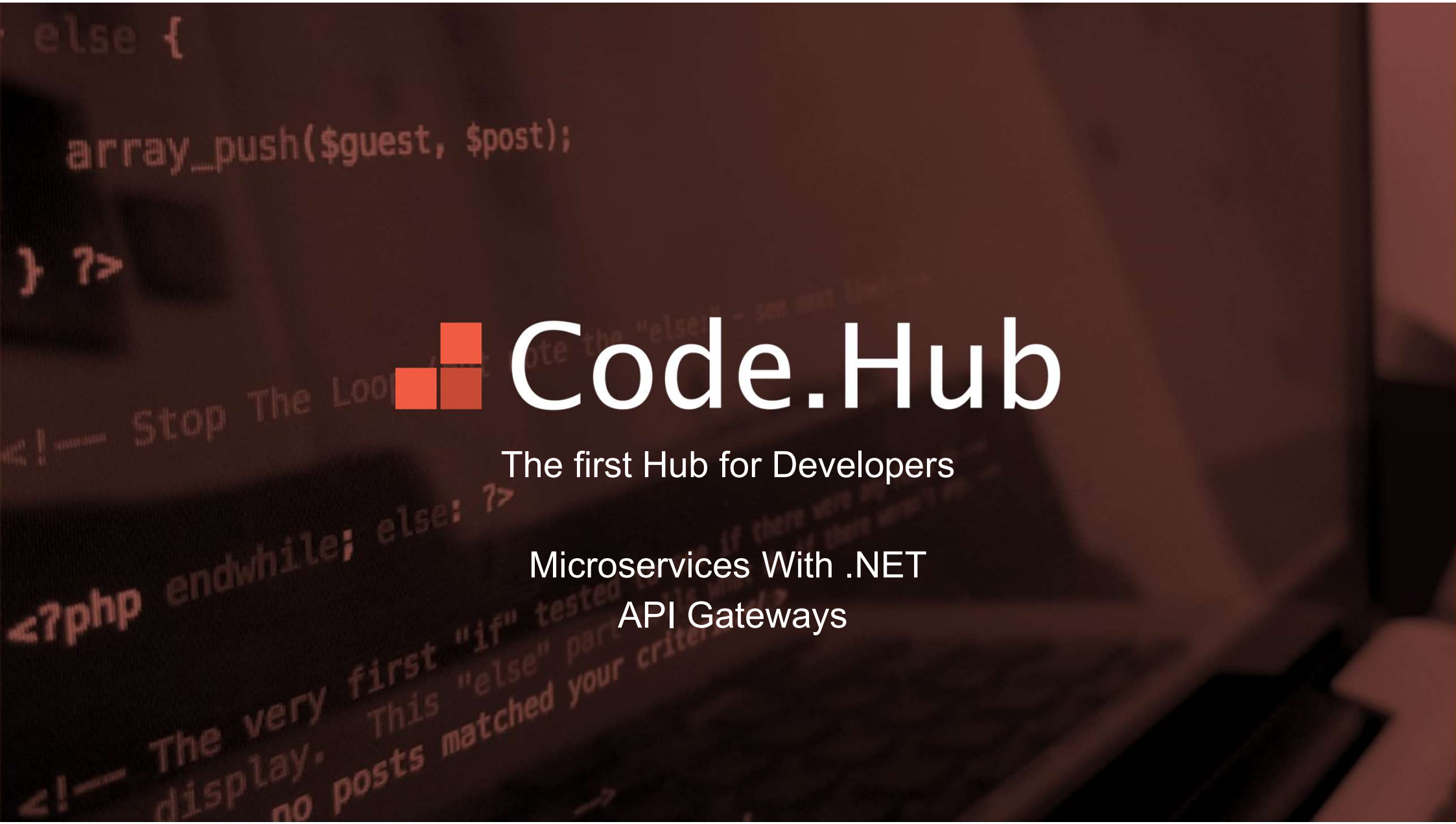
