

# **BSN-DDC 基础网络 DDC SDK 详细设计**

V2.0

北京红枣科技有限公司

2021 年 12 月

## 修改记录

日期	版本	修改说明	修改者
2021.12.9	V1.0	版本初始化	
2021.12.11	V1.0	1. 补充了调用时序图 2. 添加了交易查询、区块查询章节	
2021.12.15	V1.0	1. 添加了签名事件和数据解析章节 2. 细化了部分方法的核心逻辑说明	
2021.12.27	V1.0	1. 方法的调用者进行了阶定 2. 1155 的销毁和批量销毁调用者、入参以及核心逻辑进行了更新	
2022.1.6	V1.0	1. 附录针对武汉链的区块信息示例、交易回执信息示例以及交易信息示例进行了补充	
2022.1.10	V1.0	1. 721 添加安全生成方法 2. 1155 生成和批量生成添加附加数据参数 3. 数据解析部分 721 添加了安全生成的部分	
2022.1.11	V1.0	1. 更新 1155 批量生成和批量查询对应的 Map 类型	
2022.1.11	V1.0	1. 3.2.1.1 和 3.2.1.2 更新; 2. 3.2.2.6 和 3.2.2.7 更新; 3. 3.2.8.1.1 更新;	
2022.1.14	V1.0	1. 权限管理、费用管理、721 以及 1155 所有方法添加 sender 参数(注: 此参数用于签名事件中传递给链服务或业务系统, 由业务系统针对所传的 sender 指定账户进行交易签名)	

2022.1.17	V1.0	<ol style="list-style-type: none"> <li>1. 1155 生成和批量生成定义成了安全生成和批量安全生成，方法名也进行了调整。</li> <li>2. 2.3 节添加了接入 KEY 的参数说明</li> <li>3. 添加了 4.4 章节-离线生成账户</li> </ol>	
2022.1.19	V1.0	<ol style="list-style-type: none"> <li>1. BSN-DDC-权限管理添加跨平台授权。</li> <li>2. 721 和 1155 添加 URI 设置方法。</li> <li>3. 数据解析添加对跨平台授权以及 721 和 1155 对应的 URI 设置数据解析处理。</li> <li>4. 1155 的销毁和批量销毁添加 DDC 授权者也可以调用，以及 1155 的 URI 设置添加 DDC 的授权者也可以调用，数据解析部分，针对 URI 设置的事件解析添加了一个字段。</li> </ol>	
2022.1.20	V1.0	<ol style="list-style-type: none"> <li>1. DDC 业务费扣除事件通知添加 ddclid 字段。</li> </ol>	
2022.1.21	V1.0	<ol style="list-style-type: none"> <li>1. 添加 4.5 接入 URL 设置方法和 4.6 接入 Key 设置方法。</li> </ol>	
2022.1.23	V1.1	<ol style="list-style-type: none"> <li>1. 删除了 DID 章节。</li> <li>2. 权限管理添加了平台方批量添加账户、平台方添加链账户开关设置、平台方添加链账户开关查询、运营方批量添加链账户方法。</li> <li>3. 费用管理添加了批量充值以及批量链账户余额查询方法。</li> <li>4. 数据解析添加了添加账户开关设置、批量添加账户、批量充值方法。</li> </ol>	

		<p>5. 所有核心逻辑关于账户地址的检查描述进行了完善。</p> <p>6. 所有的合约方法对应数据上链部分都将返回值进行了删除，与合约部分保持了统一。</p> <p>7. 721 和 1155 添加最新 DDCID 查询接口。</p>	
2022.2.28	V1.1	1. 对应合约所有的查询类方法加上了 view 关键字。	
2022.3.3	V1.1	1. 权限管理添加了同步平台方 DID 方法以及数据解析部分补充了事件解析。	
2022.03.07	V1.1	<p>1. 721 的生成、安全生成以及 1155 的安全生成以及批量安全生成出掉了对 DDCURI 参数检验，其中 1155 的批量安全生成的参数名有 ddclInfo 换成了 ddcs，并对逻辑进行了完善。</p> <p>2. 数据解析中部分添加账户和批量添加账户中合约事件参数和合约保持一致。</p>	
2022.05.07	V1.1	1. 附录章节添加了 GAS 查询接口；	
2022.05.07	V2.0	<p>1. BSN-DDC-跨链应用章节以及对应的数据解析部分；</p> <p>2. 721 和 1155 部分数据解析添加了跨链锁定和解锁处理；</p>	
2022.07.15	V2.0	<p>1. DDC 跨链回滚更新成跨链通知以及相应的业务处理；</p> <p>2. 数据解析部分跨链应用部分更新了 DDC 跨链流转/生成和 DDC 跨链通</p>	

		知。	
2022.8.03	V2.0	<p>1. 721 添加了批量生成、批量安全生成、元交易生成、元交易安全生成、元交易批量生成、元交易批量安全生成、元交易转移、元交易安全转移、元交易销毁以及 Nonce 查询接口，数据解析部分同步添加了相应的数据解析章节；</p> <p>2. 1155 添加了元交易安全生成、元交易批量安全生成、元交易安全转移、元交易批量安全转移、元交易销毁、元交易批量销毁以及 Nonce 查询接口，数据解析部分同步添加了相应的数据解析章节；</p> <p>3. 计费 and 权限添加了对启用批量开关设置接口；</p> <p>4. 1155 添加了同步拥有者接口；</p> <p>5. 跨链应用对应的跨链流转接口去掉了数量参数；</p>	
2023.4.06	V2.0	1. 增加开放联盟链跨链应用	

## 目录

修改记录 .....	2
1. 编写目的 .....	7
2. 整体设计 .....	7
2.1 调用时序图 .....	7
2.2 开发语言标准 .....	7
2.3 参数格式标准 .....	7
3. 功能设计 .....	8
3.1 DDC .....	8

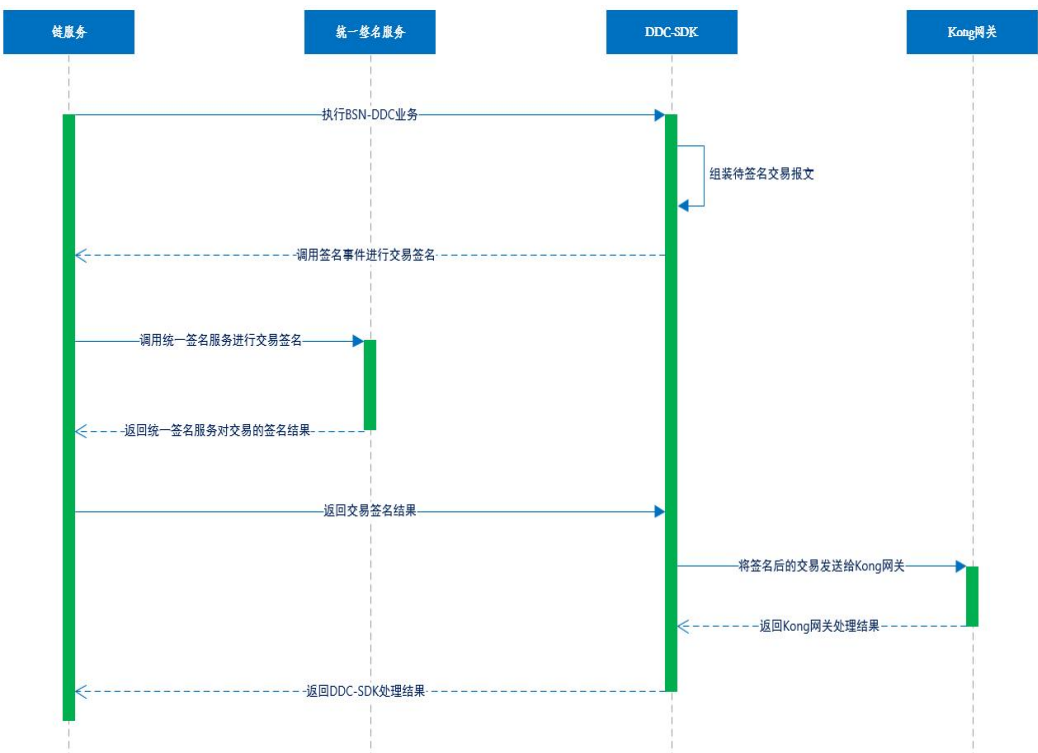
3.1.1	BSN-DDC-权限管理 .....	8
3.1.2	BSN-DDC-费用管理 .....	22
3.1.3	BSN-DDC-721 .....	32
3.1.4	BSN-DDC-1155 .....	61
3.1.5	BSN-DDC-跨链应用 .....	87
3.1.6	BSN-DDC-开放联盟链跨链应用 .....	90
3.1.7	BSN-DDC-交易查询 .....	93
3.1.8	BSN-DDC-区块查询 .....	95
3.1.9	BSN-DDC-签名事件 .....	96
3.1.10	BSN-DDC-数据解析 .....	97
4.	附录 .....	128
4.1	区块信息示例 .....	128
4.1.1	泰安链 .....	128
4.1.2	武汉链 .....	130
4.1.3	文昌链 .....	132
4.2	交易回执信息示例 .....	133
4.2.1	泰安链 .....	133
4.2.2	武汉链 .....	135
4.2.3	文昌链 .....	137
4.3	交易信息示例 .....	139
4.3.1	泰安链 .....	139
4.3.2	武汉链 .....	140
4.3.3	文昌链 .....	140
4.4	离线生成账户 .....	141
4.4.1	功能介绍 .....	141
4.4.2	API 定义 .....	141
4.5	接入 Url 设置 .....	142
4.5.1	功能介绍 .....	142
4.5.2	API 定义 .....	142
4.6	接入 Key 设置 .....	143
4.6.1	功能介绍 .....	143
4.6.2	API 定义 .....	143
4.7	GAS 查询 .....	143
4.7.1	功能介绍 .....	143
4.7.2	API 定义 .....	143

# 1. 编写目的

为了让运营方或各平台方对 DDC-SDK 整体设计有一个全面详细的了解，同时为项目的开发、测试、验证、交付等环节提供原始依据以及开发指导，特此整理 DDC-SDK 整体设计规范方案说明文档。

