

# 斗票WebSDK接口文档

## 模块：JNetwork(Class)

### 类：JNetwork

- 1. POST请求

```
static POST(url, parameters, headers){}
```

- 请求参数

请求参数	数据类型	数据说明
url	string	相对地址
parameters	string	地址参数
headers	string	头参数

- 返回数据

返回数据	数据类型	数据说明
data	*	请求返回的数据

- 调用示例

```
JNetwork.POST('/refreshlocation', {
  longitude: 122.0248313589239,
  latitude: 37.33800802417976
}, {
  'Accept': 'application/json',
  'Content-Type': 'application/json'
}).then(data => {
  console.log(data);
}, error => {
  console.log(error);
});
```

## • 2. GET请求

```
static GET(url, parameters, headers){}
```

### • 请求参数

请求参数	数据类型	数据说明
url	string	相对地址
parameters	string	地址参数
headers	string	头参数

### • 返回数据

返回数据	数据类型	数据说明
data	*	请求返回的数据

### • 调用示例

```
JNetwork.GET('/refreshlocation', {
  longitude: 122.0248313589239,
  latitude: 37.33800802417976
}).then(data => {
  console.log(data);
}, error => {
  console.log(error);
});
```

## 类：JNetworkConfig

- 1. 获取首页数据

```
static setConfig(config){}
```

- 请求参数

请求参数	数据类型	数据说明
config	<a href="#">ConfigModel</a>	object

- 返回数据

返回数据	数据类型	数据说明
-	-	-

- 调用示例

```
JNetworkConfig.setConfig({
  baseUrl: 'https://jbz-dev.idoupiao.com/openfilm/appapi'
});
```

## 类：JNetworkFilm

- 1. 获取热门电影

```
static filmHotfilms(){}
```

- 请求参数

请求参数	数据类型	数据说明
-	-	-

- 返回数据

返回数据	数据类型	数据说明
hotFilms	Array	热门影片列表
-	<a href="#">FilmDetailModel</a>	热门影片

- 调用示例

```
JNetworkFilm.filmHotfilms().then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- 2. 分页获取热门电影

```
static filmHotfilmsWithPage(cityId, page = 1){}
```

- 请求参数

请求参数	数据类型	数据说明
cityId	int	城市Id
page	int	页号

- 返回数据

返回数据	数据类型	数据说明
films	Array	获取到的影片列表
-	<a href="#">BaseFilmModel</a>	影片

- 调用示例

```
JNetworkFilm.filmHotfilmsWithPage(2, 1).then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 3. 获取待映电影

```
static filmWaitfilmsWithPage(cityId, page = 1){}
```

• 请求参数

请求参数	数据类型	数据说明
cityId	string	城市Id
page	int	页号

• 返回数据

返回数据	数据类型	数据说明
-	<a href="#">BaseFilmModel</a>	影片

• 调用示例

```
JNetworkFilm.filmWaitfilmsWithPage(2, 1).then((data) => {  
  console.log(data)  
}, error => {  
  console.log(error);  
});
```

• 4. 获取影片详情 (交行使用的)?

```
static filmDetail(filmId, platform){}
```

• 请求参数

请求参数	数据类型	数据说明
filmId	string	影片ID
platform	<a href="#">EnumPlatform</a>	平台类型

• 返回数据

返回数据	数据类型	数据说明
-	<a href="#">FilmDetailModel</a>	影片详情

- 调用示例

```
JNetworkFilm.filmDetail('248700', 'maoyan').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- 5. 获取所有电影列表 (交行使用的)?

```
static filmList(){}
```

- 请求参数

请求参数	数据类型	数据说明
-	-	-

- 返回数据

返回数据	数据类型	数据说明
upcommingFilms	Array	待映影片列表
-	<a href="#">FilmDetailModel</a>	影片
hotFilms	Array	热映影片列表
-	<a href="#">FilmDetailModel</a>	影片

- 调用示例

```
JNetworkFilm.filmList().then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

6. 获取指定影片

```
static filmWithFilmId(filmId){}
```

请求参数

请求参数	数据类型	数据说明
filmId	string	影院Id

返回数据

返回数据	数据类型	数据说明
film	Array	影片列表
-	<a href="#">FilmDetailModel</a>	影片
hotComments	Array	热门评论
-	<a href="#">CommentModel</a>	评论

调用示例

```
JNetworkFilm.filmWithFilmId('1a692bb163fa4609b59927055faab749').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```



类：JNetworkCinema

1. 获取影院详情 (暂时不可用)

