

CSDN

博客学院下载GitChat论坛

...

Spring for Android 参考手册

翻译

2011年07月20日 00:04:25

标签: 框架 / spring / android

7099

原文地址: [点击打开链接](#)

1.Sp...g for Android(SFA) 概述

1.1 前言

对于使用Spring框架的开发人员，SFA为他们在Android的环境下开发提供了有力的支持。通过内嵌的RestTemplate组件，SFA项目可为您的项目提供便利的REST客户端开发方式，这种用法很像spring的jdbc模板类，能有效的简化客户端的开发工作量。同时，SFA也将一系列健壮的、基于开放授权的Spring社交功能集成到您的应用中，这将为您提供实现流行的社交网站类似Twitter、非死不可等的授权连接提供便利之法。

2.RestTemplate 组件

2.1 前言

众所周知，Spring 的RestTemplate提供了健壮的基于Java环境的REST 客户端，而SFA的RestTemplate组件则带来了再Android环境下的类似版本。

2.2 概述

RestTemplate类是SFA RestTemplate组件包的核心类。他在概念上类似Spring框架其他同级项目中提供的模板类。RestTemplate的行为表现取决于他提供的那些回调方法，同时通过配置合适的HttpMessageConverter 类，用户可以高效的将对象转换成HTTP请求中的信息或从响应消息中将信息转换回对应的对象。

每当创建一个新的RestTemplate实例，构造器就会生成配套的辅助对象用于支持RestTemplate的功能。以下是RestTemplate组件中一些提供支持的模块介绍。

2.2.1 HTTP Client

RestTemplate 提供了抽象的RESTful HTTP 请求的模块，内部实现中，RestTemplate 封装了原生的Android HTTP 客户端连接包。这包括标准的J2SE连接组件（用SimpleClientHttpRequestFactory封装）和HttpComponents HttpClient组件（用HttpComponentsClientHttpRequestFactory封装）。默认情况下具体使用哪种ClientHttpRequestFactory 取决于您的Android的版本。

```
[java]
1. protected HttpAccessor() {
2.     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.GINGERBREAD) {
3.         this.requestFactory = new SimpleClientHttpRequestFactory();
4.     } else {
5.         this.requestFactory = new HttpComponentsClientHttpRequestFactory();
6.     }
7. }
```

Unable to Conn

The Proxy was unable to connect to the remote site. responding to requests. If you feel you have reached please submit a ticket via the link provided below.

URL: <http://pos.baidu.com/s?hei=250&wid=300&di=u%2Fblog.csdn.net%2Fyoara%2Farticle%2Fdetails%2>

联系我们

请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

他的最新文章 更多文章

springcloud 启动多网卡选择

Mysql锁查询相关指令

springcloud-config组件的配置和使用

solr最实用入门使用手册

Hystrix介绍

文章分类

hibernate4篇

HTML/JSP9篇

Javascript7篇

Java基础30篇

linux10篇

Spring10篇

展开

文章存档

2018年2月1篇

2017年2月2篇

2016年10月

2016年9月

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

第1页 共20页

2018/3/22 下午1:03

通过 `RestTemplate` (`ClientHttpRequestFactory requestFactory`) 去选择您希望的连接组件的具

联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度
©1999-2015 N版权所有
京ICP证050176号

经营性网站备案
网络110报警
中国互联网
北京互联网违法和不良信息举报中心

入 `Accept-Encoding` 参数指定传输字节流的压缩方式。目前为
组件支持在发送和接受时通过 `gzip` (仅支持 `gzip`) 格式压缩过

`RestTemplate` 需要第三方 `Json` 数据映射包的支持。目前 `SFA` 提供
`son 2.x`, 和 `Google Gson`。虽然 `Jackson` 系列是有名的 `JSON` 解析包的
的 `Android` 程序安装包。

`framework.http.converter.json` 中。

2.2.4 XML 转换

对 `XML` 转换支持同样需要第三方包。`Simple XML serializer` 包提供了不错的转换支持。
`SFA` 将支持代码放在包 `org.springframework.http.converter.xml` 中。

2.2.5 RSS 和 Atom 摘要文件支持

`RSS` 和 `Atom` 的支持也需要第三方包, `Android ROME Feed Reader` 包就提供了相应的功能。
`SFA` 将支持代码放在包 `org.springframework.http.converter.feed` 中。

2.3 所需要的包文件

- `spring-android-rest-template-{version}.jar`
- `spring-android-core-{version}.jar`

2.4 RestTemplate 构造器

下面列出四种 `RestTemplate` 的构造方法。默认的构造方法不包含消息的转换器, 因此您必须自己添加消息转换器。

```
[java]
1. RestTemplate restTemplate = new RestTemplate();
2. restTemplate.getMessageConverters().add(new StringHttpMessageConverter());
```

通过带 `boolean` 参数的构造器可以生成具有默认消息转换器的 `RestTemplate` 实例, 关于转换器的更多细节, 我们在下面讲解。

如上文所讲, 如果您需要一个特别 `ClientHttpRequestFactory`, 也可以向构造器中传入一个 `ClientHttpRequestFactory` 的实例参数。

- `RestTemplate();`
- `RestTemplate(boolean includeDefaultConverters);`
- `RestTemplate(ClientHttpRequestFactory requestFactory);`
- `RestTemplate(boolean includeDefaultConverters, ClientHttpRequestFactory requestFactory);`

2.5 RestTemplate API 简介

加入 CSDN, 享受更精准的内容推荐, 与 500 万程序员共同成长!