# 2. 整体设计

## 2.1 调用时序图



## 2.2 开发语言标准

目前使用 Java 语言开发 SDK。

## 2.3 参数格式标准

□ 时间

格式为 yyyy-MM-dd HH:mm:ss 形式的字符串，例如: 2021-05-25 12:30:59 表示 2021 年 5 月 25 日 12 时 30 分 59 秒。

#### ❑ 返回异常

当 SDK 处理功能逻辑出错时，会抛出相应的运行时异常，包含具体的错误信息。

#### ❑ 接入 Key

如果在 DDC 门户创建项目时启用了项目 KEY 值，则在做 DDC 业务的时候，请求网关时需要附加上 KEY 值参数，此参数通过 Header 进行传递(注：KEY 在设置的时候 KEY 用“x-api-key”，值根据实际情况填写)。

## 3. 功能设计

### 3.1 DDC

所有的 BSN-DDC 方法都需要调用签名事件对待签名交易进行签名，签名事件必须由业务调用者服务进行注册并实现交易签名的业务逻辑。

#### 3.1.1 BSN-DDC-权限管理

对账户进行管理，包含了账户的添加、删除、查询及更新账户状态的操作。

##### 3.1.1.1 平台方添加账户

###### 3.1.1.1.1 功能介绍

平台方可以通过调用该方法进行 DDC 账户信息的创建，上级角色可进行下级角色账户的操作，平台方通过该方法只能添加终端账户。



3.1.1.1.2 API 定义

- 方法定义：String addAccountByPlatform(String sender,String account,String accountName, String accountDID);
- 合约方法：addAccountByPlatform(address account,AccountInfo accountInfo);
- 调用者：平台方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
  2. 检查 account 为标准 address 格式，并且不能为 0 地址；
  3. 检查 accountName 不为空；
  4. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
账户地址	account	String	是	DDC 链账户地址
账户名称	accountName	String	是	DDC 账户对应的账户名称
账户 DID	accountDID	String	否	DDC 账户对应的 DID 信息

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.1.2 平台方批量添加账户

#### 3.1.1.2.1 功能介绍

平台方可以通过调用该方法进行 DDC 链账户信息的批量创建，上级角色可进行下级角色账户的操作，平台方通过该方法只能添加终端账户。

#### 3.1.1.2.2 API 定义

➤ 方法定义：String addBatchAccountByPlatform(String sender,List<AccountInfo> accounts);

➤ 合约方法：addBatchAccountByPlatform(address[] memory accounts,string[] memory accountNames,string[] memory accountDIDs);

➤ 调用者：平台方；

- 核心逻辑：
1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
  2. 检查 accounts 集合大小必须大于 0；
  3. 循环 accounts 集合，并根据索引检查 account 为标准 address 格式且不能为 0 地址以及 accountName 不为空；
  4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
账户信息列表	accounts	List<AccountInfo>	是	
AccountInfo				
账户地址	account	String	是	DDC 链账户

				地址
账户名称	accountName	String	是	DDC 账户对应的账户名称
账户 DID	accountDID	String	否	DDC 账户对应的 DID 信息

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.1.3 平台方添加链账户开关设置

#### 3.1.1.3.1 功能介绍

运营方可以通过调用该方法设置平台方添加链账户开关。

#### 3.1.1.3.2 API 定义

➤ 方法定义：String setSwitcherStateOfPlatform(String sender, Boolean isOpen);

➤ 合约方法：setSwitcherStateOfPlatform(bool isOpen);

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
2. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址

开关标识	isOpen	Boolean	是	
------	--------	---------	---	--

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.1.4 平台方添加链账户开关查询

#### 3.1.1.4.1 功能介绍

运营方可以通过调用该方法查询平台方添加链账户开关状态。

#### 3.1.1.4.2 API 定义

- 方法定义：Boolean switcherStateOfPlatform();
- 合约方法：switcherStateOfPlatform() view returns (bool);
- 调用者：运营方；
- 核心逻辑：无；
- 输入参数：

字段名	字段	类型	必传	备注

➤ 输出参数：

字段名	字段	类型	必传	备注
开关状态		Boolean	是	

### 3.1.1.5 运营方添加账户

#### 3.1.1.5.1 功能介绍

运营方可以通过调用该方法直接对平台方或平台方的终端用户进行创建。

#### 3.1.1.5.2 API 定义

- 方法定义：String addAccountByOperator(String sender,String account,String accountName,String accountDID,String leaderDID);
- 合约方法：addAccountByOperator(address account,AccountInfo accountInfo);
- 调用者：运营方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
  2. 检查 account 为标准 address 格式，并且不能为 0 地址；
  3. 检查 accountName 不为空；
  4. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
账户地址	account	String	是	DDC 链账户地址
账户名称	accountName	String	是	DDC 账户对应的账户名称
账户 DID	accountDID	String	否	DDC 账户对

				应的 DID 信息
平台方 DID	leaderDID	String	是	该普通账户对应的上级账户的 DID

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.1.6 运营方批量添加账户

#### 3.1.1.6.1 功能介绍

运营方可以通过调用该方法直接对平台方或平台方的终端用户进行批量创建。

#### 3.1.1.6.2 API 定义

➤ 方法定义：String addBatchAccountByOperator(String sender,List<AccountInfo> accounts);

➤ 合约方法：addBatchAccountByOperator(address[] memory accounts,string[] memory accountNames,string[] memory accountDIDs,string[] memory leaderDIDs);

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
2. 检查 accounts 集合大小必须大于 0；
3. 循环 accounts 集合，并根据索引检查 account 为标准 address 格式且不能为 0 地址以及 accountName 不为空；
4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
账户信息列表	accounts	List<AccountInfo>	是	
AccountInfo				
账户地址	account	String	是	DDC 链 账户地址
账户名称	accountName	String	是	DDC 账 户对应的账户名称
账户 DID	accountDID	String	否	DDC 账 户对应的 DID 信息
平台方 DID	leaderDID	String	是	该普通账户对应的上级账户的 DID

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.1.7 查询账户

#### 3.1.1.7.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法进行 DDC 账户信息的查询。

#### 3.1.1.7.2 API 定义

➤ 方法定义：AccountInfo getAccount(String account);

➤ 合约方法：getAccount(address account) view returns (AccountInfo);

➤ 调用者：运营方、平台方以及终端用户；

➤ 核心逻辑：

1. 检查 account 为标准 address 格式，并且不能为 0 地址；
2. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
账户地址	account	String	是	DDC 用户链账户地址

➤ 输出参数：

字段名	字段	类型	必传	备注
账户 DID	accountDID	String	否	DDC 账户对应的 DID 信息（普通用户可为空）
账户名称	accountName	String	是	DDC 账户对应的账户名称
账户角色	accountRole	enum	是	DDC 账户对应的身份信息。值包含： 0. Operator（运营方） 1.PlatformManager（平台方） 2.Consumer（用户方）
账户上级管理者	leaderDID	String	是	DDC 账户对应的上级管理员，账户角色为 Consumer 时必填。对于普通用户 Consumer 该值为平台管理者 PlatformManager



平台管理账户状态	platformState	enum	是	DDC 账户对应的当前账户状态（仅平台方可操作该状态）。值包含： 0.Frozen（冻结状态，无法进行 DDC 相关操作）  1.Active（活跃状态，可进行 DDC 相关操作）
运营管理账户状态	operatorState	enum	是	DDC 账户对应的当前账户状态（仅运营方可操作该状态）。值包含： 0.Frozen（冻结状态，无法进行 DDC 相关操作）  1.Active（活跃状态，可进行 DDC 相关操作）
冗余字段	field	String	否	冗余字段

### 3.1.1.8 更新账户状态

#### 3.1.1.8.1 功能介绍

运营方或平台方可以通过调用该方法对终端用户进行 DDC 账户信息状态的更改。

#### 3.1.1.8.2 API 定义

- 方法定义：String updateAccState(String sender,String account,State state,bool changePlatformState);
- 合约方法：updateAccountState(address account,State state,bool

changePlatformState);

➤ 调用者：运营方、平台方；

➤ 核心逻辑：

- 1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
- 2. 检查 account 为标准 address 格式，并且不能为 0 地址；
- 3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
账户地址	account	String	是	DDC 用户链 账户地址
状态	state	enum	是	1.Frozen（冻 结状态,无法 进行 DDC 相 关操作）  2.Active（活 跃状态,可进 行 DDC 相关 操作）
修改平台 方状态标 识	changePlatformState	Boolean	否	仅运营方调 用有效。  当调用者为 运营方且该 字段为 true 时,运营方可 修改某一账 户的 platformState 为 state

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.1.9 跨平台授权

3.1.1.9.1 功能介绍

运营方可以通过调用该方法对 DDC 的跨平台操作进行授权。

3.1.1.9.2 API 定义

➤ 方法定义：String crossPlatformApproval(String sender,String from,String to,bool approved);

➤ 合约方法：crossPlatformApproval(address from,address to,bool approved);

➤ 调用者：运营方；

- 核心逻辑：
- 1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
  - 2. 检查 from 为标准 address 格式，并且不能为 0 地址；
  - 3. 检查 to 为标准 address 格式，并且不能为 0 地址；
  - 4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
授权者	from	String	是	
接收者	to	String	是	
授权标识	approved	Boolean	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.1.10 同步平台方 DID

#### 3.1.1.10.1 功能介绍

运营方通过调用该API接口将旧平台方链账户所对应的DID同步到链上。

#### 3.1.1.10.2 API 定义

- 方法定义：String syncPlatformDID(String sender,List<String> dids);
- 合约方法：syncPlatformDID(string[] memory dids);
- 调用者：运营方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
  2. 循环 dids 集合，并根据索引检查平台方 DID 不为空；
  3. 检查签名事件是否被注册；

- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
平台方集合	dids	List<String>	是	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.1.11 启用批量设置

