

[illegible]

■ 2550mAh电

■ 日本雾霾之战：民众与政府的博弈

■ 南非猎豹冒死捕杀豪猪被扎满嘴刺

■ 别光顾着把枪口对准柴静

■ 马来西亚男子与眼镜王蛇接吻(图

■ 【早评】降息利好将会在后市中逐

■ 哪些人能从柴静的“穹顶之下”赚

■ “深圳机场撞人事件”不是一个人

■ 政采剔除国外品牌：早该说了！

■ 【DIY】做一枚宫灯迎元宵



青蛙冒险坐鳄鱼鼻子



广西北海银滩美景



萨尔瓦多城铁笼监狱



狮子金沙禁猎区旅馆



潜水员勇喂巨型双髻鲨



俄罗斯山中隐士生活

查看更多>>

谁看过这篇博文



柴小希要...



微微暖



滕格里



Geepro



旭初



Tui_GuiGe



杨258530489



J冷夜流...



清子青



Qt记事



daysMark



袁伟康1993

2月28日

2月28日

2月25日

2月16日

2月15日

2月13日

2月12日

2月11日

2月11日

2月11日

2月10日

2月10日

应用接入 > API文档 > API列表

应用接入	wiki索引
新手指引	>
政策与规范	>
API文档	▼
API导航图	
API联调工具	
API列表	
+ API调用说明	
+ API调试说明	
技术优化原则	
SDK下载	
推广渠道	>

v3/user/get_info

目录

- What's New?
- 1 功能说明

2 接口调用说明

2.1 URL

2.2 格式

2.3 HTTP请求方式

2.4 IP限制

2.5 输入参数说明

2.6 请求示例

2.7 返回参数说明

2.8 错误返回码说明

2.9 正确返回示例

2.10 错误返回示例

3 示例代码

4 接口调试

看到这里，对API的使用应该没有什么问题了。下面，我们就利用百度mp3接口来获取在线音乐。

一、调用接口获取详细数据

接口：<http://mp3.baidu.com/dev/api/?tn=getinfo&ct=0&ie=utf-8&word=&format=>

参数：

word：“歌曲名”或“歌手名”，//歌曲名

format：“json”或“xml”，//返回数据格式，Json 或 XML，默认XML

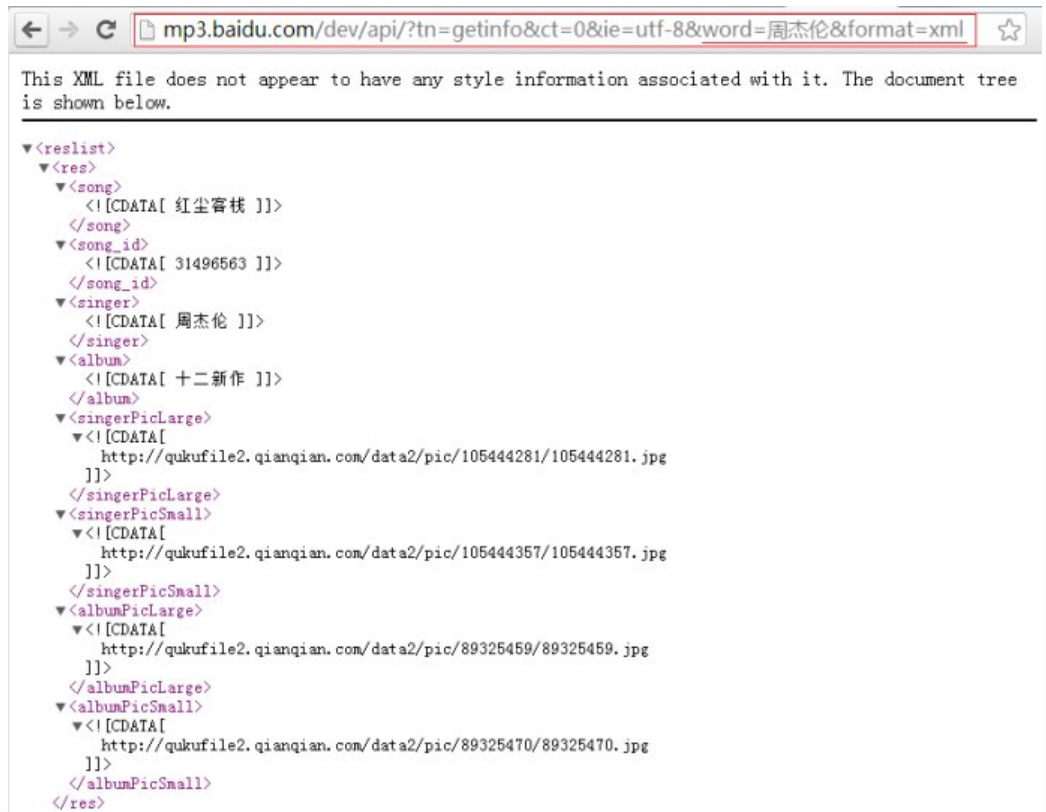
请求方式：GET

可以通过浏览器进行测试：

Json格式：



XML格式：



二、通过歌曲id (song_id) 获取歌曲下载地址

接口: <http://ting.baidu.com/data/music/links?songIds=>

参数:

songIds: "31496563" //歌曲id, 从第1个列表中得到的歌曲id

callback: "callback" //不为空时, 返回json格式数据

请求方式: GET

可以通过浏览器进行测试:



三、下载、试听歌曲, 下载歌词

mp3下载地址为songLink, 后面加xcode参数, xcode每秒自动更新一次(可以不断刷新尝试)。

歌词lrc地址为<http://ting.baidu.com>加lrcLink

如: <http://ting.baidu.com/data2/lrc/31626524/31626524.lrc>

代码如下:

```
#define MP3_URL "http://mp3.baidu.com/dev/api/?"

//设置发送的数据
QByteArray byte_array;
byte_array.append("tn=getinfo");
byte_array.append("&ct=0");
byte_array.append("&ie=utf-8");
byte_array.append("&word=" + text); //text指歌曲名称、歌手名称
byte_array.append("&format=json");

QString url = MP3_URL + QString(byte_array);
```

```
network_manager = new QNetworkAccessManager();
network_request = new QNetworkRequest();
connect(network_manager, &QNetworkAccessManager::finished, this, &HttpManager::replyFinished);
network_request->setUrl(QUrl(url));
network_manager->get(*network_request);

void HttpManager::replyFinished(QNetworkReply *reply)
{
    //获取响应的信息, 状态码为200表示正常
    QVariant status_code = reply->attribute(QNetworkRequest::HttpStatusCodeAttribute);

    //无错误返回
    if(reply->error() == QNetworkReply::NoError)
    {
        QByteArray bytes = reply->readAll(); //获取字节
        QString result(bytes); //转化为字符串

        parseJson(result); //自定义方法, 解析歌曲数据
    }
    else
    {
        //处理错误
    }
}

void HttpManager::parseJson(QString json)
{
    QByteArray byte_array;
    QJsonParseError json_error;
    QJsonDocument parse_doucment = QJsonDocument::fromJson(byte_array.append(json), &json_error);
    if(json_error.error == QJsonParseError::NoError)
    {
        if(parse_doucment.isArray())
        {
            QJsonArray array = parse_doucment.array();
            int size = array.size();
            for(int i=0; i
            {
                QJsonValue value = array.at(i);
                if(value.isObject())
                {
                    QJsonObject object = value.toObject();
                    if(object.contains("song"))
                    {
                        QJsonValue name_value = object.take("song");
                        if(name_value.isString())
                        {
                            QString name = name_value.toString();
                            QString name2 = name;
                            QListWidgetItem *list_item = new QListWidgetItem();
                            list_item->setText(name);
                            this->addItem(name);
                        }
                    }
                }
            }

            if(song_id.isEmpty())
            {
                if(object.contains("song_id"))
                {
                    QJsonValue song_value = object.take("song_id");
                    if(song_value.isString())
                    {
                        QString song_id = song_value.toString();
                    }
                }
            }
        }
    }
}
```

```
}  
}  
}  
}  
}  
}
```

这样就可以获取，音乐文件的所有信息了，然后再次请求，获取歌曲的下载链接、歌词等信息，即可实现下载及在线试听功能。

在线试听：
QMediaPlayer *player = new QMediaPlayer(this);
player->setMedia(QUrl(song_link)); //song_link为根据步骤二获取的歌曲链接
player->setVolume(50);
player->play();

效果如下：



注：
技术在于交流、沟通，转载请注明出处并保持作品的完整性。
作者：`☆奋斗ing♥孩子` 原文：http://blog.sina.com.cn/s/blog_a6fb6cc90101h3bd.html。

5

喜欢

分享： 
阅读(3619) | 评论 (4) | 收藏(1) | 已有5人转载▼ | 喜欢▼ | 打印
已投稿到：  排行榜

前一篇： Qt之QFileIconProvider续（获取文件图标、类型）
后一篇： 相遇Qt5

评论

重要提示：警惕虚假中奖信息

[发评论]

代水易带



2014-3-5 12:44

回复(1)

疯狂的石头

你这是分分钟就出个桌面软件么？Qt效率这么高呀，嘿嘿

2014-6-4 21:27

回复(1)

发评论

一去、二三里：





☐ 分享到微博

☐ 匿名评论

验证码： 请点击后输入验证码 收听验证码

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇

后一篇 >

Qt之QFileIconProvider续（获取文件图标、类型）

相遇Qt5