


[Home](#)
[Blogs](#)
[Videos](#)
[Interview Questions](#)
<https://www.edureka.co/blog/intervie>

Microservices Tutorial – Learn all about

Sahiti (<https://www.edureka.co/blog/author/sahiti-kappagantulaedureka-co/>) Jul 8, 2018

Microservices with Example

Recommended by 47 users (https://www.facebook.com/sharer/sharer.php?u=https://www.edureka.co/blog/microservices-tutorial-tutorial-tutorial-tutorial-with-example) (https://www.linkedin.com/share?url=https://www.edureka.co/blog/microservices-tutorial-tutorial-tutorial-tutorial-with-example) (https://twitter.com/edureka?ref=share) (https://www.pinterest.com/pin/1153464444444444444/)



[Add to Bookmark](#) (<https://www.edureka.co/blog/microservices-tutorial-with-example>)
 [Email this Post](#)

[8.5K](#)
[\(https://www.edureka.co/blog/microservices-tutorial-with-example#comments-wrapper\)](#) 144

https://www.edureka.co/blog/microservices-tutorial-with-example#disqus_thread

Microservices Tutorial

Microservices tutorial is the third blog of Microservices blog series. I hope that you have read my previous blog on **What is Microservices** (<https://www.edureka.co/blog/what-is-microservices/>) that explains the architecture, compares microservices with monolithic and SOA, and also explores when to use microservices with the help of use-cases.

In this Microservices tutorial, the following topics will be covered:

- Monolithic Architecture
- Challenges of Monolithic Architecture
- What is Microservices
- Microservice Architecture
- Microservices Example – Demo

Let us explore the concepts of microservices through a use-case on Mediamore.com.

Mediamore is an entertainment company which provides streaming media and videos online. It consists of various genres of TV Shows in different languages. Like many other companies mediamore started its journey with a monolithic architecture.

Let us now explore the monolithic framework of mediamore.

Monolithic Architecture



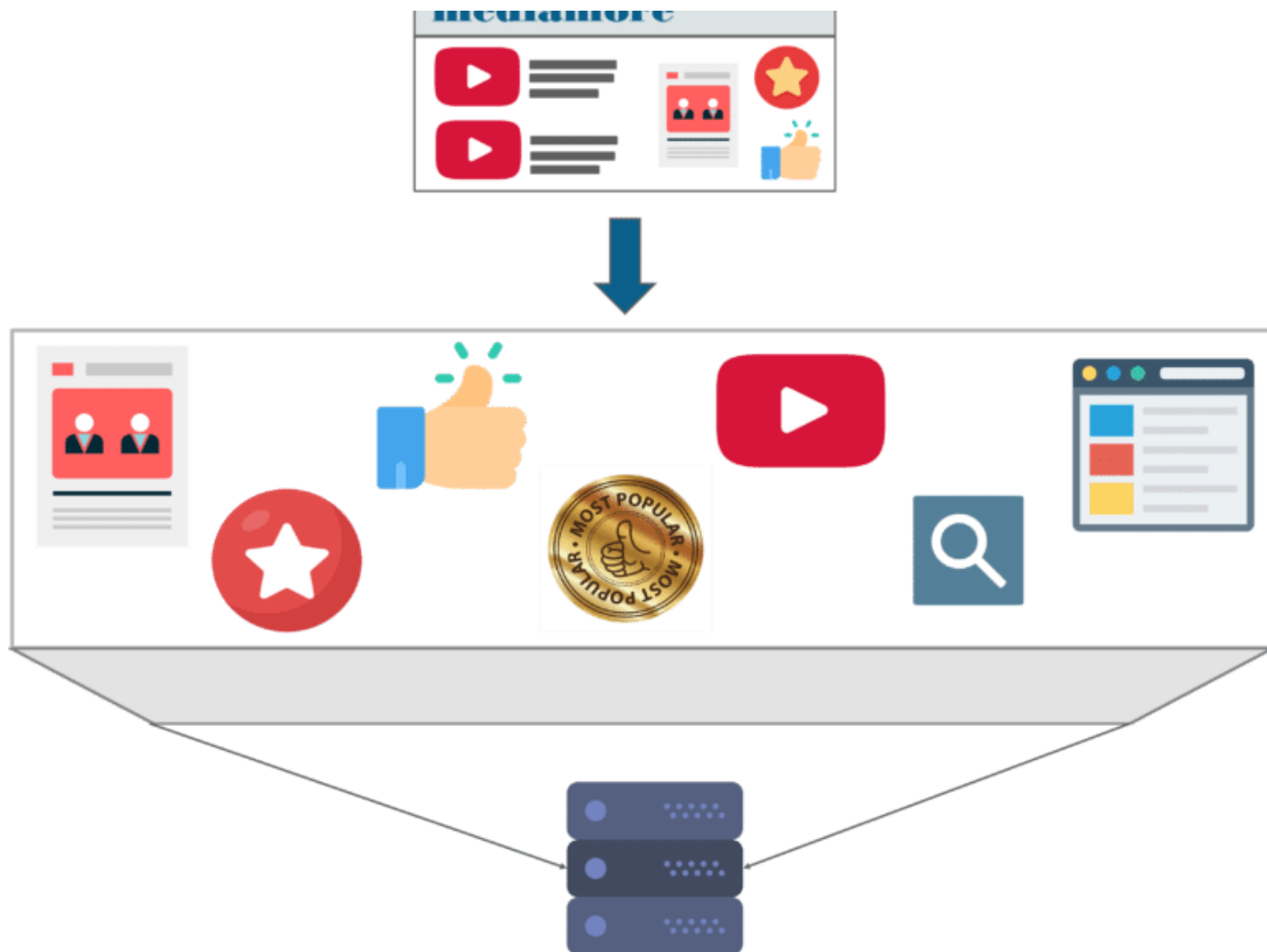


Figure 1: Monolithic Architecture of Mediamore – Microservices Tutorial

Refer to the above diagram. We can infer that all the features such as the search, user-info, recommendations, video playlist and others are put on a single database using single code.

Now, let me tell you the challenges faced by the developers while using a monolithic framework by using some scenarios.

Challenges of Monolithic Architecture

Scenario 1: Scalability: *Let's assume that the developers want to update the playlist according to most popular tv shows and also simultaneously want to update all videos to HD quality.*

The developers cannot scale the application simultaneously. New instances of the same application have to be created every time a new feature has to be developed or deployed.

Scenario 2: Agility: *Assume that developers want to make immediate changes in the application.*

The monolithic application can definitely accommodate these changes. But, the problem here is that the developers have to rebuild the code for every small change.