```
static cinemaDetail(cinemaId){}
```

请求参数

请求参数	数据类型	数据说明
cinemald	string	影院ID

• 返回数据

返回数据	数据类型	数据说明
-	-	-

• 调用示例

```
JNetworkCinema.cinemaDetail('a123123123').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 2. 根据影片Id获取影院的列表（比价）

```
static cinemaContrastNeedLocation(filmId, regionName, orderType){}
```

• 请求参数

请求参数	数据类型	数据说明
filmId	string	影片Id
regionName	string	地域名字
orderType	int	排序类型 1:距离 2:价格

• 返回数据

返回数据	数据类型	数据说明
cinemas	Array	影院列表
-	<a href="#">CinemaCompareModel</a>	影院
regions	Array	行政区列表
-	<a href="#">RegionModel</a>	行政区

• 调用示例



```
JNetworkCinema.cinemaContrastNeedLocation('1a692bb163fa4609b59927055faab749', '', 1)
    .then((data) => {
        console.log(data)
    }, error => {
        console.log(error);
    });
```

### 3. 获取影院的列表（比价，不需要影院Id）

```
static cinemaContrastNoFilmNeedLocation(regionName, orderType){}
```

#### 请求参数

请求参数	数据类型	数据说明
regionName	string	地域名字
orderType	int	排序类型 1:距离 2:价格

#### 返回数据

返回数据	数据类型	数据说明
cinemas	Array	影院列表
-	<a href="#">CinemaCompareModel</a>	影院
regions	Array	行政区列表
-	<a href="#">RegionModel</a>	行政区

#### 调用示例

```
JNetworkCinema.cinemaContrastNoFilmNeedLocation('', 1).then((data) => {
    console.log(data)
}, error => {
    console.log(error);
});
```

### 4. 指定影院的排片的电影列表及影院本身的信息(交行使用的)?

```
static cinemaScreenings(cinemaId){}
```

• 请求参数

请求参数	数据类型	数据说明
cinemald	string	影院ID

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkCinema.cinemaScreenings('a123123123').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 5. 指定影院的排片的最近日期列表(交行使用的)?

```
static cinemaScreeningsDate(cinemaId, filmId){}
```

• 请求参数

请求参数	数据类型	数据说明
cinemald	string	影院ID
filmId	string	影片ID

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkCinema.cinemaScreeningsDate('a123123', 'b123123').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 6. 指定影院的某个影片的某天的排片的详细信息(交行使用的)?

```
static cinemaScreeningsItem(cinemaId, filmId, date){}
```

• 请求参数

请求参数	数据类型	数据说明
cinemaId	string	影院ID
filmId	string	影片ID
date	string	日期

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkCinema.cinemaScreeningsItem('a123123123', 'b123123123', '2017-04-18').then
console.log(data)
}, error => {
  console.log(error);
});
```

• 7. 实时座位图(交行使用的)?

```
static cinemaSeat(type, paras){}
```

• 请求参数

请求参数	数据类型	数据说明
type	string	平台类型
paras	object	(根据不同平台变化)

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkCinema.cinemaSeat('meituan', {}).then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- 8. 影院列表（已弃用）（临时使用了）

```
static cinemaListNeedLocation(filmId){}
```

- 请求参数

请求参数	数据类型	数据说明
filmId	string	影片id

- 返回数据

返回数据	数据类型	数据说明
cinemas	Array	影院列表
-	<a href="#">CinemaModel</a>	影院
recommandCinema	Array	推荐影院

- 调用示例

```
JNetworkCinema.cinemaListNeedLocation('1a692bb163fa4609b59927055faab749').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

## 9. 整合后的影院列表(交行使用的)?

```
static newCinemaListNeedLocation(filmid, region, order, feature, inType, date){}
```

### 请求参数

请求参数	数据类型	数据说明
filmid	string	影片Id
region	string	市区
order	string	排序
feature	string	特色
inType	string	后台为了判断请求的类型（CinemaBuyList：2，CinemaListView：1）
date	string	日期

### 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

### 调用示例

```
JNetworkCinema.newCinemaListNeedLocation().then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- 10. 收藏影院

```
static cinemaFavoriteCinemaNeedLogin(cinemaId, cinemaName){}
```

- 请求参数

请求参数	数据类型	数据说明
cinemaId	string	影院Id
cinemaName	string	影院名字

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkCinema.cinemaFavoriteCinemaNeedLogin('a123123', '上海电影院').then((data) => console.log(data), error => { console.log(error); });
```