登录 注册

2016年5月 展开

他的热门文章

- Ubuntu 下安装 Zabbix 21453
- 从 Spring For Android 开始对 Android 的研究 (一) —— 一个完整的例子 18142
- mina 学习笔记四: 交互的核心 IoSession 18105
- linux 下安装 tesseract-ocr 12057
- mina 学习笔记三: 一切的源头 IoService 8661
- Spring for Android 参考手册 7096
- mina 学习笔记五: 做嫁衣的 IoFilter 和 IoListener 6888
- mina 学习笔记二: 从官方例子开始 6774
- struts1.2 bean 标签的用法 6449
- Nginx+tomcat 实现负载均衡和动静分离 6400

RestTemplate 的API方法命名遵循同样的规则。

联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度
©1999-2015 N版权所有
京ICP证050176号

经营性网站备案信息
网络110报警平台
中国互联网举报中心
不良信息举报中心

将被调用，后一部分表示将返回何种信息。比如getForObject() 响应数据转换成你指定的对象类型并返回。而postForLocation 对象转换成HTTP请求内容发送，返回的信息中将标识新的对

将被封装在RestClientException 中并被抛出，可以通过在Handler来捕获并处理。

```

1.  public <T> T doExecute(URL url, HttpMethod method, RequestCallback
2.      RequestCallback,
3.      ResponseExtractor<T> responseExtractor) throws RestClientException {
4.      Assert.notNull(url, "'url' must not be null");
5.      Assert.notNull(method, "'method' must not be null");
6.      ClientHttpResponse response = null;
7.      try {
8.          ClientHttpRequest request = createRequest(url, method);
9.          if (requestCallback != null) {
10.              requestCallback.doWithRequest(request);
11.          }
12.          response = request.execute();
13.          if (!getErrorHandler().hasError(response)) {
14.              logResponseStatus(method, url, response);
15.          }
16.          else { //如果有异常，调用相应的handler处理
17.              handleResponseError(method, url, response);
18.          }
19.          if (responseExtractor != null) {
20.              return responseExtractor.extractData(response);
21.          }
22.          else {
23.              return null;
24.          }
25.      }
26.      catch (IOException ex) {
27.          throw new ResourceAccessException("I/O error: " + ex.getMessage(), ex);
28.      }
29.      finally {
30.          if (response != null) {
31.              response.close();
32.          }
33.      }
34.  }
```

更多有关于RestTemplate 的信息请参考API Javadoc。

2.5.1 HTTP DELETE

- public void delete(String url, Object... urlVariables) throws RestClientException;
- public void delete(String url, Map<String, ?> urlVariables) throws RestClientException;
- public void delete(URL url) throws RestClientException;

2.5.2 HTTP GET

- public <T> T getForObject(String url, Class<T> responseType, Object... urlVariables) throws R

联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证051305号

经营性网站 网络110报警 中国互联网协会 不良信息举报中心

public void setHeaders(String url, Map<String, ?> urlVariables) throws RestClientException;

public void setHeaders(String url, Object... urlVariables) throws RestClientException;

public void setHeaders(String url, Map<String, ?> urlVariables) throws RestClientException;

public void setHeaders(String url, Object... urlVariables) throws RestClientException;

2.5.4 HTTP OPTIONS

- public Set<HttpMethod> optionsForAllow(String url, Object... urlVariables) throws RestClientException;
- public Set<HttpMethod> optionsForAllow(String url, Map<String, ?> urlVariables) throws RestClientException;
- public Set<HttpMethod> optionsForAllow(URI url) throws RestClientException;

2.5.5 HTTP POST

- public URI postForLocation(String url, Object request, Object... urlVariables) throws RestClientException;
- public URI postForLocation(String url, Object request, Map<String, ?> urlVariables);
- public URI postForLocation(URI url, Object request) throws RestClientException;
- public <T> T postForObject(String url, Object request, Class<T> responseType, Object... urlVariables);
- public <T> T postForObject(String url, Object request, Class<T> responseType, Map<String, ?> urlVariables);
- public <T> T postForObject(URI url, Object request, Class<T> responseType) throws RestClientException;
- public <T> ResponseEntity<T> postForEntity(String url, Object request, Class<T> responseType, Object... urlVariables);
- public <T> ResponseEntity<T> postForEntity(String url, Object request, Class<T> responseType, Map<String, ?> urlVariables) throws RestClientException;
- public <T> ResponseEntity<T> postForEntity(URI url, Object request, Class<T> responseType) throws RestClientException;

2.5.6 HTTP PUT

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册

第4页 共20页

2018/3/22 下午1:03

联系我们

request, Map<String, ?> urlVariables) throws RestClientExce



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

不良信息举报中心

```
6. // 判断给定的类型是否可以被转换器写入
7. boolean canWrite(Class<?> clazz, MediaType mediaType);
8.
9. // 返回可被该转换器支持的header头消息类型
10. List<MediaType> getSupportedMediaTypes();
11.
12. // 将给定的inputMessage消息转换成相应的类型对象并返回
13. T read(Class<? extends T> clazz, HttpInputMessage inputMessage)
14.     throws IOException, HttpMessageNotReadableException;
15.
16. // 将给定的对象转换后写入outputMessage
17. void write(T t, MediaType contentType, HttpOutputMessage outputMessage)
18.     throws IOException, HttpMessageNotWritableException;
19.
20. }
```

