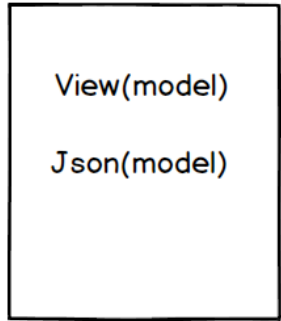
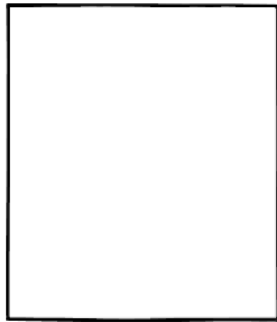


ASP.NET MVC
ASP.NET MVC

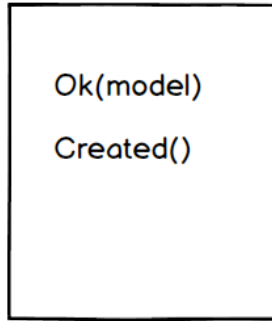


Controller



CustomersController

ASP.NET Web Api

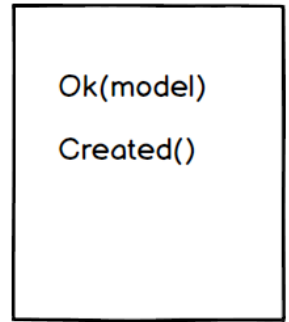


ApiController

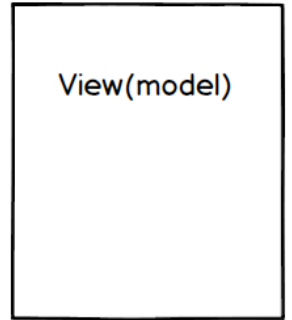


CustomersController

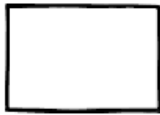
ASP.NET Core MVC



ControllerBase



Controller



GET api/customers

GET api/customers/100

POST api/customers

PUT api/customers/100

PATCH api/customers/100

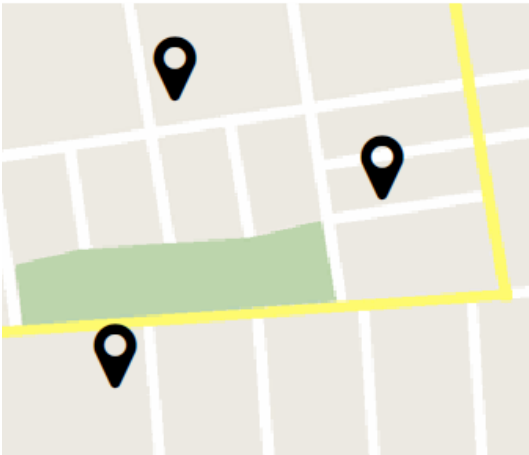
DELETE api/customers/100

GET api/customers/100/orders/last

GET api/customers?city=Poznan&street=Dworcowa

~~GET api/customers/pesel/105454545~~

~~GET api/customers?action=delete~~



GET api/customers/100

Accept: application/json

GET api/customers/100

Accept: image/png

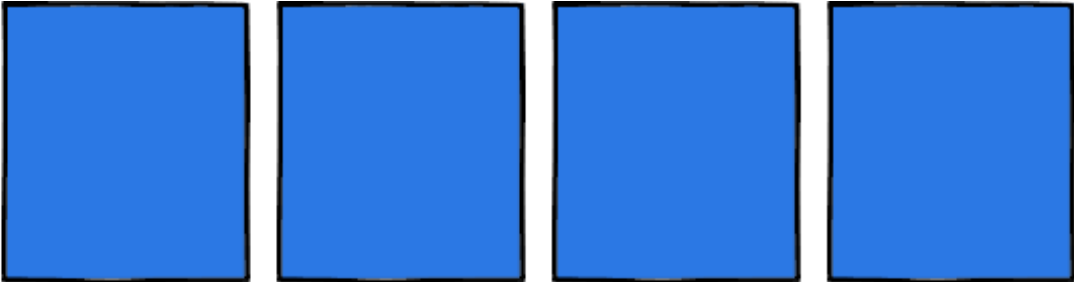
GET api/customers

Accept: kml

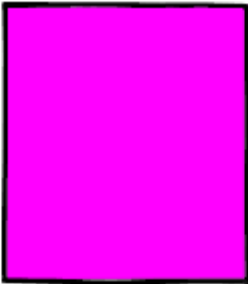


Key	Value	Cykl życia
IOrderService	FakeOrderService	Singleton
Faker<Customer>	CustomerFaker	
ICustomerService	FakeCustomerService	Transient
ICustomerService	DbCustomerService	Scoped

