

Introduction to Microservices Architecture

Somnuk Keretho, PhD
Department of Computer Engineering
Kasetsart University
sk@ku.ac.th

Agenda

1

Why Microservices?

2

What Is Microservice Architecture?

3

Features of Microservice Architecture

4

Advantages of Microservice
Architecture

5

Companies Using Microservices



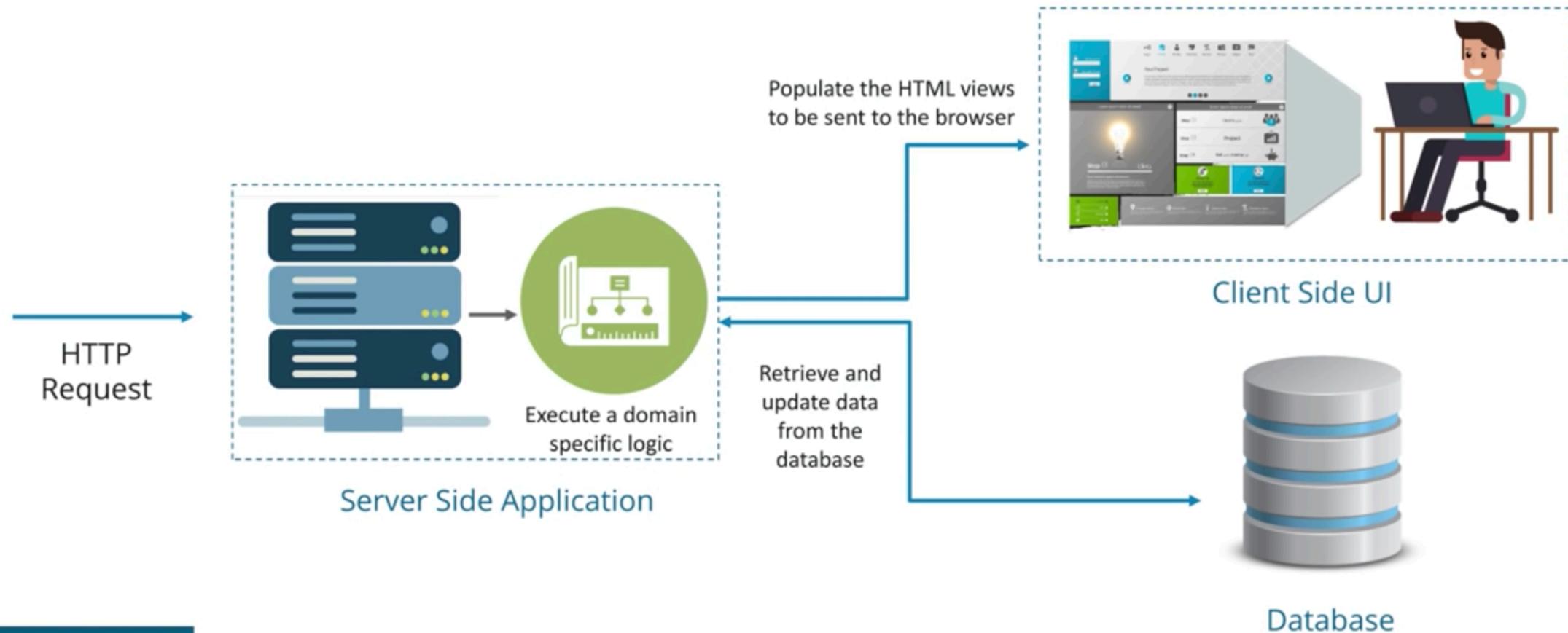
Microservice Architecture

Ref: <https://www.edureka.co/microservices-architecture-training>

Why Microservices?

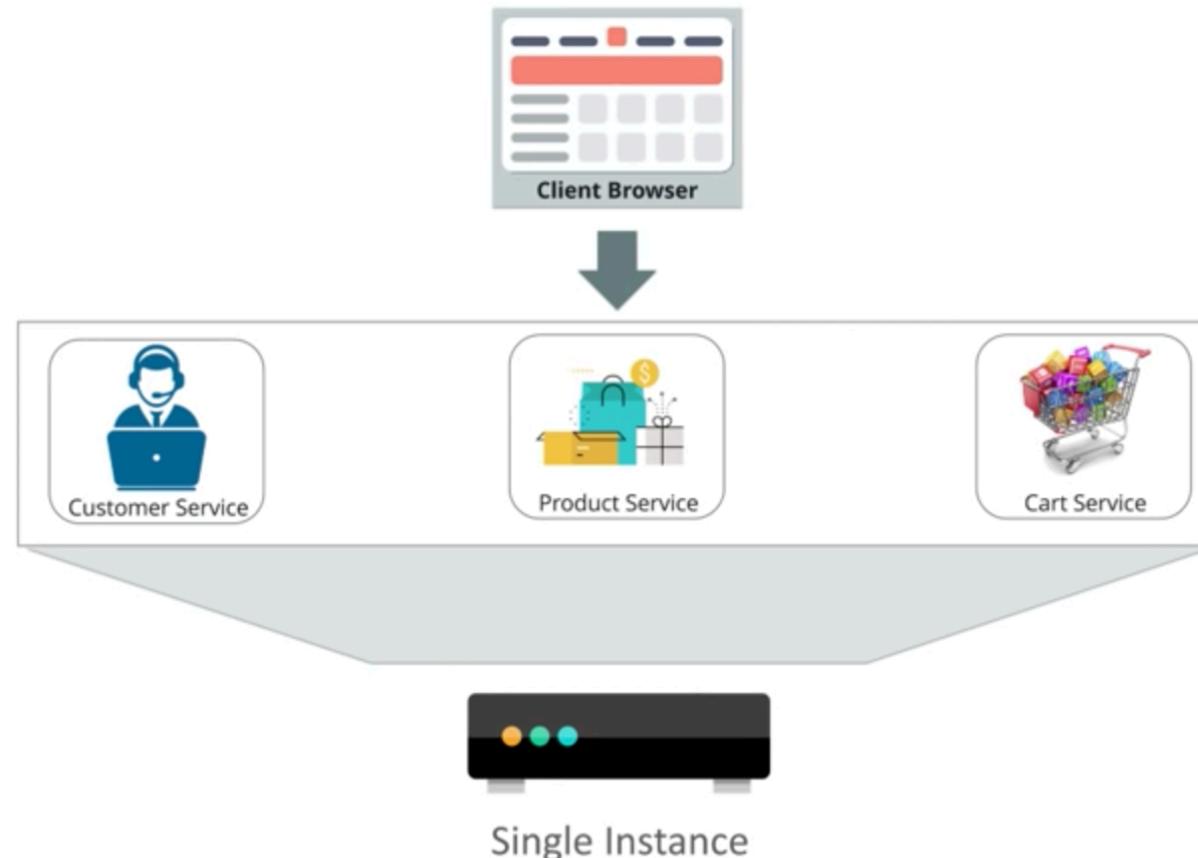
Before Microservices – Monolithic Architecture

Monolithic Architecture is like a big container wherein all the software components of an application are assembled together and tightly packaged



Monolithic Architecture - Example

Let's take a classic use case of an E-Commerce Application



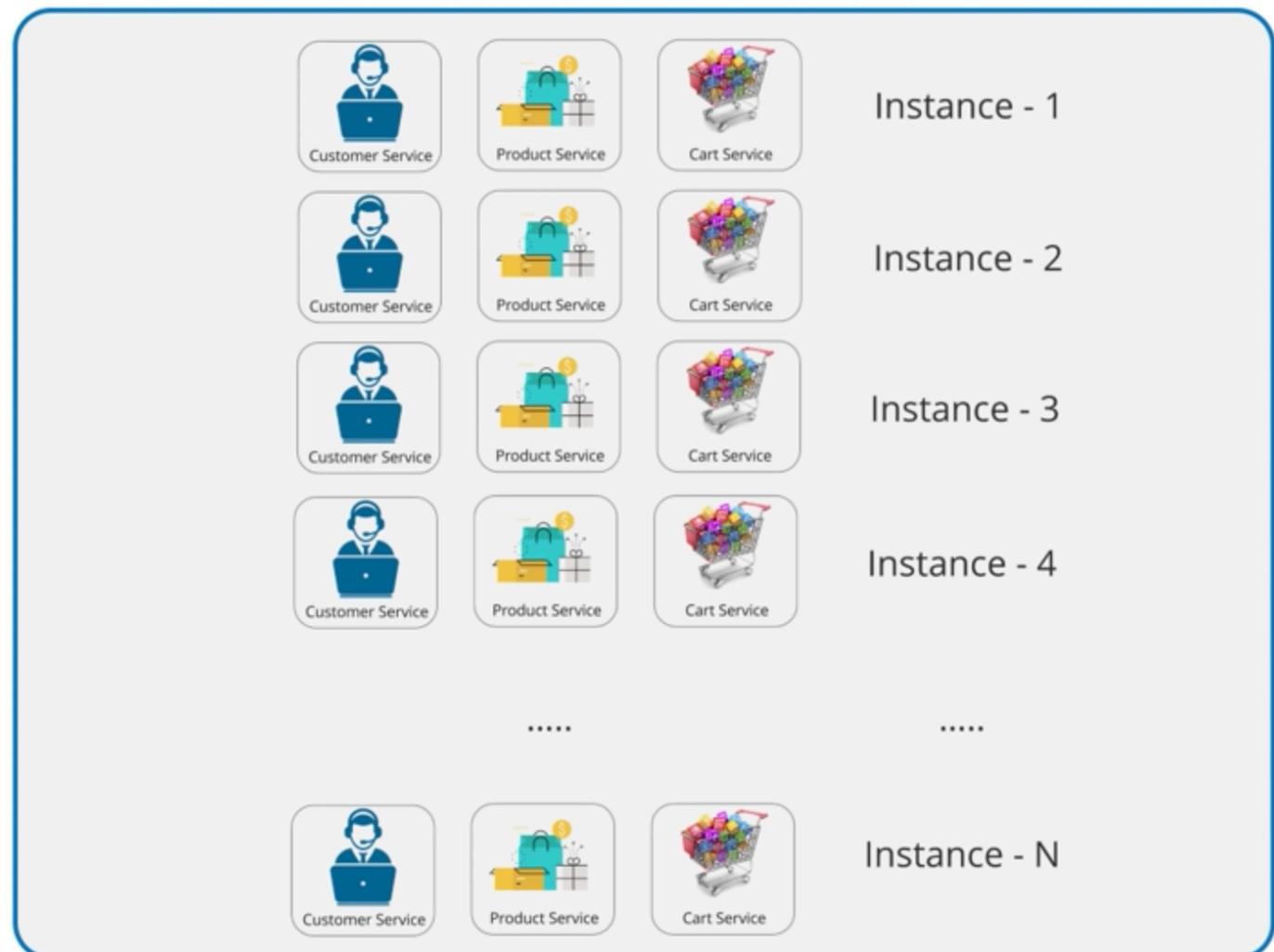
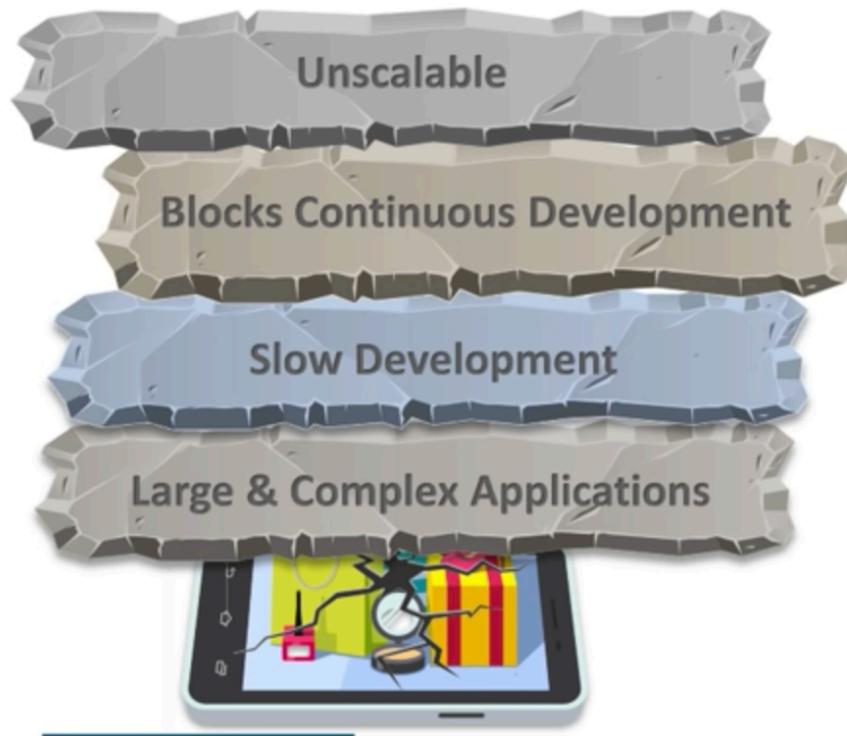
Monolithic Architecture - Challenges



