What exactly is Apache Camel?

Asked 12 years, 11 months ago Modified 8 months ago

Viewed 693k times

1643

I don't understand what exactly Camel does.

If you could give in 101 words an introduction to Camel:

- What exactly is it?
- How does it interact with an application written in Java?
- Is it something that goes together with the server?
- Is it an independent program?

Please explain what Camel is.

java apache-camel soa enterprise-integration

Share

Improve this question

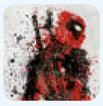
Follow

edited Jan 28, 2016 at 15:25

Ravindra babu

38.9k ● 11 ● 256 ● 219

asked Jan 13, 2012 at 2:32



Myy

18.5k ● 11 ● 40 ● 57

21 Answers

Sorted by: Highest score (default)

1406

My take to describe this in a more accessible way...

In order to understand what Apache Camel is, you need to understand what are Enterprise Integration Patterns.

Let's start with what we presumably already know: The Singleton pattern, the Factory pattern, etc; They are merely ways of organizing your solution to the problem, but they are not solutions themselves. These patterns were analyzed and extracted for the rest of us by the Gang of Four, when they published their book: Design Patterns. They saved some of us tremendous effort in thinking of how to best structure our code.

Much like the Gang of Four, Gregor Hohpe and Bobby Woolf authored the book *Enterprise Integration Patterns* (EIP) in which they propose and document a set of new patterns and *blueprints* for how we could *best* design large component-based systems, where components can be running on the same process or in a different machine.

They basically propose that we structure our system to be *message* oriented -- where components communicate with each others using messages as inputs and outputs and absolutely nothing else. They show us a complete set of patterns that we may choose from and implement in our different components that will together form the whole system.

So what is Apache Camel?

Apache Camel offers you the interfaces for the EIPs, the base objects, commonly needed implementations, debugging tools, a configuration system, and many other helpers which will save you a ton of time when you want to implement your solution to follow the EIPs.

Take MVC. MVC is pretty simple in theory and we could implement it without any framework help. But good MVC frameworks provide us with the structure ready-to-use and have gone the extra mile and thought out all the other "side" things you need when you create a large MVC project and that's why we use them most of the time.

That's exactly what Apache Camel is for EIPs. **It's a complete production-ready framework for people who want to implement their solution to follow the EIPs.**

Share Improve this answer Follow

edited Jun 1, 2023 at 8:01
Stephan Henningsen
3,783 ● 2 ● 23 ● 29

answered Jul 18, 2012 at 11:38
Amr Mostafa
23.9k ● 2 ● 31 ● 24

33 EIP is the key. If you don't understand EIP, you might use Camel like blind men and elephant(camel). EIP - eaipatterns.com – hutingung Jul 16, 2014 at 11:10

4 A little bit EIP problems: "There have been many libraries and frameworks over the years to help with inte- gration. But frequently the concepts behind the Enterprise Integration Patterns get transformed into some complex class hierarchies or objects that need to be wired together just so, and the original intentions and patterns are often lost. The developer is forced from then on to focus on the low-level detail and some complex class library API, losing the bigger picture and patterns." – Quan Nguyen Aug 16, 2016 at 6:35

796

If you have 5 to 10 minutes, I generally recommend people to read this Integration with Apache Camel by Jonathan Anstey. It's a well written piece which gives a brief introduction to and overview of some of Camel's concepts, and it implements a use case with code samples. In it, Jonathan writes:

> Apache Camel is an open source Java framework that focuses on making integration easier and more accessible to developers. It does this by providing:
>
> - concrete implementations of all the widely used Enterprise Integration Patterns (EIPs)
> - connectivity to a great variety of transports and APIs
> - easy to use Domain Specific Languages (DSLs) to wire EIPs and transports together

There is also a free chapter of Camel in Action (Camel in Action, 2nd ed. is here) which introduces Camel in the first chapter. Jonathan is a co-author on that book with me.