3.1.1.11.1 功能介绍

运营方通过调用该 API 接口对启用批量开关进行设置。

3.1.1.11.2 API 定义

- 方法定义: String setSwitcherStateOfBatch(String sender,Boolean isOpen);
- 合约方法: setSwitcherStateOfBatch(bool isOpen);
- 调用者: 运营方;
- 核心逻辑:
  - 1. 检查 sender 为标准 address 格式, 并且不能为 0 地址;
  - 2. 检查签名事件是否被注册;

➤ 输入参数:

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
开关标识	isOpen	Boolean	是	

➤ 输出参数:

字段名	字段	类型	必传	备注
		String	是	交易哈希

## 3.1.2 BSN-DDC-费用管理

### 3.1.2.1 充值

#### 3.1.2.1.1 功能介绍

运营方或平台方可以通过调用该方法为所属同一方的同一级别账户或者下级账户进行充值；

#### 3.1.2.1.2 API 定义

➤ 方法定义：String recharge(String sender,String to,BigInteger amount);

➤ 合约方法： recharge(address to,uint256 value) view returns (bool success);

➤ 调用者：运营方、平台方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，并且不能为 0 地址；
2. 检查 to 为标准 address 格式，并且不能为 0 地址；
3. 检查 amount 必须大于零；
4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
用户账户地址	to	String	是	充值账户的地址
金额	amount	BigInteger	是	转移金额

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.2.2 批量充值

#### 3.1.2.2.1 功能介绍

运营方或平台方可以通过调用该方法为所属同一方的同一级别账户或者下级账户进行批量充值;

#### 3.1.2.2.2 API 定义

➤ 方法定义: String rechargeBatch(String sender,Multimap<String,BigInteger> accounts);

➤ 合约方法: rechargeBatch(address[] memory toList,uint256[] memory amounts);

➤ 调用者: 运营方、平台方;

- 核心逻辑:
1. 检查 sender 为标准 address 格式, 并且不能为 0 地址;
  2. 检查 accounts 集合大小必须大于 0;
  3. 循环 accounts 集合, 并根据索引检查 account 为标准 address 格式且不能为 0 地址以及金额必须大于 0;
  4. 检查签名事件是否被注册;

➤ 输入参数:

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
账户充值列表	list	Multimap<String,	是	

		BigInteger>		
--	--	-------------	--	--

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.2.3 链账户余额查询

#### 3.1.2.3.1 功能介绍

运营方、平台方或终端用户可以调用方法查询指定账户的余额。

#### 3.1.2.3.2 API 定义

- 方法定义：BigInteger balanceOf(String accAddr);
- 合约方法：balanceOf(address owner) view returns (uint256 balance);
- 调用者：运营方、平台方或终端用户；
- 核心逻辑：
  1. 检查 accAddr 为标准 address 地址，并且不能为 0 地址；
  2. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
账户地址	accAddr	String	是	查询的账户地址

➤ 输出参数：

字段名	字段	类型	必传	备注
业务费余额	amount	BigInteger	是	账户所对应的业务费余额



### 3.1.2.4 批量链账户余额查询

#### 3.1.2.4.1 功能介绍

运营方、平台方或终端用户可以调用方法批量查询账户的余额。

#### 3.1.2.4.2 API 定义

➤ 方法定义：List<BigInteger> balanceOfBatch(List<String> accAddrs);

➤ 合约方法：balanceOfBatch(address[] memory accAddrs) view returns (uint256[]);

➤ 调用者：运营方、平台方或终端用户；

➤ 核心逻辑：

1. 检查 accAddrs 集合大小必须大于 0；
2. 循环 accAddrs 集合，并根据索引检查 account 为标准 address 格式且不能为 0 地址；
3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
账户地址集合	accAddrs	List<String>	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
账户余额集合		List<BigInteger>	是	

### 3.1.2.5 DDC 计费规则查询

#### 3.1.2.5.1 功能介绍

运营方、平台方或终端用户可以通过调用该方法查询指定的 DDC 业务合约的方法所对应的调用业务费用。

#### 3.1.2.5.2 API 定义

- 方法定义： `BigInteger queryFee(String ddcAddr,String sig);`
- 合约方法： `queryFee(address ddcAddr,bytes4 sig) view returns (uint amount);`
- 调用者：运营方、平台方，终端用户；
- 核心逻辑：
  1. 检查 ddcAddr 为标准 address 格式，并且不是 0 地址；
  2. 检查 sig 是有效的 4 位 Byte 的哈希格式字符串；
  3. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
业务合约地址	ddcAddr	String	是	DDC 业务合约地址
方法 ID	sig	String	是	Hex 格式的合约方法 ID，例如： 0x42966c68

- 输出参数：

字段名	字段	类型	必传	备注
业务费	amount	BigInteger	是	查询的 DDC 合约业务费

### 3.1.2.6 运营账户充值

#### 3.1.2.6.1 功能介绍

运营方可以通过调用该方法为自己的账户增加业务费。

#### 3.1.2.6.2 API 定义

- 方法定义：String selfRecharge(String sender,BigInteger amount);
- 合约方法：selfRecharge(uint amount);
- 调用者：运营方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 amount 必须大于零；
  3. 检查事件是否被注册；

- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
业务费	amount	BigInteger	是	对运营方账户进行充值的业务费

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.2.7 设置 DDC 计费规则

#### 3.1.2.7.1 功能介绍

运营方可以通过调用该方法设置指定的 DDC 主合约的方法调用费用。

#### 3.1.2.7.2 API 定义

➤ 方法定义：`String setFee(String sender,String ddcAddr,String sig,BigInteger amount);`

➤ 合约方法：`setFee(address ddcAddr,byte4 sig,uint32 amount);`

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查 ddcAddr 为标准 address 格式，且不能为 0 地址；
3. 检查 sig 是有效的 4 位 Byte 的哈希格式字符串；
4. 检查 amount 必须大于等于零；
5. 检查事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
业务合约地址	ddcAddr	String	是	DDC 业务合约地址
方法 ID	sig	String	是	Hex 格式的合约方法 ID，例如： 0x42966c68
业务费用	amount	BigInteger	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.2.8 删除 DDC 计费规则

#### 3.1.2.8.1 功能介绍

运营方可以通过调用该方法删除指定的 DDC 主合约的方法调用费用。

#### 3.1.2.8.2 API 定义

➤ 方法定义：String delFee(String sender,String ddcAddr, String sig);

➤ 合约方法：delFee(address ddcAddr,bytes4 sig);

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查 ddcAddr 为标准 address 格式，且不能为 0 地址；
3. 检查 sig 是有效的 4 位 Byte 的哈希格式字符串；
4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
业务合约地址	ddcAddr	String	是	DDC 业务合约地址
方法 ID	sig	String	是	Hex 格式的合约方法 ID，例如： 0x42966c68

--	--	--	--	--

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.2.9 删除 DDC 主合约授权

#### 3.1.2.9.1 功能介绍

运营方可以通过调用该方法删除指定的 DDC 业务合约授权。

#### 3.1.2.9.2 API 定义

➤ 方法定义：String delDDC(String sender,String ddcAddr);

➤ 合约方法：delDDC(address ddcAddr);

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查 ddcAddr 为标准 address 格式，且不能为 0 地址；
3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
业务合约地址	ddcAddr	String	是	DDC 业务合约地址

➤ 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.2.10 启用批量设置

#### 3.1.2.10.1 功能介绍

运营方通过调用该 API 接口对启用批量开关进行设置。

#### 3.1.2.10.2 API 定义

- 方法定义: String setSwitcherStateOfBatch(String sender, Boolean isOpen);
- 合约方法: setSwitcherStateOfBatch(bool isOpen);
- 调用者: 运营方;
- 核心逻辑:
  - 1. 检查 sender 为标准 address 格式, 并且不能为 0 地址;
  - 2. 检查签名事件是否被注册;
- 输入参数:

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
开关标识	isOpen	Boolean	是	

- 输出参数:

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3 BSN-DDC-721

#### 3.1.3.1 生成

##### 3.1.3.1.1 功能介绍

平台方或终端用户可以通过调用该方法进行 DDC 的生成。

##### 3.1.3.1.2 API 定义

- 方法定义：String mint(String sender,String to, String ddcURI);
- 合约方法：mint(address to, string memory ddcURI);
- 调用者：平台方、终端用户；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 资源标识符	ddcURI	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希



### 3.1.3.2 批量生成

#### 3.1.3.3.1 功能介绍

平台方或终端用户可以通过调用该方法进行 DDC 的批量生成。

#### 3.1.3.3.2 API 定义

➤ 方法定义：String mintBatch(String sender,String to,List<String> ddcURLs);

➤ 合约方法：mintBatch(address to,string[] memory ddcURLs);

➤ 调用者：平台方、终端用户；

- 核心逻辑：
1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 to 为标准 address 格式，且不能为 0 地址；
  3. 检查 ddcURLs 集合大小必须大于 0；
  4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者	to	String	是	
URI 集合	ddcURLs	List<String>	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.3.3 安全生成

3.1.3.3.1 功能介绍

平台方或终端用户可以通过调用该方法进行 DDC 的安全生成。

3.1.3.3.2 API 定义

➤ 方法定义：String safeMint(String sender,String to, String ddcURI,byte[] data);

➤ 合约方法：safeMint(address to, string memory ddcURI,bytes memory data);

➤ 调用者：平台方、终端用户；

- 核心逻辑：
- 1. 检查 sender 为标准地址格式，且不为 0 地址；
  - 2. 检查接收者账户地址为标准地址格式，且不为 0 地址；
  - 3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 资源标识符	ddcURI	String	是	
附加数据	data	byte[]	否	

➤ 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.3.4 批量安全生成

#### 3.1.3.4.1 功能介绍

平台方或终端用户可以通过调用该方法对 DDC 进行批量安全生成。

#### 3.1.3.4.2 API 定义

➤ 方法定义：String safeMintBatch(String sender,String to,List<String>

ddcURLs,byte[] data);

➤ 合约方法：safeMintBatch(address to,string[] memory ddcURLs,bytes

memory data);

➤ 调用者：平台方、终端用户；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查 to 为标准 address 格式，且不能为 0 地址；
3. 检查 ddcURLs 集合大小必须大于 0；
4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者	to	String	是	
URI 集合	ddcURLs	List<String>	是	
附加数据	data	byte[]	否	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.5 DDC 授权