appsettings.json



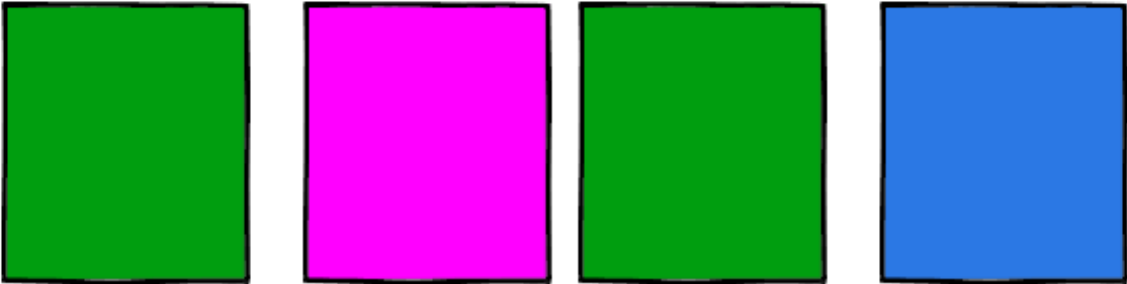
appsettings.development.json



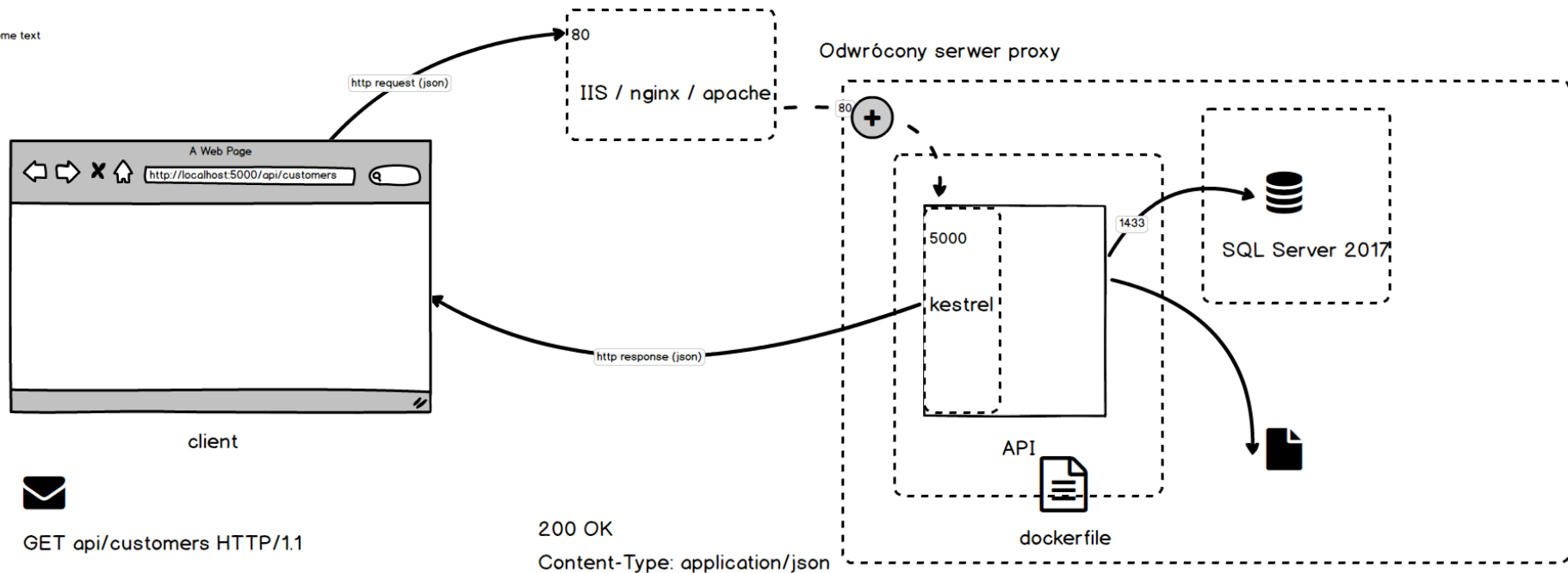
appsettings.xml



IConfiguration



Some text



GET api/customers HTTP/1.1

Host: localhost:5000

Accept: application/xml