Share Improve this answer Follow

edited Oct 5, 2020 at 10:51
Kawu
14k ● 37 ● 128 ● 202

answered Jan 13, 2012 at 10:49
Claus Ibsen
57.1k ● 7 ● 52 ● 66

61 The Camel in Action book is a very very good book to get to learn the basics and also how to use some of the more complicated features of Camel. I highly recommend it! (I am in no way affiliated with the book or publisher) – Matt Aldridge Jan 18, 2012 at 15:26

2 @Clause if want to choose between mule ESB & Camel. what should be my demacation on choosing one ove the other – kuhajeyan Jan 7, 2013 at 9:11

8 See some of the links at the *comparison to Camels competitors* at: camel.apache.org/articles.html. – Claus Ibsen Jan 7, 2013 at 13:20

1 Yes absolutely it can be used for connecting microservices, its after all just a little Java toolkit/framework. The Camel in Action 2nd edition book has a full chapter on Camel microservices. – Claus Ibsen Dec 27, 2017 at 8:13

4 The images in the "Integration with Apache Camel" article no longer work. Here is a link to an archived version where the images work: web.archive.org/web/20171029104103/https://dzone.com/articles/… – Peter V. Mørch May 27, 2021 at 16:20

680

Creating a project description should not be complicated.

I say:

> Apache Camel is messaging technology glue with routing. It joins together messaging start and end points allowing the transference of messages from different sources to different destinations. For example: JMS -> JSON, HTTP

> -> JMS or funneling FTP -> JMS, HTTP -> JMS, JSON -> JMS

Wikipedia says:

> Apache Camel is a rule-based routing and mediation engine which provides a Java object based implementation of the Enterprise Integration Patterns using an API (or declarative Java Domain Specific Language) to configure routing and mediation rules. The domain specific language means that Apache Camel can support type-safe smart completion of routing rules in your IDE using regular Java code without huge amounts of XML configuration files; though XML configuration inside Spring is also supported.

See? That wasn't hard was it?

Share Improve this answer Follow

edited Jan 6, 2015 at 22:56

Dave Jarvis
31.2k ● 43 ● 181 ● 323

answered May 31, 2012 at 15:37

David Newcomb
10.9k ● 3 ● 51 ● 63

360 Apache Camel homepage refers to this thread... They didn't manage to provide a short functional explanation of their own product. – youri Dec 13, 2013 at 8:03

17 This article is a prime example how constructive criticism, and honest effort can create sublime documentation. It is a featured on the official Camel website. But let's keep it constructive, and avoid name tagging. Documentation writers and other contributors are sometimes hard to come by with, and they deserve our respect. BTW - we have many java coding standards ... and stick with it with pride and honor ... how about a documenting standard for media like Wiki's and official Guides? – YoYo Mar 23, 2014 at 20:07

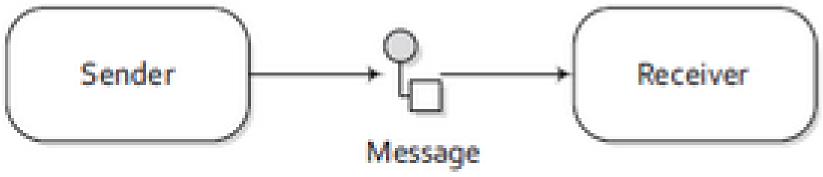
1 Is it like a fine grain reverse proxy? – Asad Hasan May 13, 2014 at 1:55

16 Now, this is what I call crisp and to the point answer. Strangely, the accepted answer looks like an advertisement. +1 – EMM Jan 3, 2015 at 9:39

8 It's 2017 now and in 3 years time all they did was add an acknowledgement that their own description is buzzword soup and add a link to this thread. They didn't even take the time to add a summary of this thread (or even just copy-paste some stuff) to the page itself. All we get is a link to SO. Come on, at some point you gotta stop defending such attitude. – Stijn de Witt Sep 28, 2017 at 7:32

202

Camel sends messages from A to B:



Why a whole framework for this? Well, what if you have:

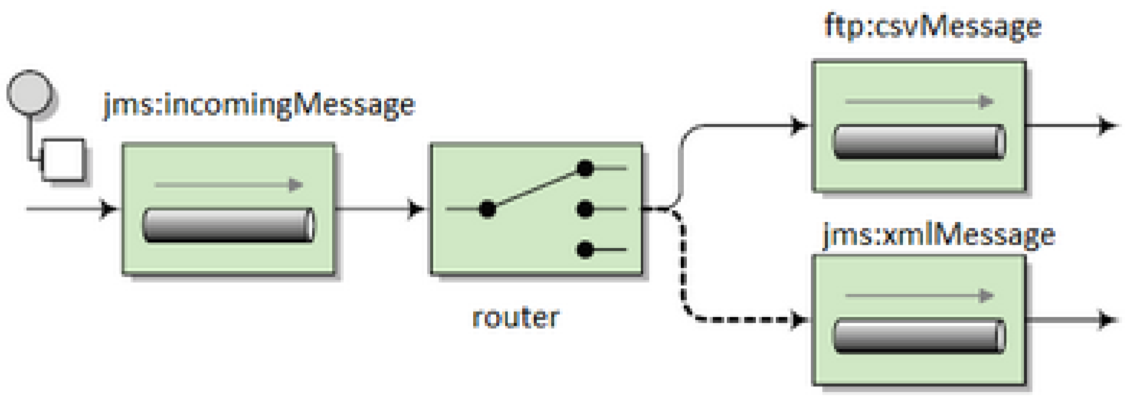
- many senders and many receivers

- a dozen of protocols (`ftp`, `http`, `jms`, etc.)
- many complex rules
    - Send a message A to Receivers A and B only
    - Send a message B to Receiver C **as XML**, but partly **translate** it, **enrich** it (add metadata) and **IF condition X**, then send it to Receiver D too, but **as CSV**.

So now you need:

- **translate** between protocols
- **glue** components together
- define routes - **what goes where**
- **filter** some things in some cases

Camel gives you the above (and more) out of the box:



with a cool DSL language for you to define the what and how:

```
new DefaultCamelContext().addRoutes(new RouteBuilder
        public void configure() {
            from("jms:incomingMessages")
```

```
                        .choice() // start router rules
                        .when(header("CamelFileName")
                                .endsWith(".xml"))
                        .to("jms:xmlMessages")
                        .when(header("CamelFileName")
                                .endsWith(".csv"))
                        .to("ftp:csvMessages");
}
```

See also this and this and Camel in Action (as others have said, an excellent book!)

Share Improve this answer Follow

answered Apr 28, 2017 at 17:53

Andrejs
11.8k ● 6 ● 47 ● 51

36 As a noob, this is the only answer on this page that made any kind of sense to me in understanding what Camel does. Thanks. – anotherDev Apr 1, 2022 at 13:06

3 THIS answer should be pinned to the top – JBM Jul 4 at 11:39

1 All the other answers talk about the concept. This is the only answer which shows what it does, and with an example of how it does it. Thank you – Pankaj Nov 14 at 8:11

122

In short:

When there is a requirement to connect / integrate systems, you will probably need to connect to some data source and then process this data to match your business requirements.

In order to do that:

1) You could develop custom program that would do it (might be time consuming and hard to understand, maintain for other developer)

2) Alternatively, you could use Apache Camel to do it in standardised way (it has most of the connectors already developed for you, you just need to set it up and plug your logic - called Process):

Camel will help you to:

1. Consume data from any source/format
2. Process this data
3. Output data to any source/format

By using Apache Camel you will make it easy to understand / maintain / extend your system to another developer.

Apache Camel is developed with Enterprise Integration Patterns. The patterns help you to integrate systems in a good way :-)

Share Improve this answer Follow

edited May 16, 2014 at 13:54

Rory Hunter
3,450 ● 1 ● 15 ● 16

answered Jun 4, 2013 at 20:48

Marcin Wasiluk
4,854 ● 4 ● 40 ● 45

That means other developers can change the logic in another programming language too? – JavaTechnical Jan 19, 2015 at 16:22

2 @JavaTechnical given the messaging pattern (EIP), you already can code different components in different languages because these messages are in language independent formats like JSON. Camel provides an easy framework to implement EIP. This is based on my understanding. Please correct me if I'm mistaken. – Dheeraj Bhaskar Apr 27, 2015 at 11:25

63

BASED ON ANALOGY

Camel based routing can be understood much easily by putting yourself in the shoes of a Airline owner (eg.: American Airlines, Jet Airways) .

Purpose of 'your airline' is to 'carry' 'passengers' from one 'city' to another one in the world. You use aircrafts from different 'aircraft companies' like Boeing, Airbus, HAL for carrying passengers.

Your airline's boards passengers using 'airports' of the from city and deboards them using the airport of the to city. A passenger may 'travel' to multiple cities, but everywhere they have to go through the airport to travel between your airline's aircraft and the city.

Note that a passenger 'departing' from the city is essentially 'arriving' into your airlines' aircraft. And a passeger 'arriving' into the city, is essentially departing from the aircraft. Since we are in the shoes of airline owner, the term 'arrival passenger' and 'departing passenger' are reversed from our conventional notions which are based on cities perspective.

