Consul is used for Service Discovery and a load balancing

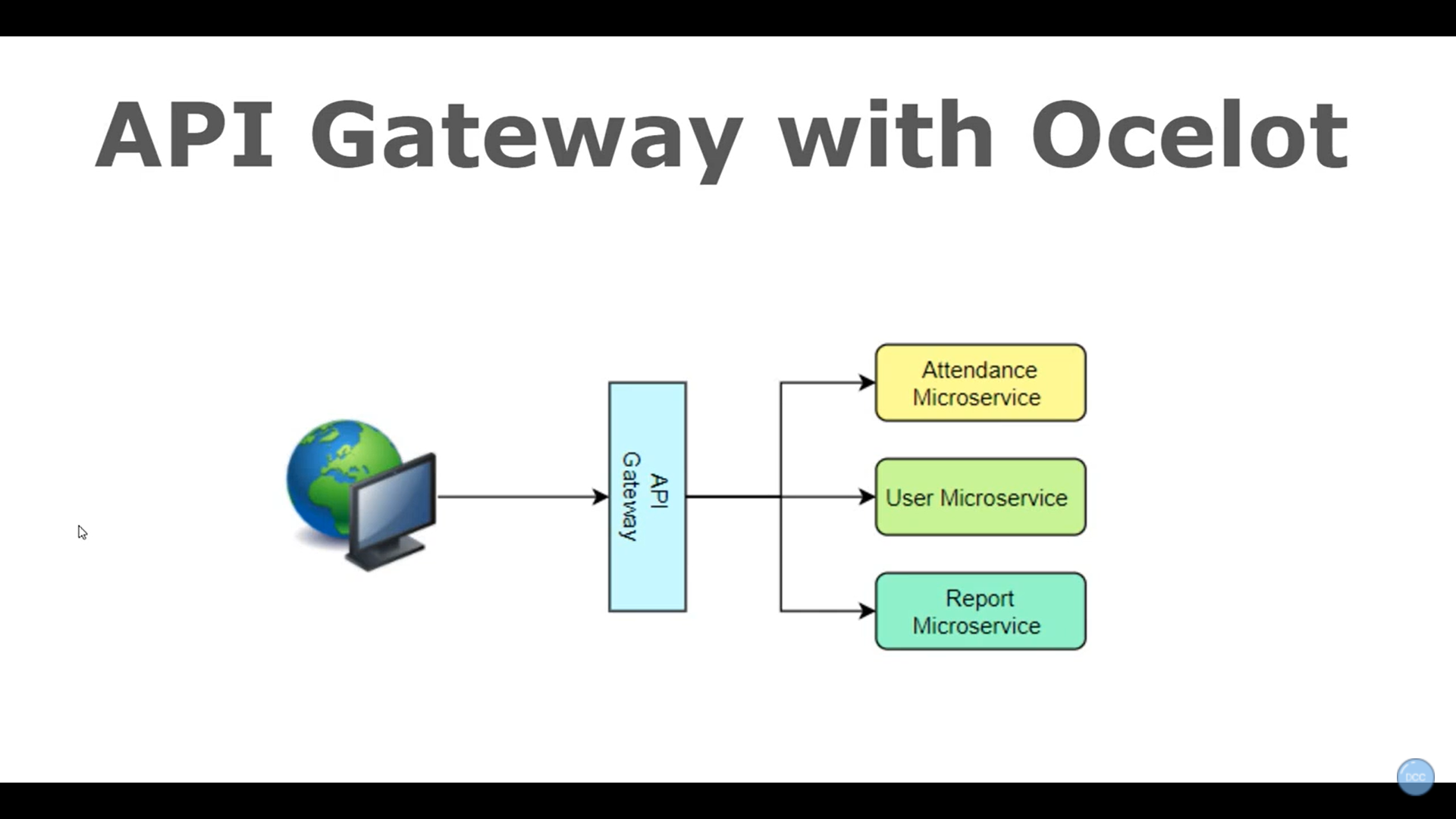
[Downloads | Consul by HashiCorp](https://www.consul.io/downloads)

