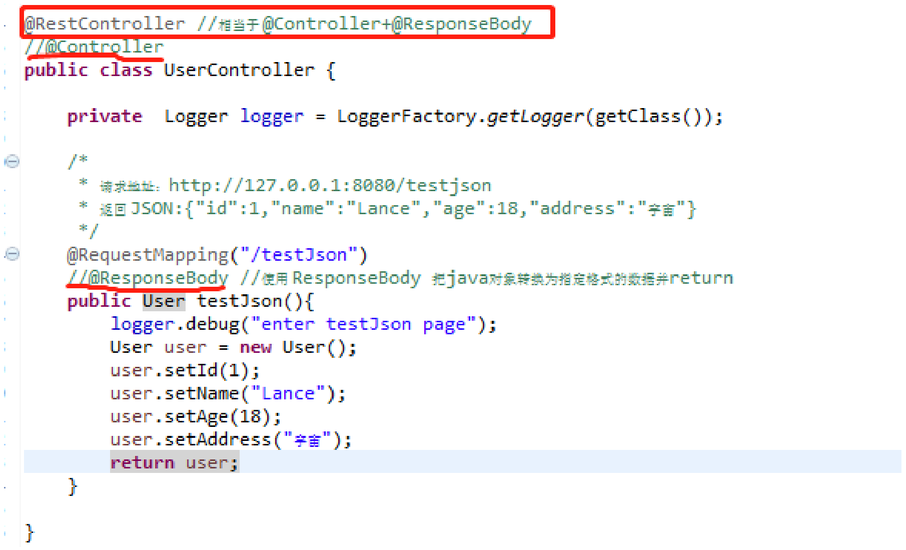
# springboot-json

下载地址：https://github.com/QiangBoCai/springbootDemo/

## 1.返回json的两种方式



```
@RestController //相当于@Controller+@ResponseBody
//@Controller
public class UserController {

    private  Logger logger = LoggerFactory.getLogger(getClass());

    /*
     * 请求地址：http://127.0.0.1:8080/testjson
     * 返回 JSON:{"id":1,"name":"Lance","age":18,"address":"宇宙"}
     */
    @RequestMapping("/testJson")
    //@ResponseBody //使用 ResponseBody 把java对象转换为指定格式的数据并return
    public User testJson(){
        logger.debug("enter testJson page");
        User user = new User();
        user.setId(1);
        user.setName("Lance");
        user.setAge(18);
        user.setAddress("宇宙");
        return user;
    }

}
```

## 2.spring boot JSON 解析包

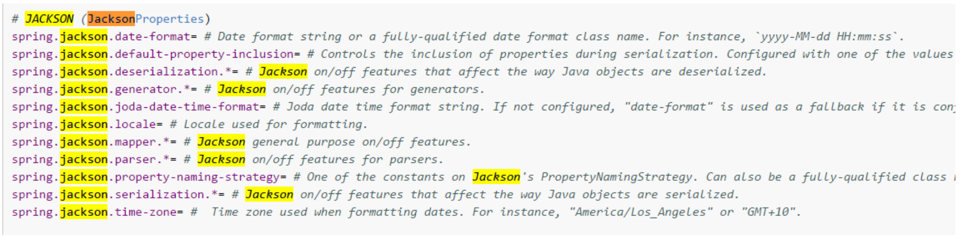
### 2.1 spring boot 使用Jackson（推荐）

Spring Boot 默认引用了JSON解析包Jackson，不需要特殊配置；

#### 2.1.1 使用application.yml配置jackson

application properties可以参考官网的详细配置，jackson部分

https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html



```
# JACKSON (JacksonProperties)
spring.jackson.date-format= # Date format string or a fully-qualified date format class name. For instance, `yyyy-MM-dd HH:mm:ss`.
spring.jackson.default-property-inclusion= # Controls the inclusion of properties during serialization. Configured with one of the values
spring.jackson.deserialization.*= # Jackson on/off features that affect the way Java objects are deserialized.
spring.jackson.generator.*= # Jackson on/off features for generators.
spring.jackson.joda-date-time-format= # Joda date time format string. If not configured, "date-format" is used as a fallback if it is con
spring.jackson.locale= # Locale used for formatting.
spring.jackson.mapper.*= # Jackson general purpose on/off features.
spring.jackson.parser.*= # Jackson on/off features for parsers.
spring.jackson.property-naming-strategy= # One of the constants on Jackson's PropertyNamingStrategy. Can also be a fully-qualified class
spring.jackson.serialization.*= # Jackson on/off features that affect the way Java objects are serialized.
spring.jackson.time-zone= #  Time zone used when formatting dates. For instance, "America/Los_Angeles" or "GMT+10".
```

#### 2.1.2 使用jackson的注解

```
@JsonIgnore//序列化时忽略该字段
//@JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss", timezone = "GMT+8")
public Date getCreateTime() {
 return createTime;
}
```

### 2.2 spring boot 使用fastjson

使用了fastjson，会替代掉Jackson,二者注解不能同时使用

#### 2.2.1 pom.xml配置fastjson依赖

```
<!-- 配置fastjson依赖 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.47</version>
</dependency>
```

#### 2.2.2 使用@Bean注入方式，替代默认的jackson库

```
@SpringBootApplication
public class Application   {

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


 @Bean //使用@Bean注入方式
    public HttpMessageConverters fastJsonHttpMessageConverters(){
        //1.需要定义一个convert转换消息的对象;
        FastJsonHttpMessageConverter fastJsonHttpMessageConverter = new FastJsonHttpMessageConver
ter();
        //2:添加fastJson的配置信息;
        FastJsonConfig fastJsonConfig = new FastJsonConfig();
        fastJsonConfig.setSerializerFeatures(SerializerFeature.PrettyFormat);
        //3处理中文乱码问题
        List<MediaType> fastMediaTypes = new ArrayList<>();
        fastMediaTypes.add(MediaType.APPLICATION_JSON_UTF8);
        //4.在convert中添加配置信息.
        fastJsonHttpMessageConverter.setSupportedMediaTypes(fastMediaTypes);
        fastJsonHttpMessageConverter.setFastJsonConfig(fastJsonConfig);
        HttpMessageConverter<?> converter = fastJsonHttpMessageConverter;

        return new HttpMessageConverters(converter);

    }

 public static void main(String[] args){
  logger.debug("enter main method");

  SpringApplication.run(Application.class, args);

 }
}
```

#### 2.2.3 使用fastjson的注解

```
//@JSONField(serialize = false)//序列化时忽略该字段
//@JSONField(format =  "yyyy-MM-dd HH:mm:ss")
public Date getEndTime() {
 return endTime;
}
```