Scenario 3: Hybrid Technologies: *Suppose developers of this application are comfortable with*

various technologies like JAVA, C++, .NET, C#.

Even though they are comfortable with various technologies, they still have to build large and complex applications on a single technology.

Scenario 4: Fault Tolerance: *Let's suppose that a specific feature is not working in the application.*

The complete system goes down because of this problem. In order to tackle this problem, the application has to be re-built, re-tested and also re-deployed.

So, how did the developers of mediamore overcome these complexities?

Developers thus decided to re-architect their monolithic application into multiple individual deployable components, called as microservices.

Here lies the million dollar question!

What is Microservices?

Microservices is an architecture wherein all the components of the system are put into individual components, which can be built, deployed, and scaled individually.

Let me explain you with a simple analogy.

You must have seen how bees build their honeycomb by aligning hexagonal wax cells. They initially start with a small section using various materials and continue to build a large beehive out of it. These cells form a pattern resulting in a strong structure which holds together a particular section of the beehive. Here, each cell is independent of the other but it is also correlated with the other cells. This means that damage to one cell does not damage the other cells, so, bees can reconstruct these cells without impacting the complete beehive.





Figure 2: Beehive Representation of Microservices – Microservices Tutorial

Refer to the above diagram. Here, each hexagonal shape represents an individual service component. Similar to the working of bees, each agile team builds an individual service component with the available frameworks and the chosen technology stack. Just as in a beehive, each service component forms a strong microservice architecture to provide better scalability. Also, issues with each service component can be handled individually by the agile team with no or minimal impact on the entire application.

The next question that may come to your mind is how do the different components of microservice architecture work together.

But, before that, let me list down the components of the microservice architecture.

Refer to the below diagram.

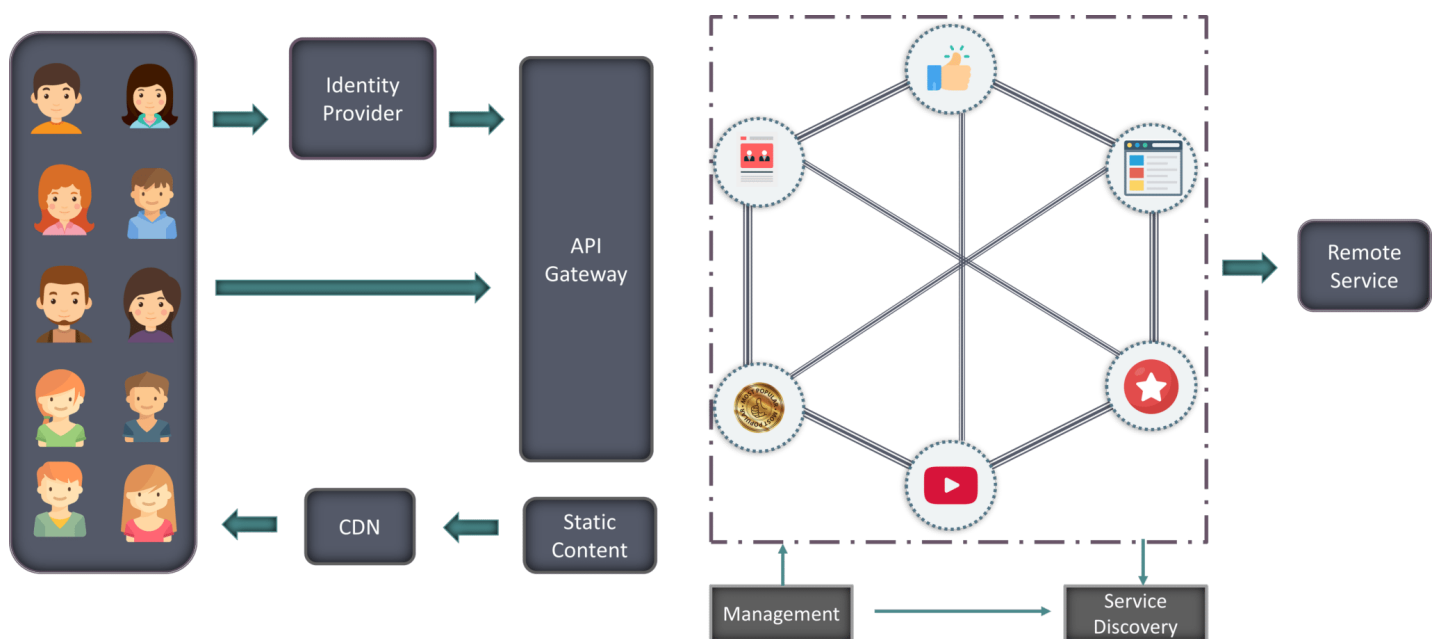


Figure 3: Microservices Architecture – Microservices Tutorial

- **Clients** – Different users from various devices send requests.
- **Identity Providers** – Authenticates user or clients identities and issues security tokens.
- **API Gateway** – Handles client requests.
- **Static Content** – Houses all the content of the system.
- **Management** – Balances services on nodes and identifies failures.
- **Service Discovery** – A guide to find the route of communication between microservices.
- **Content Delivery Networks** – Distributed network of proxy servers and their data centers.
- **Remote Service** – Enables the remote access information that resides on a network of IT devices.

Let me now brief you on how these components work together on mediamore by considering a

scenario.

Microservice Architecture

Scenario:

Alice is an avid user of mediamore. She uses mediamore regularly to watch her favorite series online. She recently missed watching an episode of her favorite TV show.

When Alice logs in to the application, she sees the most recommended content on her home page. After some searching, she finally finds her TV Show.

But, what if Alice wants to get her TV Show with a single click.

How will the developers work together to fulfill Alice's request?

Alice's request is passed on to the **Identity Provider**. Identity provider thus authenticates Alice's request by identifying her as a regular user on mediamore.

These requests are passed to the **API Gateway** which acts as an entry point for Alice to forward her requests to the appropriate microservices.

Each feature has its own working microservice, handling their own data. These microservices also have their own **load balancers** and **execution environments** to function properly.