Same 'airport' infrastructure of each city is used by 'departing' passengers and 'arrival' passengers. An airport provides 'departing infrastructure' for departing passengers, which is different from the 'arrival infrastructure' provided for arriving passengers.

Passengers can continue doing their day to their activities due to various 'amenities' provided inside the aircraft by your airlines, while travelling.

On top of that, your airline also provides lounge facilities for special treatments like 'understanding local language' and or preparing you for the 'travel'.

Lets replace few words/phrases used above with following:

your airline: Apache Camel

aircraft companies: Transport mechanisms

your airline's aircraft: Apache Camel's underlying transport mechanism

carry: route

passengers: message;

city: system;

airport: Camel Component;

understanding local languages: Type Conversions;

departing: producing, produced

arriving: consuming, consumed

travel: routed

amenities: provided

After replacing the words, here is what you get:

Purpose of 'Apache Camel' is to route 'messages' from one 'system' to another one in the world. Apache camel uses different transport mechanisms for message routing.

Apache Camel picks up messages using 'Camel based Component' of the 'from' system and drops them using the 'Camel based Component' of the 'to' system. A message may route to multiple systems, but everywhere they have to go through 'Camel based Components' to travel between 'Apache Camel's underlying transport mechanism' and the system.

Note that a message 'produced' from the system is essentially 'consumed' into Apache Camel's underlying transport mechanism'. And a message consumed by a system, is essentially produced by the 'Apache Camel's underlying transport mechanism'.

Since we are attempting to understand Camel, we must think from Camel's perspective here onwards. The meaning of the terms 'consumer message' and 'producer message' are ,thus, reversed from our conventional notions which are based on a system's perspective.

Same 'Camel based Component's ' coding infrastructure is used by 'producer message' and 'consumer message'. A 'Camel based Component' provides a 'producer endpoint' for 'producer message' and a 'consumer endpoint' for 'consumer message'.

Messages can be processed by Camel when they are being routed.

On top of this routing, Camel provides special features like 'Type Conversion' and many more...

Share Improve this answer Follow

answered Apr 13, 2015 at 12:54

DolphinJava

2,752 ● 1 ● 27 ● 38

Great explanation nice to read and easy to remember. I wonder what role has a runway or pilot, if they even exist in camel. – Stimpson Cat Oct 17, 2018 at 11:51

Nice explanation specially Airport example keep it up. if you added few more extended terms and little bit piece of code example would be really great !! Thanks – Ankur Nirmalkar Jun 27, 2019 at 8:23

55

One of the things you need to understand, before you try to understand Apache Camel, are Enterprise Integration Patterns. Not everyone in the field is actually aware of them. While you can certainly read the Enterprise Integration Patterns book, a quicker way to get up to speed on them would be to read something like the Wikipedia article on Enterprise Application Integration.

One you have read and understood the subject area, you would be much more likely to understand the purpose of Apache Camel

HTH

Share Improve this answer Follow

answered Jan 13, 2012 at 3:36

Crollster
2,771 ● 2 ● 24 ● 34

39

If you are aware of Enterprise Integration Patterns, Apache Camel is one integration framework which implements all EIPs.

And you can deploy Camel as a standalone application in a web-container.

Basically, if you have to integrate several applications with different protocols and technologies, you can use Camel.

Share Improve this answer

answered Jan 13, 2012 at 2:59

Follow

Sunil
1,248 ● 3 ● 13 ● 21

1 "implements all EIPs" is not correct: *„Because Camel implements many of the design patterns in the EIP book“*. – Gerold Broser Dec 11, 2020 at 22:18

23

A definition from another perspective:

Apache Camel is an integration framework. It consists of some Java libraries, which helps you implementing integration problems on the Java platform. What this means and how it differs from APIs on the one side and an Enterprise Service Bus (ESB) on the other side is described in my article "When to use Apache Camel".

Share Improve this answer

Follow

answered Mar 16, 2013 at 17:08

Kai Wähner
5,430 ● 4 ● 37 ● 33

22

> What exactly is it?

Apache Camel is a lightweight integration framework which implements many Enterprise Integration patterns. You can easily integrate different applications.

You can integrated with many of the technologies like HTTP, FTP, JMS, EJB, JPA, RMI, JMS, JMX, Netty etc.

> How does it interact with an application written in Java?