POST api/customers HTTP/1.1

Host: localhost:5000

Content-Type: application/json

200 OK

Content-Type: application/json

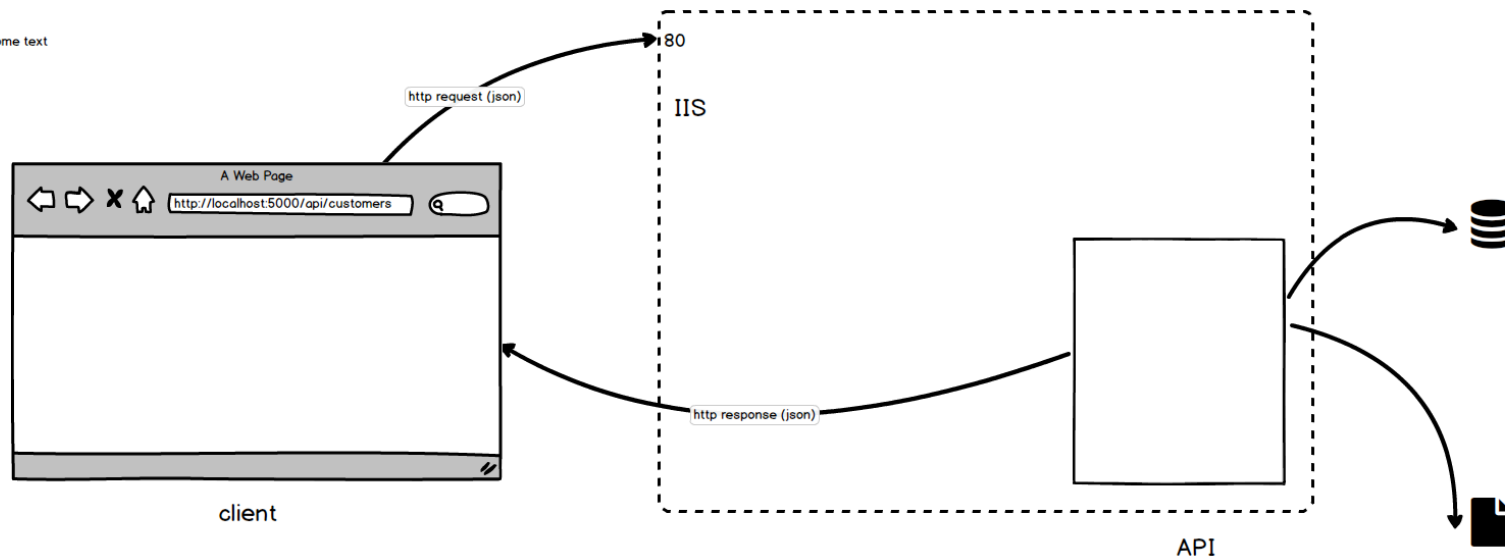


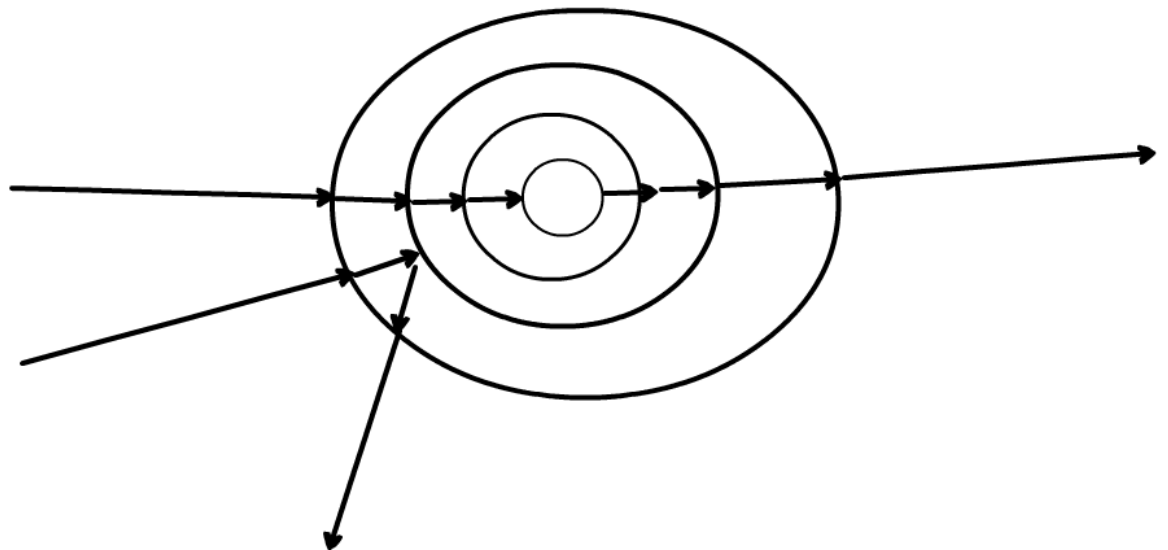
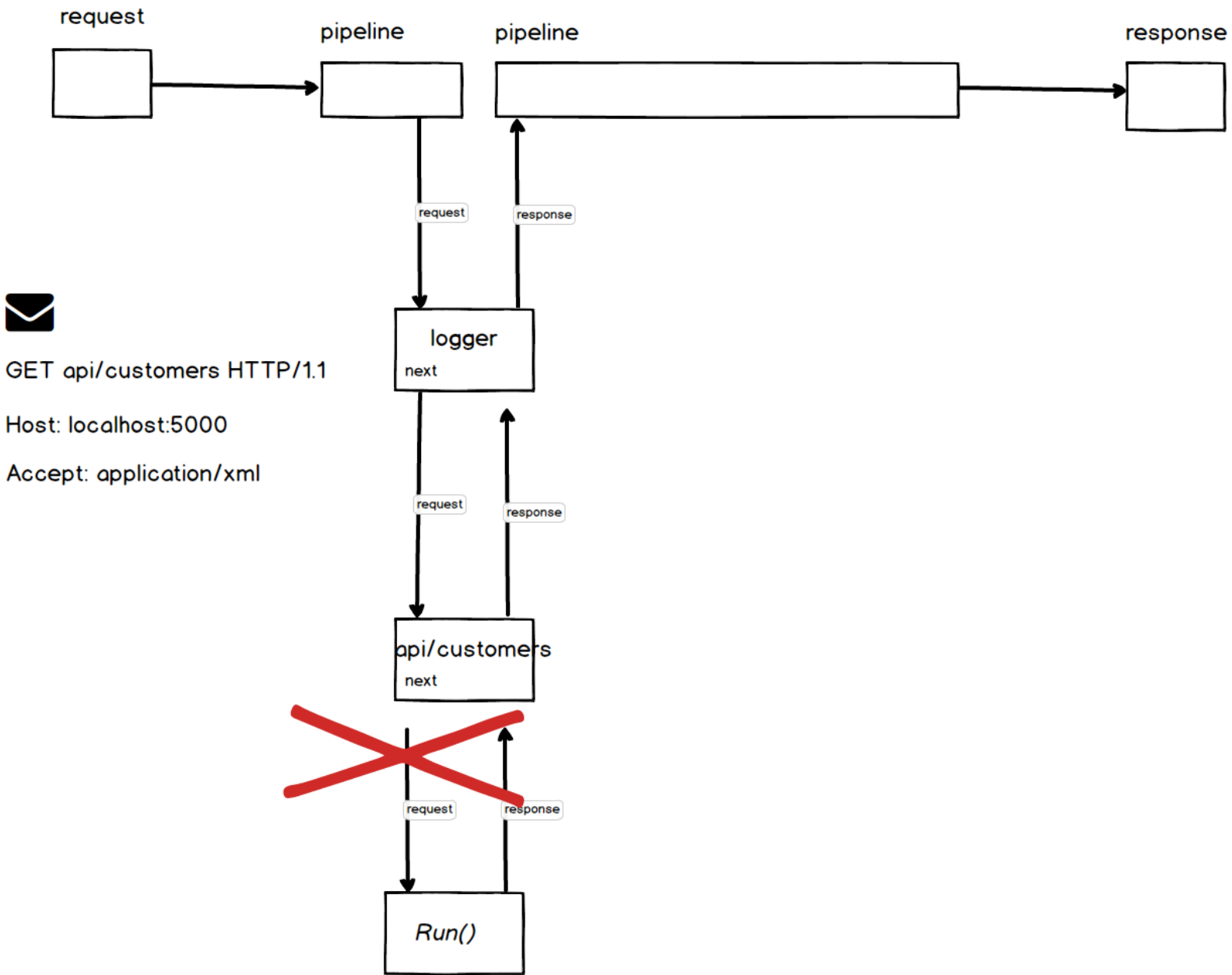
docker-compose.yml