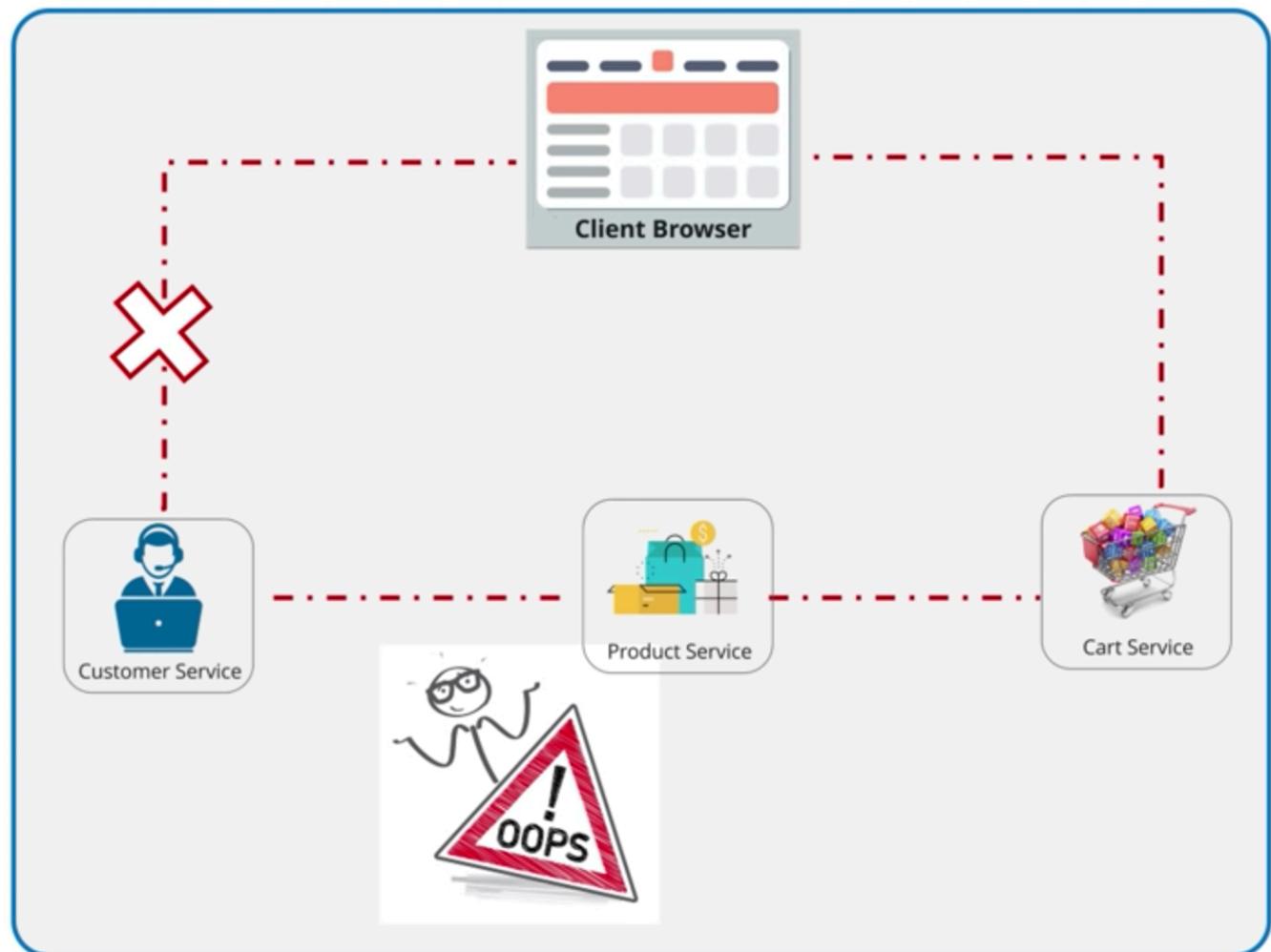
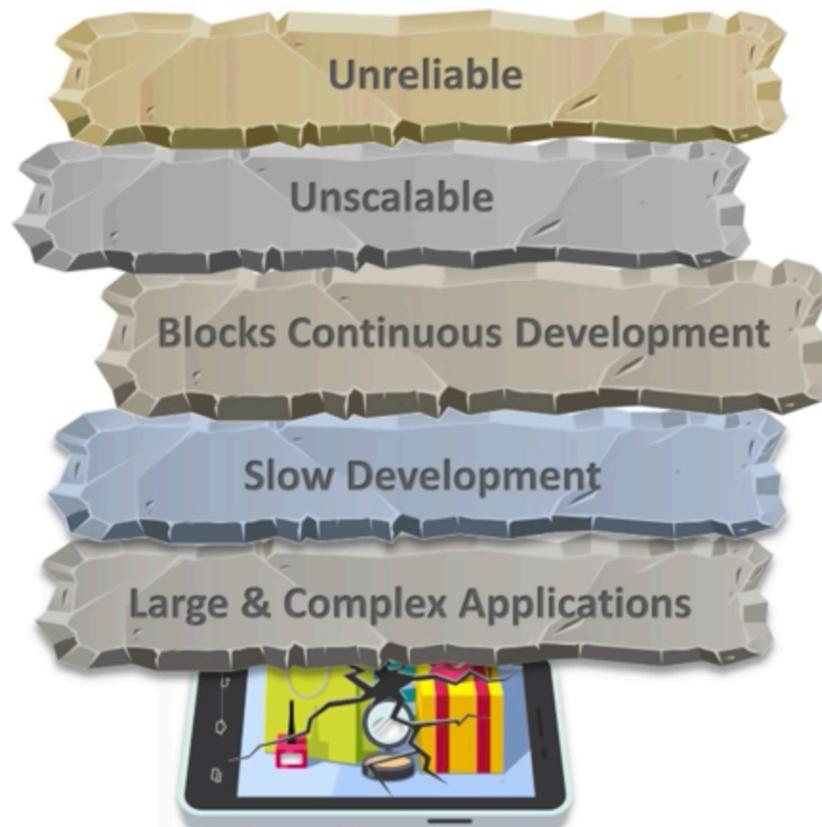
Monolithic Architecture - Challenges