Camel uses a *Java Domain Specific Language or DSL* for creating Enterprise Integration Patterns or Routes in a variety of domain-specific languages (DSL)

Enterprise Integration Pattern resolves around these concepts :

Message, End Point, Producer, Consumer, Routing, Bus, Transform and Process.

> Is it something that goes together with the server?

It acts as a bridge across multiple enterprise sub systems.

> Is it an independent program?

Apache Camel, an integration framework, integrates different independent applications.

The major advantage of Camel : You can integrate different applications with different technologies (and different protocols) by using same same concepts for every integration.

Share Improve this answer Follow

edited Apr 2, 2023 at 14:42

answered Dec 8, 2015 at 10:58

Ravindra babu
38.9k ● 11 ● 256 ● 219

I wonder why do you use the word "lightweight". My observation is that Apache Camel is actually heavy. – Krzysztof Tomaszewski Jan 8, 2020 at 14:35

"implements all Enterprise Integration patterns" is not correct: *„Because Camel implements many of the design patterns in the EIP book“*. – Gerold Broser Dec 11, 2020 at 22:19

Can I create an http endpoint in Apache Camel for external users to be able to send a request to this endpoint? I am reading the HTTP component and it says producer - what does this mean? – variable Jul 15, 2022 at 7:03

16

Most "new" things in computing aren't really new at all, they're just a mystifying wrapper around something that already well-understood. When they're hard to understand, it's usually because someone decided to invent new language terms or colonise existing terms for a different purpose (a good example of *that* is the X developers' reversal of what "client" and "server" mean.)

Camel is a Java-based wrapper/API for inter-application middleware.

Middleware is a general term for software that provides interpretation services between entities that don't share a common language or data types.

That's what Camel is, at bottom. We can flesh out the description by noting that it provides for EIP-type middleware.

It doesn't provide the middleware itself, since it can't know the details of what the applications need to communicate. But it provides the API for creating the invariant parts of that middleware (create a start point, create an end point, create conditions for starting and ending, etc)

Hope that helps.

Share Improve this answer Follow

answered Sep 8, 2015 at 13:10

MMacD
349 ● 2 ● 10

4 "Most "new" things in computing aren't really new at all, they're just a mystifying wrapper around something already well-understood." <<<< Best thing I've read all week! – Dave Feb 14, 2017 at 8:53

10

Here is another attempt at it.

You know how there are/were things like Webmethods, ICAN Seebeyond, Tibco BW, IBM Broker. They all did help with integration solutions in the enterprise. These tools are commonly known by the name Enterprise Application Integration (EAI) tools.

There were mostly drag drop tools built around these technologies and in parts you would have to write adapters in Java. These adapter code were either untested or had poor tooling/automation around testing.

Just like with design patterns in programming, you have Enterprise Integration patterns for common integration solutions. They were made famous by a book of the same name by Gregor Hohpe and Bobby Woolf.

Although it is quite possible to implement integration solutions which use one or many EIP, Camel is an attempt at doing this within your code base using one of XML, Java, Groovy or Scala.

Camel supports all Enterprise Integration Patterns listed in the book via its rich DSL and routing mechanism.

So Camel is a competing technoloy to other EAI tools with better support for testing your integration code. The code is concise because of the Domain Specific Languages (DSLs). It is readable by even business users and it is free and makes you productive.

Share Improve this answer Follow

edited Nov 27, 2013 at 20:58
Tarun Varshney
83 ● 1 ● 5

answered Nov 11, 2013 at 11:51
skipy
4,207 ● 2 ● 24 ● 21

8

There are lot of frameworks that facilitates us for messaging and solving problems in messaging. One such product is Apache Camel.

Most of the common problems have proven solutions called as design patterns. The design pattern for messaging is Enterprise Integration patterns(EIPs) which are well explained here. Apache camel help us to implement our solution using the EIPs.

The strength of an integration framework is its ability to facilitate us through EIPs or other patterns,number of transports and components and ease of development on which Apache camel stands on the top of the list

Each of the Frameworks has its own advantages Some of the special features of Apache camel are the following.

1. It provides the coding to be in many DSLs namely Java DSL and Spring xml based DSL , which are popular.
2. Easy use and simple to use.
3. Fuse IDE is a product that helps you to code through UI

Share Improve this answer Follow

edited Aug 11, 2014 at 2:50

answered May 29, 2014 at 8:52

Naveen Raj
801 ● 11 ● 26

8

In plain English, camel gets (many) things done without much of boiler plate code.