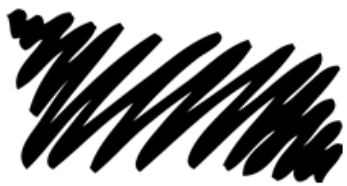
> docker-compose up

> docker-compose down

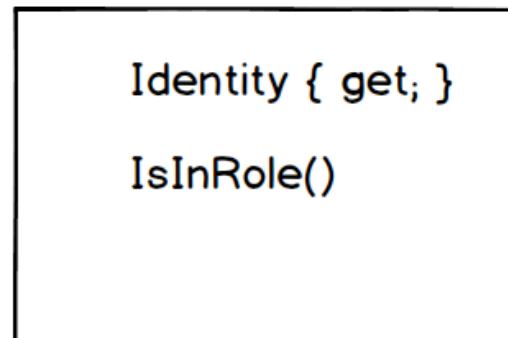
Some text



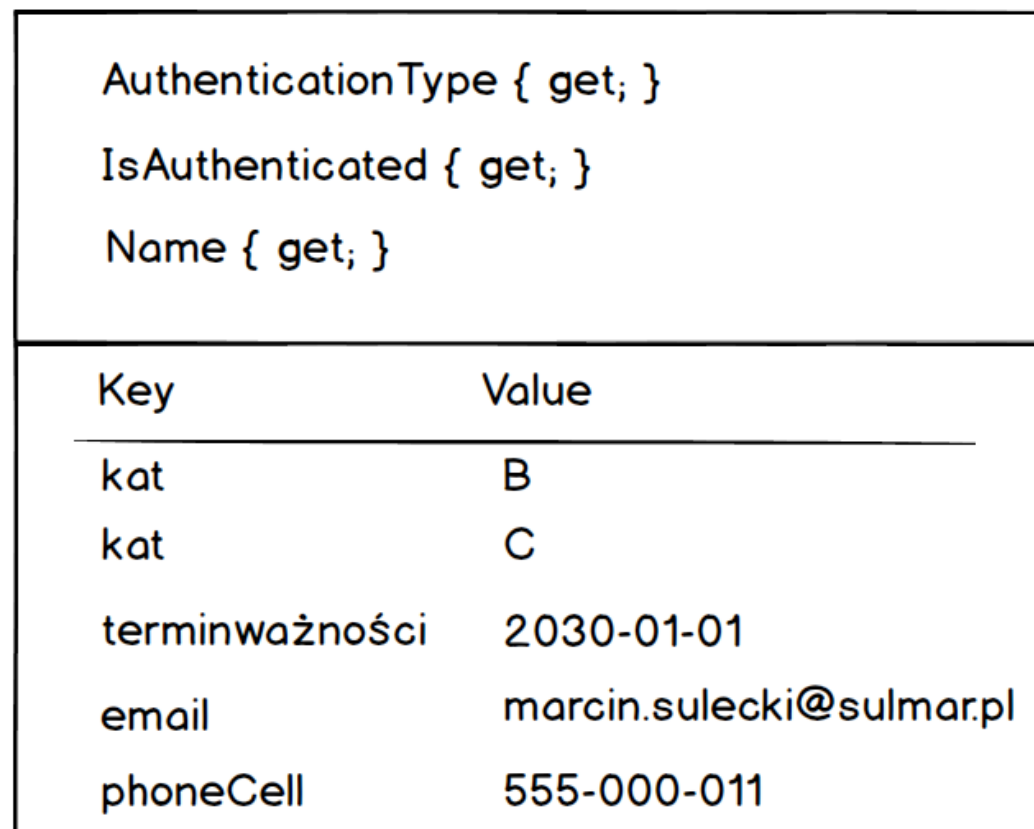




IPrincipal



IIdentity



ClaimsIdentity

OAuth 2.0

POST api/customers/createtoken

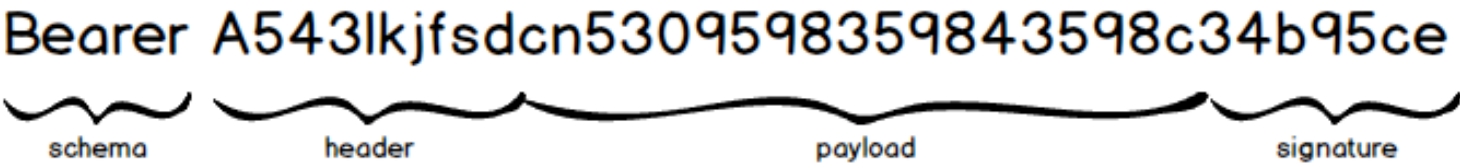
Content-Type: application/json

{ "login": "marcin", "password": "12345" }

A543lkjfsdcn5309598359843598c34b95ce

Authorization Bearer A543lkjfsdcn5309598359843598c34b95ce

JWT (Json Web Token)