其中，SFA框架已经提供了现成的MediaType(即mime类型)。

```
[java]
1. org.springframework.http.MediaType
```

2.6.1 默认的消息转换器

基于性能的选择考虑，默认的RestTemplate无参构造方法不包含任何消息转换器。但是，如果你调用了布尔版本的构造方法并传递true，则构造器会对于一些主要的mime类型增加转换器。当然您也可以编写自定义的转换器，并通过messageConverters属性添加。

RestTemplate生成的转换器包括：ByteArrayHttpMessageConverter，StringHttpMessageConverter，和ResourceHttpMessageConverter。如果Android的版本在2.2或以上，那么XmlAwareFormHttpMessageConverter 和SourceHttpMessageConverter也会被添加。否则，若在2.1版本，因为缺少对XML的必要支持，这两个转换器会被FormHttpMessageConverter 代替。我们来看看代码中是如何初始化的。

```
[java]
1. private static class DefaultMessageConverters {
2.
3.     private static final boolean javaxXmlTransformPresent =
4.         ClassUtils.isPresent("javax.xml.transform.Source", RestTemplate.class.getClassLoader());
5.
6.     private static final boolean simpleXmlPresent =
7.         ClassUtils.isPresent("org.simpleframework.xml.Serializer", RestTemplate.class.getClassLoader());
8.
9.     private static final boolean jacksonPresent =
10.        ClassUtils.isPresent("org.codehaus.jackson.map.ObjectMapper", RestTemplate.class.getClassLoader()) &&
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

12 | private static final boolean jackson2Present =

联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 版权所有
京ICP证050176号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

不良信息举报中心

32. |

if (simpleXmlPresent) {
messageConverters.add(new SimpleXmlHttpMessageConverter());
}

if (jackson2Present) {
messageConverters.add(new MappingJackson2HttpMessageConverter());
} else if (jacksonPresent) {
messageConverters.add(new MappingJacksonHttpMessageConverter());
}

if (romePresent) {
messageConverters.add(new SyndFeedHttpMessageConverter());
}
}

ient("com.fasterxml.jackson.databind.ObjectMapper", RestTem
) &&
.s.isPresent("com.fasterxml.jackson.core.JsonGenerator", Re
der());

boolean romePresent =
ient("com.google.code.rome.android.repackaged.com.sun.syndi
testTemplate.class.getClassLoader());

t(List<HttpMessageConverter<?>> messageConverters) {
l(new ByteArrayHttpMessageConverter());
l(new StringHttpMessageConverter());
l(new ResourceHttpMessageConverter());

sform 不能被类加载器找到,则退回到FormHttpMessageConverter版本
mPresent) {
..add(new SourceHttpMessageConverter<Source>());
..add(new XmlAwareFormHttpMessageConverter());

..add(new FormHttpMessageConverter());

if (simpleXmlPresent) {
messageConverters.add(new SimpleXmlHttpMessageConverter());
}

if (jackson2Present) {
messageConverters.add(new MappingJackson2HttpMessageConverter());
} else if (jacksonPresent) {
messageConverters.add(new MappingJacksonHttpMessageConverter());
}

if (romePresent) {
messageConverters.add(new SyndFeedHttpMessageConverter());
}
}

2.6.2 关于MIME的介绍

[点击打开链接](#)

2.6.3 SFA自带的消息转换器

现在来看看SFA框架中的HttpMessageConverter实现类。对于任何一种转换器都已经对应了默认的media类型(mime)，但是可以通过supportedMediaTypes属性覆盖。

ByteArrayHttpMessageConverter


该转换器可以从HTTP请求或应答中读写字节数组，默认情况下，该转换器支持所有的media types (*/*)，同时写数据会以Content-Type为application/octet-stream（任意二进制数据）的形式。

FormHttpMessageConverter

能处理格式化的数据。支持用application/x-www-form-urlencoded(格式化编码数据)和multipart/form-data（多组件格式数据）的读写转换，格式化的数据通常用一个MultiValueMap<String, String>数据结构来读写。

XmlAwareFormHttpMessageConverter

联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站备案

网络110报警

中国互联网协会

北京互联网违法和不良信息举报中心

Simple XmlSerializer

plain (原文数据) 写数据。

Simple XmlSerializer

操作，写资源支持application/octet-stream。

ax.xml.transform.Source类型，只支持DOMSource，SAXSource，情况下支持text/xml和 application/xml两种XML可扩展标记

ing类型，支持所有的text media类型(text/*)，支持Content-Type为 plain (原文数据) 写数据。

支持对XML格式的读写处理，需要第三方包Simple XML serializer。XML的映射工作可以通过该包提供的注解方式进行配置。如果有需要提供附加的控制功能，可以通过切面的形式通过一个可定制的Serializer进行功能的加强。默认情况下，支持读写application/xml，text/xml，和application/*+xml这几种MIME类型。

需要注意的是，此框架和Spring OXM并不兼容。他是用在SFA中的独立的第三方XML串行化实现。基于maven的依赖如下

