接口文档

**1**、balanceOf(address) 输入地址，可以获取该地址代币的余额接口

参数：address 用户的地址

返回值：

类型：int

说明：地址代币的余额

示例：

word\_cup.deployed().then(function(instance) {//

return instance.balanceOf("0x7be737Eaa8B85aBC0332fF596B403AcA368D0DaF");

}).then(function (res) {

console.log(Web3App.utils.toBN(res / 100000000).toString())

}).catch(function (err) {

console.log(err);

});

**2**、transfer( to, value) 用户调用transfer函数将自己的token转账给\_to地址，\_value为转账个数

参数:

to 转账到的地址

value 转账金额

示例：（接口中的from参数代表从哪个地址转账，这个就是发起转账的用户地址）

word\_cup.deployed().then(function(instance) { // ERC20 转账接口

return instance.transfer("0x7be737Eaa8B85aBC0332fF596B403AcA368D0DaF", '100000000', {from : "0x207e1452E3930e40E30881bfb9cae2aaa634739d"});

}).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err);

});

**3**、createGame（red, blue）创建比赛接口（需要设置红蓝敌对方）

参数：

red 红方

blue 蓝方

返回值：

类型: int

说明: 比赛id（后续的参加比赛，比赛设置结果，查询比赛信息都需要这个比赛id）

示例：

word\_cup.deployed().then(function(instance) {// 创建比赛接口

return instance.createGame('Russia', 'Saudi Arabia', {from : "0x207e1452E3930e40E30881bfb9cae2aaa634739d", gas: 900000});

}).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err);

});

**4**、getGame（gameid）获取比赛信息接口

参数：

gameid 比赛id

返回值：

类型 数组

说明：

第0个值 创建比赛人的地址

第1个值 比赛状态（0 进行中 1 已结束 2 已返还 只有0状态下才可以对比赛下注和设置比赛接口）

第2个值 红方下注人数

第3个值 蓝方下注人数

第4个值 总下注人数

第5个值 红方

第6个值 蓝方

第7个值 获胜方

第8个值 总奖池金额

示例：

[ '0x207e1452e3930e40e30881bfb9cae2aaa634739d',

BigNumber { s: 1, e: 0, c: [ 1 ] },

BigNumber { s: 1, e: 0, c: [ 1 ] },

BigNumber { s: 1, e: 0, c: [ 1 ] },

BigNumber { s: 1, e: 0, c: [ 2 ] },

'0x5275737369610000000000000000000000000000000000000000000000000000',

'0x5361756469204172616269610000000000000000000000000000000000000000',

'0x5275737369610000000000000000000000000000000000000000000000000000',

BigNumber { s: 1, e: 8, c: [ 200000000 ] } ]

调用示例：

word\_cup.deployed().then(function(instance) {

return instance.getGame(0);// 获取比赛详情接口

}).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err);

});

**5**、partIn(gameid, value, side) 下注接口

参数：

gameid 比赛id

value 下注金额

side 0 主场胜 1 主场平 2 主场负

示例：

word\_cup.deployed().then(function(instance) {//参加比赛接口

return instance.partIn(0, 100000000, 'Saudi Arabia', {from : "0x7be737Eaa8B85aBC0332fF596B403AcA368D0DaF", gas: 900000});

}).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err);

});

**6**、openGame(gameid, side) 比赛设置结果接口（这个接口只有创建比赛的人才能调用成功）

参数：

gameid 比赛id

side 0 主场胜 1 主场平 2 主场负

示例：

word\_cup.deployed().then(function(instance) {//比赛公布结果接口

return instance.openGame(1, 'Russia', {from : "0x207e1452E3930e40E30881bfb9cae2aaa634739d", gas: 900000});

}).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err);

console.log(1);

});

**7**、getPartIn() 获取参加的比赛id的接口

参数：无

返回值：

类型 ：数组

说明： 返回所有参加的比赛的id

示例：

word\_cup.deployed().then(function(instance) {

return instance.getPartIn{from : "0x207e1452E3930e40E30881bfb9cae2aaa634739d", gas: 900000});

}).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err);

console.log(1);

});

8、getCreate() 获取创建比赛的id接口

参数：无

返回值：

类型 ：数组

说明： 返回创建的比赛的id

示例：word\_cup.deployed().then(function(instance) {

return instance.getCreate({

from: "0x22bCcA213d04fDF51ff14915A8877172F0B3D8de"

}); // 获取比赛详情接口

}).then(function(res) {

console.log(res)

}).catch(function(err) {

console.log(err);

});