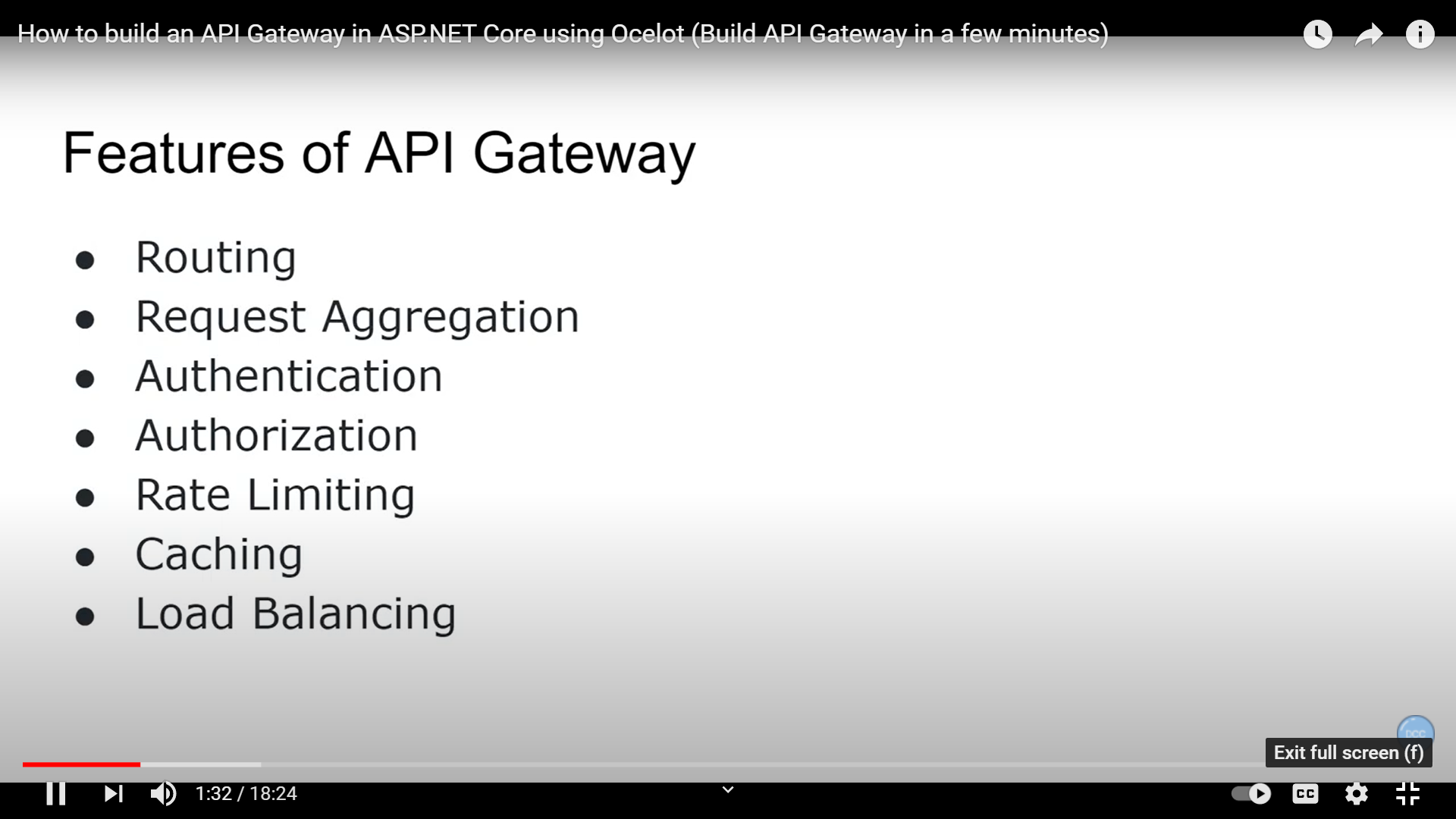
Copy the path  
Consul.exe agent –dev

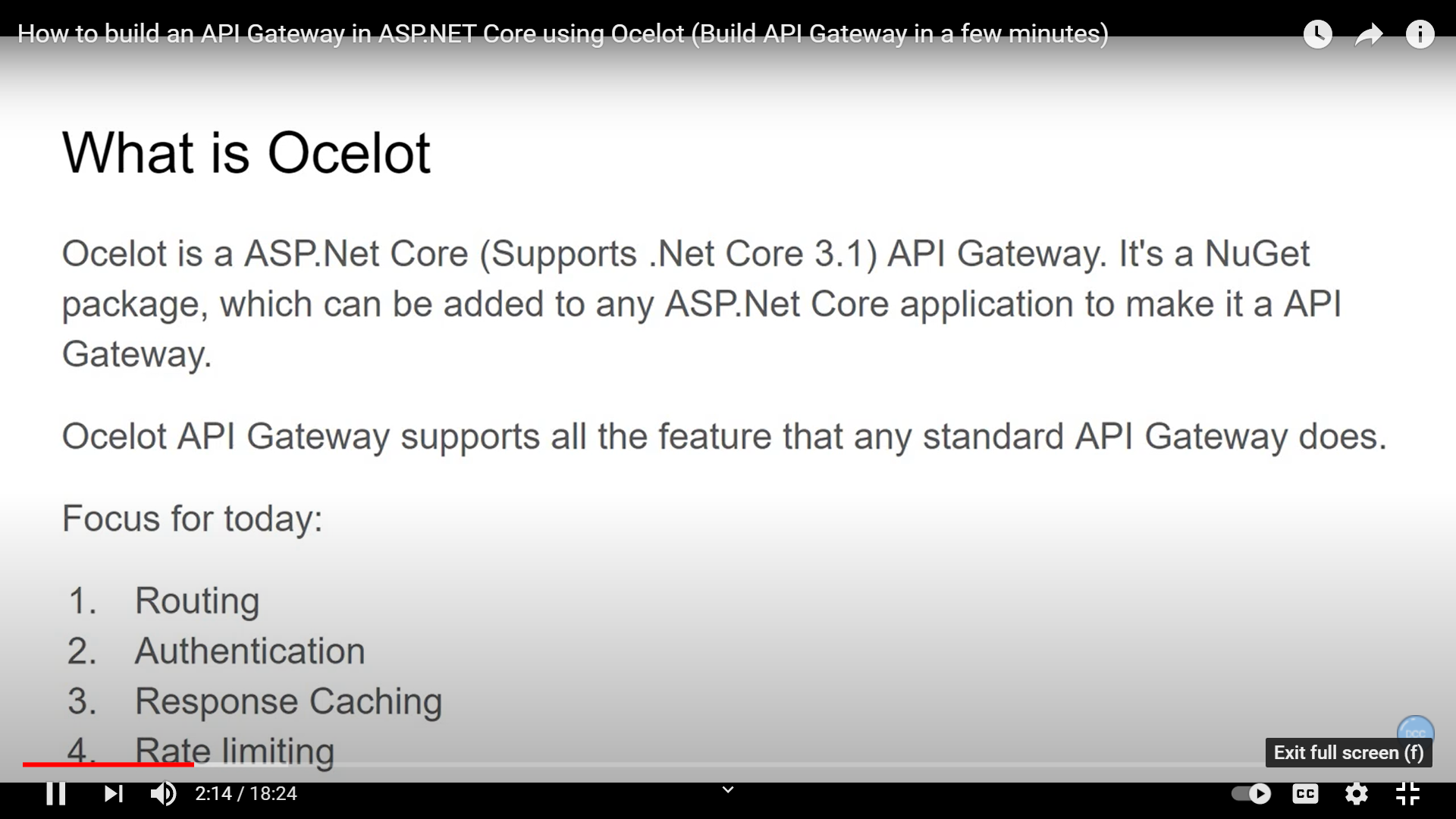
Its running on port 8500

<http://localhost:8500/ui/dc1/services>

Here we have three services . Accessing the services through external, it does not make sense that we should expose different URL’s or endpoints. API Gateway is a single entry point to all the services and based on the routes that we set, requests should be sent to particular service.







Purpose of API Gateway is to routé to the endpoint to a particular service endpoint

Create a service Web Api

Create an empty project file

In Program.cs file,

Step 1:

Add logging so that we get log messages

public static IHostBuilder CreateHostBuilder(string[] args) =>

Host.CreateDefaultBuilder(args)

.ConfigureWebHostDefaults(webBuilder =>

{

webBuilder.UseStartup<Startup>();

})

**.ConfigureLogging(logging => logging.AddConsole());**

Step 2: Install Ocelot package

Step 3: Go to Start.cs file

public void ConfigureServices(IServiceCollection services)

{

**services.AddOcelot();**

}

Step 4:

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseRouting();

app.UseEndpoints(endpoints =>

{

endpoints.MapGet("/", async context =>

{

await context.Response.WriteAsync("Hello World!");

});

});

**app.UseOcelot().Wait();**

}

Step 5: Add ocelot.json file

{

"Routes": [

{

"DownstreamPathTemplate": "/weatherforecast",

"DownstreamScheme": "https",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "44391"

}

],

"UpstreamPathTemplate": "/api/WeatherForecast",

"UpstreamHttpMethod": [ "Get" ]

},

{

"DownstreamPathTemplate": "/api/students",

"DownstreamScheme": "https",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "44391"