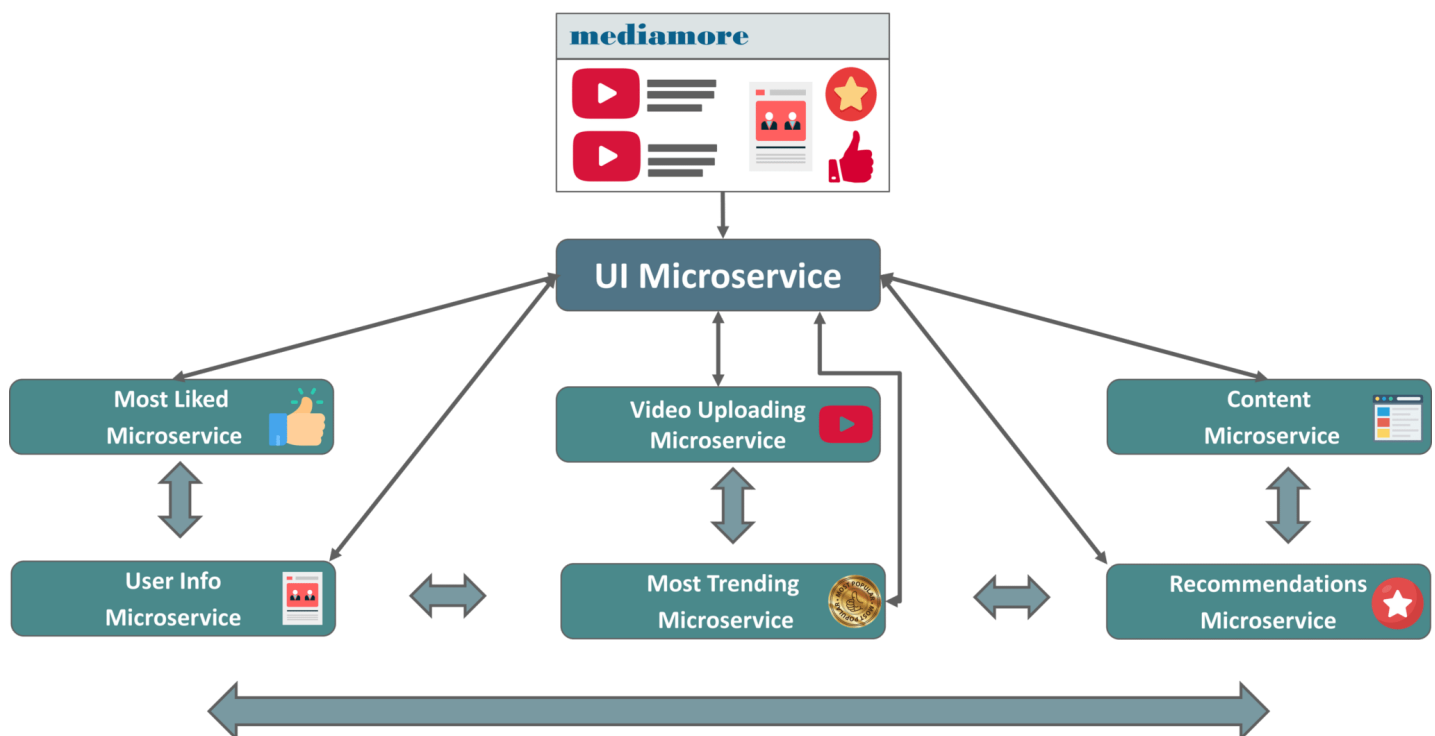


Figure 4: Microservices Architecture of Mediamore – Microservices Tutorial

Refer to the diagram below. Each microservice is handled by a small agile team such as content team, video uploading team, most trending team, search team etc.

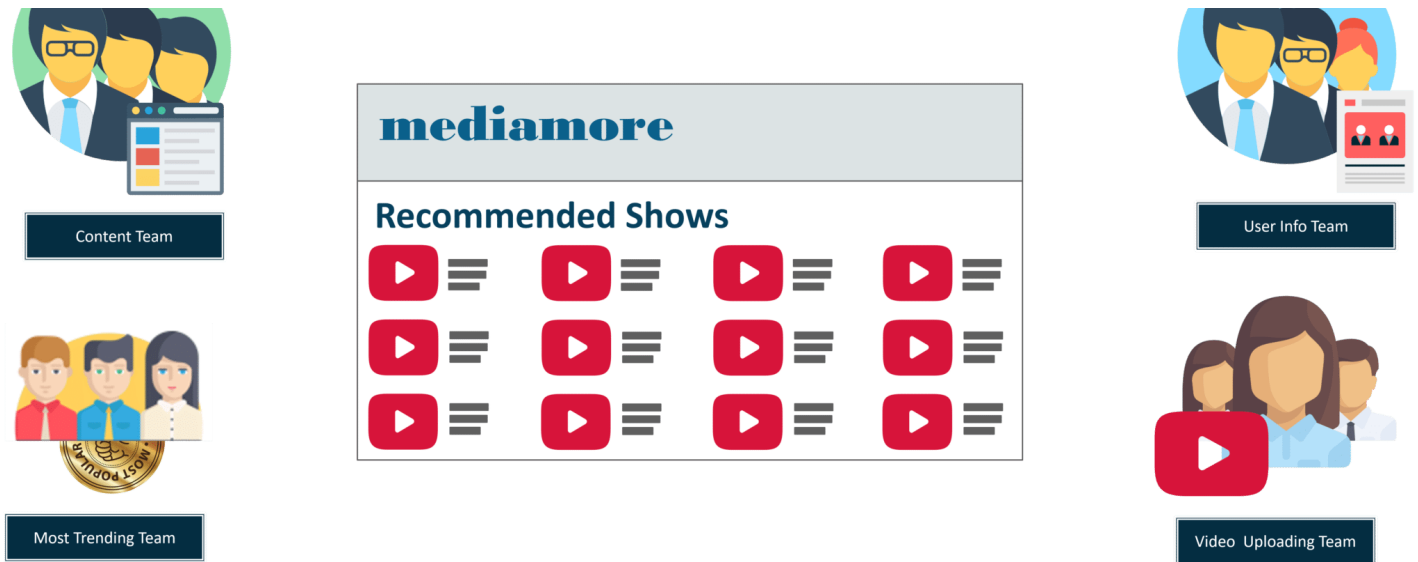


Figure 5: Division of Teams of Mediamore – Microservices Tutorial

- The content team consists of millions of TV Shows that the application provides.
- The video uploading team have the responsibility to upload all the content into the application
- The most trending team houses the most trending shows according to the geographical location of users and so on.

These small teams of developers relate each and every piece of content with the metadata that describes the searched content. Then, metadata is fed into another microservice i.e. the search function which ensures Alice's search results are captured into the content catalog.

Then, the third microservice i.e. most trending microservice captures the trending content among all the mediamore users according to their geographical locations.

