



# Moonbeam开发进阶课程

## xTokens Pallet和预编译

PureStake开发者关系团队



## 课程导航

xTokens Pallet  
概述

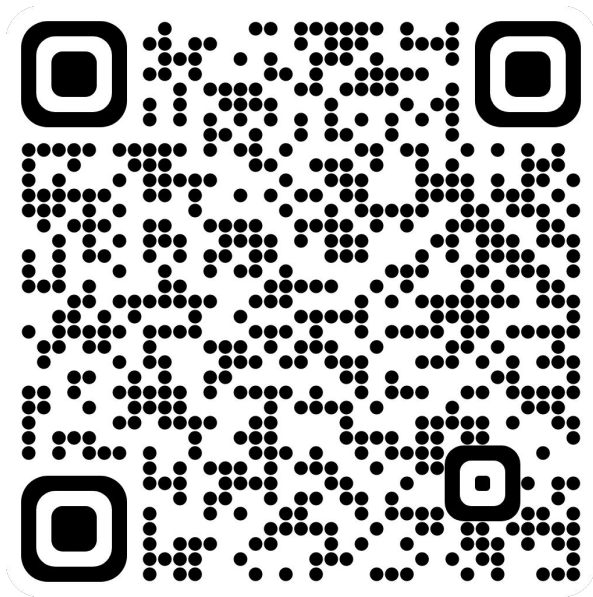
xTokens Pallet  
转帐示范

xTokens  
预编译

预编译  
调用示范

## 课程资源GitHub Repo

<https://github.com/PureStake/moonbuilders-academy/tree/main/chinese/advanced-course>





# xTokens Pallet概述

## xTokens Pallet概述

- Open-Runtime-Module-Library (ORML) 的一部分, 由Acala团队研发
- ORML是一个社区维护的 Substrate 运行时模块集合
- <https://github.com/open-web3-stack/open-runtime-module-library>



# xTokens Pallet

## 简化XCM代币转帐



**XCM通用性高 但数据结构复杂**

XCM信息格式通用性高，但是造成数据结构定义复杂，界面复杂



**xTokens是一种便利性库**

简化XCM信息发送和跨链代币转帐的Wrapper函数库



**可以通过Moonbeam EVM预编译接口使用**

适用于任何XC-20代币，可以智能合约调用

The screenshot displays the Moonbeam Alpha xTokens interface for submitting a transaction. The form is divided into several sections with numbered annotations:

- 1**: Account selection (ALICE).
- 2**: Currency selection (xTokens).
- 3**: Transfer parameters (currencyId, amount, dest, destWeight).
- 4**: OtherReserve selection (u128).
- 5**: OtherReserve value (42259045809535163221576417993425387648).
- 6**: Amount selection (u128 (Balance)).
- 7**: Destination selection (XcmVersionedMultiLocation).
- 8**: destWeight selection (u64 (Weight)).
- 9**: Submit Transaction button.

The interface also shows encoded call data and call details at the bottom.

## xTokens Pallet方法定义

# 通过资产ERC20预编译地址进行XCM转帐

```
function transfer(address currency_address, uint256 amount, Multilocation memory destination, uint64 weight ) external;
```

# 通过Multilocation进行XCM转帐

```
function transfer_multiasset(Multilocation memory asset, uint256 amount, Multilocation memory destination, uint64 weight) external;
```



# xTokens Pallet转帐示范



## 获取DEV测试代币

<https://docs.moonbeam.network/cn/builders/get-started/networks/moonbase/#get-tokens>



## 获取xcUNIT

<https://moonbeam-swap.netlify.app/#/swap>



## 连接Moonbase Alpha Relay中继链和创建账户

<https://polkadot.js.org/apps/?rpc=wss://frag-moonbase-relay-rpc-ws.g.moonbase.moonbeam.network#/accounts>



# Moonbase Alpha Relay中继链测试代币

代币名称: UNIT/xcUNIT

精度: 12位

Multilocation: { "parents": 1 }

资产

ID: 422590458095351632215764179

93425387648

Moonbase Alpha预编译地址

: 0xffffffff1fcacbd218edc0eba20fc23

08c778080

```
henryduong@WeiWorkMac xcmTools % ts-node calculateMultilocationInfo
.ts --network moonbase -a '{ "parents": 1 }'
2022-07-15 18:12:15      API/INIT: RPC methods not decorated: eth
_feeHistory, eth_maxPriorityFeePerGas, moon_isBlockFinalized, moon_
isTxFinalized
Storage Key 0x1da53b775b270400e7e61ed5cbc5a146ea70f53d5a3306ce02aaf
97049cf181a1f720ca3a567a7892f51ba4eabe649ccffffffff1fcacbd218edc0eb
a20fc2308c778080
Asset Address Precompile: 0xffffffff1fcacbd218edc0eba20fc2308c77808
0
Asset ID is 42259045809535163221576417993425387648

henryduong@WeiWorkMac xcmTools %
```

## 连接Moonbase Alpha测试网

<https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Fwss.api.moonbase.moonbeam.network#/accounts>



## Encoded Call Data

## 使用xTokens.transfer方法:

0x1e00018080778c30c20fa2ebc0ed18d2cbca1f0010a5d4e80000000000000000000000001010101003e1dc48e42109d95b71a9fb58208498  
1e81995e6e3bdca4ce3a2cb15d53e5bd5b00ca9a3b00000000

## 使用xTokens.transferMultiAsset方法:

0x1e010100010000070010a5d4e801010101003e1dc48e42109d95b71a9fb582084981e81995e6e3bdc4ce3a2cb15d53e5bd5b00ca9a3b0000000