```
[java]
1. <dependency>
2.   <groupId>org.simpleframework</groupId>
3.   <artifactId>simple-xml</artifactId>
4.   <version>${simple-version}</version>
5. </dependency>
```

MappingJackson2HttpMessageConverter

基于Jackson框架提供对请求与应答数据的Json-Object转换。对于Json的映射配置可以Jackson提供的注解支持。当串行化/反串行化需要时也支持切面式的控制管理，默认支持application/json类型的读写。

请注意该转换器和GsonHttpMessageConverter 转换器都支持application/json类型的数据转换，因此您最好只添加一款转换器，因为RestTemplate 会优先选择他最早匹配到的转换器进行数据转换，所以两款转换器都添加会导致不可预料的结果。

MappingJackson2HttpMessageConverter如果不是通过maven而是手工添加，您还需要在lib包中引入jackson-annotations和jackson-core jars。

```
[java]
1. <dependency>
2.   <groupId>com.fasterxml.jackson.core</groupId>
3.   <artifactId>jackson-databind</artifactId>
4.   <version>${jackson-version}</version>
5. </dependency>
```

也可以通过MappingJacksonHttpMessageConverter支持 Jackson JSON Processor包。同样，手工导入

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 💬 客服论坛

关于 招聘 广告服务 百度

©1999-2013 版权所有

京ICP证050055号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

不良信息举报中心

```
<!-- kson -->
<groupId>
<artifactId>
</version>
```

ageConverter的功能一直，也支持注解和切面式。

```
<!-- gson -->
<groupId>
<artifactId>
</version>
```

SyndFeedHttpMessageConverter

基于Android ROME Feed Reader包用于读写RSS和Atom feeds的转换器。数据将被转换成com.google.code.rome.android.repackaged和com.sun.syndication.feed.synd.SyndFeed类型。默认情况下支持application/rss+xml 和application/atom+xml两种MIME类型。SyndFeedHttpMessageConverter是RssChannelHttpMessageConverter和AtomFeedHttpMessageConverter的集合体，如果您只需要其中之一，您可以使用任意一个转换器替换SyndFeedHttpMessageConverter。

在Android2.1或更早，该包还依赖于JDOM包。

```
[java]
1. <dependency>
2.   <groupId>com.google.code.android-rome-feed-reader</groupId>
3.   <artifactId>android-rome-feed-reader</artifactId>
4.   <version>${android-rome-version}</version>
5. </dependency>

[java]
1. <dependency>
2.   <groupId>org.jdom</groupId>
3.   <artifactId>jdom</artifactId>
4.   <version>${jdom-fork-version}</version>
5. </dependency>
```

Android ROME Feed Reader包还没有通过Maven的中央仓库管理，因此您需要添加下面的远程仓库到您的POM配置文件中。

```
[java]
1. <!-- For developing with Android ROME Feed Reader -->
2. <repository>
3.   <id>android-rome-feed-reader-repository</id>
4.   <name>Android ROME Feed Reader Repository</name>
5.   <url>https://android-rome-feed-reader.googlecode.com/svn/maven2/releases</url>
6. </repository>
```

2.7 一些简单的例子

2.7.1 Basic Usage Example

下面的代码展示了最基本的用法

1 | String url = "http://www.hairju.com";

联系我们



请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有
京ICP证050176号

经营性网站备案
网络110报警
中国互联网
北京互联网不良信息举报中心

new RestTemplate();

ters().add(new StringHttpMessageConverter());

消息转换成 String类型数据

getForObject(url, String.class, "SpringSource");

IC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "http://www.wap
-mobile1.0.dtd">

/www.w3.org/1999/xhtml"><!--STATUS OK--><head><meta http-e
content="text/html; charset=utf-8" /><meta http-equiv="Ca
t="no-cache" /><style type="text/css">body {margin: 0;text
-size: 14px;font-family: Arial,Helvetica,LiHei Pro Medium;
m {position: relative;margin: 12px 15px 91px;height: 41px;

sion

Gzip 能有效的减少通讯传输的信息量。Gzip格式传输必须在服务器端被支持。通过将内容的请求头 Accept-Encoding 甚至为 gzip, 您可以向服务器端请求使用 gzip 压缩传输。如果服务器端支持, 返回压缩后的数据。RestTemplate 将检查 Content-Encoding 属性判断应答返回数据是否是压缩的, 是则用 GZIPInputStream 去解压缩。目前 Content-Encoding 仅支持 gzip 格式。

以下代码展示了如何告知服务器端用 gzip 传输数据。

```
[java]
1. // 添加 Accept-Encoding 消息头属性, 设置为gzip
2. HttpHeaders requestHeaders = new HttpHeaders();
3. requestHeaders.setAcceptEncoding(ContentCodingType.GZIP);
4. HttpEntity<?> requestEntity = new HttpEntity<Object>(requestHeaders);
5. //
6. RestTemplate restTemplate = new RestTemplate();
7. //
8. restTemplate.getMessageConverters().add(new StringHttpMessageConverter());
9. // 注意调用的方法
10. ResponseEntity<String> response = restTemplate.exchange(url, HttpMethod.GET, requestEntity, String.class);
```

需要注意, 如果在 2.3 或以后的版本使用基于 J2SE 的标准连接 SimpleClientHttpRequestFactory, Android 框架会自动在 Accept-Encoding 中设置 gzip, 如果您希望是 gzip 压缩失效, 您必须在 header 信息中增加其他标志值。