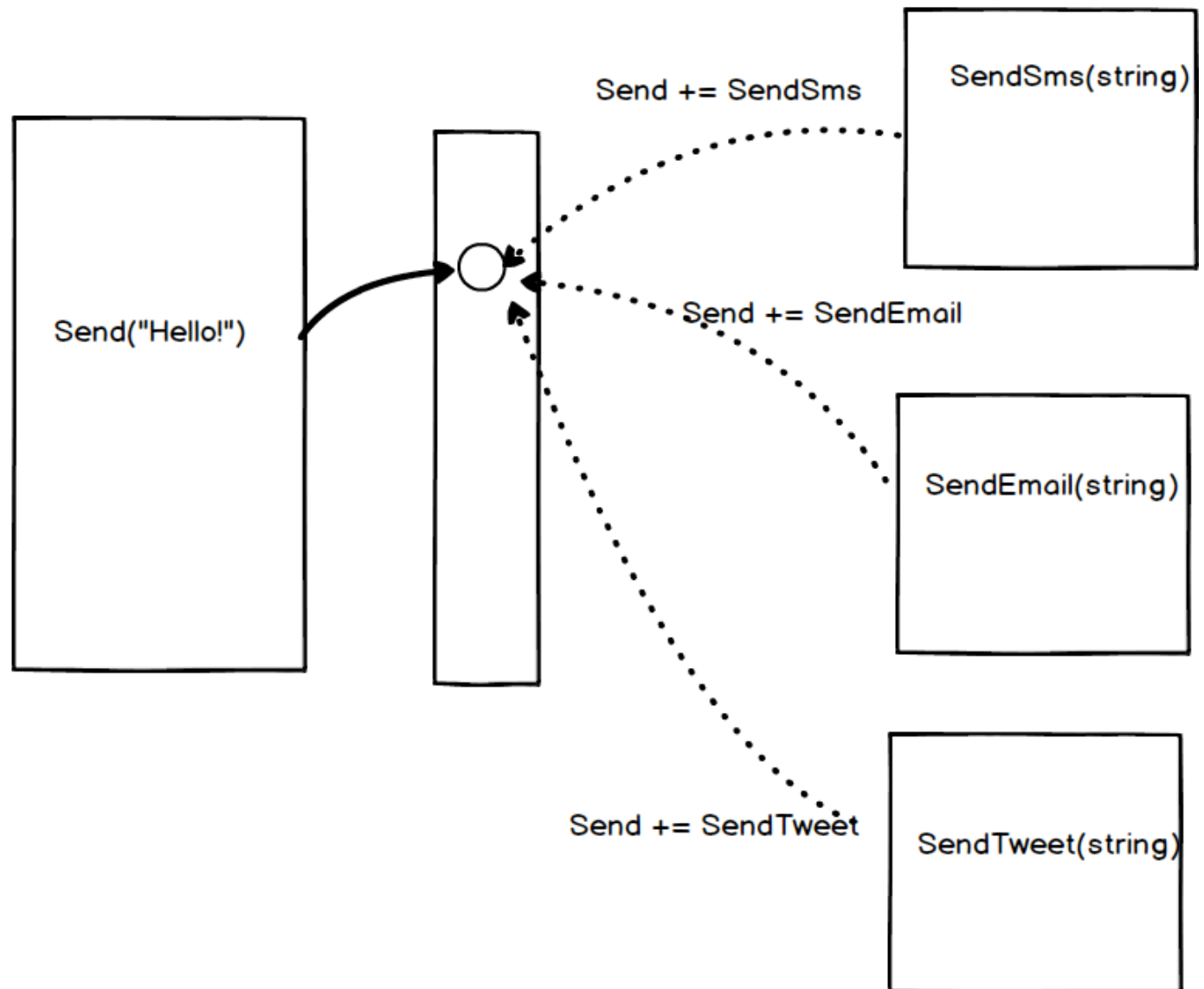
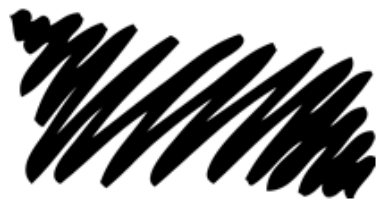