Just to give you a perspective, the Java DSL given below will create a REST endpoint which will be able to accept an XML consisting of List of Products and splits it into multiple products and invoke Process method of BrandProcessor with it. And just by adding .parallelProcessing (note the commented out part) it will parallel process all the Product Objects. (Product class is JAXB/XJC generated Java stub from the XSD which the input xml is confined to.) This much code (along with few Camel dependencies) will get the job done which used to take 100s of lines of Java code.

```
from("servlet:item-delta?matchOnUriPrefix=true&httpMet
.split(stax(Product.class))
/*.parallelProcessing()*/
.process(itemDeltaProcessor);
```

After adding the route ID and logging statement

```
from("servlet:item-delta?matchOnUriPrefix=true&httpMet
.routeId("Item-DeltaRESTRoute")
.log(LoggingLevel.INFO, "Item Delta received on Item-D
.split(stax(Product.class))
.parallelProcessing()
.process(itemDeltaProcessor);
```

This is just a sample, Camel is much more than just REST end point. Just take a look the pluggable component list http://camel.apache.org/components.html

Share Improve this answer Follow

edited Sep 20, 2017 at 10:54

David Goodwin
4,298 ● 1 ● 23 ● 12

answered May 13, 2016 at 21:41

KingJulien
303 ● 3 ● 9

8

Assume you create an ecommerce company like Amazon, and you want to only focus on strategy/choice of products to sell. unlike amazon delivery fleet, instead of yourself handling moving of goods from sellers to warehouse, making changes to it in warehouse like packaging and sending it out to other city and customers. You hire a company that does all this and just give them info of all your warehouse locations, vehicle types, delivery locations and a list of when to do what. Then they handle that by themselves, that would be Apache Camel. They take care of moving things from one end to other, once you handover stuff to them, so that you are free to focus on other things.

Share Improve this answer Follow

answered Nov 30, 2018 at 11:05

Chetan Gowda
410 ● 7 ● 15

8

101 Word Intro

Camel is a framework with a consistent API and programming model for integrating applications together. The API is based on theories in Enterprise Integration Patterns - i.e., bunch of design patterns that tend to use messaging. It provides out of the box implementations of most of these patterns, and additionally ships with over 200 different components you can use to easily talk to all kinds of other systems. To use Camel, first write your business logic in POJOs and implement simple interfaces centered around messages. Then use Camel’s DSL to create "Routes" which are sets of rules for gluing your application together.

Extended Intro

On the surface, Camel's functionality rivals traditional Enterprise Service Bus products. We typically think of a Camel Route being a "mediation" (aka orchestration) component that lives on the server side, but because it's a Java library it’s easy to embed and it can live on a client side app just as well and help you integrate it with point to point services (aka choreography). You can even take your POJOs that process the messages inside the Camel route and easily spin them off into their own remote consumer processes, e.g. if you needed to scale just one piece independently. You can use Camel to connect routes or processors through any number of different remote transport/protocols depending on your needs. Do you need an extremely efficient and fast binary protocol, or one that is more human readable and easy to debug? What if you wanted to switch? With Camel this is usually as easy as changing a line or two in your route and not changing any business logic at all. Or you could support both - you’re free to run many Routes at once in a Camel Context.

You don't really need to use Camel for simple applications that are going to live in a single process or JVM - it would be overkill. But it's not conceptually any more difficult than code you may write yourself. And if your requirements change, the separation of business logic and glue code makes it easier to maintain over time. Once you learn the Camel API, it is easy to use it like a Swiss-Army knife and apply it quickly in many different contexts to cut down on the amount of custom code you’d otherwise have to write. You can learn one flavor - the Java DSL, for example, a fluent API that's easy to chain together - and pick up the other flavors easily.

Overall Camel is a great fit if you are trying to do microservices. I have found it invaluable for evolutionary architecture, because you can put off a lot of the difficult, "easy-to-get-wrong" decisions about protocols, transports and other system integration problems until you know more about your problem domain. Just focus on your EIPs and core business logic and switch to new Routes with the "right" components as you learn more.

Share Improve this answer

Follow

edited Dec 12, 2018 at 23:56

answered Dec 12, 2018 at 23:51

cwash

4,245 ● 5 ● 46 ● 53

7

Camel helps in routing, transformation, monitoring.