#### 3.1.3.5.1 功能介绍

DDC 拥有者可以通过调用该方法对 DDC 进行授权，发起者需要是 DDC 的拥有者。

#### 3.1.3.5.2 API 定义

➤ 方法定义：String approve(String sender,String to,BigInteger ddclId);

➤ 合约方法：approve(address to,uint256 ddclId);

➤ 调用者：DDC 拥有者；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查授权者账户地址为标准 address 格式，且不能为 0 地址；
3. 检查 ddclId 是否大于 0；
4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
授权者账户	to	String	是	
DDC 唯一标识	ddclId	BigInteger	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.6 DDC 授权查询

#### 3.1.3.6.1 功能介绍

运营方、平台方或终端用户可以通过调用该方法对 DDC 进行授权查询。

#### 3.1.3.6.2 API 定义

- 方法定义：String getApproved(BigInteger ddclId);
- 合约方法：getApproved(uint256 ddclId);
- 调用者：运营方、平台方或终端用户；
- 核心逻辑：
  1. 检查 ddclId 的值是否大于 0；
  2. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
DDC 唯一标识	ddclId	BigInteger	是	

- 输出参数：

字段名	字段	类型	必传	备注
授权的账户	operator	String	是	

### 3.1.3.7 账户授权

#### 3.1.3.7.1 功能介绍

DDC 拥有者可以通过调用该方法进行账户授权，发起者需要是 DDC 的拥有者。

#### 3.1.3.7.2 API 定义

➤ 方法定义：String setApprovalForAll(String sender,String operator,bool approved);

➤ 合约方法：setApprovalForAll(address operator,bool approved);

➤ 调用者：DDC 拥有者；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查授权者账户地址为标准 address 格式，且不能为 0 地址；
3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
授权者账户	operator	String	是	
授权标识	approved	Boolean	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.8 账户授权查询

#### 3.1.3.8.1 功能介绍

运营方、平台方或终端用户可以通过调用该方法进行账户授权查询。

#### 3.1.3.8.2 API 定义

- 方法定义：Boolean isApprovedForAll(String owner,String operator);
- 合约方法：isApprovedForAll(address owner,address operator);
- 调用者：运营方、平台方或终端用户；
- 核心逻辑：
  1. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  2. 检查授权者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查签名事件是否被注册；

- 输入参数：

字段名	字段	类型	必传	备注
拥有者账户	owner	String	是	
授权者账户	operator	String	是	

- 输出参数：

字段名	字段	类型	必传	备注
授权标识	approved	Boolean	是	

### 3.1.3.9 安全转移

#### 3.1.3.9.1 功能介绍

DDC 的拥有者或授权者可以通过调用该方法对 DDC 进行安全转移。

3.1.3.9.2 API 定义

➤ 方法定义：String safeTransferFrom(String sender,String from,String to,BigInteger ddclId, byte[] data);

➤ 合约方法：safeTransferFrom(address from,address to,uint256 ddclId, byte[] data);

➤ 调用者：DDC 拥有者、DDC 授权者；

- 核心逻辑：
- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  - 3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  - 4. 检查 ddclId 的数值是否大于 0；
  - 5. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	from	String	是	
接收者账户	to	String	是	
DDC 唯一标识	ddclId	BigInteger	是	
附加数据	data	Byte[]	否	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希



3.1.3.10 转移

3.1.3.10.1 功能介绍

DDC 拥有者或授权者可以通过调用该方法对 DDC 进行转移。

3.1.3.10.2 API 定义

➤ 方法定义：String transferFrom(String sender,String from,String to,BigInteger ddclId);

➤ 合约方法：transferFrom(address from,address to,uint256 ddclId);

➤ 调用者：DDC 拥有者、DDC 授权者；

- 核心逻辑：
- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  - 3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  - 4. 检查 ddclId 的数值是否大于 0；
  - 5. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	from	String	是	
接收者账户	to	String	是	
DDC 唯一标识	ddclId	BigInteger	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.3.11 冻结

#### 3.1.3.11.1 功能介绍

运营方可以通过调用该方法对 DDC 进行冻结。

#### 3.1.3.11.2 API 定义

- 方法定义：String freeze (String sender,BigInteger ddclId);
- 合约方法：freeze(uint256 ddclId);
- 调用者：运营方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 DDCID 数值是否大于 0；
  3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 唯一标识	ddclId	BigInteger	是	DDC 唯一标识

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.12 解冻

#### 3.1.3.12.1 功能介绍

运营方可以通过调用该方法对 DDC 进行解冻。

#### 3.1.3.12.2 API 定义

- 方法定义：String unFreeze(String sender,BigInteger ddclId);
- 合约方法：unFreeze(uint256 ddclId);
- 调用者：运营方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 DDCID 数值是否大于 0；
  3. 检查签名事件是否被注册；

- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 唯一标识	ddclId	BigInteger	是	DDC 唯一标识

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.13 销毁

#### 3.1.3.13.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过调用该方法对 DDC 进行销毁。

#### 3.1.3.13.2 API 定义

- 方法定义：String burn(String sender,BigInteger ddclId);
- 合约方法：burn(uint256 ddclId);
- 调用者：DDC 拥有者、DDC 授权者；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 ddclId 的数值是否大于 0；
  3. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 唯一标识	ddclId	BigInteger	是	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.14 查询数量

#### 3.1.3.14.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法进行查询当前账户

拥有的 DDC 的数量。

### 3.1.3.14.2 API 定义

- 方法定义：BigInteger balanceOf(String owner);
- 合约方法：balanceOf(address owner) view returns (uint256);
- 调用者：运营方、平台方以及终端用户；
- 核心逻辑：
  1. 检查所有者账户地址为标准 address 地址，且不能为 0 地址；
  2. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
所有者账户	owner	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
DDC 的数量	balance	BigInteger	是	

### 3.1.3.15 查询所有者

#### 3.1.3.15.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法查询当前 DDC 的拥有者。

#### 3.1.3.15.2 API 定义

- 方法定义：String ownerOf(BigInteger ddclId);
- 合约方法：ownerOf(uint256 ddclId) view returns (address);

➤ 调用者：运营方、平台方以及终端用户；

➤ 核心逻辑：

1. 检查 ddclid 的数值是否大于 0；
2. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
DDC 唯一标识	ddclid	BigInteger	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
拥有者账户	owner	String	是	

### 3.1.3.16 获取名称

#### 3.1.3.16.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法查询当前 DDC 的名称。

#### 3.1.3.16.2 API 定义

➤ 方法定义：String name();

➤ 合约方法：name() view returns (string memory);

➤ 调用者：运营方、平台方以及终端用户；

➤ 核心逻辑：

1. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

--	--	--	--	--

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	DDC 运营方名称

### 3.1.3.17 获取符号

#### 3.1.3.17.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法查询当前 DDC 的符号标识。

#### 3.1.3.17.2 API 定义

- 方法定义：String symbol();
- 合约方法：symbol() view returns (string memory);
- 调用者：运营方、平台方以及终端用户；
- 核心逻辑：
  1. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	DDC 运营方符号

### 3.1.3.18 获取 DDCURI

#### 3.1.3.18.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法查询当前 DDC 的资源标识符。

#### 3.1.3.18.2 API 定义

- 方法定义：String ddcURI(BigInteger ddclid);
- 合约方法：ddcURI(uint256 ddclid) view returns (string memory);
- 调用者：运营方、平台方以及终端用户；
- 核心逻辑：
  1. 检查 ddclid 的数值是否大于 0;
  2. 检查签名事件是否被注册;
- 输入参数：

字段名	字段	类型	必传	备注
DDC 唯一标识	ddclid	BigInteger	是	

- 输出参数：

字段名	字段	类型	必传	备注
DDC 资源标识符	ddcURI	String	是	

### 3.1.3.19 URI 设置

#### 3.1.3.19.1 功能介绍

DDC 拥有者或 DDC 授权者通过调用该方法对 DDC 的资源标识符进



行设置。

3.1.3.19.2 API 定义

- 方法定义：String setURI(String sender,BigInteger ddclId,String ddcURI);
- 合约方法：setURI(uint256 ddclId,string memory ddcURI);
- 调用者：DDC 拥有者、DDC 授权者；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 ddclId 的数值是否大于 0；
  3. 检查 ddcURI 是否为空字符串；
  4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 唯一标识	ddclId	BigInteger	是	
DDC 资源标识符	ddcURI	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.3.20 名称符号设置

3.1.3.20.1 功能介绍

合约拥有者调用该方法对 721 的名称及符号进行设置。

3.1.3.20.2 API 定义

- 方法定义：String setNameAndSymbol(String sender,String name,String symbol);
- 合约方法： setNameAndSymbol(string memory name\_, string memory symbol\_);
- 调用者： DDC 拥有者、DDC 授权者；
- 核心逻辑：
  - 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查名称不能为空以及符号不能为空；
  - 3. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
名称	name	String	是	
符号	symbol	String	是	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.3.21 最新 DDCID 查询

3.1.3.21.1 功能介绍

运营方、平台方以及终端用户通过调用该方法对当前最新 DDCID 进行查询。

### 3.1.3.21.2 API 定义

- 方法定义：BigInteger getLatestDDCId();
- 合约方法：getLatestDDCId() view returns (uint256);
- 调用者：运营方；
- 核心逻辑：无
- 输入参数：

字段名	字段	类型	必传	备注

- 输出参数：

字段名	字段	类型	必传	备注
最新 DDCID		BigInteger	是	

### 3.1.3.22 Nonce 查询

#### 3.1.3.22.1 功能介绍

通过调用该方法对签名者账户所对应的最新 nonce 值进行查询，注：此查询只适用于发起元交易处理业务所对应的 nonce 值查询。

#### 3.1.3.22.2 API 定义

- 方法定义：BigInteger getNonce(String from);
- 合约方法：getNonce(address from) view returns (uint256);
- 调用者：所有用户；
- 核心逻辑：无
- 输入参数：