- 11. 取消收藏影院

```
static cinemaCancelFavoriteCinemaNeedLogin(cinemaId){}
```

- 请求参数

请求参数	数据类型	数据说明
cinemaId	string	影院Id

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkCinema.cinemaCancelFavoriteCinemaNeedLogin('a123123').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

类：JNetworkFilmView

• 1. 获取指定影院基础数据(如影院电影排片和影院电话及地址)

```
static filmviewFilmview(cinemaId){}
```

• 请求参数

请求参数	数据类型	数据说明
cinemaId	string	影院Id

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkFilmView.filmviewFilmview('a123123').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 2. 获取指定影院排片日期安排

```
static filmviewDate(cinemaId, filmId){}
```

• 请求参数

请求参数	数据类型	数据说明
cinemald	string	影院Id
filmId	string	影片Id

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkFilmView.filmviewDate('a123123', 'b123123').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 3. 获取指定影院排片放映厅安排

```
static filmviewItems(cinemaId, filmId, date){}
```

• 请求参数

请求参数	数据类型	数据说明
cinemald	string	影院Id
filmId	string	影片Id
date	string	日期

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例



```
JNetworkFilmView.filmviewItems('a123123', 'b123123', '2017-04-18').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

## 类：JNetworkTrade

- 1. 锁座

```
static tradeLockSeatNeedLogin(type, paras){}
```

- 请求参数

请求参数	数据类型	数据说明
type	string	平台类型
paras	object	锁座参数

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkTrade.tradeLockSeatNeedLogin('maoyan', {}).then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- 2. 取消锁座

```
static cancelLockSeatNeedLogin(orderId){}
```

- 请求参数

请求参数	数据类型	数据说明
orderId	string	订单Id

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkTrade.cancelLockSeatNeedLogin('dd11111111').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- **3. 下订单**

```
static tradeConfirmOrderNeedLogin(type, paras){}
```

- 请求参数

请求参数	数据类型	数据说明
type	string	平台类型
paras	object	下订单参数

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkTrade.tradeConfirmOrderNeedLogin('maoyan', {}).then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

#### • 4. 申请预订单

```
static tradePrePayOrderNeedLogin(orderId, payType, prizeIds, redIds){}
```

#### • 请求参数

请求参数	数据类型	数据说明
orderId	string	订单Id
payType	string	支付类型
prizeIds	string	待定
redIds	string	待定

#### • 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

#### • 调用示例

```
JNetworkTrade.tradePrePayOrderNeedLogin('d123123', 'weixin', '', '').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

## 类：JNetworkMine

#### • n. 我的订单

```
static mineOrderNeedLogin(){}
```

- 请求参数

请求参数	数据类型	数据说明
-	-	-

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkMine.mineOrderNeedLogin().then((data) => {  
  console.log(data)  
}, error => {  
  console.log(error);  
});
```

- 2. 我的收藏

```
static mineFavoriteNeedLogin(){}
```

- 请求参数

请求参数	数据类型	数据说明
-	-	-

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkMine.mineFavoriteNeedLogin().then((data) => {  
  console.log(data)  
}, error => {  
  console.log(error);  
});
```

## 类：JNetworkCity

- 1. 获取城市列表（需要定位）

```
static cityListNeedLocation(){}
```

- 请求参数

请求参数	数据类型	数据说明
-	-	-

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkCity.cityListNeedLocation().then((data) => {  
  console.log(data)  
}, error => {  
  console.log(error);  
});
```

- 2. 获取当前城市（需要定位）

```
static cityListNeedLocation(location){}
```

- 请求参数

请求参数	数据类型	数据说明
location	object	定位信息（可空）

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkCity.cityListNeedLocation({longitude:121.458858, latitude:23.484588}).then  
console.log(data)  
, error => {  
  console.log(error);  
});
```

- 3. 获取城市列表

```
static cityCities(){}
```

- 请求参数

请求参数	数据类型	数据说明
-	-	-

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkCity.cityCities().then((data) => {  
  console.log(data)  
, error => {  
  console.log(error);  
});
```

类：JNetworkAccount

1. 用户登录

```
static accountLogin(mobile, password){}
```

请求参数

请求参数	数据类型	数据说明
mobile	string	登录需要的手机号码
password	string	登录需要的密码

返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

调用示例