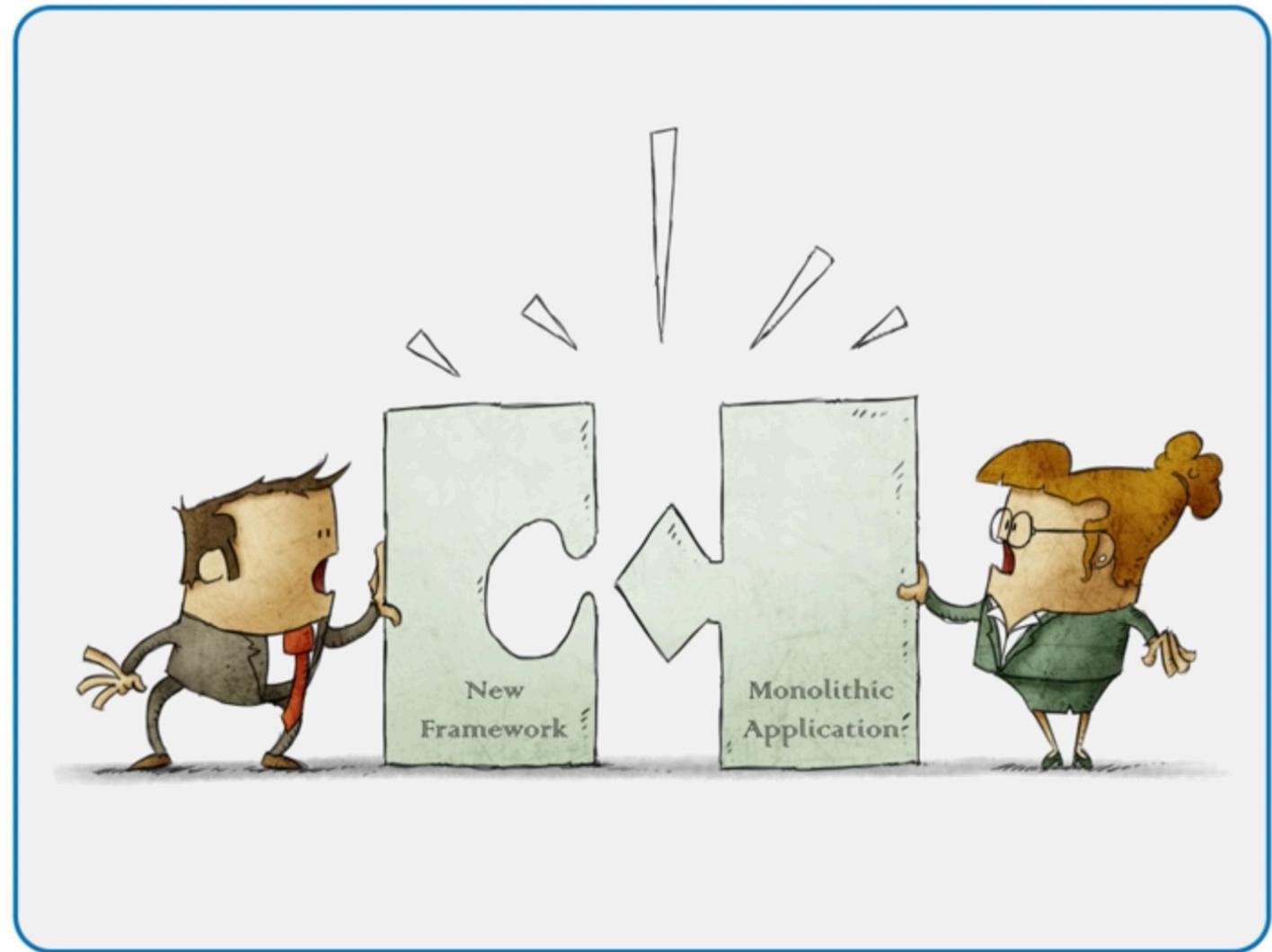
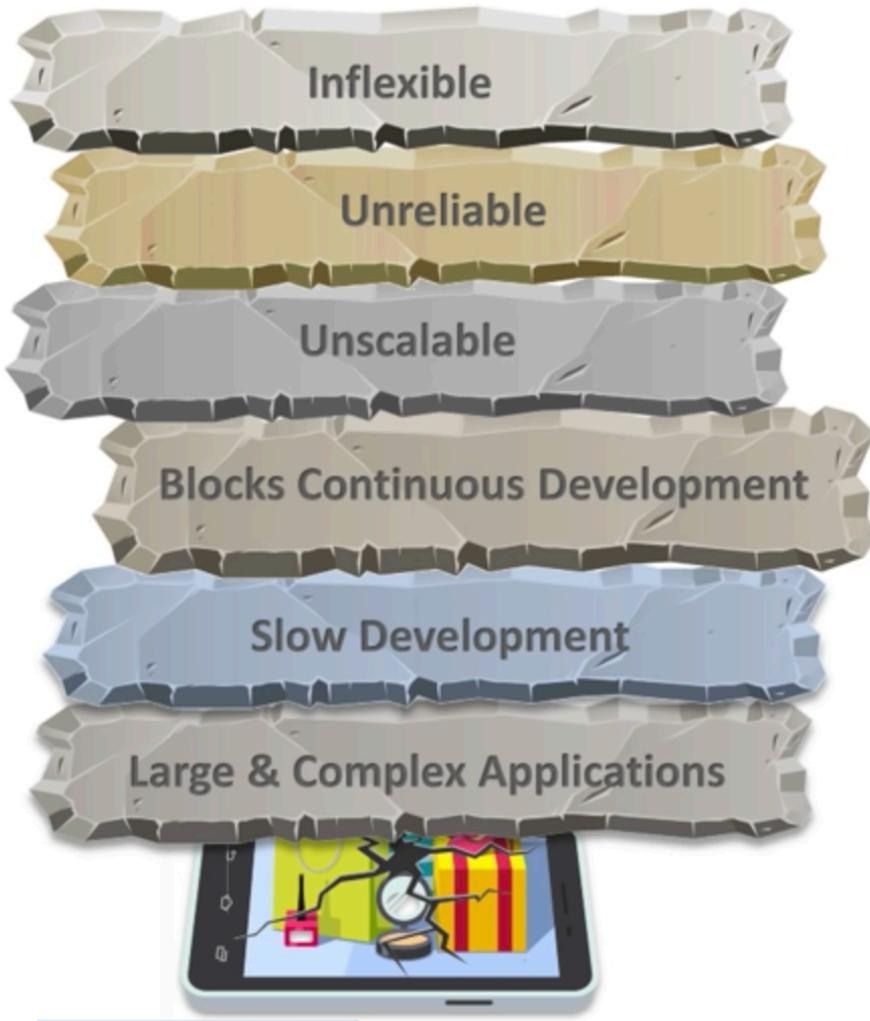
Monolithic Architecture - Challenges



Monolithic Architecture - Challenges



Monolithic Architecture - Challenges



What Is Microservice Architecture?

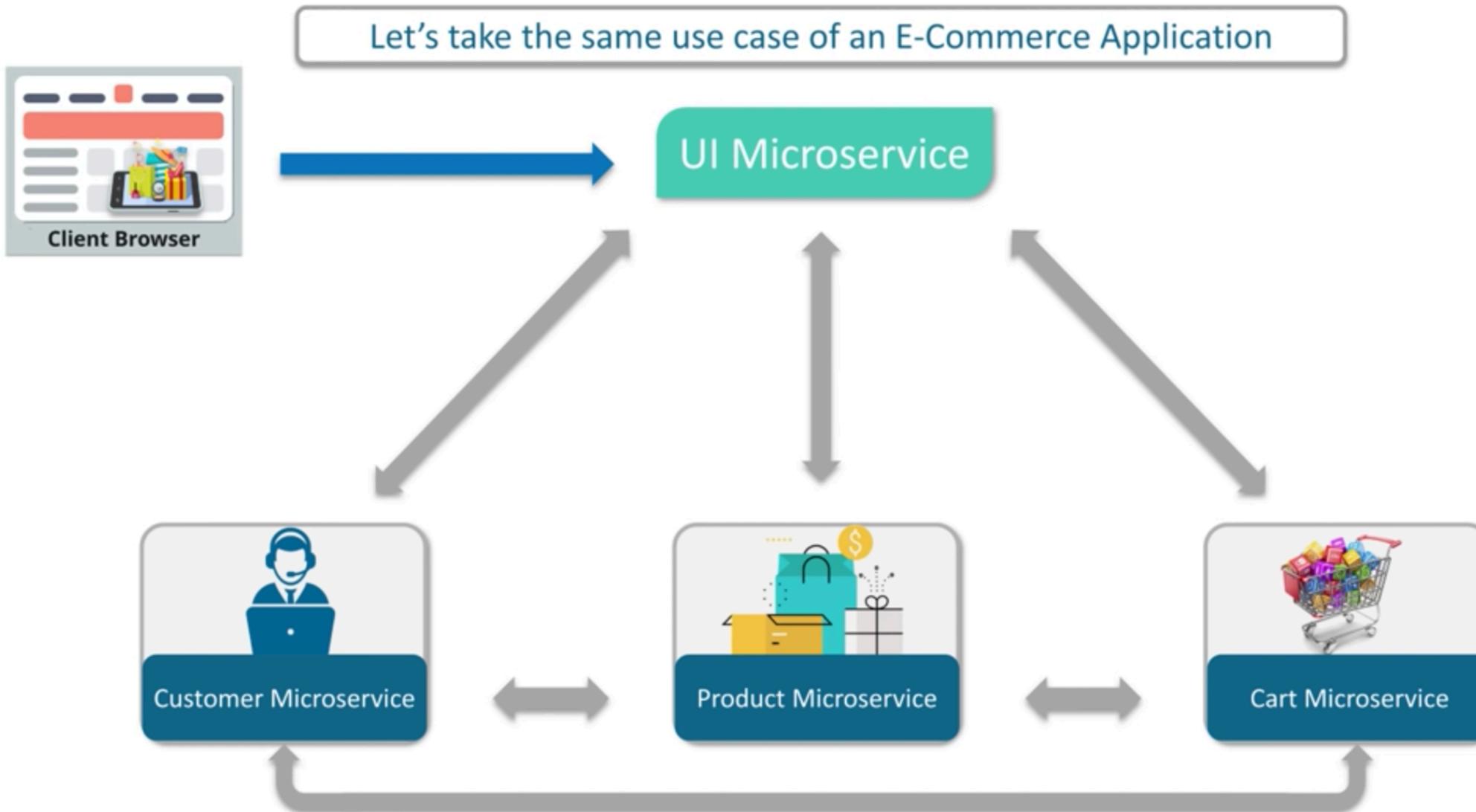
What Is Microservice Architecture?

Microservices, aka *Microservice Architecture*, is an **architectural style** that structures an application as a **collection of small autonomous services**, modelled around a **Business Domain**



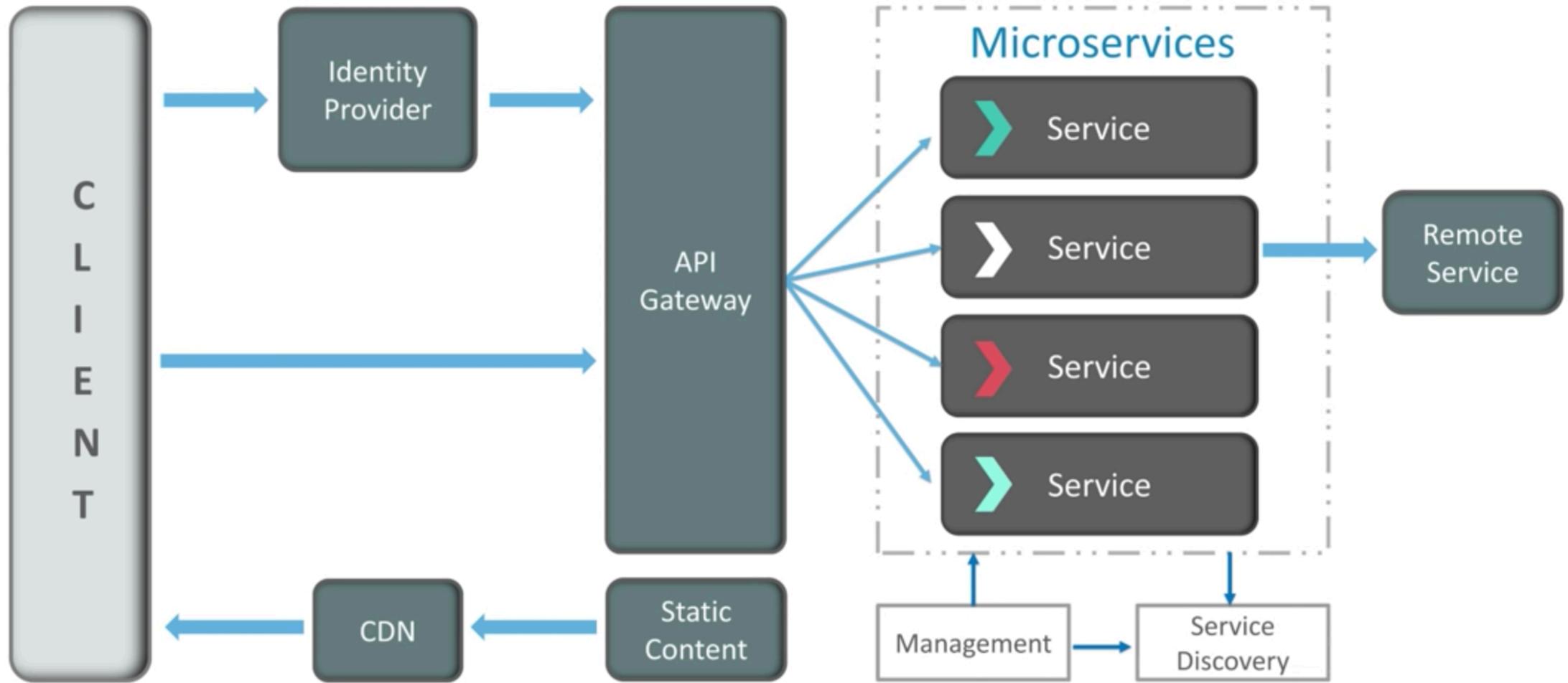
In microservices architecture, each service is **self-contained** and implements **a single business capability**.