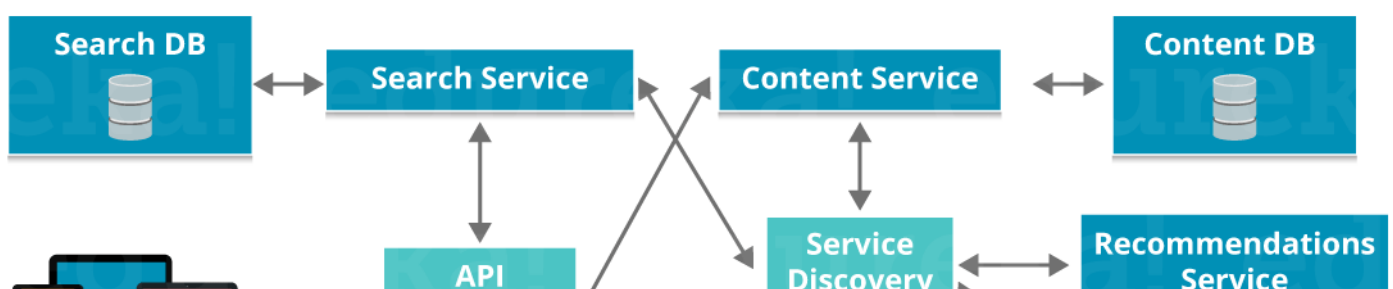
The content from this microservice is what Alice sees when she first logs into mediamore.

These individually deployable microservices are put in specific **containers** to join the application. Containers are used to deliver the code to the sector where deployment is required.

But before they join the application to work together, they have to find each other to fulfil Alice's request.

How do these microservices find each other?

Microservices use **service discovery** which acts as a guide to find the route of communication between each of them. Microservices then communicate with each other via a stateless server i.e. either by **HTTP Request/Message Bus**.



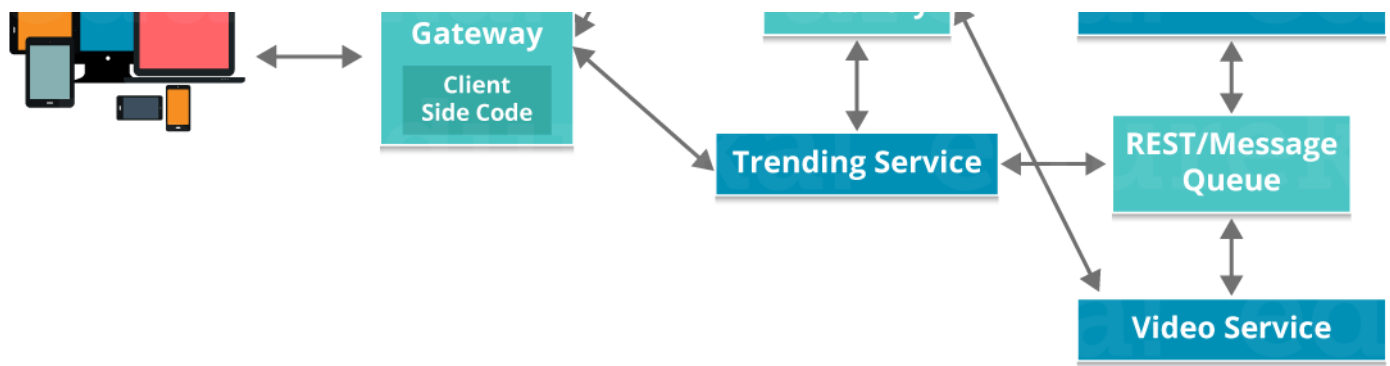


Figure 6: Communication Between Microservices – Microservices Tutorial

These microservices communicate with each other using an **Application Program Interface(API)**. After the Microservices communicate within themselves, they deploy the **static content** to a cloud-based storage service that can deliver them directly to the clients via **Content Delivery Networks (CDNs)**.

So, when Alice searches for her TV Show, the search microservice communicates with the content catalog service in API about what is Alice searching for and then these microservices compare the typed words with the metadata they already have.

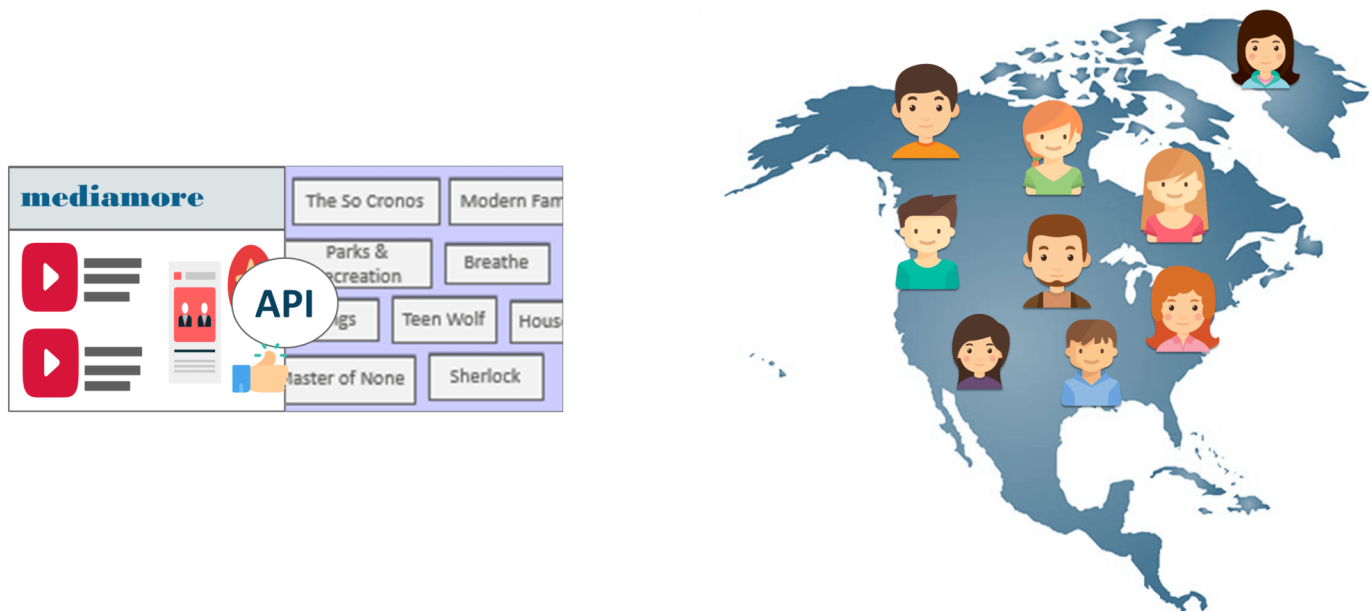


Figure 7: Representation of how to search operation is performed with the help of API – Microservices Tutorial

Once the teams of developers capture the most typed words by Alice, the analytics team update the code in recommendations microservice and compare Alice's most viewed content and preferences to popular content among other users in the same geographical region.