}

],

"UpstreamPathTemplate": "/api/students",

"UpstreamHttpMethod": [ "Get" ]

}

],

"GlobalConfiguration": {

"BaseUrl": "https://localhost:5021"

}

}

BaseUrl is where API Gateway will be running

DownStream is where you actually have to come

UpStream is the path that is mapped to API Gateway (Endpoint for API Gateway)

Step 6 : Add this json file , go to Program.cs file

public static IHostBuilder CreateHostBuilder(string[] args) =>

Host.CreateDefaultBuilder(args)

.ConfigureWebHostDefaults(webBuilder =>

{

**var env = Environment.GetEnvironmentVariable("ASPNETCORE\_ENVIRONMENT");**

**webBuilder.UseStartup<Startup>();**

**webBuilder.ConfigureAppConfiguration(config => config.AddJsonFile($"ocelot.{env}.json"));**

})

.ConfigureLogging(logging=>logging.AddConsole());

**RateLimiting :**

{

"Routes": [

{

"DownstreamPathTemplate": "/weatherforecast",

"DownstreamScheme": "https",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "44391"

}

],

"UpstreamPathTemplate": "/api/WeatherForecast",

"UpstreamHttpMethod": [ "Get" ]

},

{

"DownstreamPathTemplate": "/api/students",

"DownstreamScheme": "https",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "44391"

}

],

"UpstreamPathTemplate": "/api/students",

"UpstreamHttpMethod": [ "Get" ],

**"RateLimitOptions": {**

**"ClientWhiteList": [],**

**"EnableRateLimiting": true,**

**"Period": "5s",**

**"PeriodTimespan": 1,**

**"Limit": 1**

**}**

}

],

"GlobalConfiguration": {

"BaseUrl": "https://localhost:5021"

}

}

If you request too fast, it will give error 409

Only 1 request every 5 sec. Max ,

PeriodTimespan how much time you wait for next request

Next is Response cache, for that we need package

Ocelot.Cache.CacheManager

After that go to start.cs file

**Add services.AddOcelot().AddCacheManager(settings=>settings.WithDictionaryHandle());**

**Then go to ocelot.dev.json file,**

{

"Routes": [

{

"DownstreamPathTemplate": "/weatherforecast",

"DownstreamScheme": "https",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "44391"

}

],

"UpstreamPathTemplate": "/api/WeatherForecast",

"UpstreamHttpMethod": [ "Get" ]

},

{

"DownstreamPathTemplate": "/api/students",

"DownstreamScheme": "https",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "44391"

}

],

"UpstreamPathTemplate": "/api/students",

"UpstreamHttpMethod": [ "Get" ],

**"RateLimitOptions": {**

**"ClientWhiteList": [],**

**"EnableRateLimiting": true,**

**"Period": "5s",**

**"PeriodTimespan": 1,**

**"Limit": 1**

**},**

**“FileCacheOptions”: {“TtlSeconds”:30}**

}

],

"GlobalConfiguration": {

"BaseUrl": "https://localhost:5021"