```
[java]
1. HttpHeaders requestHeaders = new HttpHeaders();
2. //增加如下标识
3. requestHeaders.setAcceptEncoding(ContentCodingType.IDENTITY);
4. HttpEntity<?> requestEntity = new HttpEntity<Object>(requestHeaders);
5. //
6. RestTemplate restTemplate = new RestTemplate();
7. //
8. restTemplate.getMessageConverters().add(new StringHttpMessageConverter());
9. //
10. ResponseEntity<String> response = restTemplate.exchange(url, HttpMethod.GET, requestEntity, String.class);
```

2.7.3 通过 GET 方式获取 JSON 数据

首先需要定义一个与返回的 JSON 格式对应的 POJO 类。

```
[java]
1. public class Event {
2.     private Long id;
```

加入CSDN, 享受更精准的内容推荐, 与500万程序员共同成长!

登录 注册

第9页 共20页

2018/3/22 下午1:03

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 ● 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站 备案

网络110报警

中国互联网

北京互联网违法和不良信息举报中心

id) {

{

;string title) {

new RestTemplate();

ters().add(new MappingJackson2HttpMessageConverter());

```
5.
6. [] events = restTemplate.getForObject(url, Event[].class);
```

您也可以通过设置请求头Accept信息的方式

```
[java]
1. // 显式设置 Accept header
2. HttpHeaders requestHeaders = new HttpHeaders();
3. requestHeaders.setAccept(Collections.singletonList(new MediaType("application", "json")));
4. HttpEntity<?> requestEntity = new HttpEntity<Object>(requestHeaders);
5.
6. //
7. RestTemplate restTemplate = new RestTemplate();
8. //
9. restTemplate.getMessageConverters().add(new MappingJackson2HttpMessageConverter());
10. //
11. ResponseEntity<Event[]> responseEntity = restTemplate.exchange(url, HttpMethod.GET, requestEntity, Event[].class);
12. Event[] events = responseEntity.getBody();
```

其中MappingJackson2HttpMessageConverter可以换成GsonHttpMessageConverter，作为您另一种实现JSON解析的选择。

2.7.4 通过GET方式获取 XML 数据

我们使用刚刚用过的Event对象来演示XML的使用方法，注意Event的类及属性都有注解。

```
[java]
1. @Root
2. public class Event {
3.     @Element
4.     private Long id;
5.     @Element
6.     private String title;
7.
8.     public Long getId() {
9.         return id;
10.    }
11.    public void setId(Long id) {
12.        this.id = id;
13.    }
14.
15.    public String getTitle() {
16.        return title;
17.    }
18.    public String setTitle(String title) {
19.        this.title = title;
20.    }
21. }
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

[登录](#)[注册](#)

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站 信息

网络110报警平台

中国互联网协会 心

北京互联网违法和不良信息举报中心

```
2. new RestTemplate();
3. restTemplate.getMessageConverters().add(new SimpleXmlHttpMessageConverter());
4.
5. List eventList = restTemplate.getForObject(url, EventList.class);
```

和刚才一样，也可以通过设置请求头Accept信息的方式：

```
[java]
1. HttpHeaders requestHeaders = new HttpHeaders();
2. requestHeaders.setAccept(Collections.singletonList(new MediaType("application", "xml")))
3. HttpEntity<?> requestEntity = new HttpEntity<Object>(requestHeaders);
4.
5. RestTemplate restTemplate = new RestTemplate();
6. //
7. restTemplate.getMessageConverters().add(new SimpleXmlHttpMessageConverter());
8. //
9. ResponseEntity<EventList> responseEntity = restTemplate.exchange(url, HttpMethod.GET, requestEntity, EventList.class);
10. EventList eventList = responseEntity.getBody();
```

2.7.5 通过POST的方式发送 JSON 数据

我们定义了如下的POJO用于演示：

```
[java]
1. public class Message {
2.     private long id;
3.     private String subject;
4.     private String text;
5.
6.     public void setId(long id) {
7.         this.id = id;
8.     }
9.     public long getId() {
10.        return id;
11.    }
12.
13.    public void setSubject(String subject) {
14.        this.subject = subject;
15.    }
16.    public String getSubject() {
17.        return subject;
18.    }
19.
20.    public void setText(String text) {
21.        this.text = text;
22.    }
23.    public String getText() {
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

[登录](#)[注册](#)

联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证051065号

经营性网站备案信息

网络110报警平台

中国互联网举报中心

不良信息举报中心

e();

ect");

new RestTemplate();

ters().add(new MappingJackson2HttpMessageConverter());

ters().add(new StringHttpMessageConverter());

e.postForObject(url, message, String.class);

e信息也是可以的

1. 数据

2. ge message = new Message();

3. ge.setId(555);

4. ge.setSubject("test subject");

5. ge.setText("test text");

6.

7. // 设置 Content-Type

8. HttpHeaders requestHeaders = new HttpHeaders();

9. requestHeaders.setContentType(new MediaType("application", "json"));

10. HttpEntity<Message> requestEntity = new HttpEntity<Message>(message, requestHeaders);

11.

12. RestTemplate restTemplate = new RestTemplate();

13. restTemplate.getMessageConverters().add(new MappingJackson2HttpMessageConverter());

14. restTemplate.getMessageConverters().add(new StringHttpMessageConverter());

15. //

16. ResponseEntity<String> responseEntity = restTemplate.exchange(url, HttpMethod.POST, requestEntity, String.class);

17. String result = responseEntity.getBody();

2.7.6 获取 RSS 或 Atom 的订阅信息

下面是一个加载RSS订阅信息的例子：

