exception practice

demo1

错误

```
@Test
public void mytest1(){
    UserService userService = new UserServiceImpl();
    userService.sayHello("songge");
}
```

应该从 spring 容器中取得组件

修改

```
@Test
public void mytest1(){
    // 应该从 spring 容器中取得组件
    ApplicationContext applicationContext = new
ClassPathXmlApplicationContext("application.xml");
    UserService userService = (UserService) applicationContext.getBean("userServiceImpl");
    userService.sayHello("songge");
}
```

demo2

错误

```
@Repository
public class UserDaoImpl implements UserDao{
    @Override
    public void sayHello(String content) {
        System.out.println("hello " + content);
    }
}

@Repository
public class UserDaoImpl2 implements UserDao{
    @Override
    public void sayHello(String content) {
        System.out.println("hello " + content);
    }
}
```

```
public class UserServiceImpl implements UserService {
    @Autowired
    UserDao userDao;

    @Override
    public void sayHello(String content) {
        userDao.sayHello(content);
    }
}
```

容器中存在相同类型的 bean 时，需要指定 id

修改

```
public class UserServiceImpl implements UserService {
    // 存在类别相同的组件时应该指定 id
    @Resource(name = "userDaoImpl")
    UserDao userDao;

    @Override
    public void sayHello(String content) {
        userDao.sayHello(content);
    }
}
```

demo3

错误

```
@Pointcut("execution(* com..service.*(..))")
public void pointCut() {
}
```

切点表达式错误

修改

```
@Pointcut("execution(* com..service.*.*(..))")
public void pointCut() {
}
```

demo4

错误

```
@Controller
public class HelloController {

    @RequestMapping("hello")
    public BaseRespVo hello(){
        System.out.println("处理该hello请求的500问题");
        return BaseRespVo.ok();
    }
}
```

直接返回对象，需要相应注解 `@RestController` 或 `@Controller` + `@ResponseBody`

此外需要引入 JSON 解析库 Jackson

修改

```
@RestController
public class HelloController {
    @RequestMapping("hello")
    public BaseRespVo hello(){
        System.out.println("处理该hello请求的500问题");
        return BaseRespVo.ok();
    }
}
```

```
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.12.2</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
    <version>2.12.2</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-annotations</artifactId>
    <version>2.12.2</version>
</dependency>
```

demo5

错误

```
@RestController
public class HelloController {
    @RequestMapping("user/hello")
    public BaseRespVo hello(){
        System.out.println("user hello");
        return BaseRespVo.ok();
    }
}
```

```
@RequestMapping("user")
@RestController
public class UserController {
    @RequestMapping(value = "hello", method = RequestMethod.GET)
    public BaseRespVo hello() {
        System.out.println("user hello");
        return BaseRespVo.ok();
    }
}
```

一个 URL 映射到不同的 Handler 方法上

修改

分别映射到 GET 和 POST 请求上

```
@RestController
public class HelloController {
    @RequestMapping(value = "user/hello",method = RequestMethod.POST)
    public BaseRespVo hello(){
        System.out.println("user hello");
        return BaseRespVo.ok();
    }
}

@RequestMapping("user")
@RestController
public class UserController {
    @RequestMapping(value = "hello", method = RequestMethod.GET)
    public BaseRespVo hello() {
        System.out.println("user hello");
        return BaseRespVo.ok();
    }
}
```