```
JNetworkAccount.accountLogin('13764730291', '123456').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

2. 用户登出

```
static accountLogout(sessionId){}
```

请求参数

请求参数	数据类型	数据说明
sessionId	string	用户登录标识

返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkAccount.accountLogout('').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- 3. 获取验证码

```
static accountGetVerifyCode(mobile, type){}
```

- 请求参数

请求参数	数据类型	数据说明
mobile	string	接收验证码的手机号码
type	string	验证码类型 （1：注册使用 2：忘记密码使用）

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkAccount.accountGetVerifyCode('13764730291', 2).then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

- 4. 注册用户



```
static accountRegister(mobile, verifyCode, password){}
```

- 请求参数

请求参数	数据类型	数据说明
mobile	string	用户的手机号码
verifyCode	string	验证码
password	string	密码

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkAccount.accountRegister('13764730291', '112525', '123456').then((data) => console.log(data)
}, error => {
  console.log(error);
});
```

- 5. 忘记密码并且找回密码

```
static accountUpdatepass(mobile, verifyCode, password){}
```

- 请求参数

请求参数	数据类型	数据说明
mobile	string	用户的手机号码
verifyCode	string	验证码
password	string	新密码

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkAccount.accountUpdatepass('13764730291', '112525', '123456').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

## 类：JNetworkOther

- 1. 搜索

```
static search(val, lastKey){}
```

- 请求参数

请求参数	数据类型	数据说明
val	string	关键字
lastKey	string	目前无用

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkOther.search('上海').then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 2. 按城市搜索

```
static searchSearch(cityId, query, lastKey){}
```

• 请求参数

请求参数	数据类型	数据说明
cityId	string	城市Id
query	string	关键字
lastKey	string	目前无用

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkOther.searchSearch().then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

• 3. 热搜词汇

```
static hotQuery(){}
```

• 请求参数

请求参数	数据类型	数据说明
-	-	-

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkOther.hotQuery().then((data) => {  
  console.log(data)  
}, error => {  
  console.log(error);  
});
```

- 4. 广告接口

```
static bannersNeedCityIdNeedLocation(){}
```

- 请求参数

请求参数	数据类型	数据说明
-	-	-

- 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

- 调用示例

```
JNetworkOther.bannersNeedCityIdNeedLocation().then((data) => {  
  console.log(data)  
}, error => {  
  console.log(error);  
});
```

- 5. 广告栏

```
static pageBanners(){}
```

• 请求参数

请求参数	数据类型	数据说明
-	-	-

• 返回数据

返回数据	数据类型	数据说明
-	promise	异步请求封装

• 调用示例

```
JNetworkOther.pageBanners().then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

# 模块：JNetwork(Delegate)

## 代理：JNetworkDelegate

• 1. 城市信息请求参数代理

```
static cityParas(){}
```

• 回调参数

请求参数	数据类型	数据说明
-	-	-

• 返回数据

返回数据	数据类型	数据说明
-	<a href="#">CityModel</a>	城市模型

• 调用示例

```
cityParas() {  
    return store.getState().location.userLocationCity;  
}
```

## 2. 定位信息请求参数代理

```
static locationParas(){}
```

### 回调参数

请求参数	数据类型	数据说明
-	-	-

### 返回数据

返回数据	数据类型	数据说明
-	<a href="#">CoordinateModel</a>	定位信息模型

### 调用示例