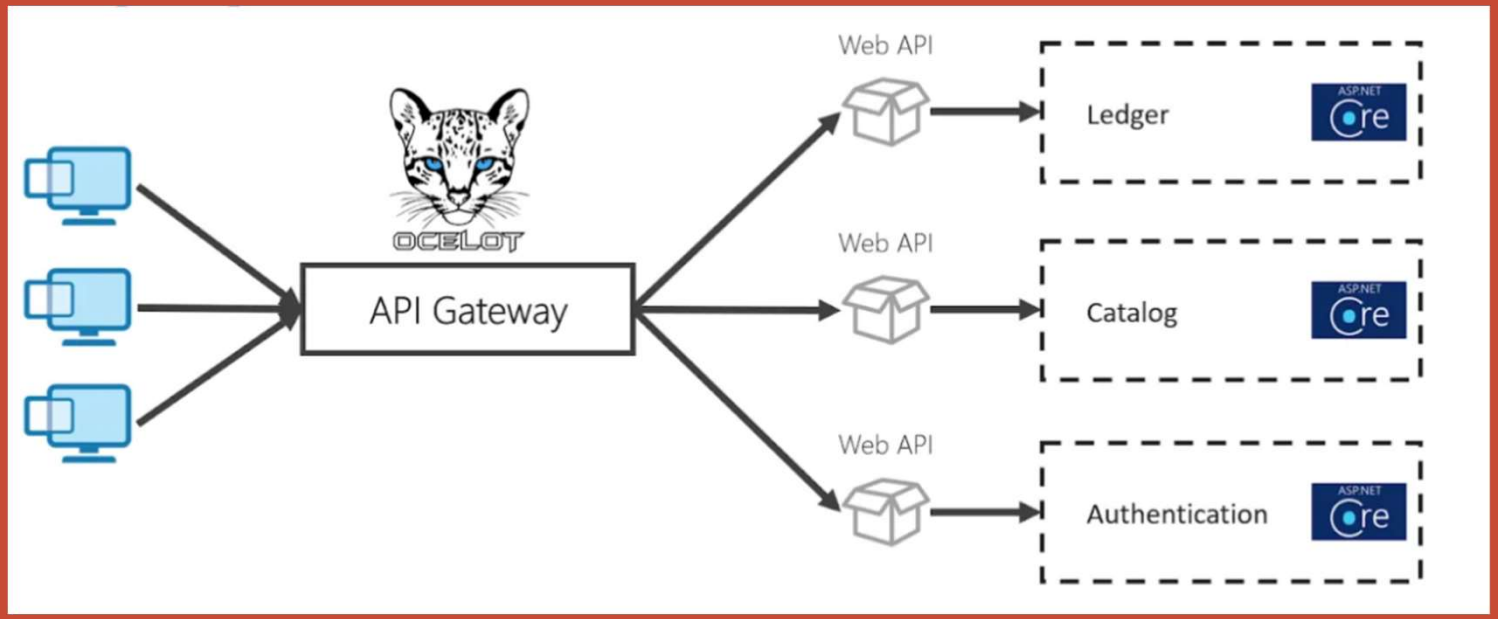