This means that the next time Alice logs on to the application, she not only sees the most popular content but also finds a personalized playlist which contains the shows she has previously viewed.

In this way, Alice's request is fulfilled by the development team in a quick manner as they did not have to build the complete application again and just had to update the code to deploy this new functionality.

So this way microservices invoke parallel environments to satisfy millions of customers with varying interests.

Now in this Microservices Tutorial, my next section will focus on Hands-On.

Subscribe to our youtube channel to get new updates..!



edureka!

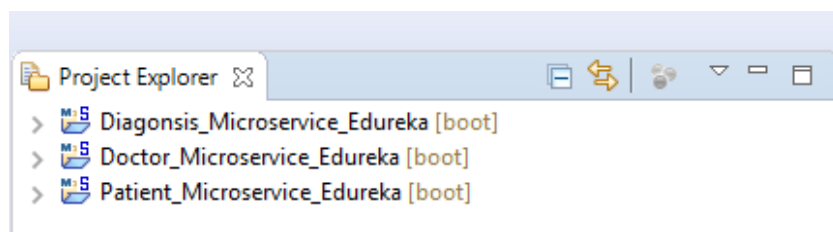
YouTube

465K

Microservices Example – Demo

To demonstrate the concepts of microservices, I have created 3 Maven Projects called as Doctor_Microservice_Edureka, Diagnosis_Microservice_Edureka, and Patient_Microservice_Edureka using Spring Boot.

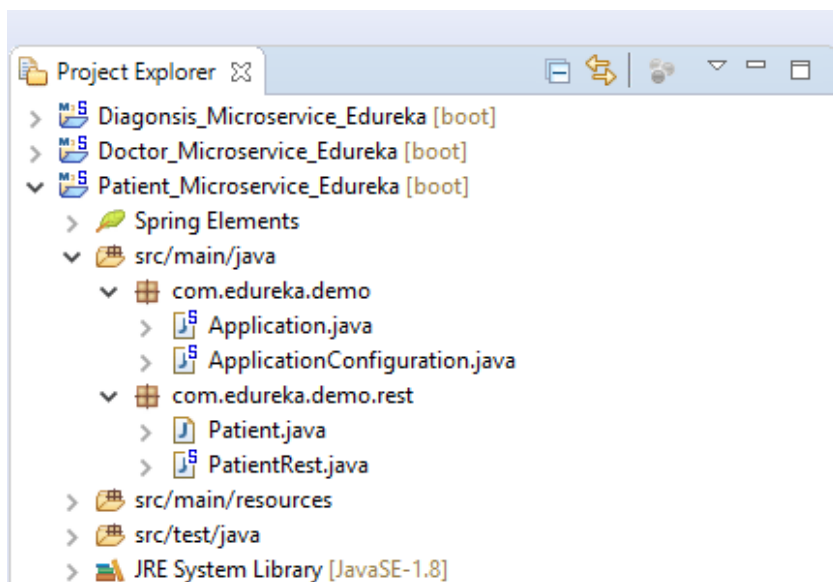
Refer to the snapshot below.

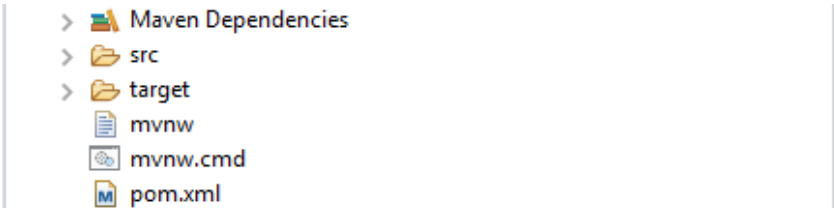


Before you understand how these 3 projects interact with each other. Let me brief you on the files of these projects.

To explain this I will consider the project **Patient_Microservice_Edureka** and list down its basic files.

Refer to the snapshot below.





- > Maven Dependencies
- > src
- > target
- mvnw
- mvnw.cmd
- pom.xml

Pom.xml – Dependencies are added for the creation of REST services.

Application.java – Identical class in all three projects. Acts as an initiator to Spring Boot.

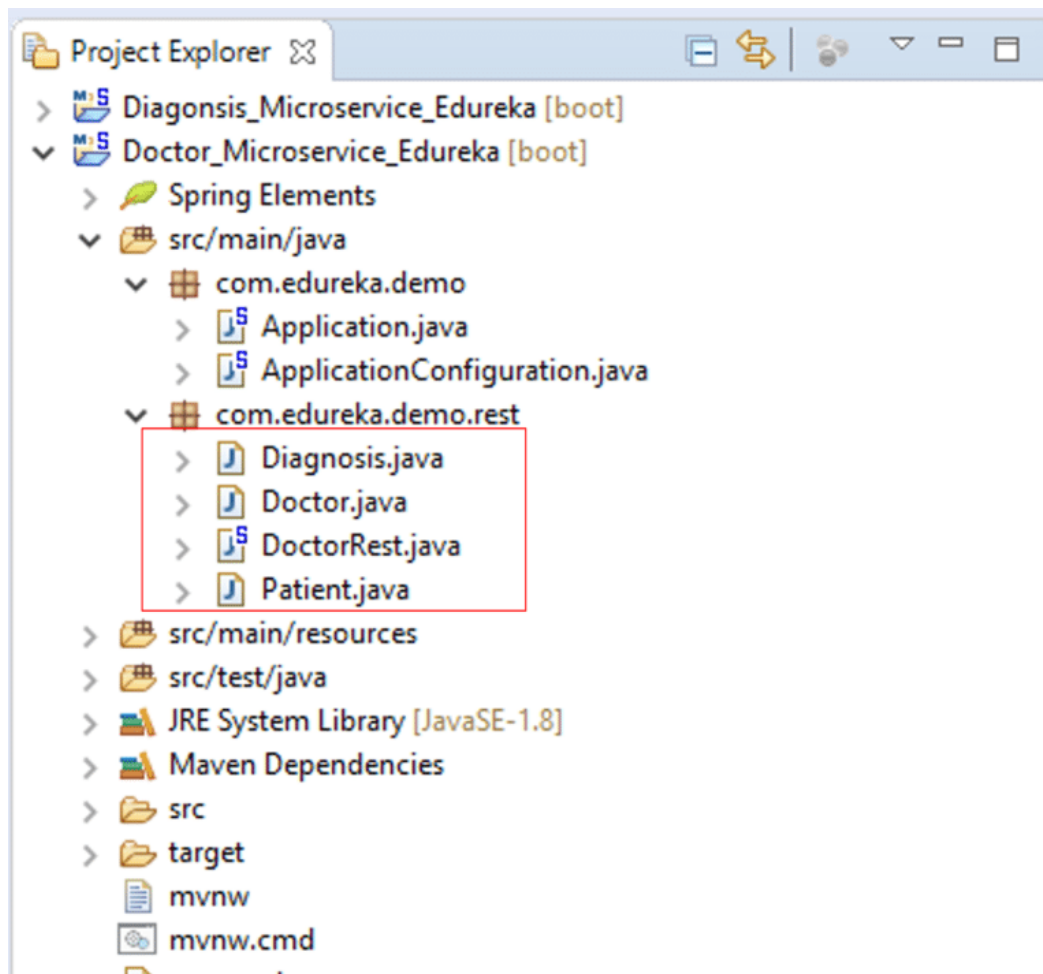
ApplicationConfiguration.java – Research configuration class responsible for exposing REST services for application users.

