

如何扩展原生 substrate模块

吴旻哲

背景

- 原生的模块满足不了要求
- 很多情况下不需要拷贝黏贴代码
- 提高成熟代码的复用率，并减少BUG

怎么做

- 创建一个新的srml模块作为原模块的代理，外部访问时使用代理模块。
 - <https://substrate.dev/docs/en/tutorials/creating-a-runtime-module>
- 在代理模块中添加原模块方法，在这个方法中添加扩展代码并调用原生模块的相应方法。（扩展decl_module!中的dispatchable方法）
- 嵌入到系统
- 使用Polkadot UI测试这个新模块

例子

- 扩展balances模块
 - 对转账做更多的限制
- 所以，创建一个balances-ext对balances模块进行扩展

如何访问其他模块

- 需要直接访问balances和sudo模块，所以
- Cargo.toml添加这两个模块的依赖（版本一致）
- 模块Trait添加其他模块的Trait进行约束：

```
pub trait Trait: balances::Trait + sudo::Trait {  
}
```

```
<balances::Module<T>>::transfer(new_origin, dest, value);
```

transfer方法

```
pub fn transfer(  
  origin,  
  dest: <T::Lookup as StaticLookup>::Source,  
  #[compact] value: T::Balance  
) {  
  // 必须是sudo用户才能执行transfer  
  let sender = ensure_signed(origin)?;  
  ensure!(sender == <sudo::Module<T>>::key(), "Sender must be the Sudo key to transfer");  
  let new_origin = system::RawOrigin::Signed(sender).into();  
  
  // 每次转账额度不能大于100万  
  let value_u64 = <T::Balance as As<u64>>::as_(value);  
  ensure!(value_u64 < 1000000, "The amount of transfer is too large");  
  
  // 调用原生模块的方法  
  <balances::Module<T>>::transfer(new_origin, dest, value);  
}
```

嵌入到系统

- runtime
 - Cargo.toml

```
srml-balances = { package = "srml-balances", git = 'https://github.com/wum'
balances = { package = "srml-balances-ext", path = '../srml/balances-ext',
```

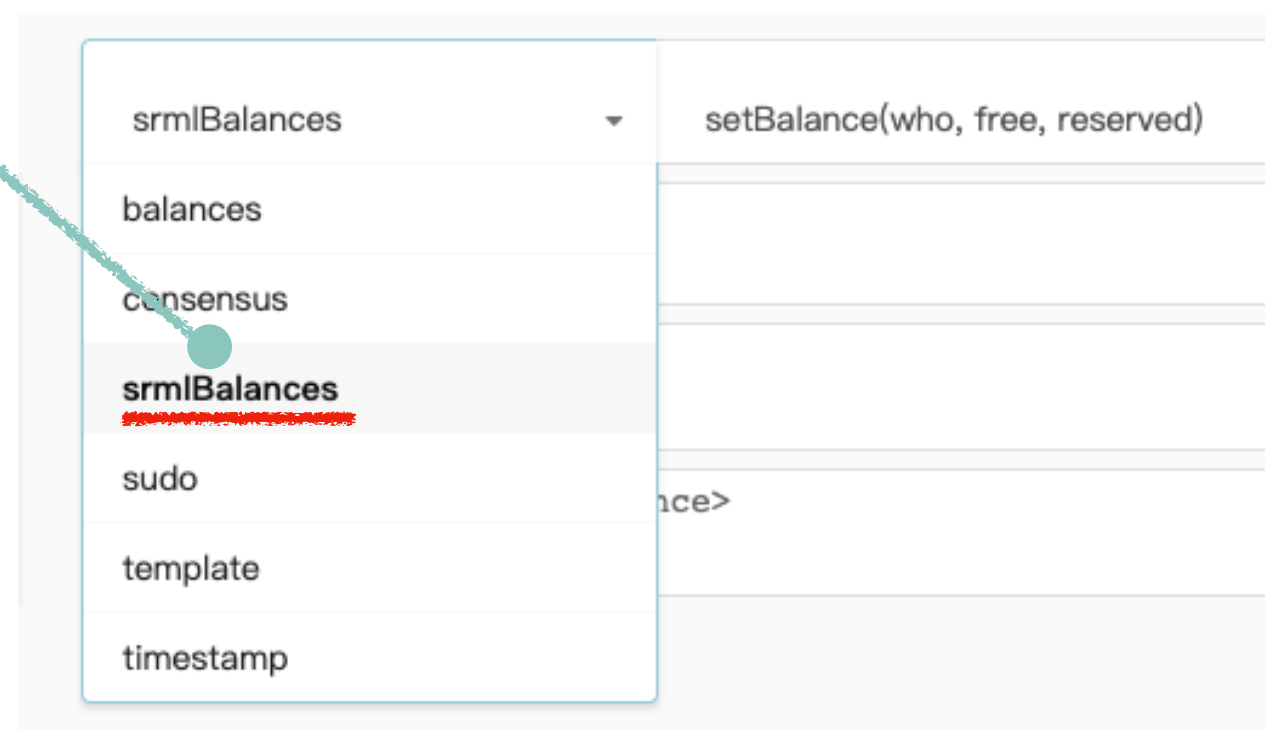
- lib.rs

```
Balances: srml_balances::{Module, Call, Storage, Config<T>, Event<T>},
BalancesExt: balances::{Module, Call},
```