Microservice Architecture - Example



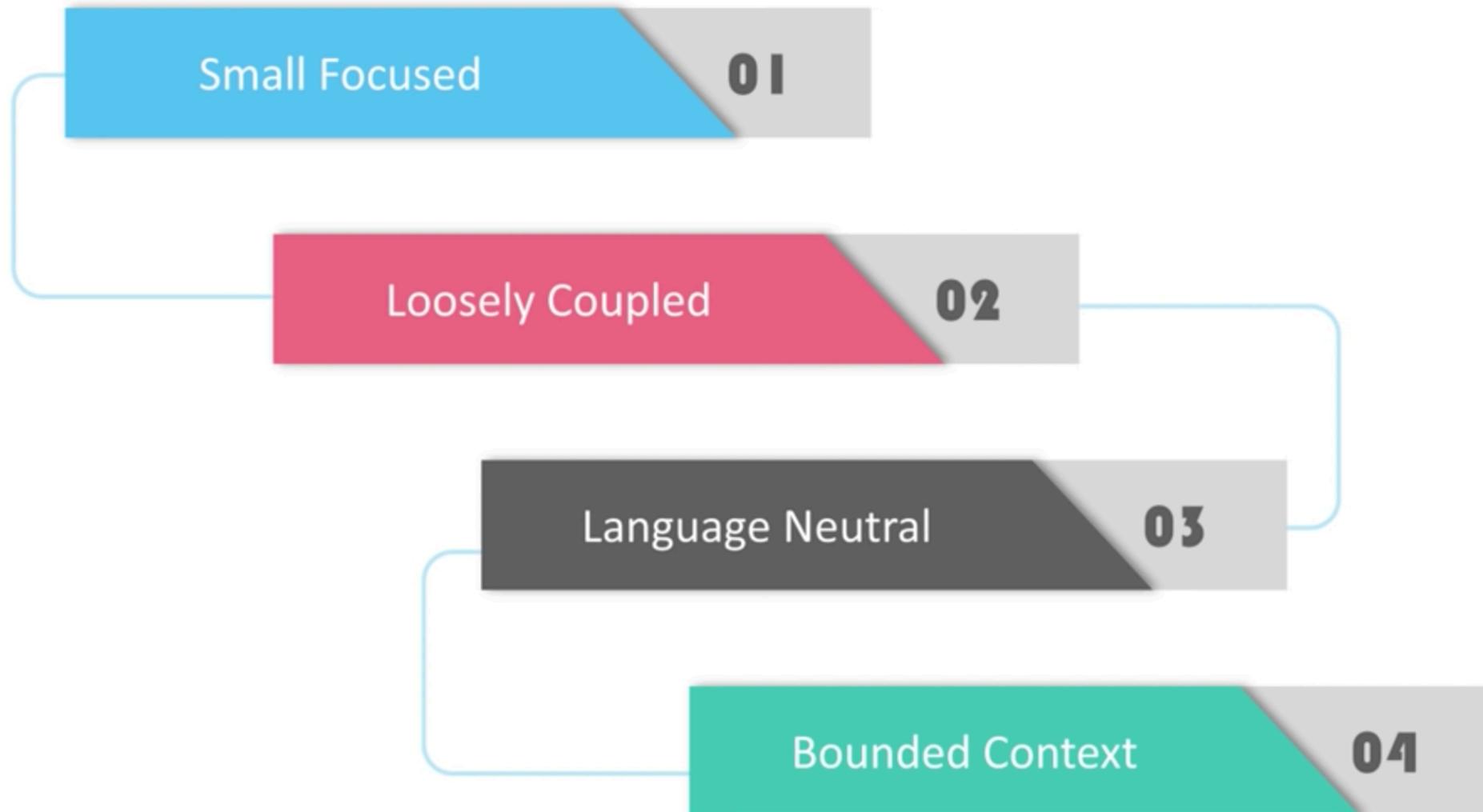
Microservice Architecture

Microservice Architecture



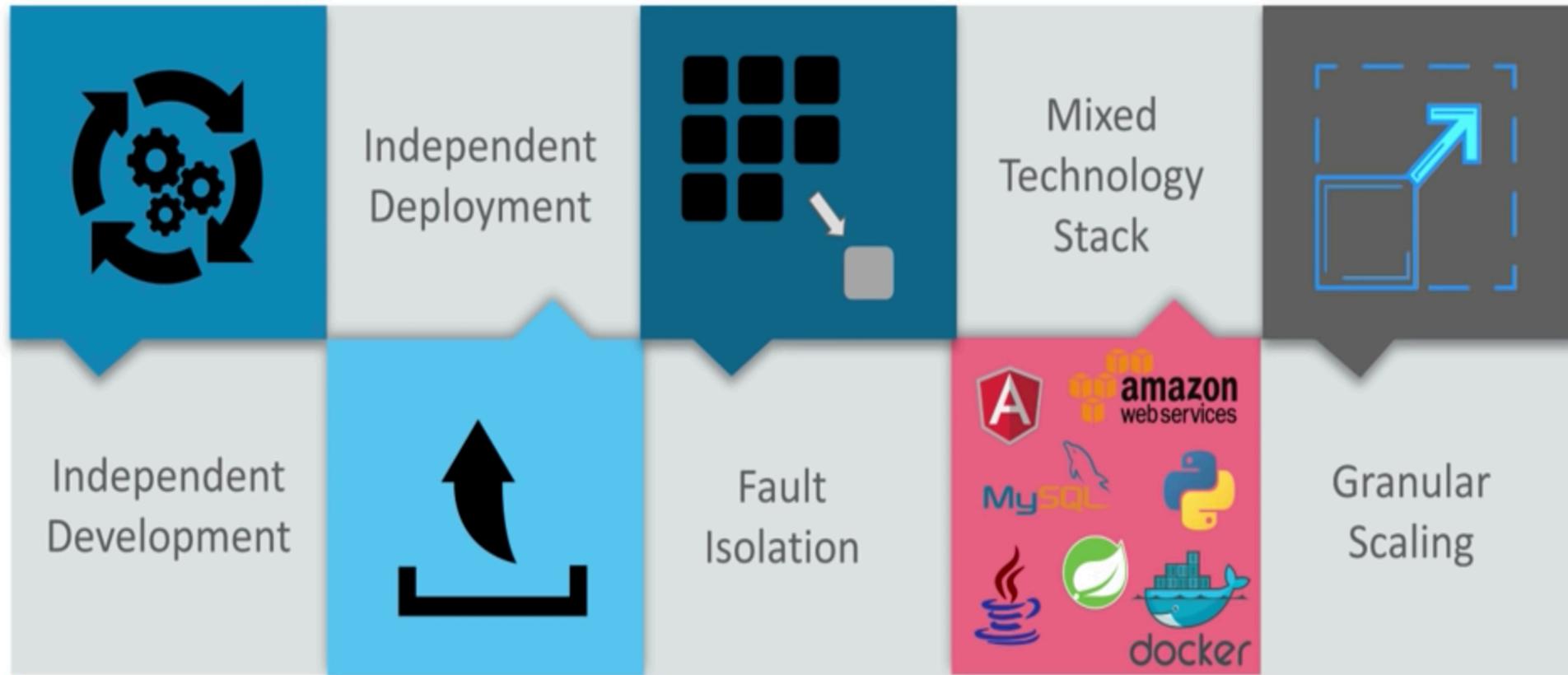
Features Of Microservice Architecture

Features Of Microservice Architecture



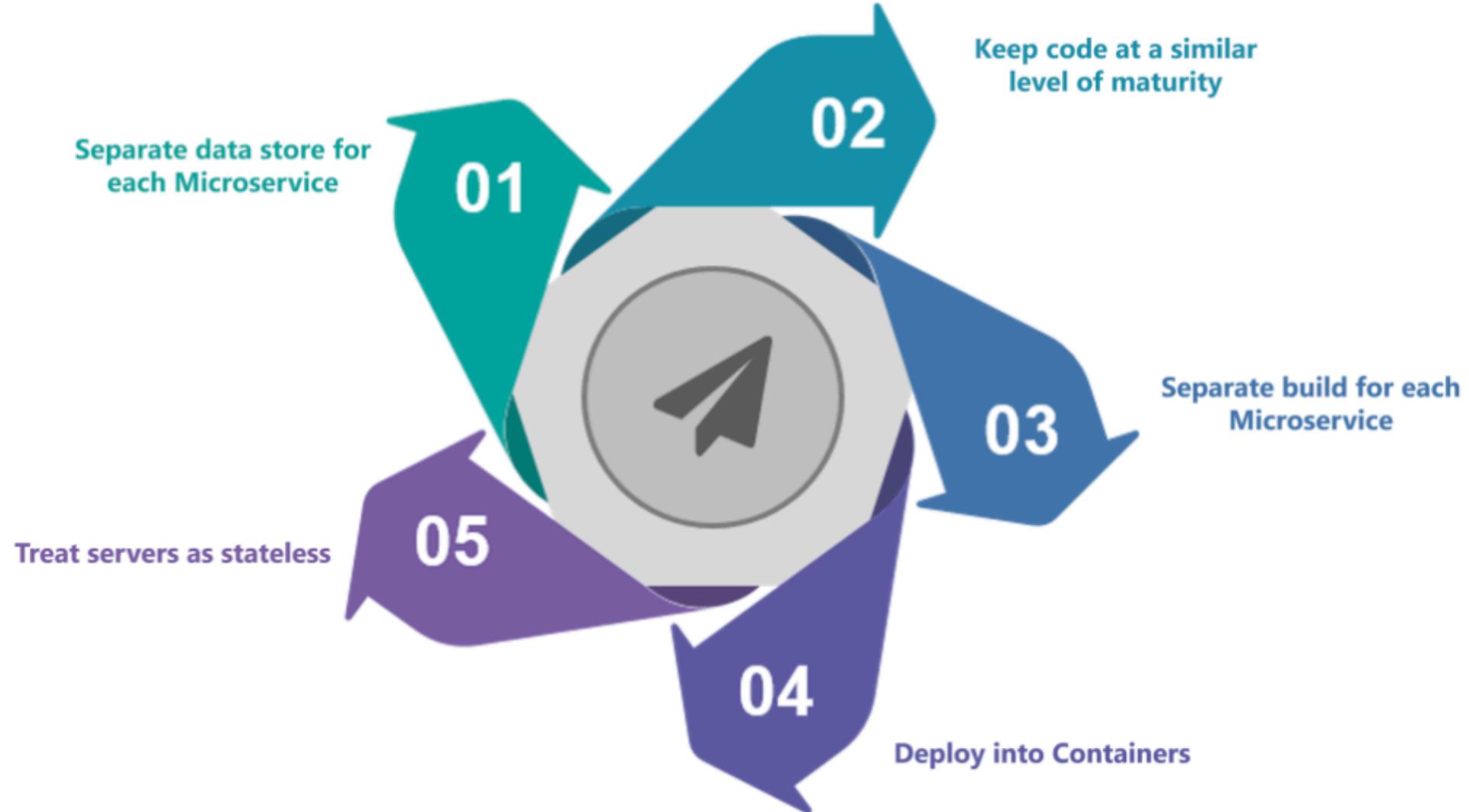
Advantages Of Microservice Architecture

Advantages Of Microservice Architecture



Best Practices to Design Microservices

Best Practices to Design Microservices

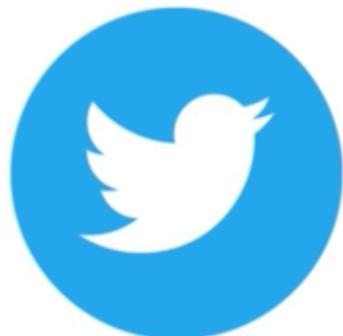


Companies Using Microservices

Companies Using Microservices



NETFLIX



UBER



ebay

GILT

the guardian



NORDSTROM

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays three Spring Boot projects:
 - Diagnosis_Microservice_Edureka [boot]**: Contains Spring Elements, src/main/java, src/main/resources, src/test/java, JRE System Library [JavaSE-1.8], Maven Dependencies, src, target, mvnw, mvnw.cmd, pom.xml.
 - Doctor_Microservice_Edureka [boot]**: Contains Spring Elements, src/main/java, src/main/resources, src/test/java, JRE System Library [JavaSE-1.8], Maven Dependencies, src, target, mvnw, mvnw.cmd, pom.xml.
 - Patient_Microservice_Edureka [boot]**: Contains Spring Elements, src/main/java (with com.edureka.demo and com.edureka.demo.rest packages), Application.java, ApplicationConfiguration.java, Patient.java, PatientRest.java.
- Editor Area:** The PatientRest.java file is the active editor, showing Java code for a REST API. The code includes methods for getting lists of patients and individual patient details by ID.
- Console:** Shows the message "No consoles to display at this time."

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Displays two projects:
 - Doctor_Microservice_Edureka [boot]:** Contains Spring Elements, src/main/java, src/main/resources, src/test/java, JRE System Library [JavaSE-1.8], Maven Dependencies, src, target, mvnw, mvnw.cmd, and pom.xml.
 - Patient_Microservice_Edureka [boot]:** Contains Spring Elements, src/main/java (with subfolders com.edureka.demo and com.edureka.demo.rest), src/main/resources, src/test/java, JRE System Library [JavaSE-1.8], Maven Dependencies, src, target, mvnw, mvnw.cmd, and pom.xml.
- Code Editor View:** Shows the PatientRest.java file with the following code:

```
52     clients.add(patient2);
53     clients.add(patient3);
54     clients.add(patient4);
55     clients.add(patient5);
56     clients.add(patient6);
57
58 }
59
60 @GET
61 @Produces(MediaType.APPLICATION_JSON)
62 public List<Patient> getClientes() {
63     return clients;
64 }
65
66 @GET
67 @Path("patient")
68 @Produces(MediaType.APPLICATION_JSON)
69 public Patient getCliente(@QueryParam("id") long id) {
70
71     Patient cli = null;
72
73     for (Patient c : clients) {
74
75         if (c.getId() == id)
76             cli = c;
77
78     }
79
80     return cli;
81 }
82
83 }
```
- Console View:** Displays the message "No consoles to display at this time."

```
Application.java ApplicationConfiguration.java Patient.java PatientRest.java
```

```
52     clients.add(patient2);
53     clients.add(patient3);
54     clients.add(patient4);
55     clients.add(patient5);
56     clients.add(patient6);

57 }

58 @GET
59 @Produces(MediaType.APPLICATION_JSON)
60 public List<Patient> getClients() {
61     return clients;
62 }

63 @GET
64 @Path("patient")
65 @Produces(MediaType.APPLICATION_JSON)
66 public Patient getCliente(@QueryParam("id") long id) {

67     Patient cli = null;

68     for (Patient c : clients) {
69
70         if (c.getId() == id)
71             cli = c;
72
73     }
74
75     return cli;
76 }
77
78 }
```

or Java similarly
for diagnosis as well.

we have defined all
the variables like, you know,