字段名	字段	类型	必传	备注
DDC 拥有者	from	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
最新 Nonce 值		BigInteger	是	

### 3.1.3.23 元交易生成

#### 3.1.3.23.1 功能介绍

终端用户可以通过授权平台方调用该方法对 DDC 进行元交易生成，  
注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.3.23.2 API 定义

➤ 方法定义：`String metaMint(String sender,String to,String ddcURL, BigInteger nonce,BigInteger deadline,byte[] sign);`

➤ 合约方法：`metaMint(address to,string memory ddcURL,uint256 nonce,uint256 deadline,bytes memory sign);`

➤ 调用者：平台方、终端用户；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
3. 检查 deadline 的数据是否大于 0；
4. 检查签名值数组长度是否大于 0；
5. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 资源标识符	ddcURI	String	是	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.24 元交易安全生成

#### 3.1.3.24.1 功能介绍

终端用户可以通过授权平台方调用该方法对 DDC 进行元交易安全生成，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.3.24.2 API 定义

➤ 方法定义：String metaSafeMint(String sender,String to, String ddcURI,

byte[] data,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaSafeMint(address to,string memory ddcURI,bytes

memory data,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：平台方、终端用户；

➤ 核心逻辑：

- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
- 2. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
- 3. 检查 deadline 的数据是否大于 0；
- 4. 检查签名值数组长度是否大于 0；
- 5. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 资源标识符	ddcURI	String	是	
附加数据	data	byte[]	否	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.25 元交易批量生成

#### 3.1.3.25.1 功能介绍

终端用户可以通过授权平台方调用该方法对 DDC 进行元交易批量生成，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

3.1.3.25.2 API 定义

- 方法定义：String metaMintBatch(String sender,String to,List<String> ddcURLs,BigInteger nonce,BigInteger deadline,byte[] sign);
- 合约方法：metaMintBatch(address to,string[] memory ddcURLs,uint256 nonce,uint256 deadline,bytes memory sign);
- 调用者：平台方、终端用户；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 to 为标准 address 格式，且不能为 0 地址；
  3. 检查 ddcURLs 集合大小必须大于 0；
  4. 检查 deadline 的数据是否大于 0；
  5. 检查签名值数组长度是否大于 0；
  6. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者	to	String	是	
URI 集合	ddcURLs	List<String>	是	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.3.26 元交易批量安全生成

3.1.3.26.1 功能介绍

终端用户可以通过授权平台方调用该方法对 DDC 进行元交易批量安全生成，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

3.1.3.26.2 API 定义

➤ 方法定义：String metaSafeMintBatch(String sender,String to,List<String> ddcURLs,byte[] data,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaSafeMintBatch(address to,string[] memory ddcURLs,bytes memory data,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：平台方、终端用户；

- 核心逻辑：
- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查 to 为标准 address 格式，且不能为 0 地址；
  - 3. 检查 ddcURLs 集合大小必须大于 0；
  - 4. 检查 deadline 的数据是否大于 0；
  - 5. 检查签名值数组长度是否大于 0；
  - 6. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址



接收者	to	String	是	
URI 集合	ddcURLs	List<String>	是	
附加数据	data	byte[]	否	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.27 元交易转移

#### 3.1.3.27.1 功能介绍

DDC 拥有者或 DDC 授权者通过授权平台方调用该方法对 DDC 进行元交易转移，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.3.27.2 API 定义

➤ 方法定义：String metaTransferFrom(String sender,String from,String to,BigInteger ddclD,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaTransferFrom(address from,address to,uint256 ddclD,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：平台方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
4. 检查 ddclid 的数值是否大于 0；
5. 检查 deadline 的数据是否大于 0；
6. 检查签名值数组长度是否大于 0；
7. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	from	String	是	
接收者账户	to	String	是	
DDC 唯一标识	ddclid	BigInteger	是	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.28 元交易安全转移

#### 3.1.3.28.1 功能介绍

DDC 的拥有者或授权者可以通过授权平台方调用该方法对 DDC 进行元交易安全转移，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

3.1.3.28.2 API 定义

➤ 方法定义：String metaSafeTransferFrom(String sender,String from,String to,BigInteger ddclId,byte[] data,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaSafeTransferFrom(address from,address to,uint256 ddclId,bytes memory data,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：DDC 拥有者、DDC 授权者；

- 核心逻辑：
- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  - 3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  - 4. 检查 ddclId 的数值是否大于 0；
  - 5. 检查 deadline 的数据是否大于 0；
  - 6. 检查签名值数组长度是否大于 0；
  - 7. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	from	String	是	
接收者账户	to	String	是	
DDC 唯一标识	ddclId	BigInteger	是	
附加数据	data	Byte[]	否	
nonce 值	nonce	BigInteger	是	

过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.3.29 元交易销毁

#### 3.1.3.29.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过授权平台方调用该方法对 DDC 进行元交易销毁，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.3.29.2 API 定义

➤ 方法定义：String metaBurn(String sender, BigInteger ddclId, BigInteger nonce, BigInteger deadline, byte[] sign);

➤ 合约方法：metaBurn(uint256 ddclId, uint256 nonce, uint256 deadline, bytes memory sign);

➤ 调用者：DDC 拥有者、DDC 授权者；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查 ddclId 的数值是否大于 0；
3. 检查 deadline 的数据是否大于 0；
4. 检查签名值数组长度是否大于 0；
5. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 唯一标识	ddcId	BigInteger	是	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

## 3.1.4 BSN-DDC-1155

### 3.1.4.1 安全生成

#### 3.1.4.1.1 功能介绍

平台方或终端用户可以通过调用该方法对 DDC 进行安全生成。

#### 3.1.4.1.2 API 定义

➤ 方法定义：`String safeMint(String sender,String to,BigInteger amount,String ddcURI,byte[] data);`

➤ 合约方法：`safeMint(address to,uint256 amount, string memory ddcURI, bytes memory data);`

➤ 调用者：平台方、终端用户；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
3. 检查需要生成的 DDC 数量是否大于 0；
4. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 数量	amount	BigInteger	是	
DDCURI	ddcURI	String	是	
附加数据	data	byte[]	否	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.2 批量安全生成

#### 3.1.4.2.1 功能介绍

平台方或终端用户可以通过调用该方法对 DDC 进行批量安全生成。

#### 3.1.4.2.2 API 定义

➤ 方法定义：`String safeMintBatch(String sender,String to,Multimap<BigInteger,String> ddcs,byte[] data);`

➤ 合约方法：`safeMintBatch(address to,uint256[] amounts, string[] ddcURIs, bytes memory data);`

- 调用者：平台方、终端用户；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查 ddcs 集合大小必须大于 0；
  4. 循环 ddcs 集合，挨个检查 DDC 数量必须大于。
  5. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 信息合计	ddcInfo	Multimap<BigInteger,String>	是	
附加数据	data	byte[]	否	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.3 账户授权

#### 3.1.4.3.1 功能介绍

DDC 拥有者可以通过调用该方法进行账户授权，发起者需要是 DDC 的拥有者。

3.1.4.3.2 API 定义

- 方法定义：String setApprovalForAll(String sender,String operator, Boolean approved);
- 合约方法：setApprovalForAll(address operator, bool approved);
- 调用者：DDC 拥有者;
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址;
  2. 检查授权者账户地址为标准 address 格式，且不能为 0 地址;
  3. 检查签名事件是否被注册;
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
授权者账户	operator	String	是	
授权标识	approved	Boolean	是	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.4.4 账户授权查询

3.1.4.4.1 功能介绍

运营方、平台方或终端用户可以通过调用该方法进行账户授权查询。



3.1.4.4.2 API 定义

- 方法定义： Boolean isApprovedForAll(String owner,String operator);
- 合约方法： isApprovedForAll(address owner,address operator) view

returns (bool);

- 调用者：运营方、平台方或终端用户；
- 核心逻辑：
  1. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  2. 检查授权者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查签名事件是否被注册；

- 输入参数：

字段名	字段	类型	必传	备注
拥有者账户	owner	String	是	
授权者账户	operator	Boolean	是	

- 输出参数：

字段名	字段	类型	必传	备注
		Boolean	是	

3.1.4.5 安全转移

3.1.4.5.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过调用该方法对 DDC 进行安全转移。

3.1.4.5.2 API 定义

- 方法定义：String safeTransferFrom(String sender,String from,String to,BigInteger ddclId,BigInteger amount,byte[] data);
- 合约方法：safeTransferFrom(address from,address to,uint256 ddclId,uint256 amount,bytes memory data);
- 调用者：DDC 拥有者、DDC 授权者；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  4. 检查 DDCID 数值是否大于 0；
  5. 检查 DDC 转移所对应的数量是否大于 0；
  6. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	from	String	是	
接收者账户	to	String	是	
DDCID	ddclId	BigInteger	是	
数量	amount	BigInteger	是	DDCID 所对应的数量
附加数据	data	byte[]	否	

- 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.4.6 批量安全转移

#### 3.1.4.6.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过调用该方法对 DDC 进行批量安全转移。

#### 3.1.4.6.2 API 定义

➤ 方法定义：String safeBatchTransferFrom(String sender,String from,String to,Map<BigInteger,BigInteger> ddcs, byte[] data);

➤ 合约方法：safeBatchTransferFrom(address from, address to,uint256[] ddclids,uint256[] amounts,bytes memory data);

➤ 调用者：DDC 拥有者、DDC 授权者；

- 核心逻辑：
1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  4. 检查转移的 ddcs 集合大小是否大于 0；
  5. 检查转移的 ddcs 集合中每个 DDCID 是否大于 0；
  6. 检查转移的 ddcs 集合中每个 DDC 数量是否大于 0；
  7. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址

拥有者账户	from	String	是	
接收者账户	to	String	是	
拥有者 DDCID 集合	ddcs	Map<BigInteger, BigInteger>	是	
附加数据	data	byte[]	否	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.7 冻结

#### 3.1.4.7.1 功能介绍

运营方可以通过调用该方法对 DDC 进行冻结。

#### 3.1.4.7.2 API 定义

- 方法定义：String freeze(String sender, BigInteger ddclid);
- 合约方法：freeze(uint256 ddclid);
- 调用者：运营方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查 DDCID 数值是否大于 0；
  3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址