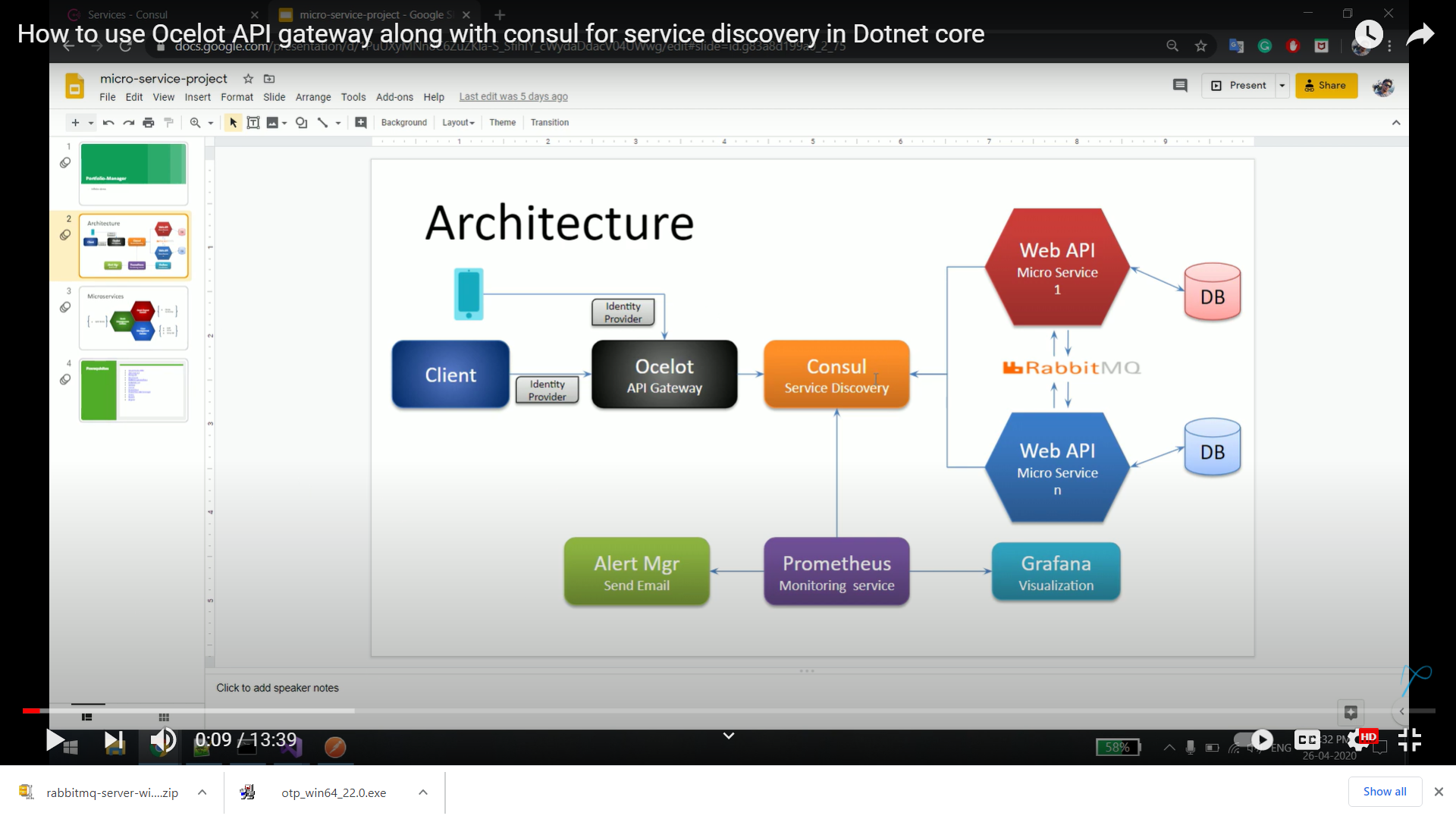
}

}

To Access Swagger

<https://localhost:44351/swagger/index.html>

Ocelot & Consul together



We need to packages

Ocelot

Ocelot.Provider.Consul

Add ocelot.json file in API Gateway

{

"ReRoutes": [

{

"UseServiceDiscovery": true,

"DownstreamPathTemplate": "/{url}",

"DownstreamScheme": "http",

"ServiceName": "students",

"LoadBalancerOptions": {

"Type": "LeastConnection"

},

"UpstreamPathTemplate": "/gateway/um/{url}",

"UpstreamHttpMethod": [ "Get" ],

"ReRoutesCaseSensitive": false

},

{

"UseServiceDiscovery": true,

"DownstreamPathTemplate": "/{url}",

"DownstreamScheme": "http",

"ServiceName": "empployees",

"LoadBalancerOptions": {

"Type": "LeastConnection"

},

"UpstreamPathTemplate": "/gateway/src/{url}",

"UpstreamHttpMethod": [ "Get" ],

"ReRoutesCaseSensitive": false

}

],

"GlobalConfiguration": {

"ServiceDiscoveryProvider": {

"Host": "localhost",

"Port": 8500,

"Type": "Consul"

}

}

}

Add in Api Gateway, start.cs file

public void ConfigureServices(IServiceCollection services)

{

**services.AddOcelot()**

**.AddConsul();**

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseRouting();

app.UseEndpoints(endpoints =>

{

endpoints.MapGet("/", async context =>

{

await context.Response.WriteAsync("Hello World!");

});

});

**app.UseOcelot().Wait();**

}

}

UpStream : Endpoint of API Gateway

DownStream : The Actual Path of Service

[**https://localhost:5021/api/emps**](https://localhost:5021/api/emps)

[**http://localhost:59751/api/employees**](http://localhost:59751/api/employees)

[**https://localhost:5021/api/weather**](https://localhost:5021/api/weather)

<http://localhost:59751/weatherforecast>

**https://localhost:5021/api/library**

http://localhost:58763/api/libraries