

# Burnettzhong's Blog

## 为基于Spring Boot的微服务架构搭建一套自动化、集中管理的API文档中心

📅 2017-04-08

### 简介

本篇文章将阐述如何通过使用我开发Parliament，与Swagger、Keyhole Software提供的工具，搭建一套自动发布、集中管理的API文档中心。

### 背景介绍

Spring Boot与Spring Cloud等项目为我们搭建微服务架构提供了很大的便利。但是微服务架构的劣势之一就是增加了治理的复杂度。

众所周知，微服务架构中的各个应用是独立开发、部署的。当微服务数量到达一定的数量后，集中管理API文档的难度越来越大。

为了减轻文档压力，很多Spring Boot项目集利用Swagger自动生成API描述页面。然而Swagger并没有解决API集中管控的问题，各个团队必须清楚的知道其他团队的Swagger页面地址，这势必完成了维护成本。这也是Parliament项目初衷。

### 解决方案

我开发了一个简单的API文档中心服务器Parliament。Parliament集成了Swagger UI，用于显示和调试API。API信息发布采用的是Keyhole Software开发的khs-spring-boot-publish-swagger-starter。需要注意的是原始的khs-spring-boot-publish-swagger-starter存在几个bug，请暂时使用我提供的版本。

背后的逻辑是：

1. 由Swagger2负责生成API描述，
2. 当Spring Boot应用每次启动时，通过khs-spring-boot-publish-swagger-starter将API描述发送到Parliament服务器。
3. Parliament服务器存储并整理API描述，并提供UI展示给各个团队开发人员。

这样我们就拥有了一个集中的、自动化的API文档中心。

## 开始搭建

在部署Parliament之前，我们需要先安装MongoDB。关于如何安装、配置MongoDB不在此文章范围。安装好MongoDB后，进行如下步骤。

### I. 部署Parliament

```
$git clone https://github.com/burnettzhong/parliament.git
$cd parliament
```

修改`application.properties`，配置数据源, 请根据实际的MongoDB信息修改：

```
spring.data.mongodb.host = {MongoDB address}
spring.data.mongodb.database = {MongoDB database name}
spring.data.mongodb.port = {MongoDB port}
```

构建并启动Parliament:

```
$mvn package && java -jar target/Parliament-1.0.0.jar
```

现在可以打开<http://localhost:8080/index.html>，当然在基本是个空白页面，因为我们还没有发布任何API信息到Parliament。

### II. 让Spring Boot应用发布API信息

这里我们需要加入两个依赖库，一个是SpringFox的Swagger2，另一个是khs-spring-boot-publish-swagger-starter。

注意：请不要使用来自Keyhole Software的khs-spring-boot-publish-swagger-starter，其中有两个bug。只有他们合并我们pull request，才可以直接使用。

1. 在Keyhole Software在Maven公共库中更新之前，我们需要手动安装我的版本(1.0.2)到本地Maven库中：

```
git clone https://github.com/burnettzhong/khs-spring-boot-publish-swagger-starter
cd khs-spring-boot-publish-swagger-starter
mvn clean install
```

## 2. 在Spring Boot应用中增加Swagger2和khs-spring-boot-publish-swagger-starter依赖

```
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger2</artifactId>
  <version>2.6.1</version>
</dependency>
<dependency>
  <groupId>com.keyholesoftware</groupId>
  <artifactId>khs-spring-boot-publish-swagger-starter</artifactId>
  <version>1.0.2</version>
</dependency>
```

## 3. 在application.yml或application.properties中配置Parliament地址：

```
swagger:
  publish:
    publish-url: http://{Parliament server address}/swagger/publish/
    swagger-url: http://127.0.0.1:${server.port}/v2/api-docs
```

这里有两个地址，publish-url是我们的Parliament部署地址；swagger-url为获取API描述地址，一般来讲都是本应用地址。

## 1. 最后通过注解配置Spring Boot应用：

```
@EnableSwagger2
@PublishSwagger
public class ServiceApplication {
    ...
}
```

@EnableSwagger2启用Swagger2生成API描述，@PublishSwagger让应用在启动的时候将API描述JSON发布到Parliament。

关于API描述的配置，Swagger2提供了很丰富的功能，请参考[Springfox Reference Documentation](#)。

至此，全部安装部署结束。在启动配置好的Spring Boot应用后，再打开<http://localhost:8080/index.html>就可以看到我们的API信息了。

Parliament根据API的base path进行分类，点击某个base path可以查看更详细的API信息(这里采用的是Swagger UI)。

## 后记

1. 如果需要，可以将Parliament作为一个微服务注册到Eureka或者Zookeeper中。
2. 因为Parliament主要是在内网使用，目前没有进行安全验证。未来会引入对应用的认证，和用户管理。
3. 未来Parliament还将增加API使用统计的功能，完善整体API治理。

欢迎关注并贡献代码到<https://github.com/burnettzhong/parliament>

# API, Mircoservice

© 2017 ♥ Han Zhong