DDC 唯一标识	ddcId	BigInteger	是	DDC 唯一标识
----------	-------	------------	---	----------

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.8 解冻

#### 3.1.4.8.1 功能介绍

运营方可以通过调用该方法对 DDC 进行解冻。

#### 3.1.4.8.2 API 定义

➤ 方法定义：String unFreeze(String sender, BigInteger ddcId);

➤ 合约方法：unFreeze(uint256 ddcId);

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查 DDCID 数值是否大于 0；
3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 唯一标识	ddcId	BigInteger	是	DDC 唯一标识

➤ 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.4.9 销毁

#### 3.1.4.9.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过调用该方法对 DDC 进行销毁。

#### 3.1.4.9.2 API 定义

- 方法定义：String burn(String sender,String owner,BigInteger ddclId);
- 合约方法：burn(address owner,uint256 ddclId);
- 调用者：DDC 拥有者、DDC 授权者；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查需要销毁的 DDCID 集合长度是否大于 0；
  4. 检查签名事件是否被注册；

- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	owner	String	是	
DDCID	ddclId	BigInteger	是	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.10 批量销毁

#### 3.1.4.10.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过调用该方法对 DDC 进行批量销毁。

#### 3.1.4.10.2 API 定义

➤ 方法定义：String burnBatch(String sender,String owner,List<BigInteger> ddclDs);

➤ 合约方法：burnBatch(address owner,uint256[] ddclDs);

➤ 调用者：DDC 拥有者、DDC 授权者；

- 核心逻辑：
1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查需要销毁的 DDCID 集合大小是否大于 0；
  4. 检查需要销毁的 DDCID 集合中每个 DDCID 数值是否大于 0；
  5. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	owner	String	是	
DDCID 集合	ddclDs	List<BigInteger>	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.4.11 查询数量

#### 3.1.4.11.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法进行查询当前账户拥有的 DDC 的数量。

#### 3.1.4.11.2 API 定义

- 方法定义：BigInteger balanceOf(String owner, BigInteger ddclId);
- 合约方法：balanceOf(address owner, uint256 ddclId) view returns (uint256);
- 调用者：运营方、平台方以及终端用户；
- 核心逻辑：
  1. 检查所有者账户地址为标准 address 格式，且不能为 0 地址；
  2. 检查 DDCID 集合长度是否大于 0；
  3. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
所有者账户	owner	String	是	
DDCID	ddclId	BigInteger	是	

- 输出参数：

字段名	字段	类型	必传	备注
数量		BigInteger	是	所有者账户所对应的 DDCID 所拥



				用的数量
--	--	--	--	------

3.1.4.12 批量查询数量

3.1.4.12.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法进行批量查询账户拥有的 DDC 的数量。

3.1.4.12.2 API 定义

➤ 方法定义: `List<BigInteger> balanceOfBatch(Multimap<String, BigInteger> ddcs);`

➤ 合约方法: `balanceOfBatch(address[] memory owners,uint256[] memory ddclids) view returns (uint256[] memory);`

➤ 调用者: 运营方、平台方以及终端用户;

- 核心逻辑:
- 1. 检查 ddcs 集合大小是否大于 0;
  - 2. 循环 ddcs 集合, 根据索引检查 ddcs 集合中拥有者账户地址为标准 address 格式且不能为 0 地址, 以及检查 DDCID 数值必须大于 0;
  - 3. 检查签名事件是否被注册;

➤ 输入参数:

字段名	字段	类型	必传	备注
拥有者 DDCID 集合	ddcs	Multimap<String, BigInteger>	是	

➤ 输出参数:

字段名	字段	类型	必传	备注
数量集合		List<BigInteger>	是	拥有者账户所对应的每个 DDCID 所拥用的数量

### 3.1.4.13 获取 DDCURI

#### 3.1.4.13.1 功能介绍

运营方、平台方以及终端用户可以通过调用该方法进行查询当前 DDC 的资源标识符。

#### 3.1.4.13.2 API 定义

- 方法定义：String ddcURI(BigInteger ddclId);
- 合约方法：ddcURI(uint256 ddclId) view returns (string memory);
- 调用者：运营方、平台方以及终端用户；
- 核心逻辑：
  1. 检查 DDCID 数值是否大于 0；
  2. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
DDCID	ddclId	BigInteger	是	

- 输出参数：

字段名	字段	类型	必传	备注
DDCURI		String	是	

3.1.4.14 URI 设置

3.1.4.14.1 功能介绍

DDC 拥有者或 DDC 授权者通过调用该方法对 DDC 的资源标识符进行设置。

3.1.4.14.2 API 定义

- 方法定义：String setURI(String sender,String owner,BigInteger ddclId,String ddcURI);
- 合约方法：setURI(address owner,uint256 ddclId,string memory ddcURI);
- 调用者：DDC 拥有者、DDC 授权者；
- 核心逻辑：
  - 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查 owner 地址为标准 address 格式，且不能为 0 地址；
  - 3. 检查 ddclId 数值是否大于 0；
  - 4. 检查 ddcURI 是否为空字符串；
  - 5. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 拥有者	owner	String	是	
DDC 唯一标识	ddclId	BigInteger	是	
DDC 资源标识符	ddcURI	String	是	

- 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.15 最新 DDCID 查询

#### 3.1.4.15.1 功能介绍

运营方、平台方以及终端用户通过调用该方法对当前最新 DDCID 进行查询。

#### 3.1.4.15.2 API 定义

- 方法定义：BigInteger getLatestDDCId();
- 合约方法：getLatestDDCId() view returns (uint256);
- 调用者：运营方；
- 核心逻辑：无
- 输入参数：

字段名	字段	类型	必传	备注

- 输出参数：

字段名	字段	类型	必传	备注
最新 DDCID		BigInteger	是	

### 3.1.4.16 Nonce 查询

#### 3.1.4.16.1 功能介绍

通过调用该方法对签名者账户所对应的最新 nonce 值进行查询，注：

此查询只适用于发起元交易处理业务所对应的 nonce 值查询。

### 3.1.4.16.2 API 定义

- 方法定义：BigInteger getNonce(String from);
- 合约方法：getNonce(address from) view returns (uint256);
- 调用者：所有用户；
- 核心逻辑：无
- 输入参数：

字段名	字段	类型	必传	备注
DDC 拥有者	from	String	是	

- 输出参数：

字段名	字段	类型	必传	备注
最新 Nonce 值		BigInteger	是	

### 3.1.4.17 元交易安全生成

#### 3.1.4.17.1 功能介绍

终端用户可以通过授权平台方调用该方法对 DDC 进行元交易安全生成，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.4.17.2 API 定义

- 方法定义：String metaSafeMint(String sender,String to,BigInteger amount,String ddcURI,byte[] data,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaSafeMint(address to,uint256 amount,string memory ddcURI,bytes memory data,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：平台方、终端用户；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
3. 检查需要生成的 DDC 数量是否大于 0；
4. 检查 deadline 的数据是否大于 0；
5. 检查签名值数组长度是否大于 0；
6. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 数量	amount	BigInteger	是	
DDCURI	ddcURI	String	是	
附加数据	data	byte[]	否	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.4.18 元交易批量安全生成

3.1.4.18.1 功能介绍

终端用户可以通过授权平台方调用该方法对 DDC 进行元交易批量安全生成，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

3.1.4.18.2 API 定义

➤ 方法定义：String metaSafeMintBatch(String sender,String to,Multimap<BigInteger,String> ddcs,byte[] data,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaSafeMintBatch(address to,uint256[] memory amounts,string[] memory ddcURLs,bytes memory data,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：平台方、终端用户；

- 核心逻辑：
- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  - 3. 检查 ddcs 集合大小必须大于 0；
  - 4. 循环 ddcs 集合，挨个检查 DDC 数量必须大于；
  - 5. 检查 deadline 的数据是否大于 0；
  - 6. 检查签名值数组长度是否大于 0；
  - 7. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

调用者	sender	String	是	调用者地址
接收者账户	to	String	是	
DDC 信息合计	ddcInfo	Multimap<BigInteger,String>	是	
附加数据	data	byte[]	否	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.19 元交易安全转移

#### 3.1.4.19.1 功能介绍

DDC 拥有者或 DDC 授权者通过授权平台方调用该方法对 DDC 进行元交易安全转移，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.4.19.2 API 定义

➤ 方法定义：String metaSafeTransferFrom(String sender,String from,String to,BigInteger ddcId,BigInteger amount,byte[] data,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaSafeTransferFrom(address from,address to,uint256



ddcid,uint256    amount,bytes    memory    data,uint256    nonce,uint256

deadline,bytes memory sign);

- 调用者：平台方；
- 核心逻辑：
  1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
  4. 检查 ddcid 的数值是否大于 0；
  5. 检查 amount 的数值是否大于 0；
  6. 检查 deadline 的数据是否大于 0；
  7. 检查签名值数组长度是否大于 0；
  8. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	from	String	是	
接收者账户	to	String	是	
DDC 唯一标识	ddcid	BigInteger	是	
数量	amount	BigInteger	是	DDCID 所对应的数量
附加数据	data	byte[]	否	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

- 输出参数：

字段名	字段	类型	必传	备注
-----	----	----	----	----

		String	是	交易哈希
--	--	--------	---	------

### 3.1.4.20 元交易批量安全转移

#### 3.1.4.20.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过授权平台方调用该方法对 DDC 进行元交易批量转移，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.4.20.2 API 定义

➤ 方法定义：String metaSafeBatchTransferFrom(String sender,String from,String to,Map<BigInteger,BigInteger> ddcs,byte[] data,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaSafeBatchTransferFrom(address from,address to,uint256[] memory ddcls,uint256[] memory amounts,bytes memory data,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：DDC 拥有者、DDC 授权者；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；
2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
3. 检查接收者账户地址为标准 address 格式，且不能为 0 地址；
4. 检查转移的 ddcs 集合大小是否大于 0；
5. 检查转移的 ddcs 集合中每个 DDCID 是否大于 0；
6. 检查转移的 ddcs 集合中每个 DDC 数量是否大于 0；
7. 检查 deadline 的数据是否大于 0；
8. 检查签名值数组长度是否大于 0；