Code.Hub
The first Hub for Developers
Microservices With .NET
API Gateways

1. ASP.NET Core Web API revisited
2. Ocelot
3. API Gateway Patterns
4. Load Balancing
5. Request/Response Transformation
6. API Gateway Security

Agenda

Code.Hub



Api Basics

Code.Hub

Minimal API in ASP.NET Core

```
app.MapGet("/books", async (ApplicationDbContext dbContext) =>
{
    var books = await dbContext.Books.ToListAsync();
    return Results.Ok(books);
});

app.MapGet("/books/{id}", async (int id, ApplicationDbContext dbContext) =>
{
    var book = await dbContext.Books
            .Include(_ => _.Author)
            .FirstOrDefaultAsync(_ => _.Id == id);
    if (book is null)
        return Results.NotFound();
    return Results.Ok(book);
});
```

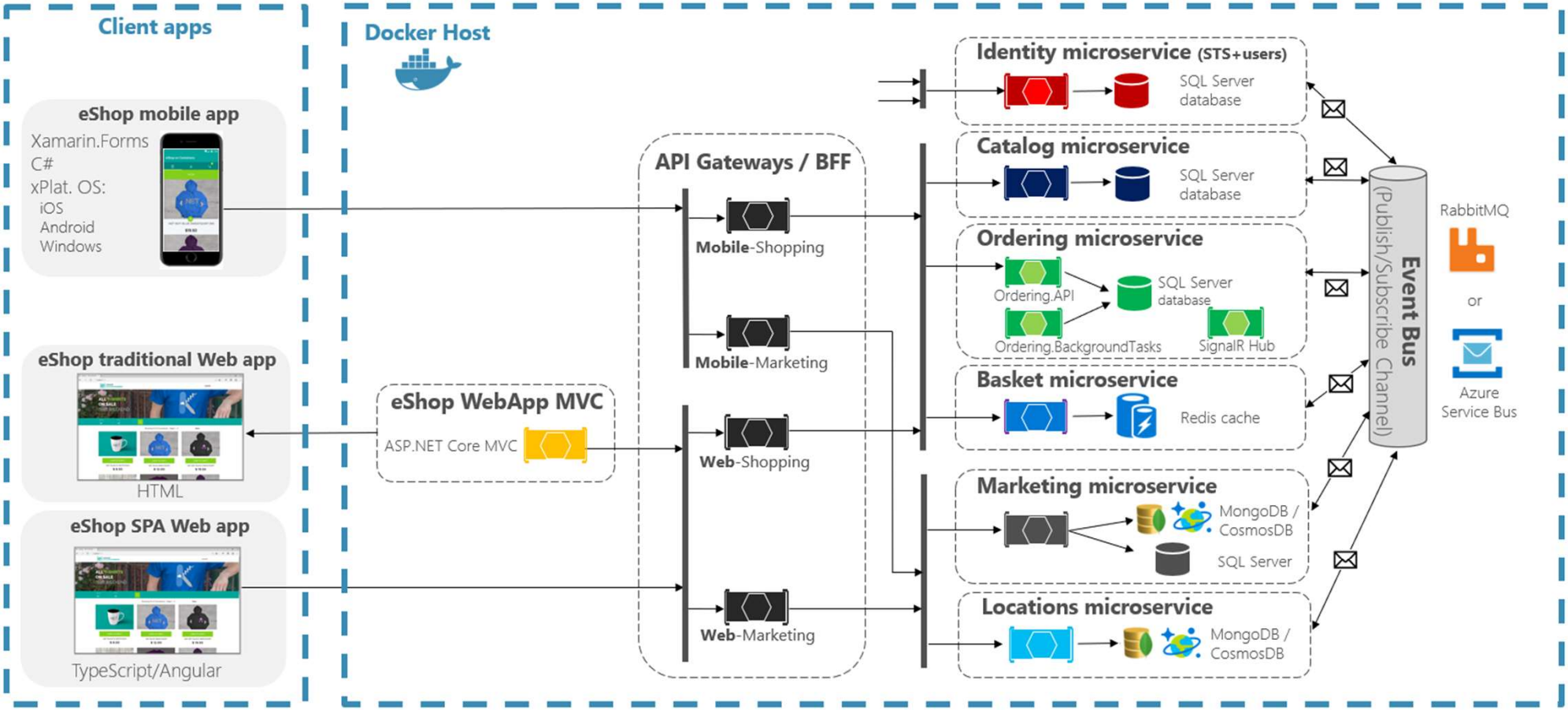
Code.Hub

Key Functions and Features

- Routing and Aggregation
- Protocol Translation
- Authentication and Authorization
- Traffic Management and Load Balancing
- Caching and Performance Optimization
- Monitoring and Analytics

eShopOnContainers reference application

(Development environment architecture)

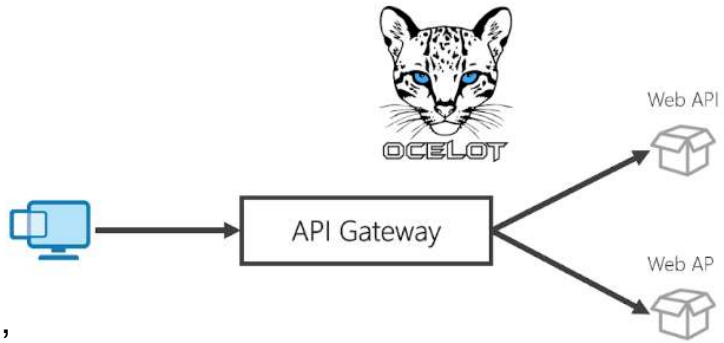


Code.Hub

Ocelot

ReRoute configuration file

```
{   "Routes": [
    {       "DownstreamPathTemplate": "/api/product",
      "DownstreamScheme": "https",
      "DownstreamHostAndPorts": [
        {     "Host": "localhost",        "Port": 44337      }
],
      "UpstreamPathTemplate": "/gateway/product",
      "UpstreamHttpMethod": [ "POST", "PUT", "GET"
]
    },
    {       "DownstreamPathTemplate":
"/api/customer",
      "DownstreamScheme": "https",
      "DownstreamHostAndPorts": [
        {     "Host": "localhost",    "Port": 44338      }
],
      "UpstreamPathTemplate": "/gateway/customer",
      "UpstreamHttpMethod": [ "POST", "PUT", "GET"
]
    } ] }
```



- DownstreamPathTemplate
- DownstreamScheme
- DownstreamHostAndPorts
- UpstreamPathTemplate
- UpstreamHttpMethod