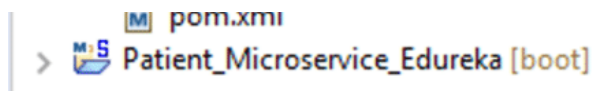
Patient.java – A simple class consisting of input such as the patient's name, id, email.

PatientRest.java – Starts the implementation of the REST services in the project.

In this way, similar files are created for the other 2 projects with some additional files in Doctor_Microservice.

REST services are thus created to search patients and the diagnosis. Keys of patient and diagnosis are passed as a parameter to a method(PatientDetails) in Doctor_Microservice. This method gets the data of the patients and diseases.





Refer to the snapshot above. Here, we observe that **Patient** and **Diagnosis** classes are included in **Doctor_Microservice_Edureka**. These classes are cloned from their original projects.

Then to start the REST services, I have initialized **Patient_Microservice_Edureka** on port **8081**, **Diagnosis_Microservice_Edureka** on **8082** and the **Doctor_Microservice_Edureka** on port **8083**.

With the above configurations, when I run each service simultaneously, 3 different console windows run in the Eclipse console window.

As you can see below that after we run the projects, Spring Boot generates a boot log. This log consists information on initializing tomcat and its associated resources. The last line of the log indicates us whether our application has started or not.

```

:: Spring Boot :: (v2.0.0.RELEASE)

2018-03-14 16:04:31.860 INFO 3760 --- [main] com.edureka.demo.Application : Starting Application on DESKTOP-DRSRV9P with PID 3760 (C:\Users\Sahit
2018-03-14 16:04:31.863 INFO 3760 --- [main] com.edureka.demo.Application : No active profile set, falling back to default profiles: default
2018-03-14 16:04:31.934 INFO 3760 --- [main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationCon
2018-03-14 16:04:32.535 INFO 3760 --- [main] o.s.b.f.s.DefaultListableBeanFactory : Overriding bean definition for bean 'requestContextFilter' with a dif
2018-03-14 16:04:32.733 INFO 3760 --- [main] f.a.AutowiredAnnotationBeanPostProcessor : JSR-330 'javax.inject.Inject' annotation found and supported for auto
2018-03-14 16:04:33.102 INFO 3760 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2018-03-14 16:04:33.124 INFO 3760 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2018-03-14 16:04:33.125 INFO 3760 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.28
2018-03-14 16:04:33.130 INFO 3760 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal perfo
2018-03-14 16:04:33.227 INFO 3760 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2018-03-14 16:04:33.228 INFO 3760 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 1299 ms
2018-03-14 16:04:33.542 INFO 3760 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet com.edureka.demo.ApplicationConfiguration$JerseyConfig mapped
2018-03-14 16:04:33.543 INFO 3760 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
2018-03-14 16:04:33.547 INFO 3760 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [/]
2018-03-14 16:04:33.547 INFO 3760 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: [/]
2018-03-14 16:04:33.547 INFO 3760 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [/]
2018-03-14 16:04:33.547 INFO 3760 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [/]
2018-03-14 16:04:33.770 INFO 3760 --- [main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.c
2018-03-14 16:04:33.833 INFO 3760 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto public org.springframework.http.ResponseEntity
2018-03-14 16:04:33.834 INFO 3760 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[/error]" onto public org.springframework
2018-03-14 16:04:33.872 INFO 3760 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springf
2018-03-14 16:04:33.873 INFO 3760 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework
2018-03-14 16:04:33.911 INFO 3760 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.spr
2018-03-14 16:04:34.059 INFO 3760 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2018-03-14 16:04:34.098 INFO 3760 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8081 (http) with context path ''
2018-03-14 16:04:34.102 INFO 3760 --- [main] com.edureka.demo.Application : Started Application in 2.572 seconds (JVM running for 3.447)

```

To test if the patient and diagnosis services are functioning properly, one a browser (Mozilla Firefox) and go to **http://localhost:8081/** and **http://localhost:8082/** URLs. You should get the outputs as shown below:

localhost:8081
JSON Raw Data Headers
Save Copy

0:
id: 1
name: "Joey"
email: "patient1@edureka.co"
1:
id: 2
name: "Rachel"
email: "patient2@edureka.co"

