

Hyperchain 事件订阅/消息推送使用手册

作者：马晓敏

创建：2017-08-01

修改：2017-08-24

类别：使用手册

[Hyperchain 事件订阅/消息推送使用手册](#)

[一、功能概述](#)

[二、具体使用](#)

[2.1 订阅一个事件](#)

[2.2 消息推送格式](#)

[2.3 取消订阅](#)

[三、兼容JSONRPC接口](#)

[四、事件类型](#)

[4.1 Block事件](#)

[4.2 SystemStatus事件](#)

[4.3 Logs事件](#)

[五、接口概览](#)

[sub_subscribe](#)

[sub_unsubscribe](#)

[sub_getAllSubscription](#)

一、功能概述

Websocket事件订阅的消息格式与JSONRPC请求参数的格式一致，并且兼容JSONRPC接口。比如，订阅Block事件，当产生一个新区块的时候平台会向客户端推送通知。

二、具体使用

2.1 订阅一个事件

订阅一个事件并且返回订阅id：

```
# 事件订阅请求
> {"jsonrpc":"2.0","method": "sub_subscribe","id":
1,"namespace":"global","params":["block", false]}
# 返回值
> {
  "jsonrpc":"2.0",
  "namespace":"global",
  "id":1,
  "code":0,
  "message":"SUCCESS","result":"0xb539ffc3ff69acbl32a7ef15d47d4ebd"
}
```

- **method:** `sub_subscribe` 表明这是一个事件订阅的请求，这个字段是固定的。
- **params:** 数组的第一个元素必须为要订阅的事件的名称，如果要传入参数，则从数组第二个元素开始，这里的 `false` 表示只返回最新区块的hash，而不返回区块里的具体内容。

如果**params**第一个元素为空，将返回error:

```
# 事件订阅请求
>> {"jsonrpc":"2.0","method": "sub_subscribe","id":
1,"namespace":"global","params":[]}
# 返回值
<< {
  "jsonrpc":"2.0",
  "code":-32602,
  "message":"Please give a subscription name as the first param"
}
```

如果订阅的事件不存在，将返回error:

```
# 事件订阅请求
>> {"jsonrpc":"2.0","method": "sub_subscribe","id":
1,"namespace":"global","params":["helloworld"]}
# 返回值
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "id":1,
  "code":-32601,
  "message":"The method sub_helloworld does not exist/is not available"
}
```

2.2 消息推送格式

收到一个block事件的通知:

```
<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "result": {
    "event": "block",
    "subscription": "0xb539ffc3ff69acbf132a7ef15d47d4ebd",

    "data": "0x18fc1e0b7ce7235f0fab5c7b5a96519245c5ccd957fb9a6ddc2c7646cc95e30b"
  }
}
<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "result": {
    "event": "block",
    "subscription": "0xb539ffc3ff69acbf132a7ef15d47d4ebd",

    "data": "0xc7c777de90124e01df28935c8b2e35cb45278bb3673e36aad7e1f153b9ea0092"
  }
}
```

result:

- `event` 表明该通知对应 `block` 事件。
- `subscription` 表明该通知对应哪条订阅请求。
- `data` 平台推送给客户端的具体数据。block事件推送给客户端的是最新区块的哈希。

2.3 取消订阅

取消订阅请求:

```
# 取消订阅请求
>> {"jsonrpc": "2.0", "method": "sub_unsubscribe", "id":
1, "namespace": "global", "params": ["0xb539ffc3ff69acbf132a7ef15d47d4ebd"]}
# 返回值
<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "id": 1,
  "code": 0,
  "message": "SUCCESS",
  "result": true
}
```

至此，id为 `0xb539ffc3ff69acbf132a7ef15d47d4ebd` 的订阅已经取消，当再次取消订阅的时候，系统将返回error:

```
# 取消订阅请求
>> {"jsonrpc":"2.0","method": "sub_unsubscribe","id":
1,"namespace":"global","params":["0xb539ffc3ff69acb132a7ef15d47d4ebd"]}
# 返回值
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "id":1,
  "code":-32012,
  "message":"subscription not found"
}
```

三、兼容JSONRPC接口

例子：

```

# 获取最新区块请求
>> {"jsonrpc":"2.0","method":"block_latestBlock","params":[],"id":1}
# 返回值
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "id":1,
  "code":0,
  "message":"SUCCESS",
  "result":{
    "version":"1.3",
    "number":"0x1",

    "hash":"0xe2cf16a370f7b37ff642b572071fec67e3df7fc570a1edad804a2eef35aca467",

    "parentHash":"0x000000000000000000000000000000000000000000000000000000000000000000000000",
    "writeTime":1499065883616093925,
    "avgTime":"0x10",
    "txcounts":"0x1",

    "merkleRoot":"0x5114ce7b1d2f61780e021c0c4d2bd3292b709369269cd4197b563760ec6383e1",
    "transactions":[
      {
        "version":"1.3",

        "hash":"0xb1ece94abd486528a7d44218abd1e6e3044f5916da61f6ece8953bdb162d7a85",
        "blockNumber":"0x1",

        "blockHash":"0xe2cf16a370f7b37ff642b572071fec67e3df7fc570a1edad804a2eef35aca467",

        "txIndex":"0x0",
        "from":"0x17d806c92fa941b4b7a8ffffc58fa2f297a3bffc",
        "to":"0xaeccd2fd1118334402c5de1cb014a9c192c498df",
        "amount":"0x0",
        "timestamp":1499065882558000000,
        "nonce":4763941686685489,
        "executeTime":"0x10",
        "payload":"0x0a9ae69d"
      }
    ]
  }
}