9. 检查签名事件是否被注册;

➤ 输入参数:

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	from	String	是	
接收者账户	to	String	是	
拥有者 DDCID 集合	ddcs	Map<BigInteger,BigInteger>	是	
附加数据	data	byte[]	否	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数:

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.21 元交易销毁

#### 3.1.4.21.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过授权平台方调用该方法对 DDC 进行元交易销毁, 注: 调用此接口所消耗的能量值和业务费对应的是平台方账户。

#### 3.1.4.21.2 API 定义

➤ 方法定义: String metaBurn(String sender,String owner,BigInteger

ddcid, BigInteger nonce, BigInteger deadline, byte[] sign);

➤ 合约方法： metaBurn(address owner,uint256 ddcid,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者： DDC 拥有者、DDC 授权者；

➤ 核心逻辑：

- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
- 2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
- 3. 检查需要销毁的 DDCID 集合长度是否大于 0；
- 4. 检查 deadline 的数据是否大于 0；
- 5. 检查签名值数组长度是否大于 0；
- 6. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
拥有者账户	owner	String	是	
DDCID	ddcid	BigInteger	是	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

3.1.4.22 元交易批量销毁

3.1.4.22.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过授权平台方调用该方法对 DDC 进行元交易批量销毁，注：调用此接口所消耗的能量值和业务费对应的是平台方账户。

3.1.4.22.2 API 定义

➤ 方法定义：String metaBurnBatch(String sender,String owner,List<BigInteger> ddclDs,BigInteger nonce,BigInteger deadline,byte[] sign);

➤ 合约方法：metaBurnBatch(address owner,uint256[] memory ddclDs,uint256 nonce,uint256 deadline,bytes memory sign);

➤ 调用者：DDC 拥有者、DDC 授权者；

- 核心逻辑：
- 1. 检查 sender 为标准 address 格式，且不能为 0 地址；
  - 2. 检查拥有者账户地址为标准 address 格式，且不能为 0 地址；
  - 3. 检查需要销毁的 DDCID 集合大小是否大于 0；
  - 4. 检查需要销毁的 DDCID 集合中每个 DDCID 数值是否大于 0；
  - 5. 检查 deadline 的数据是否大于 0；
  - 6. 检查签名值数组长度是否大于 0；
  - 7. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地

				址
拥有者账户	owner	String	是	
DDCID 集合	ddclds	List<BigInteger>	是	
nonce 值	nonce	BigInteger	是	
过期时间	deadline	BigInteger	是	
签名值	sign	byte[]	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		String	是	交易哈希

### 3.1.4.23 同步拥有者

#### 3.1.4.23.1 功能介绍

运营方可以通过调用该方法对旧 DDC 所对应的拥有者数据同步给链上。

#### 3.1.4.23.2 API 定义

➤ 方法定义：String syncDDCOwners(String sender,List<BigInteger>

ddclds,List<List<String>> owners);

➤ 合约方法：syncDDCOwners(uint256[] memory ddclds,address[][]

memory owners);

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准 address 格式，且不能为 0 地址；

2. 检查需要同步的 DDCID 集合大小是否大于 0;
3. 检查需要同步的 DDCID 集合中每个 DDCID 数值是否大于 0;
4. 检查需要同步的 DDC 所对应的拥有者集合长度是否大于 0;
5. 检查需要同步的 DDC 所对应的拥有者集合的每项长度是否大于 0
6. 检查签名事件是否被注册;

➤ 输入参数:

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDCID 集合	ddcIds	List<BigInteger>	是	
拥有者列表	owners	List<List<String>>	是	

➤ 输出参数:

字段名	字段	类型	必传	备注
		String	是	交易哈希

## 3.1.5 BSN-DDC-跨链应用

### 3.1.5.1 DDC 跨链流转

#### 3.1.5.1.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过该方法调用 DDC 跨链应用合约进行 DDC 的跨链流转。

#### 3.1.5.1.2 API 定义

➤ 方法定义: String crossChainTransfer(String sender,DDCType ddcType,String signer,String to,BigInteger ddId,byte[] data,BigInteger toChainID,String

toCCAddr,String funcName);

➤ 合约方法： crossChainTransfer(DDCType ddcType,address signer,address to,uint256 ddclId,bytes memory data,uint64 toChainID,address toCCAddr,string memory funcName);

➤ 调用者： DDC 拥有者、DDC 授权者；

➤ 核心逻辑：

- 1. 检查 sender 为标准地址格式，且不能为 0 地址；
- 2. 检查 ddcType 类型是否为 721 或 1155（0、721； 1、1155）；
- 3. 检查 singer 为标准地址格式，且不能为 0 地址；
- 4. 检查 to 为标准地址格式，且不能为 0 地址；
- 5. 检查 ddclId 的数值是否大于 0；
- 6. 检查 toChainID 的数值是否大于 0；
- 7. 检查 toCCAddr 为标准地址格式，且不能为 0 地址；
- 8. 检查 funcName 字符串不可为空；
- 9. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 类型	ddcType	DDCType	是	
目标链签名者账户	signer	String	是	
目标链接收者账户	to	String	是	
DDC 唯一标识	ddclId	BigInteger	是	
附加数据	data	byte[]	否	
目标链链 ID	toChainID	BigInteger	是	



目标链合约	toCCAddr	String	是	
目标链方法名	funcName	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		交易哈希	String	是

### 3.1.5.2 DDC 跨链通知

#### 3.1.5.2.1 功能介绍

运营方可以通过该方法调用 DDC 跨链应用合约进行 DDC 的跨链通知。

#### 3.1.5.2.2 API 定义

➤ 方法定义：String UpdateCrossChainStatus(String sender,BigInteger crossChainID,State state,String remark);

➤ 合约方法：updateCrossChainStatus(uint256 crossChainID,State state,string memory remark);

➤ 调用者：运营方；

➤ 核心逻辑：

1. 检查 sender 为标准地址格式，且不能为 0 地址；
2. 检查 crossChainID 的数值是否大于 0；
3. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址

跨链 ID	crossChainID	BigInteger	是	
状态	state	State	是	1: 跨链成功 2: 跨链失败
备注	remark	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
		交易哈希	String	是

## 3.1.6 BSN-DDC-开放联盟链跨链应用

### 3.1.6.1 DDC 跨链申请

#### 3.1.6.1.1 功能介绍

DDC 拥有者或 DDC 授权者可以通过该方法调用 DDC 开放联盟链跨链应用合约进行 DDC 的跨链申请。

#### 3.1.6.1.2 API 定义

- 方法定义：String crossChainTransfer(String sender, DDCType ddcType, BigInteger ddclId, Boolean isLock, BigInteger toChainId, String to, String data)
- 合约方法：function crossChainTransfer(DDCType ddcType,uint256 ddclId,bool isLock,uint64 toChainID,address to,bytes memory data)
- 调用者：DDC 拥有者、DDC 授权者
- 核心逻辑：
  1. 检查 sender 为标准地址格式，且不能为 0 地址；

2. 检查 ddcType 类型是否为 721 或 1155 (0、721; 1、1155) ;
3. 检查 ddclId 的数值是否大于 0;
4. 检查 isLock 的值不可为空;
5. 检查 toChainId 的是指是否大于 0;
6. 检查 to 为标准 address 格式, 且不能为 0 地址;

➤ 输入参数:

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
DDC 类型	ddcType	DDCType	是	
DDC 唯一标识	ddclId	BigInteger	是	
是否锁定	isLock	Boolean	是	
目标链链 ID	toChainId	BigInteger	是	
目标链接收者 账户	to	String	是	
附加数据	data	byte[]	是	

➤ 输出参数:

字段名	字段	类型	必传	备注
交易哈希		String		

### 3.1.6.2 DDC 跨链通知

#### 3.1.6.2.1 功能介绍

运营方可以通过该方法调用 DDC 跨链应用合约进行 DDC 的跨链通知。

### 3.1.6.2.2 API 定义

➤ 方法定义：String UpdateCrossChainStatus(String sender,BigInteger crossChainID,State state,String remark);

➤ 合约方法：updateCrossChainStatus(uint256 crossChainID,State state,string memory remark);

➤ 调用者：运营方

➤ 核心逻辑：

1. 检查 sender 为标准地址格式，且不能为 0 地址；

2. 检查 crossChainID 的数值是否大于 0；

3. 检查 remark 的值不可为空；

➤ 输入参数：

字段名	字段	类型	必传	备注
调用者	sender	String	是	调用者地址
跨链 ID	crossChainID	BigInteger	是	
状态	state	State	是	1: 跨链成功 2: 跨链失败
备注	remark	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
交易哈希		String		

### 3.1.7 BSN-DDC-交易查询

#### 3.1.7.1 查询交易信息

##### 3.1.7.1.1 功能介绍

运营方或平台方根据交易哈希对交易信息进行查询。

##### 3.1.7.1.2 API 定义

➤ 方法定义：String getTransByHash(String txHash)

➤ 调用者：平台方、运营方；

➤ 核心逻辑：

1. 根据交易哈希查询的交易信息（不同框架自定义输出参数）
2. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
交易哈希	txHash	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
交易信息	txInfo	String	是	

#### 3.1.7.2 查询交易回执

##### 3.1.7.2.1 功能介绍

运营方或平台方根据交易哈希对交易回执信息进行查询。

### 3.1.7.2.2 API 定义

- 方法定义：String getTransReceipt(String txHash)
- 调用者：平台方、运营方；
- 核心逻辑：
  1. 根据交易 hash 查询交易回执（不同框架自定义输出参数）
  2. 检查签名事件是否被注册；
- 输入参数：