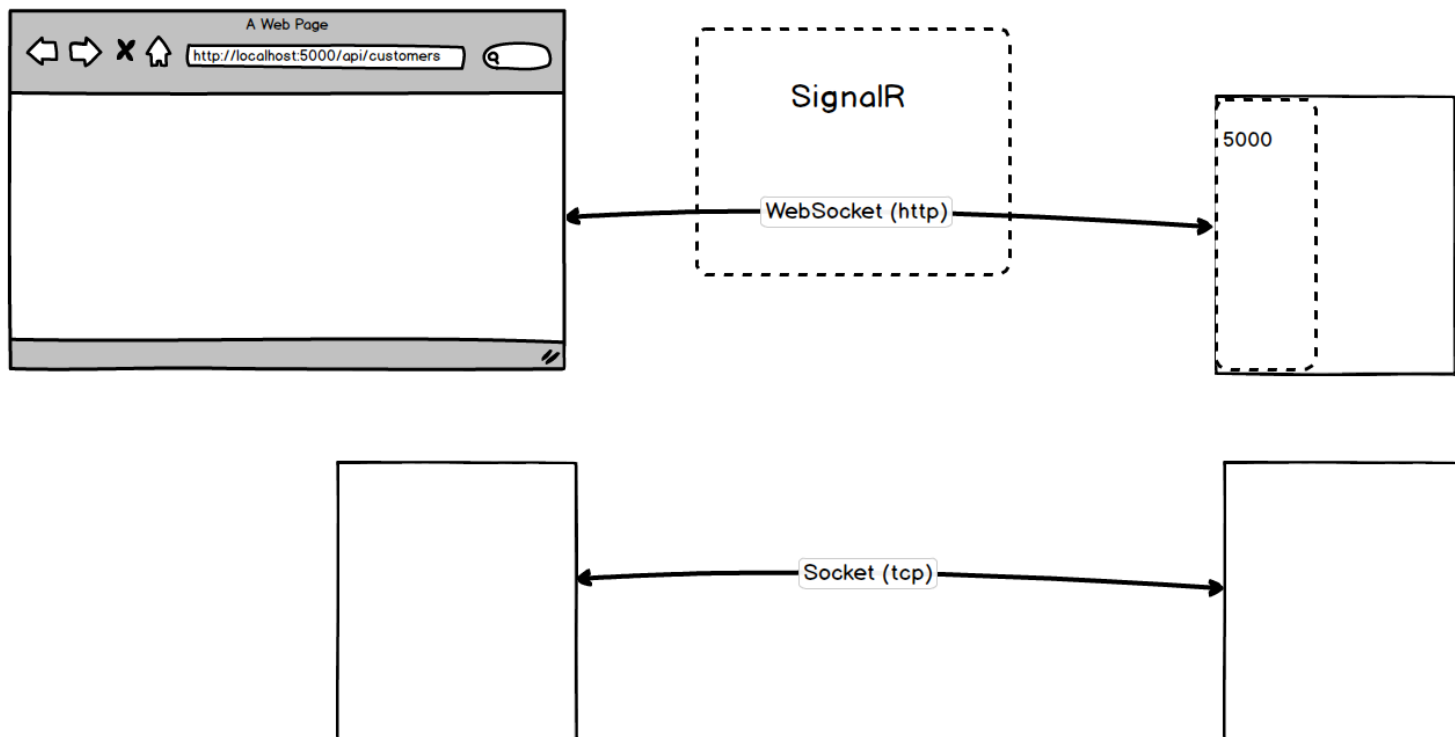
Basic

Key	Value
Authorization	<u>Basic</u> <u>base64(login:password)</u> schema parameter

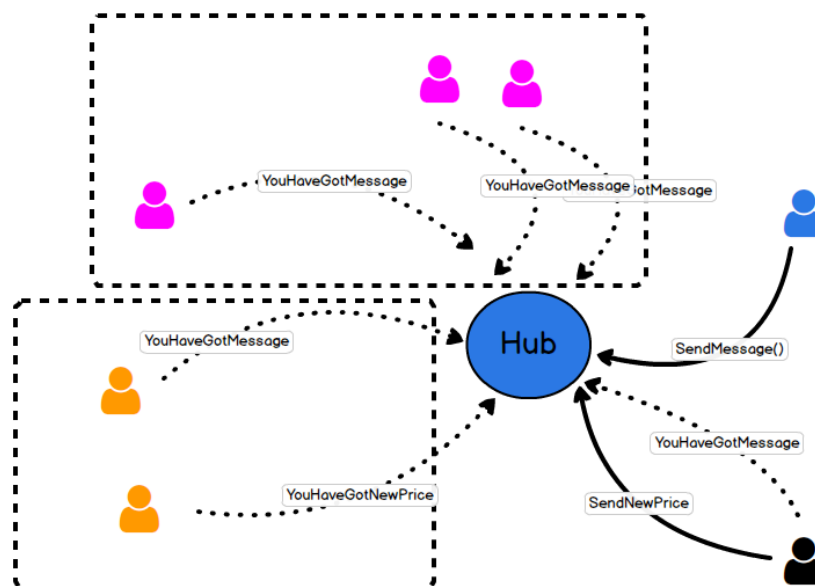
Bearer

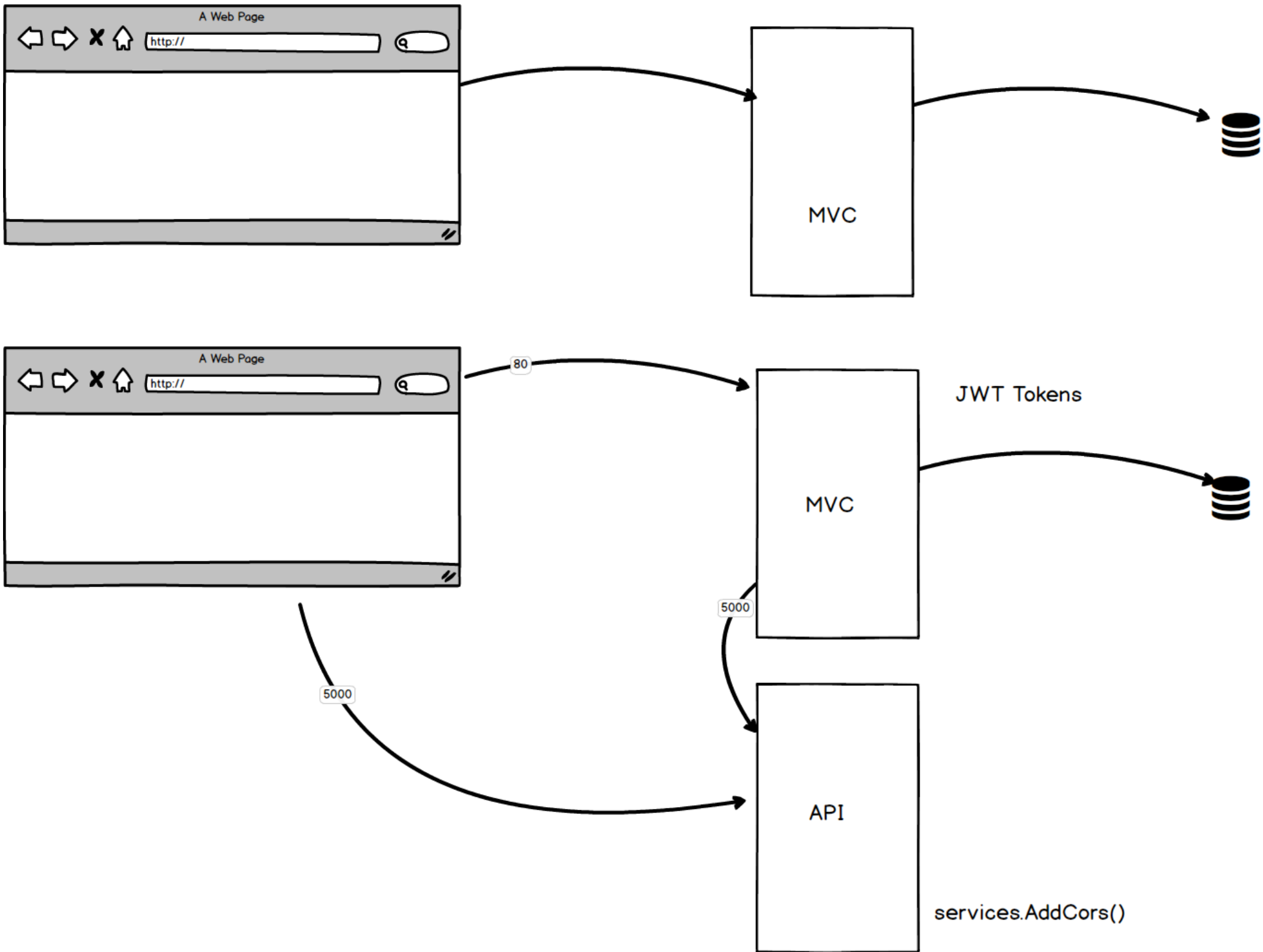
Key	Value
Authorization	<u>Bearer</u> <u>{token}</u> schema parameter