# xTokens预编译

## 在Moonbeam EVM里调用xTokens功能

- ○ ○ ○ 便利的数据结构
  - ○ ○ Multilocation, Multiasset, Currency

- Multilocation, Multiasset, Currency

- 可以与Solidity智能合约交互

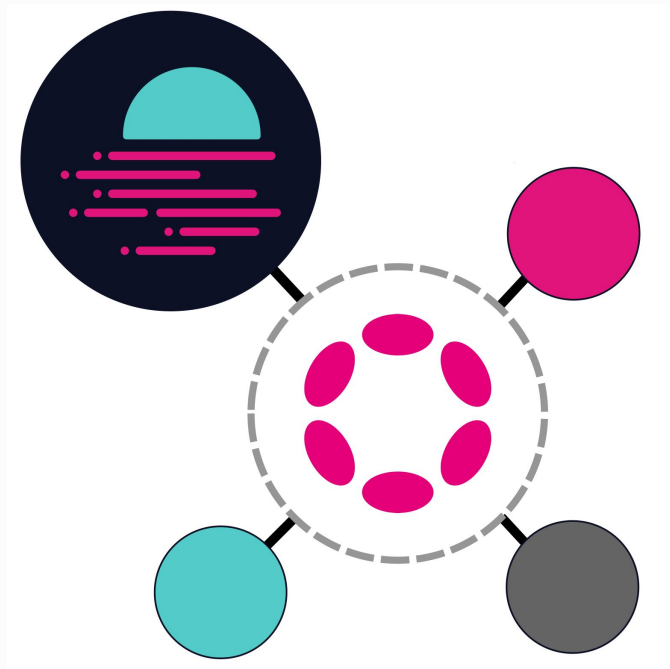
○○ xTokens预编译可以被智能合约直接调用

- 支持transfer和transferMultiasset等接口

### 接收Multilocation或ERC20地址为参数

○○○ xTokens预编译地址:

**XORKEYS**按端译地址：  
0x00000000000000000000000000000000000000804





# xTokens数据结构

## Multilocation定义:

```
struct Multilocation {  
    uint8 parents;  
    bytes [] interior;  
}
```

parents: [0, 1], 0为本地, 1为穿越中继链

## MultiAsset定义:

```
struct MultiAsset {  
    Multilocation location;  
    uint256 amount;  
}
```

## Currency定义:

```
struct Currency {  
    //ERC20合约地址  
    address currency_address;  
    uint256 amount;  
}
```

## Interior数组结构解码:

Array	Size	Interior Value
[]	0	Here
[XYZ]	1	X1
[XYZ, ABC]	2	X2
[XYZ, ... N]	N	XN

- 数组长度对应Interior的X1, X2...字段数量
- 空数组对应 Interior值: "Here"

# xTokens数据结构

Interior数组元素解码:

- 前两位字节为数据类型选择器(见右图), 后面字节代表字段内容
- 账户类型(AccountId32, AccountKey20)的最后两位字节代表网络类型(Network Any 用“00”)代表

例子:

- Parachain ID 2023 : "0x00+000007E7" (7E7是2023的16进制代表)
- Pallet Instance 3: "0x04+03"
- AccountKey20/以太坊格式地址  
0x578002f699722394afc52169069a1FfC98DA36f1 :  
"0x03+578002f699722394afc52169069a1FfC98DA36f1+00"

Byte Value	Selector	Data Type
0x00	Parachain	bytes4
0x01	AccountId32	bytes32
0x02	AccountIndex64	u64
0x03	AccountKey20	bytes20
0x04	PalletInstance	byte
0x05	GeneralIndex	u128
0x06	GeneralKey	bytes[]

# xTokens预编译示例

## 从Moonbase Alpha平行链转帐到中继链



### Transfer方法

XC-20地址, 数额, 目的地  
MultiLocation, 重量



### Transfer Multi Asset方法

资产MultiLocation, 数额, 目的地  
MultiLocation, 重量

#### Example, to Relay Chain

```
XC-20 Address =  
0xFfFfFfFf1FcaCBd218EDc0EbA20Fc2308C778080  
Amount (dec) = 100000000000  
Destination ML = [ 1,  
["0x019c6af76cd6513e44e421ab8dc8f9d86e102200eb2e55cbc41  
a2d5ce214ee254f00"] ]  
Weight = 4000000000
```

#### Example, to Relay Chain

```
Asset ML = [ 1, [] ]  
...
```



# xTokens预编译Remix示范

## xTokens预编译接口合约

<https://github.com/PureStake/moonbeam/blob/master/precompiles/xtokens/Xtokens.sol>



# 智能合约调用示例

```
// SPDX-License-Identifier: GPL-3.0-only
pragma solidity >=0.8.0;
import "./xtokens.sol";
import "./ERC20.sol";

contract xcUnitBridge {

    Xtokens public xTokens;
    IERC20 public xcUnit;
    address public constant xtokensPrecompileAddress = 0x00000000000000000000000000000000000000000000000000000000000000804;
    address public constant xcUnitERC20Address = 0xFfFfFfFf1FcCBd218EDc0EbA20Fc2308C778080;

    constructor() {
        // Initializes the xTokens precompile
        xTokens = Xtokens(xtokensPrecompileAddress);
        xcUnit = IERC20(xcUnitERC20Address);
    }

    function send_tokens(Xtokens.Multilocation memory destination, uint256 amount) external {
        //The user needs to approve the appropriate allowance separately
        xcUnit.transferFrom(msg.sender, address(this), amount);
        xTokens.transfer(xcUnitERC20Address, amount, destination, 4000000000);
    }
}
```