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Shows the project structure for "Patient_Microservice_Edureka". It includes the main source folder "src/main/java" which contains packages "com.edureka.demo" and "com.edureka.demo.rest". Inside "com.edureka.demo", there are files "Application.java" and "ApplicationConfiguration.java". Inside "com.edureka.demo.rest", there are files "Patient.java" and "PatientRest.java". Other visible files include "Application.java", "ApplicationConfiguration.java", "Patient.java", and "PatientRest.java" in the root "src/main/java" folder, along with "mvnw", "mvnw.cmd", and "pom.xml".
- Code Editor View:** The "PatientRest.java" file is open in the editor. The code implements a REST API for managing patients. It defines a list of patients and handles GET requests to retrieve all patients or a specific patient by ID.

```
46     Patient patient6 = new Patient();
47     patient6.setId(006);
48     patient6.setName("Monica");
49     patient6.setEmail("patient6@edureka.co");
50
51     clients.add(patient1);
52     clients.add(patient2);
53     clients.add(patient3);
54     clients.add(patient4);
55     clients.add(patient5);
56     clients.add(patient6);
57
58 }
59
60 @GET
61 @Produces(MediaType.APPLICATION_JSON)
62 public List<Patient> getClientes() {
63     return clients;
64 }
65
66 @GET
67 @Path("patient")
68 @Produces(MediaType.APPLICATION_JSON)
69 public Patient getCliente(@QueryParam("id") long id) {
70
71     Patient cli = null;
72
73     for (Patient c : clients) {
74
75         if (c.getId() == id)
76             cli = c;
77     }
78 }
```

- Console View:** The console shows the message "No consoles to display at this time."

I won't explain the code here.

72:1

eclipse-workspace - Patient_Microservice_Edureka/src/main/java/com/edureka/demo/rest/PatientRest.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer Application.java ApplicationConfiguration.java Patient.java PatientRest.java

```
46     Patient patient6 = new Patient();
47     patient6.setId(006);
48     patient6.setName("Monica");
49     patient6.setEmail("patient6@edureka.co");
50
51     clients.add(patient1);
52     clients.add(patient2);
53     clients.add(patient3);
54     clients.add(patient4);
55     clients.add(patient5);
56     clients.add(patient6);
57
58 }
59
60 @GET
61 @Produces(MediaType.APPLICATION_JSON)
62 public List<Patient> getClientes() {
63     return clients;
64 }
65
66 @GET
67 @Path("patient")
68 @Produces(MediaType.APPLICATION_JSON)
69 public Patient getCliente(@QueryParam("id") long id) {
70
71     Patient cli = null;
72
73     for (Patient c : clients) {
74
75         if (c.getId() == id)
76             cli = c;
77     }
78 }
```

Console No consoles to display at this time.

Alt+Enter to properties click

Project Explorer X

- New
- Go Into
- Show In Alt+Shift+W >
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path >
- Refactor Alt+Shift+T >
- Import...
- Export...
- Refresh F5
- Close Project
- Close Unrelated Projects
- Show in Remote Systems view
- Validate
- Coverage As
- Run As
- Debug As
- Profile As
- Restore from Local History...
- Maven
- Team
- Compare With
- Configure
- Source
- Spring Tools

Quick Access

java ApplicationConfiguration.java Patient.java PatientRest.java

```
Patient patient6 = new Patient();
patient6.setId(006);
patient6.setName("Monica");
patient6.setEmail("patient6@edureka.co");

clients.add(patient1);
clients.add(patient2);
clients.add(patient3);
clients.add(patient4);
clients.add(patient5);
clients.add(patient6);

SET
@Produces(MediaType.APPLICATION_JSON)
public List<Patient> getClientes() {
    return clients;
}

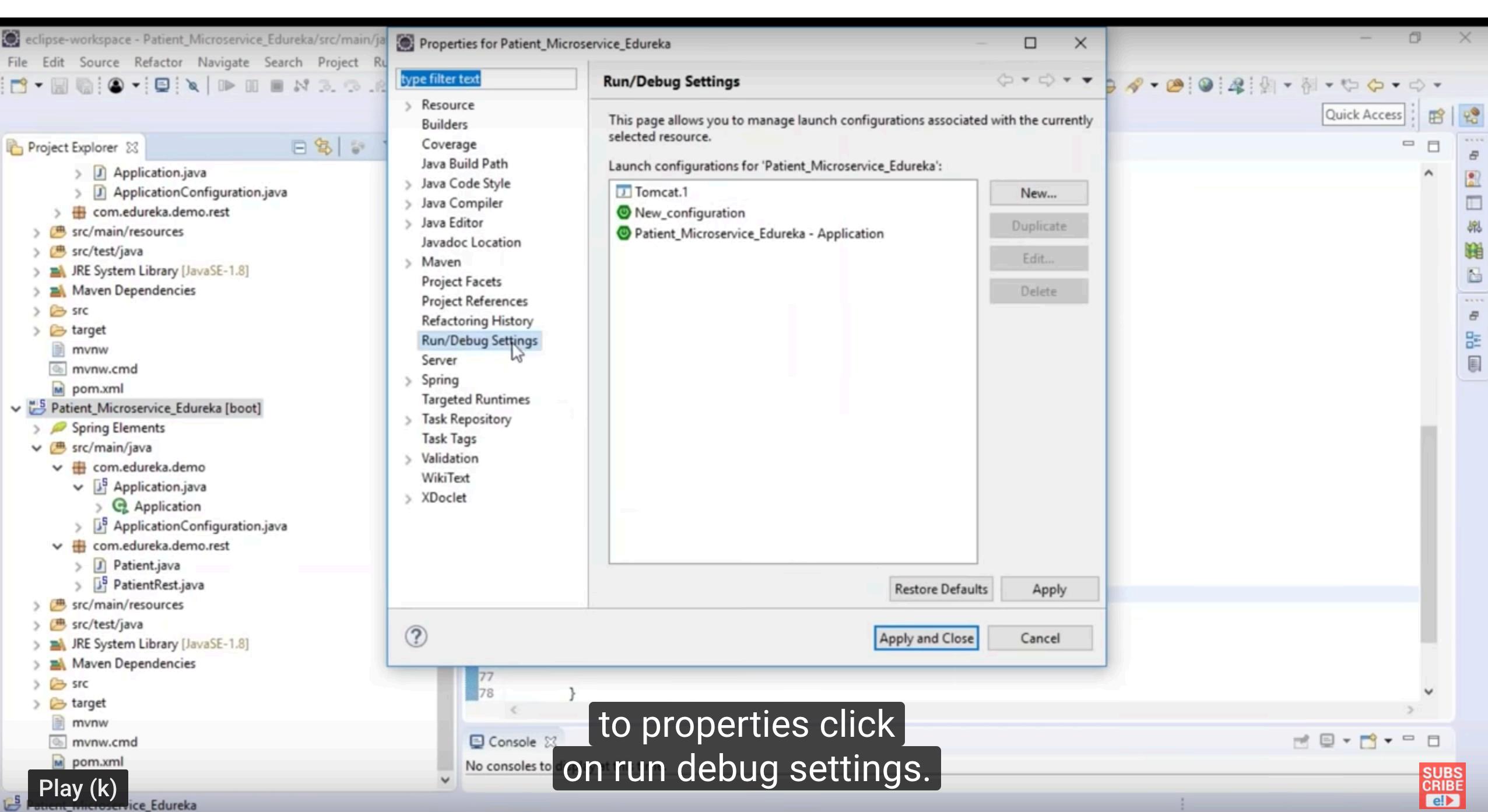
SET
@Path("patient")
@Produces(MediaType.APPLICATION_JSON)
public Patient getCliente(@QueryParam("id") long id) {
    Patient cli = null;

    for (Patient c : clients) {
        if (c.getId() == id)
            cli = c;
    }
}
```