```
[java]
1. RestTemplate restTemplate = new RestTemplate();
2. //
3. restTemplate.getMessageConverters().add(new SyndFeedHttpMessageConverter());
4.
5. // Make the HTTP GET request, marshaling the response to a SyndFeed object
6. SyndFeed feed = restTemplate.getForObject(url, SyndFeed.class);

有可能您需要调整SyndFeedHttpMessageConverter指定的MIME类型，比如默认是接受application/rss+xml 和application/atom+xml类型，但有些RSS订阅信息是使用text/xml类型的，下面的代码将演示如何设置。
```

```
[java]
1. SyndFeedHttpMessageConverter syndFeedConverter = new SyndFeedHttpMessageConverter();
2. //设置 指定的MIME类型
3. syndFeedConverter.setSupportedMediaTypes(Collections.singletonList(MediaType.TEXT_XML));
4.
5. RestTemplate restTemplate = new RestTemplate();
6. restTemplate.getMessageConverters().add(syndFeedConverter);
7. //
8. SyndFeed feed = restTemplate.getForObject(url, SyndFeed.class);
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册

第12页 共20页

2018/3/22 下午1:03

RestTemplate可以通过抛出HttpClientErrorException来处理。您可以通过调用getStatusCode()您可以确定具体的原因及确定如何处理他



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
14. ...
15. ...
16. ...
    log.e(TAG, e.getLocalizedMessage(), e);
    // 处理401 Unauthorized
```

3. Auth Module

3.1 Introduction

目前很多应用都需要通过外部的web服务以获得数据。这些web服务有可能是第三提供的，例如Twitter；也有可能是您从家里连接到公司网络服务。在这些情况下，通过web服务访问数据，您必须进行身份验证以及授权您移动设备上的应用程序可以访问这些数据。spring-android-auth组件的目标就是为获取web服务的验证提供简便的框架支持。

现在有许多关于授权认证的方法和协议，有自定义的、有专用的也有开放标准的。现在一种迅速流行的协议是OAuth。

OAuth是一种开放的协议，他允许用户向第三方应用或网站授权去访问在另一个网站上的受限（需要授权）的服务，第三方应用将获得一个认证令牌用于访问受限的服务。通过使用这个认证令牌策略，用户的登录信息不回存储在第三方而且只在访问该受限服务时有效。

3.2 Overview

现在spring-android-auth的版本通过整合 Spring Social组件以支持Android应用中的OAuth 1.x 或2.0的协议。他包含一个 SQLite 的数据仓库和与Android系统兼容的Spring Security encryption（Spring安全加密组建）。Spring Social组件可以是您的应用方便迅速的建立起与Software-as-a-Service (SaaS)服务提供商，例如Twitter、非死不可等之间的连接，从而简便的调用他们提供的服务API。为了Spring Social组件能有效的使用，以下类需要简单介绍：

3.2.1 SQLite Connection Repository


SQLiteConnectionRepository 实现了Spring Social组件中的ConnectionRepository 接口，他用来将连接信息持久化到Android设备中的SQLite 数据库。连接库是专门为单一用户设计的用于多账号多服务的信息的保存。

如果您的设备被多人使用，那么SQLiteUsersConnectionRepository 类支持多用户账号管理。不过这种情形很少见，毕竟大部分人不会和别人共用手机或移动设备。

3.2.2 Encryption

Java版的Spring Security Crypto组件并不支持Android环境。为了有效利用Spring Security组件的加密目的，我们特意为Android环境提供了AndroidEncryptors类。AndroidEncryptors 使用一

联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

• jackson-core-asl-{version}.jar

3.4 使用案例

3.4.1 Initializing the SQLite Database

SQLiteConnectionRepositoryHelper 继承自SQLiteOpenHelper。为了避免内存泄露等问题，您最好通过传递Application Context类型的上下文引用给SQLiteConnectionRepositoryHelper构造器用于生成新的实例。当应用第一次使用SQLiteConnectionRepositoryHelper 时，一个名为spring_social_connection_repository.sqlite的数据库文件将被生成用于存储连接信息。

```
[java]
1. Context context = getApplicationContext();
2. SQLiteOpenHelper repositoryHelper = new SQLiteConnectionRepositoryHelper(context);
```

3.4.2 Single User App Environment

这个例子将演示如何为多连接建立一个连接库。为了建立链接库，您需要以下对象：

- ConnectionFactoryRegistry connectionFactoryRegistry;
- SQLiteOpenHelper repositoryHelper;
- ConnectionRepository connectionRepository;

ConnectionFactoryRegistry 对象为应用存储了不同的连接信息。

```
[java]
1. connectionFactoryRegistry = new ConnectionFactoryRegistry();
```

Spring Social组件提供了一些常用服务的连接工厂类，此外，您也可以基于Spring Social 创建自己的连接工厂。下面这个例子是Twitter的连接工厂

```
[java]
1. // 加密的ID和密钥
2. String consumerToken = "YR571S2JivB0fyJS5MEg";
3. String consumerTokenSecret = "Kb8HS0luftwCJX3qVoyiLUMfZDtK1EozFoUkjlNLMx4";
4.
5. // 提供Twitter的连接工厂
6. TwitterConnectionFactory twitterConnectionFactory;
7. twitterConnectionFactory = new TwitterConnectionFactory(consumerToken, consumerTokenSecret);
```

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 ● 客服论坛

关于 招聘 广告服务 百度

©1999-2015 版权所有

京ICP证050176号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



lConnectionFactory(facebookConnectionFactory);

立商的连接信息准备连接库。

对象