```
locationParas() {  
    return {latitude: 41.816804,  
            longitude: 123.426065};  
}
```

## 模块：JNetwork(Model)

### 模型：CityModel

### 模型：ConfigModel

#### 描述

Network配置模型

#### 属性

所属	参数	数据类型	数据说明	数据示例
-	baseUrl	string	网络请求基础地址	http:https://jbz-dev.idoupiao.com/openfilm/
-	delegate	id < <a href="#">JNetworkDelegate</a> >	网络请求代理	-

## 模型：CoordinateModel

- 描述

定位模型

- 属性

所属	参数	数据类型	数据说明	数据示例
-	latitude	double	纬度	122.024831358
-	longitude	double	经度	37.3380080241

## 模型：BaseFilmModel

- 描述

影片基础模型

- 属性

所属	参数	数据类型	数据说明	数据示例
-	name	string	名称	速度与激情8
-	grade	string	评分	9.2
-	dimensional	string	放映类型	2D/3D/IMAX 3D/中国巨幕/全景声
-	description	string	描述	车主要黑化 家族被击垮
-	director	string	导演	F·加里·格雷
-	actors	string	演员	范·迪塞尔,杰森·斯坦森,道恩·强森
-	showDate	string	日期	2017-04-14
-	duration	string	时长(单位:分钟)	136
-	bigImg	string	大图	http://sep9.cn/c4gwtj
-	id	string	金保证Id	1a692bb163fa4609b59927055faab749

模型：FilmDetailModel : BaseFilmModel

- 描述

影片详情模型

- 属性



所属	参数	数据类型	数据说明	数据示例
-	category	string	类型	"动作,惊悚,犯罪"
-	language	string	语言	英语
-	area	string	上映国家	大陆上映
-	info	string	信息	在这一部中>
-	isShow	bool	是否已经上映	true
-	smallImg	string	小图	http://omdf7d8sn.bkt.clouddn.com/film/bd/速度与激情8
-	photos	string	图片列表(','号分离)	http://sep9.cn/x4mbtk,http://sep9.cn/3q2ax1
-	bdId	string	糯米Id	94866
-	dcId	string	单车Id	911
-	myId	string	猫眼Id	248700
-	mzId	string	卖座Id	3456
-	wpld	string	网票Id	26608
-	spiderId	string	蜘蛛Id	201704760207
-	createdTime	int	创建时间	1492286400000

模型：CommentModel

- 描述
  - 影片评论模型
- 属性

所属	参数	数据类型	数据说明	数据示例
-	nick	string	类型	cuppi
-	insertTime	string	插入时间	2017-04-19 01:50:26
-	score	string	评分	5
-	sureViewed	string	?	1
-	approve	int	点赞	709
-	filmId	int	?	248700
-	oppose	int	?	0
-	spoiler	int	?	0
-	id	int	?	102450041
-	time	string	评论时间	2017-04-14 02:57
-	content	string	内容	从2001年的速1一直追到2017年的速8，从不会飙车的毛头小子，到独挡一面团队领袖。保罗的离去让

模型：BaseCinemaModel

• 描述

影院基础模型

• 属性

所属	参数	数据类型	数据说明	数据示例
-	name	string	名称	大地影院罗宾森店
-	address	string	地址	上海市嘉定区城中路138号罗宾森广场4楼
-	distance	int	距离	12351.6
-	compareCount	int	平台数量	4
-	cinemaNames	Array < <a href="#">EnumPlatform</a> >	平台名字列表	["dazhong","baidu"]
-	showTime	string	?	2017-04-19 18:15
-	id	string	金保证Id	802965
-	minJbzPrice	int	金保证价格	33
-	price	int	价格	33
-	minPrice	int	最低价格	33

## 模型：CinemaModel : BaseCinemaModel

- 描述

影院比价模型

- 属性

所属	参数	数据类型	数据说明	数据示例
-	regionName	string	行政区	宝山区
-	longitude	double	经度	121.412
-	latitude	double	纬度	31.4595
-	cityId	int	城市Id	2
-	cinemaId	string	影院Id	800620
-	cinemaName	string	影院名称	上海亿博国际影城
-	filmId	string	影片Id	1a692bb163fa4609b59927055faab749
-	compareNames	string	比价名称列表	spider,baidu
-	filmName	string	?	速度与激情8
-	highPrice	int	?	3500
-	createdTime	string	?	2017-04-12 09:33:09
-	showDate	string	?	2017-04-15
-	lowPrice	int	?	3300

## 模型：CinemaCompareModel : [BaseCinemaModel](#)

- 描述

影院比价的模型

- 属性

所属	参数	数据类型	数据说明	数据示例
-	jbzShowTime	string	?	2017-04-19 18:15
-	addDc	bool	?	true
-	type	string	当前模型的平台	baidu
-	priority	int	优先级	2
-	~[wpld, spld, mzld, dcld, myld, mtld, dzld, bdld]	string	平台商的Id	870

## 模型：RegionModel

- 描述

行政区模型

- 属性

所属	参数	数据类型	数据说明	数据示例
-	ID	string	行政区Id	289
-	Name	string	行政区名字	宝山区

## 模块：JNetwork(Enum)

### 枚举：EnumPlatform

- 描述

平台枚举

- 属性

类型	枚举	说明
string	wangpiao	网票
*	maizuo	卖座
*	spider	蜘蛛
*	danche	单车
*	maoyan	猫眼
*	dazhong	大众
*	meituan	美团