So all I have to do
is right click go

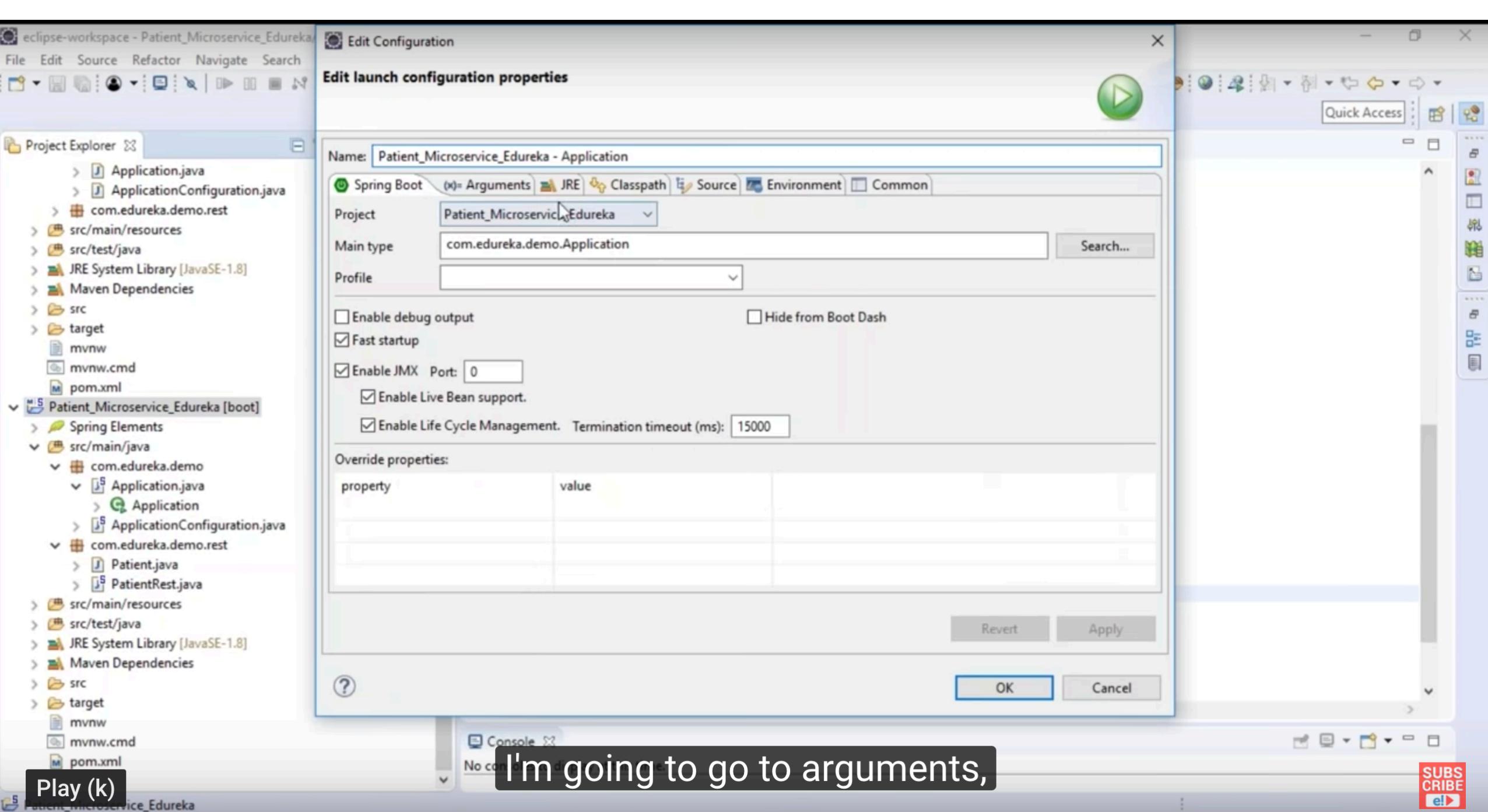
Properties Alt+Enter

SUBSCRIBE



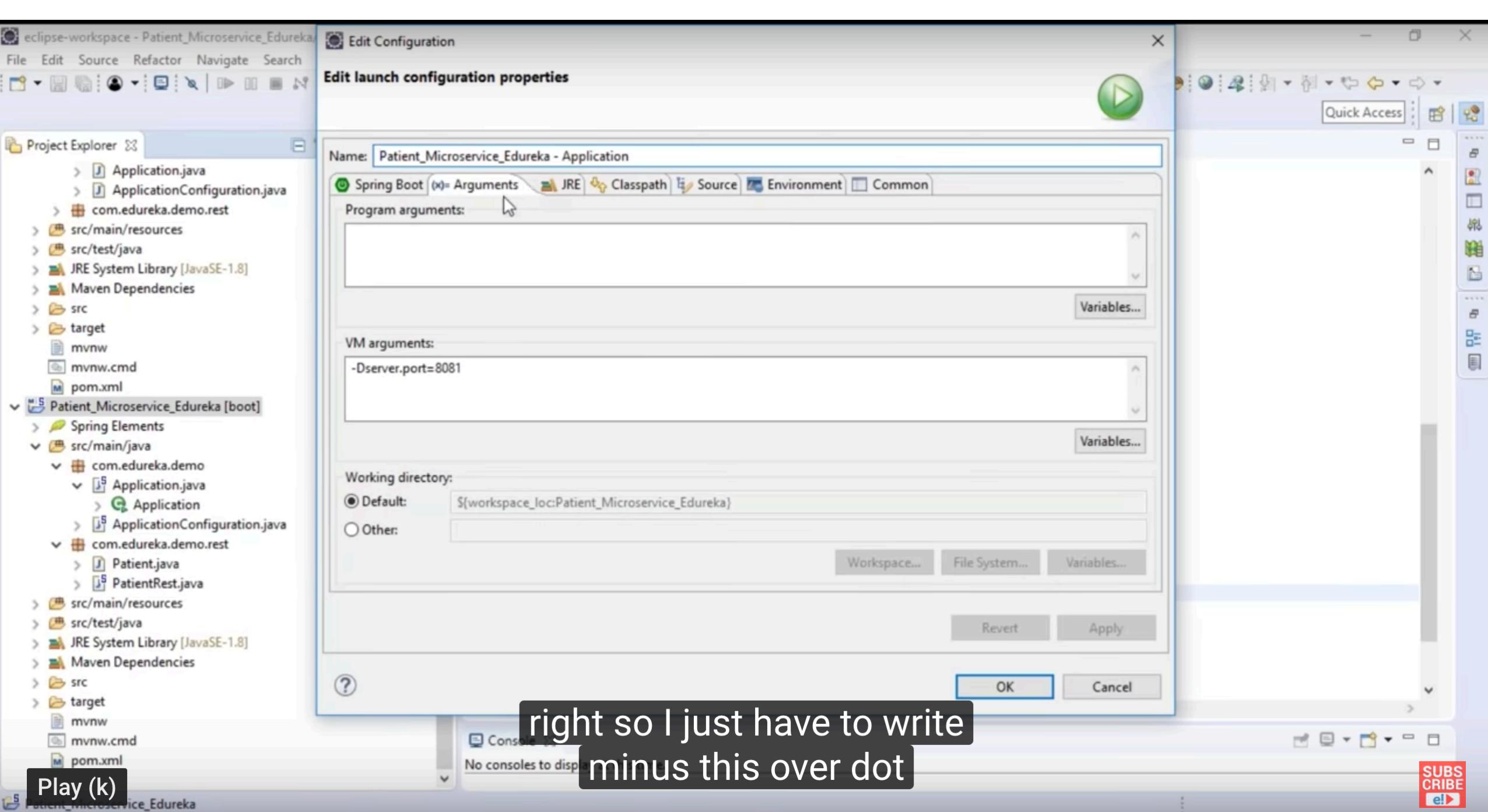
Play (k)

SUBSCRIBE
eD



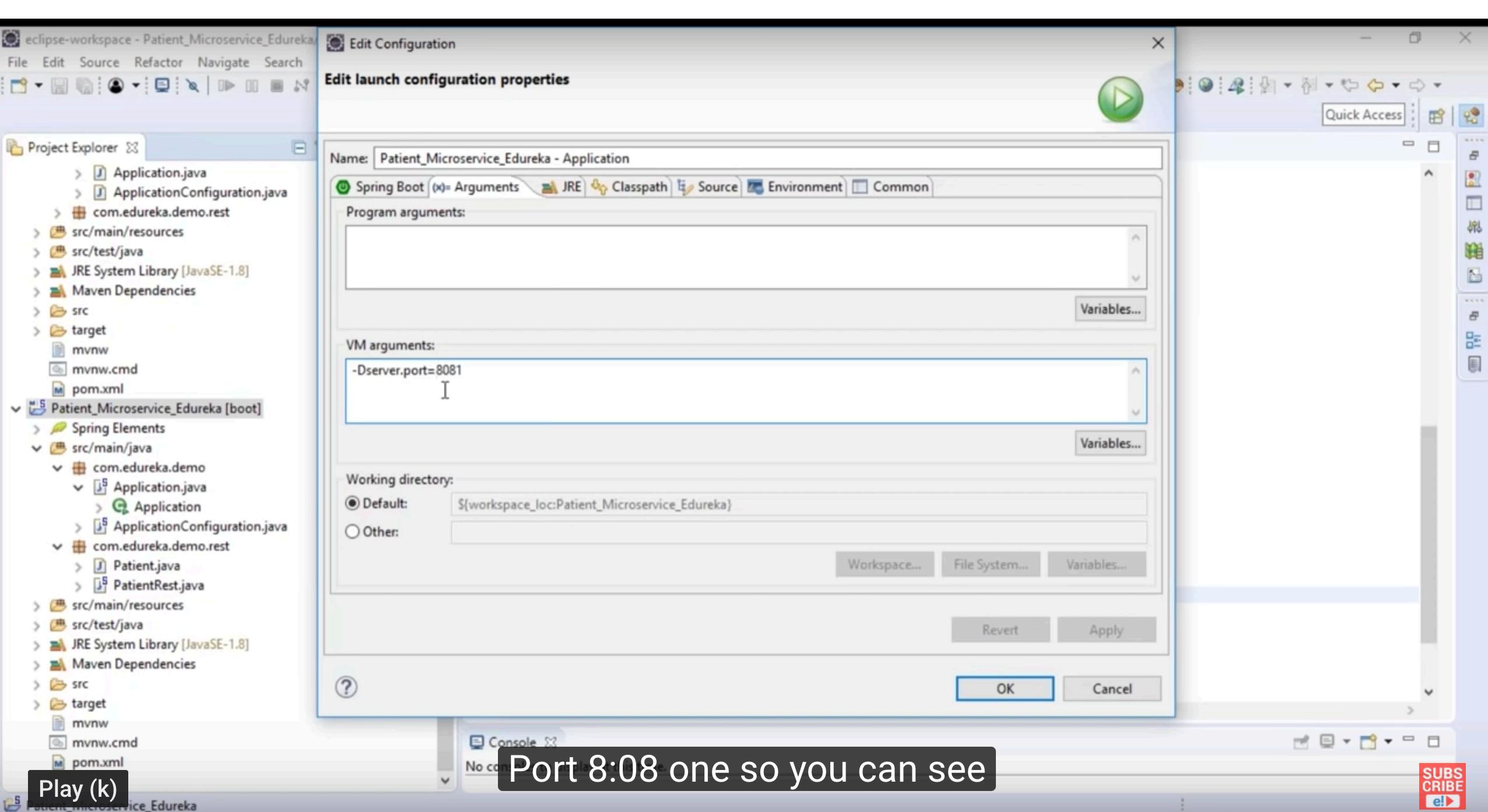
Play (k)