```
ionContext();
lper = new SQLiteConnectionFactoryHelper(context);
```

加密OAuth信息

```
: AndroidEncryptors.noOpText();
```

```
.onRepository = new SQLiteConnectionFactoryHelper(repositoryHe
```

```
', textEncryptor);
```

3.4.3 OAuth 数据

通过 `TextEncryptor` 类 `Spring Social` 组件支持加密用户的 OAuth 连接信息然后保存进 `ConnectionFactory` 中。`password` 和 `salt` (佐料, 乱序值) 可用于生成加密后的密钥。`salt` 值必须是16进制的、随机的和应用中保持一致且通用的。

把加密后的 OAuth 信息存储到数据库中也不是一个绝对的安全策略。您应该始终谨记肯定存在一些工具可以把 `.dex` 文件转换成 `JAR` 文件, 然后反编译出源码。特别的如果您编写的代码是客户端的, 那么他比服务器端的代码更容易受到破解。下面是个使用的例子。

```
[java]
1. String password = "password";
2. String salt = "5c0744940b5c369b";
3. TextEncryptor textEncryptor = AndroidEncryptors.text(password, salt);
4. ConnectionRepository repository = new SQLiteConnectionFactoryHelper(repositoryHelper,
5.     connectionFactoryRegistry, textEncryptor);
```

在开发阶段也许您希望避免使用加密以方便调试和查看数据库中的 OAuth 数据, 那么下面这个例子可以给你一些参考。

```
[java]
1. TextEncryptor textEncryptor = AndroidEncryptors.noOpText();
2. ConnectionRepository repository = new SQLiteConnectionFactoryHelper(repositoryHelper,
3.     connectionFactoryRegistry, textEncryptor);
```

3.4.4 创建一个 基于 OAuth 1.0a 协议的连接

我们将通过接下来的几个步骤展示如何创建一个对 `Twitter` 的连接。第一步是获取上一步已经创建的 `TwitterConnectionFactory` 对象。

```
[java]
1. TwitterConnectionFactory connectionFactory;
2. ConnectionRepository repository = (TwitterConnectionFactory) connectionFactoryRegistry.getConnectionFactory(Twitter.class);
```

第二步, 生成一次性的请求令牌, 用于后续的步骤。

```
[java]
1. OAuth1Operations oauth = connectionFactory.getOAuthOperations();
2.
3. // 回调url用于应用程序响应OAuth校验
4. String callbackUrl = "x-org-springsource-android-showcase://twitter-oauth-response";
5.
```

加入CSDN, 享受更精准的内容推荐, 与500万程序员共同成长!

登录

注册

一旦您有了这个url, 您可以在浏览器中显示, 让用户登录并

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 ● 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站备案

网络110报警

中国互联网协会

北京互联网违法和不良信息举报中心

7. <intent-filter>
8. <activity>

响应 <Activity>必须要用"oauth_verifier"键从请求中获得OAuth校验。

```
[java]
1. Uri uri = getIntent().getData();
2. String oauthVerifier = uri.getQueryParameter("oauth_verifier");
```

一旦你拥有了OAuth校验, 你就可以验证刚刚生成的请求令牌了。

```
[java]
1. AuthorizedRequestToken authorizedRequestToken = new AuthorizedRequestToken(requestToken, verifier);
```

现在, 您可以用请求令牌生成访问令牌了, 一旦你有了访问令牌, 请求令牌就不再需要了, 您可以安全的丢弃他。

```
[java]
1. OAuth10Operations oauth = connectionFactory.getOAuthOperations();
2. OAuthToken accessToken = oauth.exchangeForAccessToken(authorizedRequestToken, null);
```

最后一步, 我们创建Twitter连接并把它保存到连接库中。

```
[java]
1. Connection<TwitterApi> connection = connectionFactory.createConnection(accessToken);
2. connectionRepository.addConnection(connection);
```

3.4.5 Establishing an OAuth 2.0 connection

接下来基于OAuth2.0协议的方式创建非死不可的连接。请注意不同的服务供应商实现是不同的, 您需要针对OAuth2.0的供应商最适当的调整。

第一步获取factory, 这一部和上面的一样, 注意先前要先生成factory。

