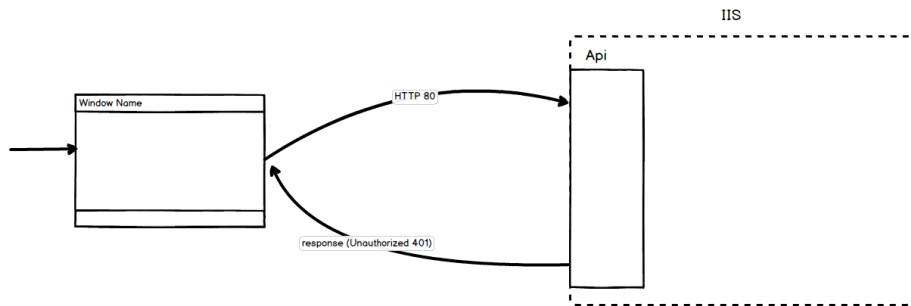


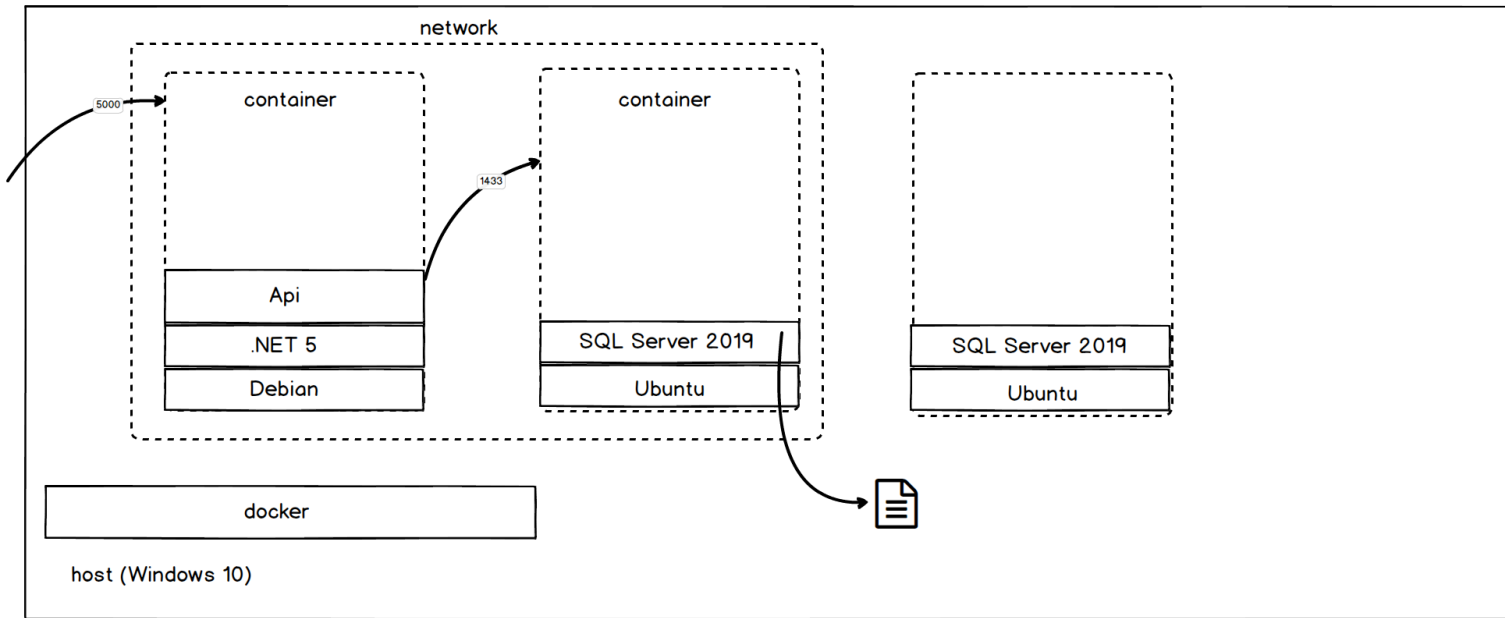
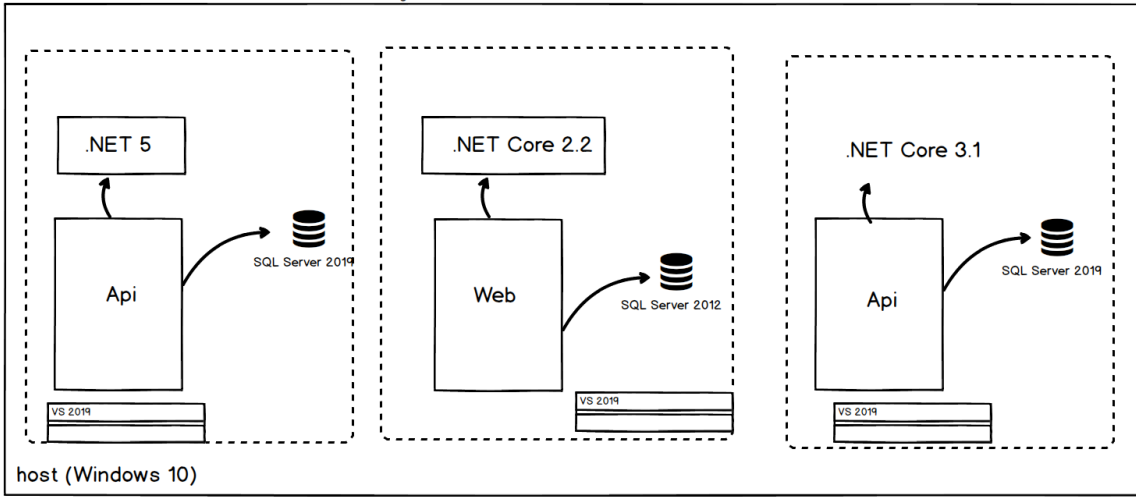
Controller
IActionResult View()



ASP.NET Core Web API







 dockerfile
image



docker-compose up / down