SUBSCRIBE

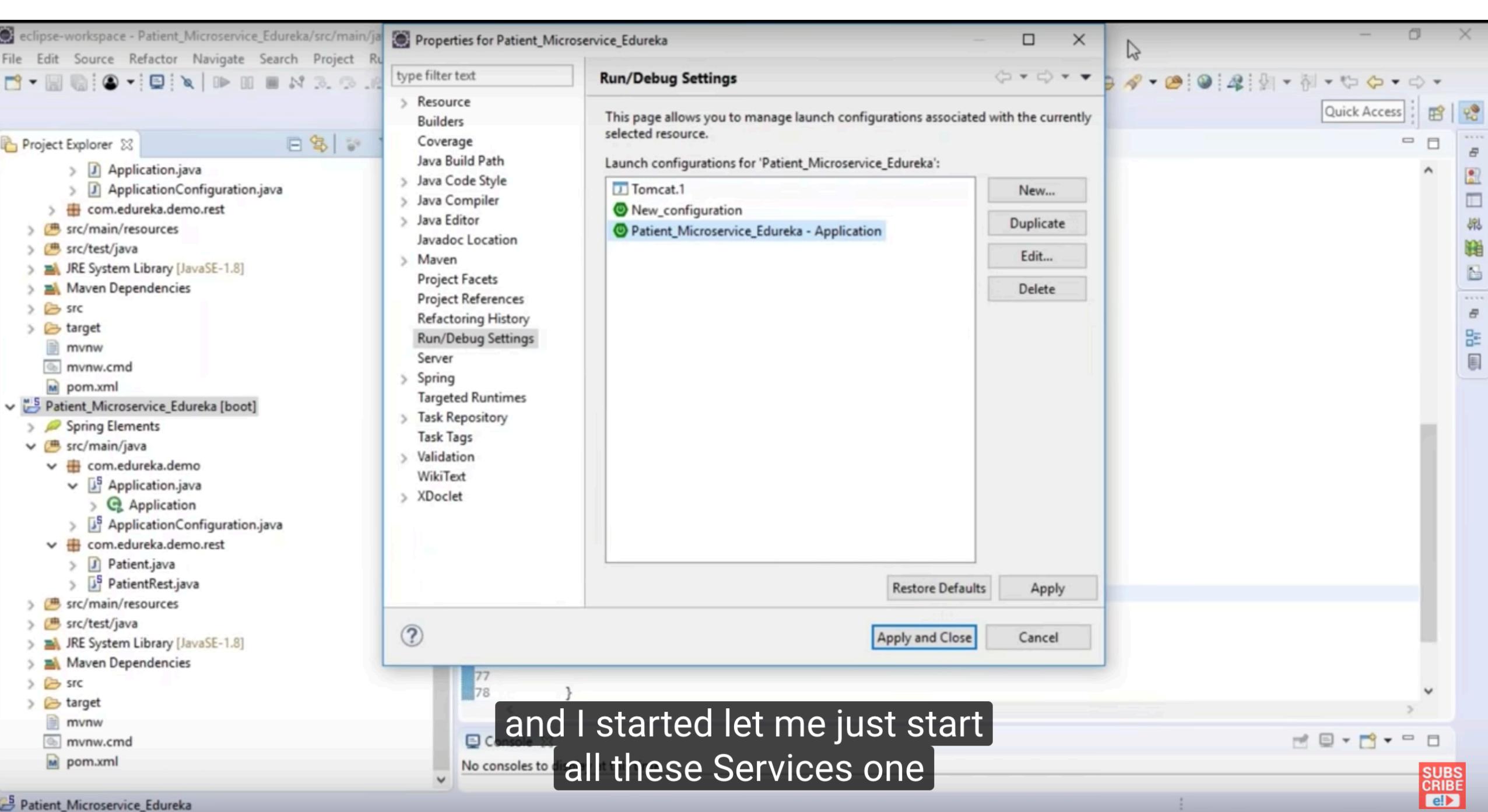


Play (k)

SUBSCRIBE



Port 8:08 one so you can see



Project Explorer

- New
- Go Into
- Show In Alt+Shift+W >
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path >
- Refactor Alt+Shift+T >
- Import...
- Export...
- Refresh F5
- Close Project
- Close Unrelated Projects
- Show in Remote Systems view
- Validate
- Coverage As
- Run As
- Debug As
- Profile As
- Restore from Local History...
- Maven
- Team
- Compare With
- Configure
- Source
- Spring Tools

```

    patient patient6 = new Patient();
    patient6.setId(006);
    patient6.setName("Monica");
    patient6.setEmail("patient6@edureka.co");

    clients.add(patient1);
    clients.add(patient2);
    clients.add(patient3);
    clients.add(patient4);
    clients.add(patient5);
    clients.add(patient6);

    return clients(MediaType.APPLICATION_JSON);
}

@GET("/patients/{id}")
public Patient getPatient(@PathParam("id") Long id) {
    return clients.get(id);
}

@POST("/patients")
public Patient addPatient(@RequestBody Patient patient) {
    clients.add(patient);
    return patient;
}

@PUT("/patients/{id}")
public Patient updatePatient(@PathParam("id") Long id, @RequestBody Patient patient) {
    Patient existingPatient = clients.get(id);
    if (existingPatient != null) {
        existingPatient.setName(patient.getName());
        existingPatient.setEmail(patient.getEmail());
        existingPatient.setId(patient.getId());
    }
    return existingPatient;
}

@DELETE("/patients/{id}")
public void deletePatient(@PathParam("id") Long id) {
    clients.remove(id);
}

```

Let's click on run a spring boot app similarly click

1 Run on Server Alt+Shift+X, R

2 Java Application Alt+Shift+X, J

3 JUnit Test Alt+Shift+X, T

4 Maven build Alt+Shift+X, M

5 Maven build... Alt+Shift+X, M

6 Maven clean Alt+Shift+X, M

7 Maven generate sources Alt+Shift+X, M

8 Maven test Alt+Shift+X, M

9 Maven test Alt+Shift+X, M

Project Explorer

- New
- Go Into
- Show In Alt+Shift+W
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path >
- Refactor Alt+Shift+T >
- Import...
- Export...
- Refresh F5
- Close Project**
- Close Unrelated Projects**
- Show in Remote Systems view
- Validate
- Coverage As
- Run As
- Debug As
- Profile As
- Restore from Local History...
- Maven
- Team
- Compare With
- Configure
- Source
- Spring Tools
- Properties Alt+Enter

on right click on this

Quick Access

File ApplicationConfiguration.java Patient.java PatientRest.java

```
Patient patient6 = new Patient();
patient6.setId(006);
patient6.setName("Monica");
patient6.setEmail("patient6@edureka.co");

clients.add(patient1);
clients.add(patient2);
clients.add(patient3);
clients.add(patient4);
clients.add(patient5);
clients.add(patient6);

}

@GET
@Produces(MediaType.APPLICATION_JSON)
public List<Patient> getClientes() {
    return clients;
}

@GET
@Path("patient")
@Produces(MediaType.APPLICATION_JSON)
public Patient getCliente(@QueryParam("id") long id) {

    Patient cli = null;

    for (Patient c : clients) {

        if (c.getId() == id)
            cli = c;
    }
}
```

SUBSCRIBE

Project Explorer

- New
- Go Into
- Show In Alt+Shift+W >
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+ Down
- Build Path >
- Refactor Alt+Shift+T >
- Import...
- Export...
- Refresh F5
- Close Project
- Close Unrelated Projects
- Show in Remote Systems view
- Validate
- Coverage As >
- Run As > **Run As**
- Debug As
- Profile As
- Restore from Local History...
- Maven
- Team
- Compare With
- Configure
- Source
- Spring Tools
- Properties Alt+Enter

Quick Access

ion.java ApplicationConfiguration.java Patient.java PatientRest.java

```
Patient patient6 = new Patient();
patient6.setId(006);
patient6.setName("Monica");
patient6.setEmail("patient6@edureka.co");

clients.add(patient1);
clients.add(patient2);
clients.add(patient3);
clients.add(patient4);
clients.add(patient5);
clients.add(patient6);

}

@GET
@Produces(MediaType.APPLICATION_JSON)
public List<Patient> getClientes() {
    return clients;
}

@GET
@Path("patient")
@Produces(MediaType.APPLICATION_JSON)
long id) {
```

1 Run on Server Alt+Shift+X, R
2 Java Application Alt+Shift+X, J
3 JUnit Test Alt+Shift+X, T
4 Maven build Alt+Shift+X, M
5 Maven build...
6 Maven clean
7 Maven generate-sources
8 Maven test
9 Maven test

java\jre1.8.0_161\bin\javaw.exe (16-Mar-2018, 12:20:45 PM)

and I'll just run it as spring boot up again,

SUBSCRIBE

Project Explorer

- Diagnosis_Microservice
- src/main/java
 - com.edureka
 - Application
 - Application
- src/main/resources
- src/test/java
- JRE System Library
- Maven Dependencies
- src
- target
 - mvnw
 - mvnw.cmd
 - pom.xml

Doctor_Microservice

- Spring Elements
- src/main/java
 - com.edureka
 - Application
 - Application
 - com.edureka
- src/main/resources
- src/test/java
- JRE System Library
- Maven Dependencies
- src
- target
 - mvnw
 - mvnw.cmd
 - pom.xml

New

Go Into

Show In

Alt+Shift+W >

Copy Ctrl+C

Copy Qualified Name

Paste Ctrl+V

Delete Delete

Remove from Context Ctrl+Alt+Shift+Down

Build Path >

Refactor Alt+Shift+T >

Import...

Export...