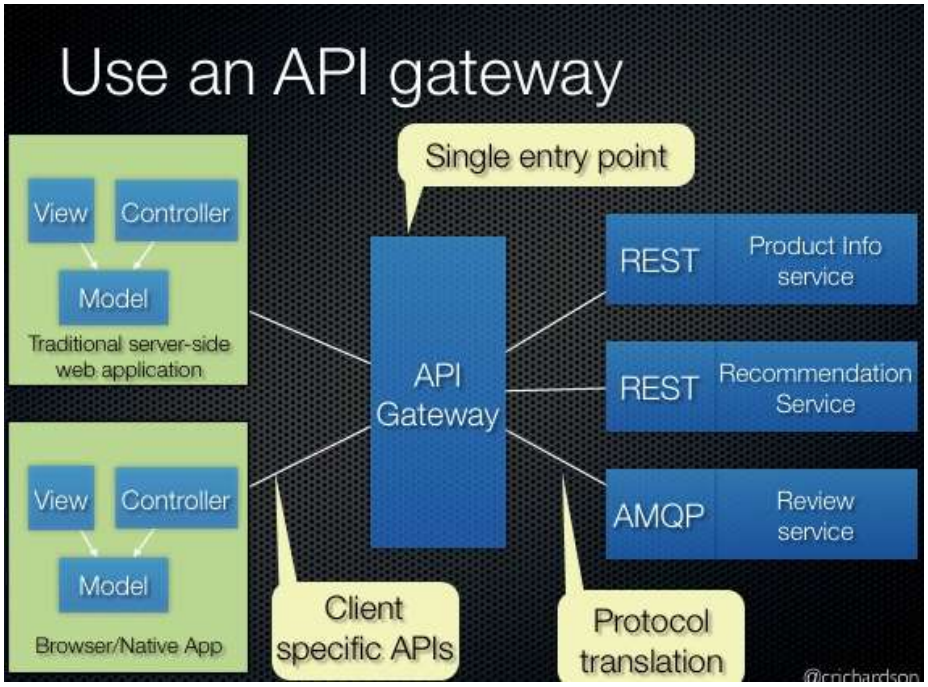
Code.Hub

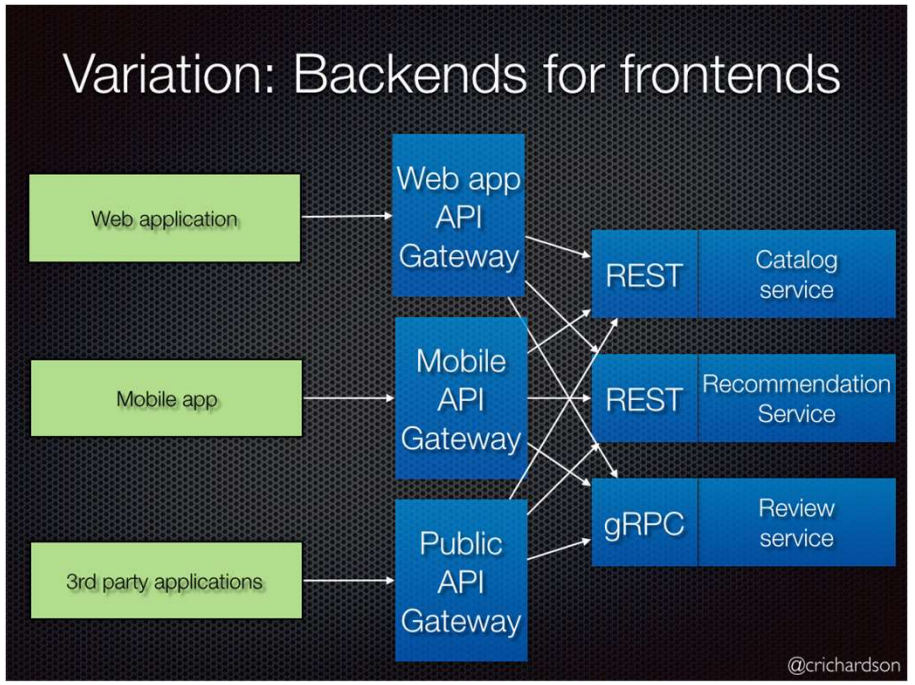
1. API Gateway Patterns
2. Load Balancing
3. Request/Response Transformation

API development

Code.Hub

API Gateway Patterns





Code.Hub

Load balancing

Types supported for Load Balancing by Ocelot

- LeastConnection - tracks which services are dealing with requests and sends new requests to service with least existing requests. The algorithm state is not distributed across a cluster of Ocelot's.
- RoundRobin - loops through available services and sends requests. The algorythm state is not distributed across a cluster of Ocelot's.
- NoLoadBalancer - takes the first available service from config or service discovery.
- CookieStickySessions - uses a cookie to stick all requests to a specific server.

custom load balancing is also supported

Request/Response Transformation

Using configuration
add a header to your upstream request

add a header to your upstream request

Using middleware

Code.Hub

Thank you!

Code.Hub