```

四、事件类型

4.1 Block事件

事件描述

客户端可订阅block事件，当区块链产生一个新区块的时候，平台会向订阅者推送最新区块。

订阅接口

```
{ "jsonrpc": "2.0", "method": "sub_subscribe", "id": 1, "namespace": "global", "params": ["block", <param>] }
```

- 参数：<param> 是布尔类型，为true表示通知信息中包括最新区块详细信息，为false表示通知信息中只返回最新区块哈希。
- 返回值：订阅id。

例子1 订阅一个只返回最新区块哈希的block事件：

```
# 订阅一个事件
>> { "jsonrpc": "2.0", "method": "sub_subscribe", "id": 1, "namespace": "global", "params": ["block", false] }
<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "id": 1,
  "code": 0,
  "message": "SUCCESS",
  "result": "0xbba4bf8a0e4358618776ee9196aleda5"
}

# 通知
<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "result": {
    "event": "block",
    "subscription": "0xbba4bf8a0e4358618776ee9196aleda5",

    "data": "0xcc6d22af42e3cd0241f9f1a1a166764e642de2c6c2c86a5b422268c2e91f16e6"
  }
}
<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "result": {
    "event": "block",
    "subscription": "0xbba4bf8a0e4358618776ee9196aleda5",

    "data": "0x45cff716aa02dacffce2e50f218caeac5896912a873d85e5a1f8d211d34dac68"
  }
}
```

例子2 订阅一个返回最新区块详细信息的block事件：

订阅一个事件

```
>> {"jsonrpc":"2.0","method": "sub_subscribe","id":
1,"namespace":"global","params":["block", true]}
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "id":1,
  "code":0,
  "message":"SUCCESS",
  "result":"0xd0f98f277f33b1125b2a5f5ac2cbc5b"
}
```

通知

```
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "result":{
    "event":"block",
    "subscription":"0xd0f98f277f33b1125b2a5f5ac2cbc5b",
    "data":{
      "version":"1.3",
      "number":"0x1",

"hash":"0xcc6d22af42e3cd0241f9f1a1a166764e642de2c6c2c86a5b422268c2e91f16e6",

"parentHash":"0x0000000000000000000000000000000000000000000000000000000000000000
0000",
      "writeTime":1499157819652020682,
      "avgTime":"0x11",
      "txcounts":"0x1",

"merkleRoot":"0xe6d8cd299eed0bae53a2439d3e5a1fb85a47edc50161fc53774266d946a2
3b15"
    }
  }
}
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "result":{
    "event":"block",
    "subscription":"0xd0f98f277f33b1125b2a5f5ac2cbc5b",
    "data":{
      "version":"1.3",
      "number":"0x2",

"hash":"0x45cff716aa02dacffce2e50f218caeac5896912a873d85e5a1f8d211d34dac68",
```

```

"parentHash": "0xcc6d22af42e3cd0241f9f1a1a166764e642de2c6c2c86a5b422268c2e91f16e6",
  "writeTime": 1499157864811500662,
  "avgTime": "0x4",
  "txcounts": "0x1",

  "merkleRoot": "0xe6d8cd299eed0bae53a2439d3e5a1fb85a47edc50161fc53774266d946a23b15"
}
}
}

```

4.2 SystemStatus事件

事件描述

客户端可订阅感兴趣的平台systemStatus事件，当平台有异常抛出或者系统状态改变的时候，便会向订阅者主动推送消息。

订阅接口

```

{"jsonrpc": "2.0", "method": "sub_subscribe", "id": 1, "namespace": "global", "params": ["exception", <filterCriteria>]}

```

- 参数： <filterCriteria> 是一个JSON对象,用于用户指定exception的过滤条件：
 - modules：一维数组，表示要订阅哪些模块的异常信息，若为空，则表示订阅所有模块。
 - modules_exclude：一维数组，表示排除哪些模块的异常信息，若为空，则表示不排除。
 - subtypes：一维数组，表示要订阅模块下面的哪一类异常信息，若为空，则表示订阅所有类型。
 - subtypes_exclude：一维数组，表示要排除模块下面的哪一类异常信息，若为空，则表示不排除。
 - error_codes：一维数组，表示要订阅指定的具体哪一条异常信息，若为空，则表示订阅所有异常。
 - error_codes_exclude：一维数组，表示要排除指定的具体哪一条异常信息，若为空，则表示不排除。
- 返回值：
 - module：异常所属模块。
 - subType：异常所属类型。
 - errorCode：异常错误码。
 - message：具体异常信息。
 - date：异常抛出日期。

例子：


```

# 订阅一个事件
>> {"jsonrpc":"2.0","method": "sub_subscribe","id":
1,"namespace":"global","params":["exception", {}]}
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "id":1,
  "code":0,
  "message":"SUCCESS",
  "result":"0xf29d389449a2d82acfb471a0e24212c7"
}

# 通知
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "result":{
    "event":"exception",
    "subscription":"0xf29d389449a2d82acfb471a0e24212c7",
    "data":{
      "module":"executor",
      "subType":"viewchange",
      "errorCode":0,
      "message":"requried to reset status to 0",
      "date":"2017-07-05T10:21:55.593401475+08:00"
    }
  }
}
}

```

(后期需补充平台所有已知异常的列表.....)