localhost:8082
JSON Raw Data Headers
Save Copy

0:
id: 1
name: "Viral Fever"
description: "Caused by viruses are among and muscle aches."
1:
id: 2
name: "Pneumonia"

```

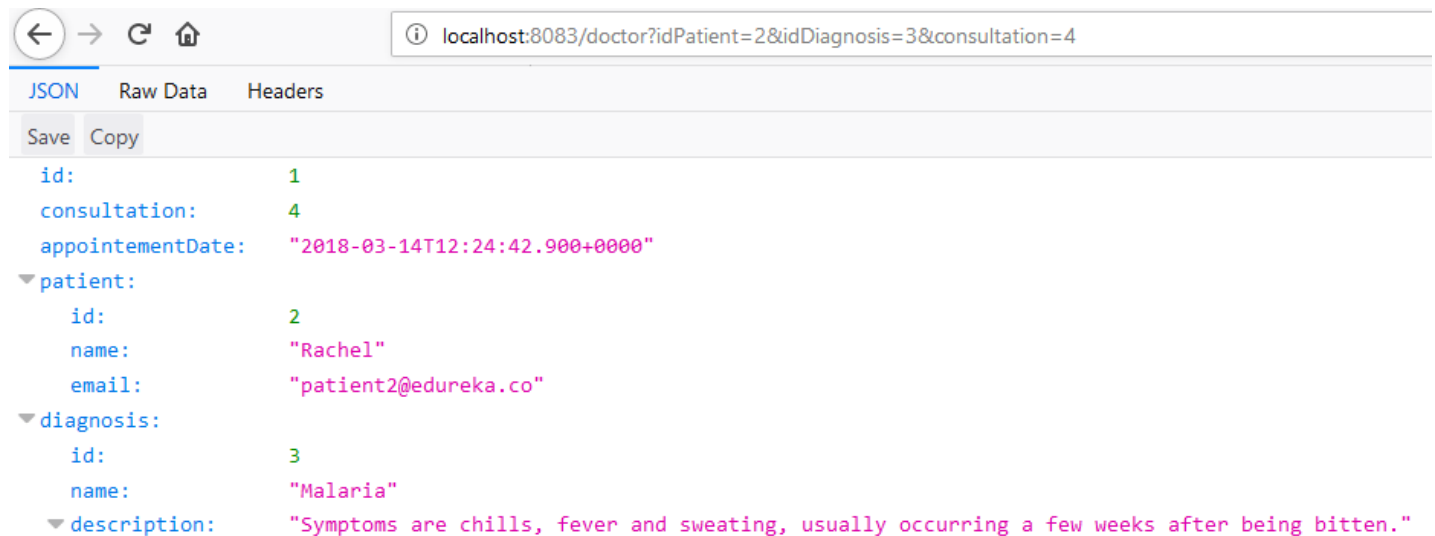
▼ 2:
  id:      3
  name:    "Phoebe"
  email:   "patient3@edureka.co"
▼ 3:
  id:      4
  name:    "Chandler"
  email:   "patient4@edureka.co"
▼ description: "Symptoms include a cough wit
▼ 2:
  id:      3
  name:    "Malaria"
▼ description: "Symptoms are chills, fever a
▼ 3:
  id:      4
  name:    "Typhoid"
▼ description: "Symptoms include high fever,

```

Finally, to test the functionality of the **Doctor_Microservice_Edureka**, I simulated the information of the patient with id 2, suffering from a disease of id 3 and consulting a doctor with consultation 4.

The URL used is as follows: **http://localhost:8083/doctor?idPatient=2&idDiagnosis=3&consultation=4**

Refer to the snapshot below for the output.



In this way, 3 Microservices interact with each other to produce the desired results.

I hope you have enjoyed reading this **Microservices Tutorial**. We have seen a simple example to understand how different microservices communicate with each other.

If you want the source code of the example shown, please comment out in the comments section.

After this Microservices Tutorial, we will be getting into the depth of practical implementations on microservices with the next blog i.e. Microservices using SpringBoot. Stay tuned!

Want To Explore More About Microservices?

View Batches Now

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

*Got a question for us? Please mention it in the comments section of " **Microservice Tutorial**" and I will get back to you.*



About Sahiti (12 Posts (<https://www.edureka.co/blog/author/sahiti-kappagantulaedureka-co/>))

[illegible]

<https://www.youtube.com/watch?v=7wvW9eUkacQ>

Share on

with- with- with- with-
example) example) example) example)

[← PREVIOUS](#)

NEXT >

Got your brain cells running?
Stay tuned to latest technology updates

Enter your Email Address

SUBSCRIBE

Related Posts



Microservices Tutorial – Learn all about Microservices with Example



(<https://www.edureka.co/blog/microservices-tutorial-with-example>)

Comments

25 Comments

144 Comments

<https://www.edureka.co/blog/>

Login

Recommend 7

Share

Sort by Best



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name



anibxl abril • 14 days ago

Excellent Article. Can you please send source code to my email : anibxlabril@gmail.com

2 • Reply • Share



EdurekaSupport Mod **anibxl abril** • 10 days ago

Hey anibxl, sure. We have sent the files to you. Kindly check and let us know if you need any help! :)

Do check out our Microservice course here: <https://www.edureka.co/micr...>

Thanks! :)

• Reply • Share



Annapurna • 13 days ago

HI Excellent Article .Could you please send the source code to my email :annapurna.prk@gmail.com

1 • Reply • Share



EdurekaSupport Mod **Annapurna** • 10 days ago

Hey Annapurna, thank you for watching our video. We have shared the files with you. Check out our blog on the top 10 reasons to learn Microservices: <https://www.edureka.co/blog...>

Do subscribe, like and share to stay connected with us. Cheers :)

• Reply • Share



mustak • 14 days ago

Interesting. Can you please share source code

1 • Reply • Share



EdurekaSupport Mod → mustak • 10 days ago

Hey Mustak,sure. Mention your email address and we will send it over. Cheers :)

^ | v • Reply • Share ›



Kolie Kay • 15 days ago

Interesting read. Really would like to take a look at the code to see what it is all about. replic8tor at gmail dot com.

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → Kolie Kay • 14 days ago

Hey Kolie, we have sent the codes to you. Do let us know if you need anything else! We are glad that you liked our content :)

^ | v • Reply • Share ›



Biplob Dn • 15 days ago

Hi, Nice article with simple example on microservices....do mail me the source code @ biplobdeb16@gmail.com.

Thanks,

Biplob

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → Biplob Dn • 14 days ago

Hey Biplob, we have sent the codes. We are glad that you liked our content. Do check out our Microservices Training program here: <https://www.edureka.co/micr...>
Hope this helps :)

^ | v • Reply • Share ›



Ramiah C • 15 days ago

Great article and easy to understand .

Please share the source code to ramaiah121@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → Ramiah C • 14 days ago

Hey Ramiah, we have shared the codes. Do check and let us know if you need anything else. Cheers :)

^ | v • Reply • Share ›



Bidyadhar Tripathy • 15 days ago

Hi,

This is a nice article with explanation.

Can you please send source code to my email : bidyaasp@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → Bidyadhar Tripathy • 14 days ago

Hev Bidvadar, thank you for reading our posts. We have sent the codes to you. Do

subscribe and stay connected with us. Cheers :)

^ | v • Reply • Share ›



Ramesh • 16 days ago

Good series of articles on Microservices.

