

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	参数类型	参数说明
url	string	相对地址
parameters	string	地址参数
headers	string	头参数

- 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

- 调用示例

```
JNetwork.POST('/main/hotfilms', {a: 1}, {'Accept': 'application/json','Content-Type': 'application/json'},  
console.log(data)  
, error => {  
  console.log(error);  
});
```

- 2. GET请求

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

- 请求参数

请求参数	参数类型	参数说明
url	string	相对地址
parameters	string	地址参数
headers	string	头参数

- 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

- 调用示例

```
JNetwork.GET('/main/hotfilms', {a: 1}, {'Accept': 'application/json', 'Content-Type': 'application/json'},  
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	参数类型	参数说明
config	ConfigModel	object

- 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

- 调用示例

```
JNetworkConfig.setConfig({baseUrl: 'https://jbz-dev.idoupiao.com/openfilm',delegat
  console.log(data)
}, error => {
  console.log(error);
});
```

类：JNetworkHome

- 1. 获取首页数据

```
static homeMainPageNeedLocation(){}
```

- 请求参数

请求参数	参数类型	参数说明
-	-	-

- 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

- 调用示例

```
JNetworkHome.homeMainPageNeedLocation().then((data) => {
  console.log(data)
}, error => {
  console.log(error);
});
```

类：JNetworkFilm

• 1. 获取热门电影

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

• 请求参数

请求参数	参数类型	参数说明
-	-	-

• 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

• 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	参数类型	参数说明
cityId	string	城市Id
page	int	页号

• 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

• 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	参数类型	参数说明
cityId	string	城市Id
page	int	页号

• 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

• 调用示例

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

• 4. 获取影片详情

```
static filmDetail(platformFilmId, platformId){}
```

• 请求参数

请求参数	参数类型	参数说明
platformFilmId	string	影片ID
platformId	string	平台id

• 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

• 调用示例

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

• 5. 获取所有电影列表

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

• 请求参数

请求参数	参数类型	参数说明
filmId	string	影院Id

• 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

• 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	参数类型	参数说明
filmId	string	影院Id

返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

调用示例

```
JNetworkFilm.filmWithFilmId('a123321321').then((data) => {  
  console.log(data)  
}, error => {  
  console.log(error);  
});
```

类：JNetworkCinema

1. 获取影院详情

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

请求参数

请求参数	参数类型	参数说明
cinemald	string	影院ID

- 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

- 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	参数类型	参数说明
filmId	string	影片Id
regionName	string	地域名字
orderType	int	排序类型 1:距离 2:价格

- 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

- 调用示例


```
JNetworkCinema.cinemaContrastNeedLocation('a123123123', '浦东新区', 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	参数类型	参数说明
regionName	string	地域名字
orderType	int	排序类型 1:距离 2:价格

• 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

• 调用示例

```
JNetworkCinema.cinemaContrastNoFilmNeedLocation('浦东新区', 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

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

• 请求参数

请求参数	参数类型	参数说明
cinemald	string	影院ID

- 返回参数

返回参数	参数类型	参数说明
-	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

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

- 请求参数

请求参数	参数类型	参数说明
cinemald	string	影院ID
filmId	string	影片ID

- 返回参数

返回参数	参数类型	参数说明
-	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

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

请求参数

请求参数	参数类型	参数说明
cinemaId	string	影院ID
filmId	string	影片ID
date	string	日期

返回参数

返回参数	参数类型	参数说明
-	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

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

请求参数

请求参数	参数类型	参数说明
type	string	平台类型
paras	object	(根据不同平台变化)

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
filmId	string	影片id

- 返回参数

返回参数	参数类型	参数说明
-	promise	异步请求封装

- 调用示例

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

9. 整合后的影院列表

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

请求参数

请求参数	参数类型	参数说明
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

返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
cinemaId	string	影院Id
cinemaName	string	影院名字

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
cinemaId	string	影院Id

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
cinemaId	string	影院Id

• 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
cinemald	string	影院Id
filmId	string	影片Id

• 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
cinemald	string	影院Id
filmId	string	影片Id
date	string	日期

• 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
type	string	平台类型
paras	object	锁座参数

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
orderId	string	订单Id

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
type	string	平台类型
paras	object	下订单参数

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
orderId	string	订单Id
payType	string	支付类型
prizeIds	string	待定
redIds	string	待定

• 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
location	object	定位信息（可空）

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
mobile	string	登录需要的手机号码
password	string	登录需要的密码

返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
sessionId	string	用户登录标识

返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
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）

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
mobile	string	用户的手机号码
verifyCode	string	验证码
password	string	密码

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
mobile	string	用户的手机号码
verifyCode	string	验证码
password	string	新密码

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
val	string	关键字
lastKey	string	目前无用

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
cityId	string	城市Id
query	string	关键字
lastKey	string	目前无用

• 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

• 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

- 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

• 返回参数

返回参数	参数类型	参数说明
-	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	参数类型	参数说明
-	-	-

• 返回参数

返回参数	参数类型	参数说明
-	CityModel	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	参数类型	参数说明
-	-	-

返回参数

返回参数	参数类型	参数说明
-	CoordinateModel	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	参数类型	参数说明
-	baseUrl	string	网络请求基础地址
-	delegate	id < JNetworkDelegate >	网络请求代理



模型：CoordinateModel

- 描述

定位模型

- 属性

所属	参数	参数类型	参数说明
-	latitude	double	纬度
-	longitude	double	经度