Refresh F5

Close Project

Close Unrelated Projects

Show in Remote Systems view

Validate

Coverage As

Run As >

- 1 Run on Server Alt+Shift+X, R
- 2 Java Application Alt+Shift+X, J
- 3 JUnit Test Alt+Shift+X, T
- 4 Maven build Alt+Shift+X, M
- 5 Maven build... Alt+Shift+X, M
- 6 Maven clean Alt+Shift+X, C
- 7 Maven generate-sources Alt+Shift+X, G
- 8 Maven install Alt+Shift+X, I
- 9 Maven test Alt+Shift+X, T

Source

Spring Tools

Properties Alt+Enter

Quick Access

Code Editor

Patient patient6 = new Patient();
patient6.setId(006);
patient6.setName("Monica");
patient6.setEmail("patient6@edureka.co");

clients.add(patient1);
clients.add(patient2);
clients.add(patient3);
clients.add(patient4);
clients.add(patient5);
clients.add(patient6);

GET
Produces(MediaType.APPLICATION_JSON)
public List<Patient> getClientes() {
 return clients;
}

GET
Path("patient")
Produces(MediaType.APPLICATION_JSON)
long id) {

System Files\Java\jre1.8.0_161\bin\javaw.exe (16-Mar-2018, 12:20:53 PM)

SUBSCRIBE

Project Explorer

- Diagnosis_Microservice_Edureka [boot]
 - Spring Elements
 - src/main/java
 - com.edureka.demo
 - Application.java
 - ApplicationConfiguration.java
 - com.edureka.demo.rest
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml
- Doctor_Microservice_Edureka [boot]
 - Spring Elements
 - src/main/java
 - com.edureka.demo
 - Application.java
 - ApplicationConfiguration.java
 - com.edureka.demo.rest
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml

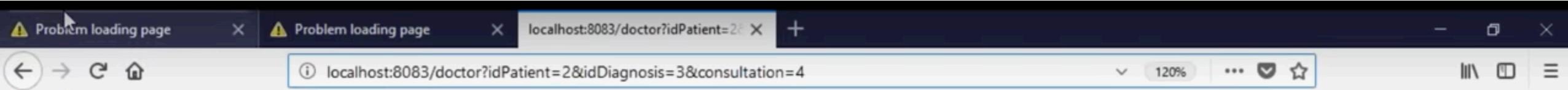
```
46     Patient patient6 = new Patient();
47     patient6.setId(006);
48     patient6.setName("Monica");
```

Console

Diagnosis_Microservice_Edureka - Application [Spring Boot App] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (16-Mar-2018, 12:20:59 PM)

```
\\|_)|_|_)|_|_|_|_|(|_|_| ) ) )
` |__| .|_|_|_|_|_\_,|_| / / /
=====|=====|/_=/ / / /
:: Spring Boot :: (v2.0.0.RELEASE)
```

```
2018-03-16 12:21:01.252 INFO 440 --- [           main] com.edureka.demo.Application        : Starting Application on DESI
2018-03-16 12:21:01.256 INFO 440 --- [           main] com.edureka.demo.Application        : No active profile set, falling back to default profiles: default
2018-03-16 12:21:01.335 INFO 440 --- [           main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@6353333: startup date [Fri Mar 16 12:21:01 IST 2018]; root of context hierarchy
2018-03-16 12:21:02.087 INFO 440 --- [           main] o.s.b.f.s.DefaultListableBeanFactory : Overriding bean definition for bean 'rootConfigProcessor' in source [org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@6353333: startup date [Fri Mar 16 12:21:01 IST 2018]; root of context hierarchy]
2018-03-16 12:21:02.455 INFO 440 --- [           main] f.a.AutowiredAnnotationBeanPostProcessor : JSR-330 'javax.inject.Inject' annotation found and豆子自动注入无名 Bean
2018-03-16 12:21:03.122 INFO 440 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port: 8080
2018-03-16 12:21:03.164 INFO 440 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2018-03-16 12:21:03.165 INFO 440 --- [           main] o.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/9.0.0.M1
2018-03-16 12:21:03.171 INFO 440 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performance in production environments was loaded
2018-03-16 12:21:03.330 INFO 440 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2018-03-16 12:21:03.330 INFO 440 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization started
2018-03-16 12:21:03.731 INFO 440 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet com.edureka.demo.Application mapped to [/]
2018-03-16 12:21:03.732 INFO 440 --- [ost-startStop-1] o.s.b.w.servlet.DispatcherServlet : DispatcherServlet with name 'dispatcherServlet' mapped to [/]
2018-03-16 12:21:03.736 INFO 440 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to URL patterns: [/*]
2018-03-16 12:21:03.736 INFO 440 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to URL patterns: [/*]
2018-03-16 12:21:03.736 INFO 440 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to URL patterns: [/*]
2018-03-16 12:21:03.737 INFO 440 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to URL patterns: [/*]
2018-03-16 12:21:04.011 INFO 440 --- [           main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: [com.edureka.demo.Application]
2018-03-16 12:21:04.091 INFO 440 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[{"path": "/error"}]" onto pul
2018-03-16 12:21:04.092 INFO 440 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[{"path": "/*", "produces": "application/json"}, {"path": "/*", "produces": "application/xml"}]" onto pul
2018-03-16 12:21:04.133 INFO 440 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler
2018-03-16 12:21:04.133 INFO 440 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto han
2018-03-16 12:21:04.190 INFO 440 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico]
```



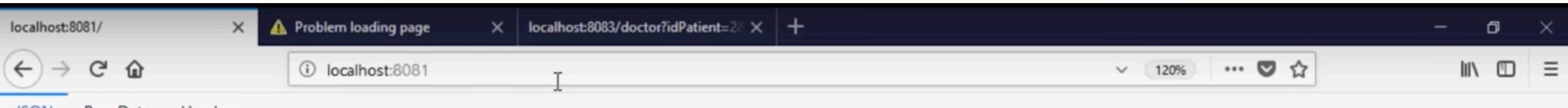
JSON Raw Data Headers

[Save](#) [Copy](#)

 Filter JSON

```
id: 1
consultation: 4
appointmentDate: "2018-03-16T05:19:40.600+0000"
patient:
  id: 2
  name: "Rachel"
  email: "patient2@edureka.co"
diagnosis:
  id: 3
  name: "Malaria"
description: "Symptoms are chills, fever and sweating, usually occurring a few weeks after being bitten."
```

All right.



```
Save Copy ▾ 0:  
    id:      1  
    name:    "Joey"  
    email:   "patient1@edureka.co"  
▼ 1:  
    id:      2  
    name:    "Rachel"  
    email:   "patient2@edureka.co"  
▼ 2:  
    id:      3  
    name:    "Phoebe"  
    email:   "patient3@edureka.co"  
▼ 3:  
    id:      4  
    name:    "Chandler"  
    email:   "patient4@edureka.co"  
▼ 4:  
    id:      5  
    name:    "Ross"  
    email:   "patient5@edureka.co"  
▼ 5:  
    id:      6  
    name:    "Monica"  
    email:   "patient6@edureka.co"
```

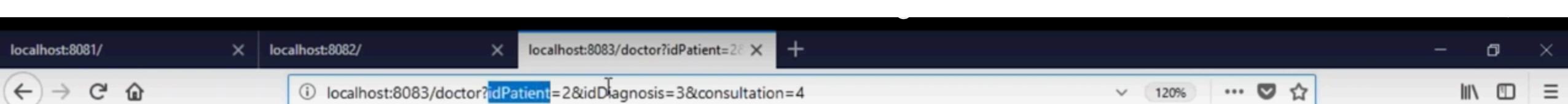
Let me reload this page again.

The screenshot shows a browser window with three tabs open. The active tab is titled 'localhost:8082' and displays a JSON array of four disease entries. Each entry contains an 'id', 'name', and 'description'. The 'description' field for each disease is a detailed paragraph. The browser interface includes standard navigation buttons, a zoom level of 120%, and a 'Filter JSON' button.

```
[{"id": 1, "name": "Viral Fever", "description": "Caused by viruses are among the most frequent causes of fever in adults. Common symptoms can include a runny nose, sore throat, cough, hoarseness, and muscle aches."}, {"id": 2, "name": "Pneumonia", "description": "Symptoms include a cough with phlegm or pus, fever, chills and difficulty breathing."}, {"id": 3, "name": "Malaria", "description": "Symptoms are chills, fever and sweating, usually occurring a few weeks after being bitten."}, {"id": 4, "name": "Typhoid", "description": "Symptoms include high fever, headache, stomach pain, weakness, vomiting and loose stools."}]
```

Right so are high of the diagnosis Microservice





JSON Raw Data Headers

```
Save Copy Filter JSON

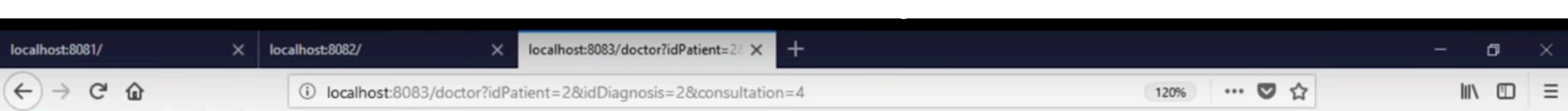
id: 1
consultation: 4
appointmentDate: "2018-03-16T05:19:40.600+0000"
patient:
  id: 2
  name: "Rachel"
  email: "patient2@edureka.co"
diagnosis:
  id: 3
  name: "Malaria"
  description: "Symptoms are chills, fever and sweating, usually occurring a few weeks after being bitten."
```

If you notice I have written
patient ID right diagnosis

edureka!

SUBSCRIBE





JSON Raw Data Headers

```
Save Copy Filter JSON

id: 2
consultation: 4
appointmentDate: "2018-03-16T06:52:10.708+0000"
patient:
  id: 2
  name: "Rachel"
  email: "patient2@edureka.co"
diagnosis:
  id: 2
  name: "Pneumonia"
  description: "Symptoms include a cough with phlegm or pus, fever, chills and difficulty breathing."
```

as to write sobre child
has been diagnosed

The screenshot shows a browser window with three tabs open: 'localhost:8081/' (closed), 'localhost:8082/' (closed), and 'localhost:8083/doctor?idPatient=2'. The active tab displays a JSON object representing a medical consultation record. The JSON structure includes fields for the consultation itself (id: 3, consultation: 4, appointmentDate: "2018-03-16T06:52:20.004+0000") and its patient (id: 2, name: "Rachel", email: "patient2@edureka.co"). It also includes a diagnosis (id: 1, name: "Viral Fever") and a detailed description of the condition.

```
localhost:8083/doctor?idPatient=2 X +  
localhost:8083/doctor?idPatient=2&idDiagnosis=1&consultation=4  
120% ⚙️ ⚡ ⚪  
JSON Raw Data Headers  
Save Copy Filter JSON  
id: 3  
consultation: 4  
appointmentDate: "2018-03-16T06:52:20.004+0000"  
patient:  
  id: 2  
  name: "Rachel"  
  email: "patient2@edureka.co"  
diagnosis:  
  id: 1  
  name: "Viral Fever"  
description: "Caused by viruses are among the most frequent causes of fever in adults. Common symptoms can include a runny nose, sore throat, cough, hoarseness, and muscle aches."
```

She has been diagnosed
with viral fever here.]



Thank You