It uses Routes; which can be described as :

When service bus receives particular message, it will route it through no of services/broker destinations such as queue/topics. This path is known as route.

Example: *your stock application has got some input by analyst, it will be processed through the application/web component and then result will be published to all the interested/registered members for particular stock update.*

Share Improve this answer

Follow

answered Aug 20, 2014 at 14:34

sa_nyc

981 ● 1 ● 13 ● 24

7

Apache Camel is a Java framework for Enterprise integration. Eg:- if you are building a web application which interacts with many vendor API's we can use the camel as the External integration tool. We can do more with it based on the use case. Camel in Action from Manning publications is a great book for learning Camel. The integrations can be defined as below.

Java DSL

```
from("jetty://0.0.0.0:8080/searchProduct").routeId("se
    .log(LoggingLevel.INFO, "searchProducts request Re
${body}")
    .bean(Processor.class,
"createSearchProductsRequest").removeHeaders("CamelHtt
    .setHeader(Exchange.HTTP_METHOD,
constant(org.apache.camel.component.http4.HttpMethods.
    .to("http4://" + preLiveBaseAPI + searchProductsUr
                    + "&bridgeEndpoint=true")
    .bean(Processor.class, "buildResponse").log(Loggin
products finished");
```

This is to just create a REST API endpoint which in turn calls an external API and sends the request back

Spring DSL

```
<route id="GROUPS-SHOW">
    <from uri="jetty://0.0.0.0:8080/showGroups" />
    <log loggingLevel="INFO" message="Reqeust recevice
-> ${body}" />
    <to uri="direct:auditLog" />
    <process ref="TestProcessor" />
</route>
```

Coming to your questions

1. What exactly is it? Ans:- It is a framework which implements Enterprise integration patterns
2. How does it interact with an application written in Java? Ans:- it can interact with any available protocols like http, ftp, amqp etc
3. Is it something that goes together with the server? Ans:- It can be deployed in a container like tomcat or can be deployed independently as a java process
4. Is it an independent program? Ans:- It can be.

Hope it helps

Share Improve this answer Follow

answered Mar 1, 2018 at 15:48

Khader M A
6,211 ● 3 ● 20 ● 19

5

Yes, this is probably a bit late. But one thing to add to everyone else's comments is that, Camel is actually a toolbox rather than a complete set of features. You should bear this in mind when developing and need to do various transformations and protocol conversions.

Camel itself relies on other frameworks and therefore sometimes you need to understand those as well in order to understand which is best suited for your needs. There are for example multiple ways to handle REST. This can get a bit confusing at first, but once you starting using and testing you will feel at ease and your knowledge of the different concepts will increase.

Share Improve this answer Follow

answered Sep 22, 2015 at 5:43

Souciance Eqdam Rashti
3,193 ● 3 ● 17 ● 35

4

Its like a pipeline connecting

```
From---->To
```

In between u can add as many channels and pipes. The faucet can be of any type automatic or manual for flow of data and a route to channelize the flow.

It supports and have implementation for all types and kinds of processing. And for same processing many approaches because it has many components and each component can also provide the desired output using different methods under it.

For instance, File transfer can be done in camel with types file moved or copied and also from folder, server or queue.

```
-from-->To
   - from-->process-->to
   - from-->bean-->to
   - from-->process-->bean-->to
   -from-->marshal-->process-->unmarshal-->to
```

From/to----folder, direct, seda, vm can be anything

Share Improve this answer Follow

answered Sep 7, 2017 at 8:23

javalearner_heaven
503 ● 2 ● 6 ● 21

3

Another point of view (based on more fundamental mathematical topics)

The most general computing platform is a Turing Machine.

There is a problem with the Turing machine. All the input/output data stays inside the turing machine. In the real world there are input sources and output sinks external to our Turing machine, and in general governed by systems outside of our control. That is, those external system will send/receive data at will in any format with any desired data-scheduler.

Question: How do we manage to make independent turing-machines speak to each other in the most-general way so that each turing-machine sees their peers as either a source of input-data or sink of output-data?

Answer: Using something like camel, mule, BizTalk or any other ESB that abstract away the data handling between completing distinct "physical" (or virtual software) turing machines.

Share Improve this answer

Follow

edited Aug 3, 2020 at 16:54

izilotti
4,927 ● 1 ● 50 ● 56

answered Dec 20, 2018 at 8:50

earizon
2,208 ● 20 ● 32