

# 1 - Before Beginning

It's good to have some basic understanding of some concepts before we get our hands on doing it.

## 1.1 Basic idea of Microservice

Before the appearance of Spring Cloud, we used to build a web application in a single application, that is to put all the stuff we need into one project. Usually this type of single application will follow a MVC architecture.

Spring Cloud aims to split a big single system into multiple small pieces. Each piece run in their own independent processes. Different pieces communicate and cooperate through the HTTP-based RESTful API. In this way we can decoupling a big single Enterprise Application. And such designing have many merits obviously.

Now the “small piece” mentioned before is given a name of "micro service". Actually, micro service is just a conventional java web app, nothing else. But the responsibility each service contains should be as few as possible. Ideally, each service should contain only one type of function. And expose the function(we called services) it provides to outside by providing a http address. When we have this concept of micro service, we can analyze the old big single application, find out how many functions it provides in total, then we classify all the functions, put the same type of function in to one place. Then build a bunch of java web application, each of them should contain the same type of function which has been classified by us before. And then we should expose the function(hereinafter referred to as service) to outside by binding the service to an Http address. In this way, we already make a web application into a micro service. By building a bunch of micro services, we can finally build up a system of micro services, or we can say a cloud of micro services, contains all the functions we need. This will be a better replace of the old big single application.

If we are using building tool such as Maven or Gradle. Then, instead of creating microservice by building a single java web app, we can create microservice by creating a module. Module is nothing but a java web app. However it is put under a super project. In conventional big single java application, there is only one entry method. By running this entry method, we start a whole application. But in a project contains modules, each module has its own entry method. Although all the modules should be put under one big project, yet each module can be turned on or off independently without any impact on other modules. Each module can be built up using Spring Boot, just like the way we built Spring Boot Application before.

## 1.2 What is Spring Cloud

And what is Spring Cloud? Spring Cloud is a rapid development framework that provide a lot of useful out-of-

the-box components, such as Eureka, Ribbon, etc. It can help construct microservices in a better way with a lot of ease.