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Ramesh • 14 days ago

Hey Ramesh, thank you for appreciating our efforts! Cheers :)

^ | v • Reply • Share ›



prathibha gowda • 16 days ago

Nice article.

Please send the code to email id :prathibha187@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ prathibha gowda • 14 days ago

Hey Prathibha, we have sen the codes to you. Do subscribe and stay connected with us. Cheers :)

^ | v • Reply • Share ›



Veerabhadra • 19 days ago

please share me the source code veeru.mallavarapu@yahoo.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Veerabhadra • 14 days ago

Hey Veerabhadra, we have shared the codes with you. We hope you liked our blog, cheers :)

^ | v • Reply • Share ›



Veerabhadra ➔ EdurekaSupport • 13 days ago

So where do I find the code base now ?

^ | v • Reply • Share ›



prakash barad • 21 days ago

thanks for great blog on Microservices with good real life example..

Please send code to email id: pdbarad1234@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ prakash barad • 14 days ago

Hey Prakash, thank you for appreciating our work. We have sent the codes to you. Do let us know if you need any help :)

^ | v • Reply • Share ›



Shiva Agrawal • 22 days ago

Hello Edureka,

Can you please send me the code of the above example to my email address:

shiva.agrawal@one97.net

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Shiva Agrawal • 14 days ago

Hey Shiva, we have shared the codes with you. Check out our Microservices course here:

<https://www.edureka.co/micr...>

Hope this helps :)

^ | v • Reply • Share ›



Alex • a month ago

Excellect article! Could you send me the source code please? My email is:

djalex_trelos@hotmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Alex • 14 days ago

Hey Alex, the codes have been sent. We are glad you found our blog useful. Subscribe and stay tuned for more updates on our channel. Cheers :)

^ | v • Reply • Share ›



Alexandros Rontogiannis • a month ago

Nice job! Could you send me the source code of the example to rontogiannis@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Alexandros Rontogiannis • 14 days ago

Hey Alexandros, we have sent the codes to you. Do check out our Microservices course here: <https://www.edureka.co/micr...>

Hope this helps :)

^ | v • Reply • Share ›



Rajyalakshmi S • a month ago

Good one. I was just checking for what is micro-services and also want to implement one sample to better understand how it works. I found this which is very helpful. Please send me code to:

raji.sapineni@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Rajyalakshmi S • 14 days ago

Hey Rajyalakshmi, we have sent the codes to you. Do subscribe and stay connected with us. Cheers :)

^ | v • Reply • Share ›



HAPPY • a month ago

Nice article.

Please send code to email id: azadkatiyar64@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → HAPPY • 14 days ago

Hey HAPPY, we have sent the codes to you. Do subscribe and stay connected with us. Cheers :)

^ | v • Reply • Share ›



Ramanathan Venugopal • a month ago

please send the source code to vramschennai@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → Ramanathan Venugopal • 14 days ago

Hey Ramanathan, we have shared the codes with you, do check out our Microservices course as well. You will find it worth pursuing: <https://www.edureka.co/micr...> Cheers :)

^ | v • Reply • Share ›



Silvia Joyce • a month ago

Very informative. Could you please send me the source code to silviajoyce26@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → Silvia Joyce • 14 days ago

Hey Silvia, thank you for reading our blog. We are glad you found it informative. We have sent the source code to you, do let us know if you need anything else. Cheers :)

^ | v • Reply • Share ›



akash • a month ago

Nice article.

Please send code to email id: akashdahr.apssdc@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → akash • a month ago

Hey Akash, thank you for giving our blog a read. We have sent the codes to you. Do let us know if you need anything else. :)

Meanwhile, check out our Microservices training program here:

<https://www.edureka.co/micr...>

^ | v • Reply • Share ›



Krishna Pandey • a month ago

Please send me the source code of This example at: ksnapd@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod → Krishna Pandey • a month ago

Hey Krishna, thank you for giving our blog a read. We have sent the codes to you. Do let us know if you need anything else. :)

Meanwhile, check out our Microservices training program here:

<https://www.edureka.co/micr...>

^ | v • Reply • Share ›



Roberto Pérez • a month ago

Excelent article, i'm reading it and learning new stuff about this architecture.

Congratulations and keep up the good work!

PS: please send the code to: rfperezc@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Roberto Pérez • a month ago

Hey Ads, thank you for giving our blog a read. We are glad that you liked our content. We have sent the codes to you. Do let us know if you need anything else. :)

Meanwhile, check out our Microservices training program here:

<https://www.edureka.co/micr...>

^ | v • Reply • Share ›



Tamar Bachaliashvili • a month ago

Hey, i need source code please. tbachaliashvili9@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Tamar Bachaliashvili • a month ago

Hey Tamar, thank you for giving our blog a read. We have sent the codes to you. Do let us know if you need anything else. :)

Meanwhile, check out our Microservices training program here:

<https://www.edureka.co/micr...>

^ | v • Reply • Share ›



Ragalatha P • a month ago

Hi! Please share the code

rglatha09@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ Ragalatha P • a month ago

Hey Ragalatha, thank you for giving our blog a read. We have sent the codes to you. Do let us know if you need anything else. :)

Meanwhile, check out our Microservices training program here:

<https://www.edureka.co/micr...>

^ | v • Reply • Share ›



raj • a month ago

Hi,

Please share the code rajkp499@gmail.com

1 ^ | v • Reply • Share ›



EdurekaSupport Mod ➔ raj • a month ago

Hey Raj, thank you for giving our blog a read. We have sent the codes to you. Do let us know if you need anything else. :)

Meanwhile, check out our Microservices training program here:

<https://www.edureka.co/micr...>

^ | v • Reply • Share ›



Avatar

This comment was deleted.

Load more comments

ALSO ON [HTTPS://WWW.EDUREKA.CO/BLOG/](https://www.edureka.co/blog/)

Java Thread Tutorial: Creating Threads and Multithreading in Java

AWS DevOps: Introduction to DevOps on AWS

edureka!

© 2014 Brain4ce Education Solutions Pvt. Ltd.

**Subscribe
to our newsletter**

Enter your Email Address

SUBSCRIBE

Related Course



Learn Java From Experts!

Explore Now

(<https://www.edureka.co/java-j2ee-soa-training>)

(<https://www.edureka.co/java-j2ee-soa-training>)



Master In Spring From Experts!

Explore Now

(<https://www.edureka.co/spring-framework>)

(<https://www.edureka.co/spring-framework>)

SEE ALL COURSES → (<https://www.edureka.co/all-courses>)

Get Live Training & Certification!