```
[java]
1. FacebookConnectionFactory connectionFactory;
2. connectionFactory = (FacebookConnectionFactory) connectionFactoryRegistry.getConnectionFactory(Facebook.class);
```

指定重定向URL, 在非死不可的情景里, 我们需要使用客户端认证流程。为了获取一个访问令牌, 非死不可会重定向到一个在URI的信息段中包含访问令牌的成功页面上。

请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站备案信息

网络110报警平台

中国互联网举报中心

不良信息举报中心

im,offline_access,read_stream,user_about_me";

方便在手机尺寸上显示，您还需要在request中增加额外的参数。下面的代码会向您展示Spring Social组件如何支持传递该参数。

additionalParameters = new LinkedMultiValueMap<String, String>();

additionalParameters.add("play", "touch");

页面url，并用浏览器或者webView中。

new OAuth2Parameters(redirectUri, scope, null, additionalParameters);

2. OAuth2Operations oauth = connectionFactory.getOAuth2Operations();

3. String authorizeUrl = oauth.buildAuthorizeUrl(GrantType.IMPLICIT_GRANT, parameters);

那接下来一步就是用户在webView或浏览器中登录并授权您的应用。浏览器将会重定向到刚刚指定的页面。如果认证通过，成功页面的URI片段中将会包含access_token 参数，获取访问令牌然后创建非死不可的连接吧。

[java]

1. AccessGrant accessGrant = new AccessGrant(accessToken);

2. Connection<FacebookApi> connection = connectionFactory.createConnection(accessGrant);

3. connectionRepository.addConnection(connection);

4.完整的例子

完整的例子可参考[sample Android application](#).

[java]

1. \$ git clone git://github.com/SpringSource/spring-android-samples.git

您目前尚未登录，请[登录](#)或[注册](#)后进行评论

Spring中文参考手册

zwhfyy

2006年10月14日 22:06

3636

Spring中文参考手册得到Spring Framework开发团队的直接授权和大力的支持，其目的是在中文世界推广优秀的开源技术。本次翻译活动由满江红开放技术研究组织(<http://www.redsa...>)

Spring Framework 开发参考手册中文(在线HTML)

Spring Framework 开发参考手册中文(在线HTML)Spring Framework 开发参考手册AuthorsRod Johnson, Juergen Hoeller, Alef Ar...

zfrong

2009年03月09日 11:00

27959

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册

第17页 共20页

2018/3/22 下午1:03

联系我们



请扫描二维码联系客服
✉ webmaster@csdn.net
☎ 400-660-0108
💬 QQ客服 ● 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有
京ICP证051305号

经营性网站备案信息

网络110报警平台

中国互联网举报中心

不良信息举报中心

Spring Boot 参考手册_中英文对照_Part IV_26

 Quinc 2017年01月09日 21:08 1651

【参与翻译】Spring4.x框架参考文档

 isea533 2014年12月31日 08:53 32466

Spring4.x框架参考文档为了不出现重复翻译Spring4参考文档，专门发此文进行通知，同时也希望有兴趣参与翻译的参与进来。 项目地址：<http://git....>

Spring Framework 开发参考手册 之一 简介

第 1 章 简介1.1. 概览Spring包含许多功能和特性，并被很好地组织在下图所示的七个模块中。本节将依次介绍每个模块。Spring框架概览 Core包是框架的最基础部分，并提供依赖注入（D...

 pioneer 2005年04月28日 14:25 1278

Spring Framework 开发参考手册 - Spring Framework reference 2.0.5 参考...

原文：<http://doc.javanb.com/spring-framework-reference-zh-2-0-5/> Spring Framework 开发参考手册 - Spr...

 GeneralYY0 2012年11月30日 14:55 3618

Spring Boot 2.0.0参考手册_中英文对照_Part III_14-18

Spring Boot 2.0.0参考手册_中英文对照_Part III_14-18

 Quincuntial 2016年11月06日 23:08 1612

SpringBoot 中文手册 --Part IV --27

 bingfengzhihuo521 2017年10月20日 17:52 82

受 <http://blog.csdn.net/column/details/13024.html> 专栏启发，对后续部分进行翻译，方便使用时查阅；本文翻译自 <https://docs.spring...>

Spring框架参考手册_5.0.0_中英文对照版_Part II_3.1

这一章包含了Spring框架的控制反转(IoC)原理的实现。

 Quincuntial 2016年09月10日 16:49 1547

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服

💬 客服论坛

关于

招聘

广告服务

百度

©1999-20

N版权所有

京ICP证05

3号

经营性网站

网络110报警

中国互联网

北京互联网

不良信息举报中心



中文版_Part I

8102

中文版

sekift

2017年11月15日 12:33

674

1.0参考文档的中文版，还是全部都翻译了。这书近1000页，这货都翻

/yuedu.baidu.co...

Python 手册

Jerry_1126

2014年11月19日 06:49

3134

用于记录

用PYTHON编程的时候，用于方便记忆的“备忘录”，完整的参考手册，请参考Python的标准文档

https://www.python.org/doc/...

Spring Boot 2.0.0参考手册_中文版_Part IV_26

Spring Boot 2.0.0参考手册_中文版_Part IV_26

Quincuntial

2017年01月09日 21:05

1807

Spring Boot 2.0.0参考手册_中文版_Part II_11-12

Spring Boot

Quincuntial

2016年10月14日 21:31

2146

Spring Security 参考手册 (一)

bigKylin

2017年05月13日 10:11

1710

Spring Security 参考手册 Ben AlexLuke TaylorRob WinchGunnar Hillert Table of Contents 前言入门简介 Sp...



Unable to Connect to Site

《python参考手册(第四版)》【PDF】下载

《python参考手册(第四版)》【PDF】下载链接: https://u253469.pipipan.com/fs/253469-230382222 内容介绍本书是权威的Python语...

zhiyong499

2017年12月09日 21:39

217

读《VBScript程序员参考手册》，做的笔记

在图书馆能找到的关于VBScript的书，大部分都是05年、03年。这本是09年，但是，依然太旧太旧。日新月异，还是不要学习VBScript，去学习其他语言吧。以下是我觉得值得一记的地方，与诸...

tiantuanzi

2015年11月28日 19:38

822

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录 注册

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

👤 QQ客服 🗨 客服论坛

关于 招聘 广告服务 百度

©1999-2015 N版权所有

京ICP证050176号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心



图书

shareid=125052&uk=721744522 C# 4.0完全参考手册目录图书信息