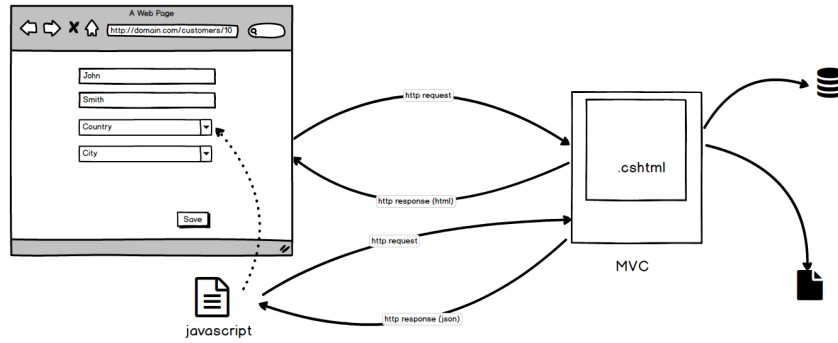
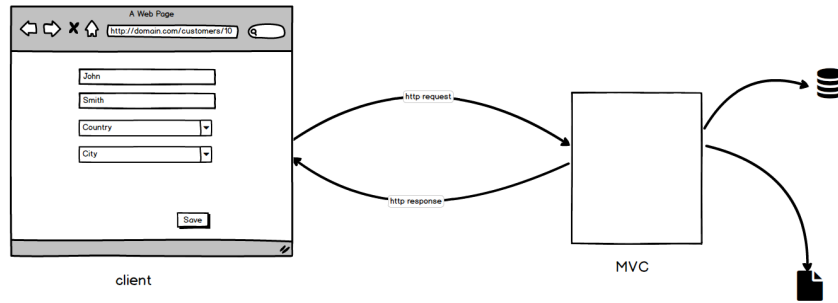




Wzorzec Mediator



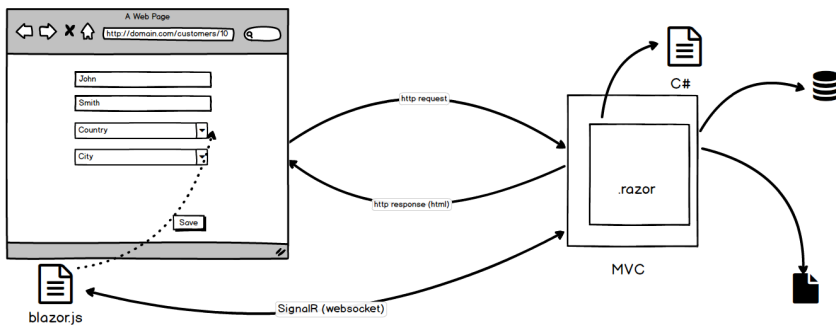




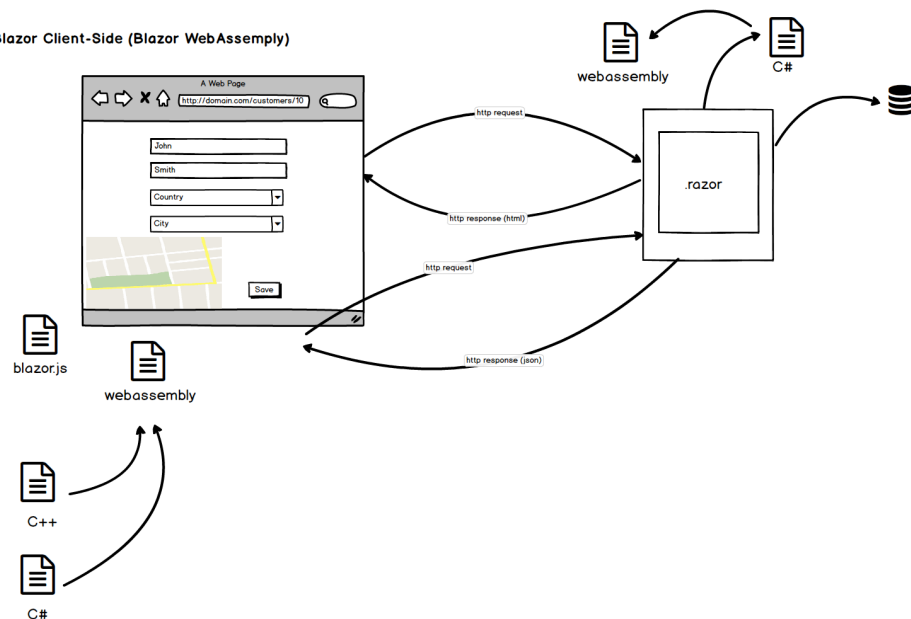
Angular (google) typescript -> javascript