4.3 Logs事件

事件描述

客户端可订阅感兴趣的虚拟机log事件，当hyperchain处理完一笔交易并在虚拟机运行过程中产生了log事件，便会向订阅者主动推送消息。

订阅接口

```

{"jsonrpc":"2.0","method": "sub_subscribe","id":
1,"namespace":"global","params":["logs", <filterCriteria>]}

```

- 参数： `<filterCriteria>` 是一个JSON对象,用于用户指定log的过滤条件：
 - `fromBlock`：起始区块号，为空则默认为没有区块下限限制。
 - `toBlock`：终止区块号，为空则默认为没有区块上限限制。
 - `addresses`：一维数组，地址列表，若为空，则默认接收所有来自任意一个合约的log事件；若不为空，则只接收来自地址列表中的合约的log事件。
 - `topics`：`common.Hash` 的二维数组，用于表示事件topic过滤条件。

其中topics一项较为复杂，是一个32字节hash的2维数组。若topics为空，则默认没有topic过滤条件；若不为空，则根据topic过滤条件对产生的log事件进行筛选。

一个实例的topics如下所示：

- []：表示没有topic过滤条件。
 - [A]：表示事件的第一个topic为A，事件的其余topic可以为任意值。
 - [null, B]：表示事件的第一个topic可以为任意值，第二个topic为B，剩余的topic为任意值。
 - [A, B]：表示事件的第一个topic为A，第二个topic为B，其余topic可以为任意值。
 - [[A, B], [A, B]]：表示事件的第一个topic为A或B，第二个topic为A或B
- 返回值：
 - `address`：产生事件的合约地址。
 - `topics`：一系列的topic。
 - `data`：数据段。
 - `blockNumber`：所属区块号。
 - `blockHash`：所属区块哈希。
 - `txHash`：所属交易哈希。
 - `txIndex`：所属交易在当前区块交易列表中的偏移量。
 - `index`：该日志在本条交易产生的所有日志中的偏移量。

例子：

```

# 订阅一个事件
>> {"jsonrpc":"2.0","method": "sub_subscribe","id":
1,"namespace":"global","params":["logs", {}]}
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "id":1,
  "code":0,
  "message":"SUCCESS",
  "result":"0xdd061810765e621fd93291a0146b2e10"
}

# 通知
<< {
  "jsonrpc":"2.0",
  "namespace":"global",
  "result":{
    "event":"logs",
    "subscription":"0xdd061810765e621fd93291a0146b2e10",
    "data":[
      {
        "address":"0x313bbf563991dc4c1be9d98a058a26108adfcf81",
        "topics":
[ "0x24abdb5865df5079dcc5ac590ff6f01d5c16edbc5fab4e195d9febd1114503da" ],

        "data":"0000000000000000000000000000000000000000000000000000000000000064",
        "blockNumber":4,

        "blockHash":"0xee93a66e170f2b20689cc05df27e290613da411c42a7bdfa951481c08fdef
b16",

        "txHash":"0xa676673a23f33a95a1a5960849ad780c5048dff76df961e9f78329b201670ae2
",
        "txIndex":0,
        "index":0
      }
    ]
  }
}

```

其他：

一个log事件的topic至多有4个，其中第一个topic的内容为产生该事件的函数签名，其余三个为用户自定义的topic内容，因此用户在订阅时，过滤条件不宜超过4个，且第一个topic筛选可以置空。

五、接口概览

sub_subscribe

订阅一个事件，例子见第二章或第四章。

sub_unsubscribe

退订一个事件，例子见第二章或第四章。

sub_getAllSubscription

获取当前已订阅的所有事件。

- 参数：无
- 返回值：对象数组。详见以下例子。

例子1:

```
>> {"jsonrpc": "2.0", "method": "sub_getAllSubscription", "id":
1, "namespace": "global", "params": []}

<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "id": 1,
  "code": 0,
  "message": "SUCCESS",
  "result": [
    {
      "subId": "0xc624794b6a328d430ab7160e3c2ec38",
      "event": "block"
    }, {
      "subId": "0xe3c86cf287a784fe852665700ba106da",
      "event": "block"
    }
  ]
}
```

例子2: 无订阅的事件

```
>> {"jsonrpc": "2.0", "method": "sub_getAllSubscription", "id":
1, "namespace": "global", "params": []}

<< {
  "jsonrpc": "2.0",
  "namespace": "global",
  "id": 1,
  "code": 0,
  "message": "SUCCESS",
  "result": []
}
```

