[**第二十三篇： 异步方法**](http://blog.csdn.net/forezp/article/details/71024169)

参考：<http://blog.csdn.net/forezp/article/details/71024169>

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/71024169>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp)

这篇文章主要介绍在springboot 使用异步方法，去请求github api.

创建工程

在pom文件引入相关依赖：

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

</dependency>

创建一个接收数据的实体：

@JsonIgnoreProperties(ignoreUnknown=true)

public class User {

private String name;

private String blog;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getBlog() {

return blog;

}

public void setBlog(String blog) {

this.blog = blog;

}

@Override

public String toString() {

return "User [name=" + name + ", blog=" + blog + "]";

}

}

创建一个请求的　githib的service:

@Service

public class GitHubLookupService {

private static final Logger logger = LoggerFactory.getLogger(GitHubLookupService.class);

private final RestTemplate restTemplate;

public GitHubLookupService(RestTemplateBuilder restTemplateBuilder) {

this.restTemplate = restTemplateBuilder.build();

}

@Async

public Future<User> findUser(String user) throws InterruptedException {

logger.info("Looking up " + user);

String url = String.format("https://api.github.com/users/%s", user);

User results = restTemplate.getForObject(url, User.class);

// Artificial delay of 1s for demonstration purposes

Thread.sleep(1000L);

return new AsyncResult<>(results);

}

}

通过，RestTemplate去请求，另外加上类@Async 表明是一个异步任务。

开启异步任务：

@SpringBootApplication

@EnableAsync

public class Application extends AsyncConfigurerSupport {

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

@Override

public Executor getAsyncExecutor() {

ThreadPoolTaskExecutor executor = new ThreadPoolTaskExecutor();

executor.setCorePoolSize(2);

executor.setMaxPoolSize(2);

executor.setQueueCapacity(500);

executor.setThreadNamePrefix("GithubLookup-");

executor.initialize();

return executor;

}

}

通过@EnableAsync开启异步任务；并且配置AsyncConfigurerSupport，比如最大的线程池为2.

测试

测试代码如下：

@Component

public class AppRunner implements CommandLineRunner {

private static final Logger logger = LoggerFactory.getLogger(AppRunner.class);

private final GitHubLookupService gitHubLookupService;

public AppRunner(GitHubLookupService gitHubLookupService) {

this.gitHubLookupService = gitHubLookupService;

}

@Override

public void run(String... args) throws Exception {

// Start the clock

long start = System.currentTimeMillis();

// Kick of multiple, asynchronous lookups

Future<User> page1 = gitHubLookupService.findUser("PivotalSoftware");

Future<User> page2 = gitHubLookupService.findUser("CloudFoundry");

Future<User> page3 = gitHubLookupService.findUser("Spring-Projects");

// Wait until they are all done

while (!(page1.isDone() && page2.isDone() && page3.isDone())) {

Thread.sleep(10); //10-millisecond pause between each check

}

// Print results, including elapsed time

logger.info("Elapsed time: " + (System.currentTimeMillis() - start));

logger.info("--> " + page1.get());

logger.info("--> " + page2.get());

logger.info("--> " + page3.get());

}

}

启动程序，控制台会打印：

2017-04-30 13:11:10.351 INFO 1511 — [ GithubLookup-1] com.forezp.service.GitHubLookupService : Looking up PivotalSoftware   
2017-04-30 13:11:10.351 INFO 1511 — [ GithubLookup-2] com.forezp.service.GitHubLookupService : Looking up CloudFoundry   
2017-04-30 13:11:13.144 INFO 1511 — [ GithubLookup-2] com.forezp.service.GitHubLookupService : Looking up Spring-Projects

耗时：3908

分析：可以卡的前面2个方法分别在GithubLookup-1 和GithubLookup-2执行，第三个在GithubLookup-2执行，注意因为在配置线程池的时候最大线程为2.如果你把线程池的个数为3的时候，耗时减少。

如果去掉@Async，你会发现，执行这三个方法都在main线程中执行。耗时总结，如下：

2017-04-30 13:13:00.934 INFO 1527 — [ main] com.forezp.service.GitHubLookupService : Looking up PivotalSoftware   
2017-04-30 13:13:03.571 INFO 1527 — [ main] com.forezp.service.GitHubLookupService : Looking up CloudFoundry   
2017-04-30 13:13:04.865 INFO 1527 — [ main] com.forezp.service.GitHubLookupService : Looking up Spring-Projects

耗时：5261

通过这一个小的栗子，你应该对异步任务有了一定的了解。

参考资料

<https://spring.io/guides/gs/async-method/>

源码下载

<https://github.com/forezp/SpringBootLearning>

优秀文章推荐：

* 更多springboot 教程：[springBoot非官方教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70341818)
* 更多springcoud 教程：[史上最简单的 SpringCloud 教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70148833)