- chain_spec.rs

```
srml_balances: Some(BalancesConfig {
  transaction_base_fee: 1,
  transaction_byte_fee: 0,
```

Multi Instance情况下?



测试

- 运行起来
 - `./scripts/build.sh`
 - `cargo build`
 - `./target/debug/testchain purge-chain -dev`
 - `./target/debug/testchain -dev`
- <https://polkadot.js.org/apps>



using the selected account
ALICE

5GrwvaEF5zXb26Fz9rcQpDWS57CtERHpNehXCPcNoHGKutQY ▾

with an account balance

0

Unit

with a transaction nonce

0

submit the following extrinsic
balances ▾

transfer(dest, value)

transfer ▾



dest: Address
BOB

transfer(dest, value)

transfer

5FHnew46xGxgs5m01ve04sBTyGBzmst0spzC9Z0hJjM694ty ▾

value: Compact<Balance>

0

Unit ▾

Submit Inherent

or

Submit Transaction

⋮ Explorer

👤 Accounts

📁 Address book

📌 Transfer

📊 Chain state

🔄 Extrinsic

🔒 Sudo

selected state query ?
srmlBalances ▾

creationFee(): Balance

The fee required to

creationFee(): Balance

The fee required to

existentialDeposit(): Balance

The minimum amount requ

freeBalance(AccountId): Balance

The 'free' balance

locks(AccountId): Vec<BalanceLock>


Any liquidity locks on som



reservedBalance(AccountId): Balance

The amount of the balanc

totalIssuance(): Balance

The total units i



	<div>balances.transfer finalized</div> 
	<div>system.ExtrinsicFailed extrinsic event</div> 

b26Fz9rcQpDWS57CtERHpNehXCPcNoHGKutQY ▼

```
2019-08-01 14:50:30 Idle (0 peers), best: #45 (0x913d...4e
2019-08-01 14:50:30 Prepared block for proposing at 46 [
13d...4efe; extrinsics: [0x6ff9...cf1e, 0x7847...c087]]
2019-08-01 14:50:30 Pre-sealed block for proposal at 46.
xde1eacd9bc3b856edb3950d3854a1a18d0c5de4563ec0947ee1cca3
Runtime: The amount of transfer is too large
2019-08-01 14:50:30 Imported #46 (0x864f...808d)
2019-08-01 14:50:35 Idle (0 peers), best: #46 (0x864f...80
2019-08-01 14:50:40 Starting consensus session on top of
```

局限

- 设计上没有为扩展作深入考虑
- 非pub的方法的扩展需要特殊方式

```
// 调用原生模块的方法  
<balances::Module<T>>::transfer(new_origin, dest, value);  
// 如果原生模块的方法不是pub的，必须使用如下方法进行方法  
balances::Call::<T>::transfer(new_origin, dest, value).dispatch(origin);
```

- sudo方法怎么办

```

1 impl<T: Trait<I>, I: Instance> ::srml_support::dispatch::Dispatchable for Call<T, I> {
2     type Trait = T;
3     type Origin = T::Origin;
4     fn dispatch(self, _origin: Self::Origin) -> ::srml_support::dispatch::Result {
5         match self {
6             Call::transfer(dest, value) => <Module<T, I>>::transfer(_origin, dest, value),
7             Call::set_balance(who, free, reserved) => {
8                 system::ensure_root(_origin)?;
9                 <Module<T, I>>::set_balance(who, free, reserved)
10            }
11            Call::__PhantomItem(_, _) => ::std::rt::begin_panic_fmt(
12                &::std::fmt::Arguments::new_v1(
13                    &["internal error: entered unreachable code: "],
14                    &match (&"__PhantomItem should never be used.",) {
15                        (arg0,) => [::std::fmt::ArgumentV1::new(arg0, ::std::fmt::Display::fmt)],
16                    },
17                ),
18                &("srml/balances/src/lib.rs", 361u32, 1u32),
19            ),
20        }
21    }
22 }

```

Tip: 查看宏展开后代码的简便方法

```
cargo expand | vi
```

```
:set ft=rust
```

资源

- <https://www.shawntabrizi.com/substrate/extending-substrate-runtime-modules/>
- <https://github.com/wuminzhe/substrate-tools>