React (facebook)

Blazor Server-Side (Blazor Server)



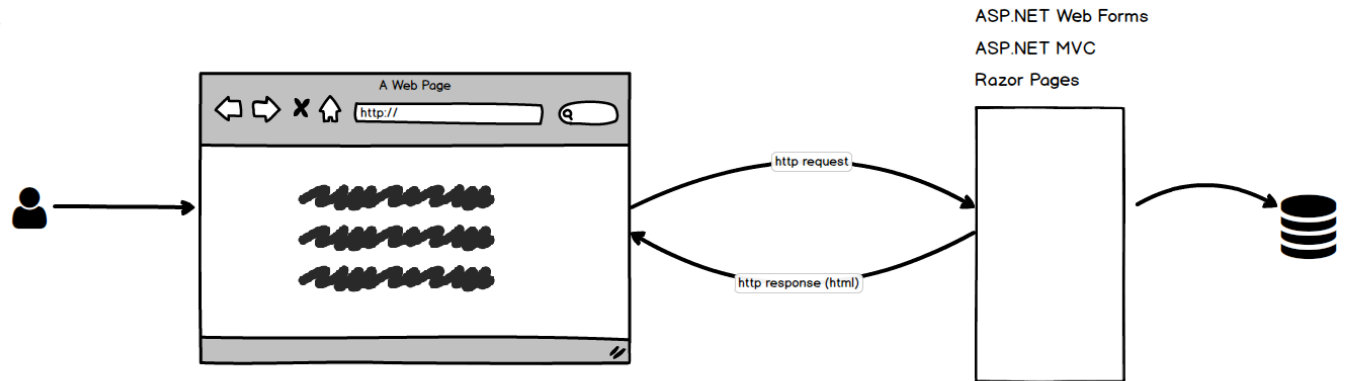
Blazor Client-Side (Blazor WebAssembly)



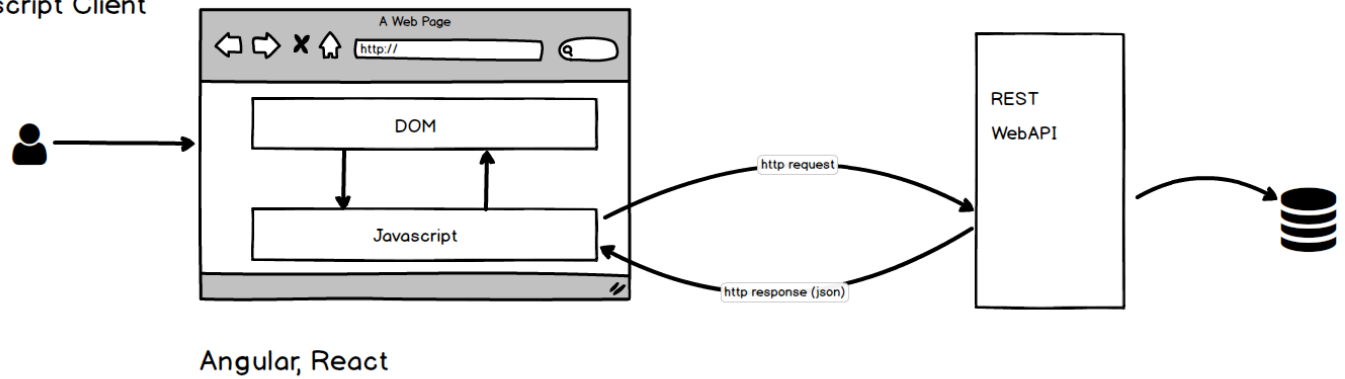


Blazor Architecture

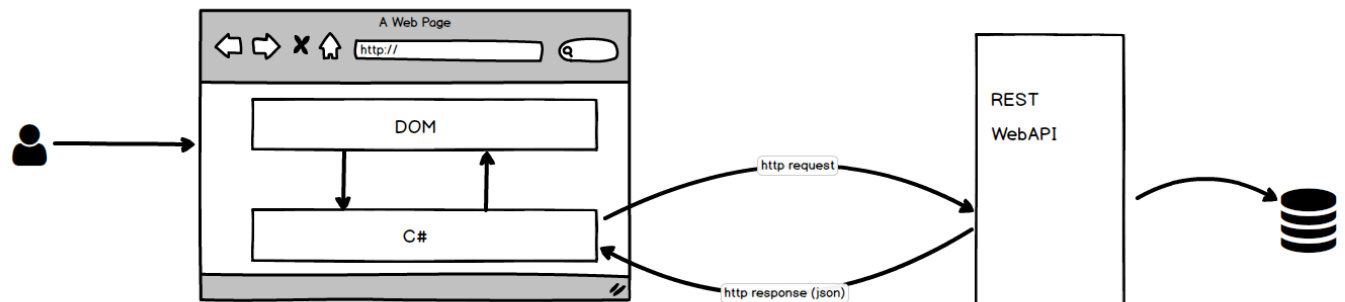
1. HTML



2. Javascript Client



3 Blazor Client Web Assembly



4 Blazor Server