字段名	字段	类型	必传	备注
交易哈希	txHash	String	是	

- 输出参数：

字段名	字段	类型	必传	备注
交易回执	txReceipt	String	是	

### 3.1.7.3 查询交易状态

#### 3.1.7.3.1 功能介绍

运营方或平台方根据交易哈希查询交易状态是否成功。

#### 3.1.7.3.2 API 定义

- 方法定义：Boolean getTransByStatus(String txHash)
- 调用者：平台方、运营方；
- 核心逻辑：
  1. 根据交易哈希查询的交易是否成功；
  2. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
交易哈希	txHash	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
交易是否成功	txStatus	Boolean	是	

## 3.1.8 BSN-DDC-区块查询

### 3.1.8.1 获取区块信息

#### 3.1.8.1.1 功能介绍

运营方或平台方根据区块高度对区块信息进行查询，并解析区块数据返回给运营方或平台方。

#### 3.1.8.1.2 API 定义

➤ 方法定义：String getBlockByNumber(BigInteger blockNumber)

➤ 调用者：运营方、平台方；

➤ 核心逻辑：

1. 根据区块高度查询区块信息（不同框架自定义输出参数）；
2. 检查签名事件是否被注册；

➤ 输入参数：

字段名	字段	类型	必传	备注
区块高度	blockNumber	BigInteger	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
区块信息	blockInfo	String	是	

## 3.1.9 BSN-DDC-签名事件

### 3.1.9.1 功能介绍

此事件是通用事件，所有的上链待签名交易报文需调用此事件进行签名，业务调用方需要注册此签名事件，并在实现的签名事件中实现签名逻辑，并将最终签名后的结果返回给 DDC-SDK。

### 3.1.9.2 事件定义

- 输入参数：签名事件类
- 输出参数：签名结果
- `String signEvent(SignEvent event);`

### 3.1.9.3 数据结构

#### ➤ SignEvent

字段名	字段	类型	必传	备注
签名者	sender	String	是	
待签名交易	unSignTrans	Object	是	



### 3.1.10BSN-DDC-数据解析

#### 3.1.10.1 权限数据

##### 3.1.10.1.1 添加账户开关设置

###### 3.1.10.1.1.1 功能说明

用于对 BSN-DDC-权限合约进行平台方添加链账户开关设置所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

###### 3.1.10.1.1.2 合约事件

- SetSwitcherStateOfPlatform(address indexed operator, bool isOpen)

###### 3.1.10.1.1.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	
是否开通	isOpen	Boolean	是	

##### 3.1.10.1.2 添加账户

###### 3.1.10.1.2.1 功能说明

用于对 BSN-DDC-权限合约进行添加账户所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

###### 3.1.10.1.2.2 合约事件

- AddAccount(address indexed caller,address indexed account)

### 3.1.10.1.2.3 数据结构

字段名	字段	类型	必传	备注
签名者	sender	String	是	
链账户地址	account	String	是	添加的链账户地址

### 3.1.10.1.3 批量添加账户

#### 3.1.10.1.3.1 功能说明

用于对 BSN-DDC- 权限合约进行平台方或运营方批量添加账户所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.1.3.2 合约事件

➤ AddBatchAccount(address indexed operator,address[] indexed accounts);

#### 3.1.10.1.3.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	
链账户地址集合	accounts	List<String>	是	

### 3.1.10.1.4 更新账户状态

#### 3.1.10.1.4.1 功能说明

用于对 BSN-DDC- 权限合约进行更新账户状态所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.1.4.2 合约事件

- UpdateAccountState(address indexed account,IAuthorityData.State platformState,IAuthorityData.State operatorState)

#### 3.1.10.1.4.3 数据结构

字段名	字段	类型	必传	备注
链账户地址	account	String	是	添加的链账户地址
平台管理 账户状态	platformState	enum	是	DDC 账户对应的 当前账户状态（仅 平台方可操作该状 态）。值包含： 1.Frozen（冻结状 态，无法进行 DDC 相关操作）  2.Active（活跃状 态，可进行 DDC 相关操作）
运营管理 账户状态	operatorState	enum	是	DDC 账户对应的 当前账户状态（仅 运营方可操作该状 态）。值包含： 1.Frozen（冻结状 态，无法进行 DDC 相关操作）  2.Active（活跃状 态，可进行 DDC 相关操作）

#### 3.1.10.1.5 跨平台授权

##### 3.1.10.1.5.1 功能说明

用于对 BSN-DDC-权限合约进行跨平台授权所产生的交易回执

中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.1.5.2 合约事件

- CrossPlatformApproval(address indexed from,address indexed to, bool approved)

#### 3.1.10.1.5.3 数据结构

字段名	字段	类型	必传	备注
授权账户	from	String	是	
接收账户	to	String	是	
授权标识	approved	Boolean	是	

### 3.1.10.1.6 同步平台方 DID

#### 3.1.10.1.6.1 功能说明

用于对 BSN-DDC-权限合约进行同步平台 DID 所产生的交易回  
执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.1.6.2 合约事件

- SyncPlatformDID(address indexed operator,string[] dids)

#### 3.1.10.1.6.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	
平台方 DID 集合	dids	List<String>	是	

## 3.1.10.2 充值数据

### 3.1.10.2.1 充值

#### 3.1.10.2.1.1 功能说明

用于对 BSN-DDC-计费合约进行充值所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.2.1.2 合约事件

➤ Recharge(address indexed from,address indexed to,uint256 value)

#### 3.1.10.2.1.3 数据结构

字段名	字段	类型	必传	备注
原链账户地址	from	String	是	业务费转出方链账户地址
目标链账户地址	to	String	是	业务费转入方链账户地址
业务费	amount	BigInteger	是	充值的业务费金额

### 3.1.10.2.2 批量充值

#### 3.1.10.2.2.1 功能说明

用于对 BSN-DDC-计费合约进行批量充值所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.2.2.2 合约事件

➤ RechargeBatch(address indexed from,address[] indexed toList,uint256[] amounts)

### 3.1.10.2.2.3 数据结构

字段名	字段	类型	必传	备注
原链账户地址	from	String	是	业务费转出方链账户地址
账户列表	accounts	Multimap<String, BigInteger>	是	Key: 接收者账户 Value: 业务费额

### 3.1.10.2.3 DDC 业务费扣除

#### 3.1.10.2.3.1 功能说明

用于对 BSN-DDC-计费合约进行 DDC 业务费扣除所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.2.3.2 合约事件

- Pay(address indexed from,address indexed ddcAddr,bytes4 sig,uint32 amount,uint256 ddclId)

#### 3.1.10.2.3.3 数据结构

字段名	字段	类型	必传	备注
链账户地址	from	String	是	扣除业务费方链账户地址
业务合约	ddcAddr	String	是	业务合约地址
方法签名	sig	String	是	业务合约所对应的方法签名
业务费	amount	BigInteger	是	充值的业务费金额
DDC 唯	ddclId	BigInteger	是	

一标识				
-----	--	--	--	--

3.1.10.2.4 设置 DDC 计费规则

3.1.10.2.4.1 功能说明

用于对 BSN-DDC-计费合约进行设置 DDC 计费规则所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.2.4.2 合约事件

➤ SetFee(address ddcAddr,byte4 sig,uint amount)

3.1.10.2.4.3 数据结构

字段名	字段	类型	必传	备注
业务合约	ddcAddr	String	是	业务合约地址
方法签名	sig	String	是	业务合约所对应的方法签名
业务费	amount	BigInteger	是	对应的业务费金额

3.1.10.2.5 删除 DDC 计费规则

3.1.10.2.5.1 功能说明

用于对 BSN-DDC-计费合约进行删除 DDC 计费规则所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.2.5.2 合约事件

➤ DelFee(address ddcAddr,bytes4 sig)

### 3.1.10.2.5.3 数据结构

字段名	字段	类型	必传	备注
业务合约	ddcAddr	String	是	业务合约地址
方法签名	sig	String	是	业务合约所对应的方法签名

### 3.1.10.2.6 删除 DDC 合约授权

#### 3.1.10.2.6.1 功能说明

用于对 BSN-DDC-计费合约进行按合约删除 DDC 计费规则所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.2.6.2 合约事件

➤ DelDDC(address ddcAddr)

#### 3.1.10.2.6.3 数据结构

字段名	字段	类型	必传	备注
业务合约	ddcAddr	String	是	业务合约地址

### 3.1.10.3 721 数据

#### 3.1.10.3.1 生成

##### 3.1.10.3.1.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。



### 3.1.10.3.1.2 合约事件

➤ Transfer(address(0),to,ddcId)

### 3.1.10.3.1.3 数据结构

字段名	字段	类型	必传	备注
接收账户地址	to	String	是	
DDCID	ddcId	BigInteger	是	

## 3.1.10.3.2 安全生成

### 3.1.10.3.2.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 安全生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.3.2.2 合约事件

➤ Transfer(address(0),to,ddcId)

### 3.1.10.3.2.3 数据结构

字段名	字段	类型	必传	备注
接收账户地址	to	String	是	
DDCID	ddcId	BigInteger	是	

## 3.1.10.3.3 批量生成

### 3.1.10.3.3.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 批量生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.3.3.2 合约事件

➤ TransferBatch(operator,address(0),to,ddclDs)

### 3.1.10.3.3.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	
接收账户地址	to	String	是	
DDCID 列表	ddclDs	List<BigInteger>	是	

## 3.1.10.3.4 批量安全生成

### 3.1.10.3.4.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 批量安全生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.3.4.2 合约事件

➤ TransferBatch(operator,address(0),to,ddclDs)

### 3.1.10.3.4.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	
接收账户地址	to	String	是	
DDCID 列表	ddclDs	List<BigInteger>	是	

### 3.1.10.3.5 转移

#### 3.1.10.3.5.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.5.2 合约事件

➤ Transfer(from, to, ddclid)

#### 3.1.10.3.5.3 数据结构

字段名	字段	类型	必传	备注
拥有账户地址	from	String	是	
接收账户地址	to	String	是	
DDCID	ddclid	BigInteger	是	

### 3.1.10.3.6 安全转移

#### 3.1.10.3.6.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 安全转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.6.2 合约事件

➤ Transfer(from, to, ddclid)

#### 3.1.10.3.6.3 数据结构

字段名	字段	类型	必传	备注
拥有账户地	from	String	是	

址				
接收账户地址	to	String	是	
DDCID	ddcid	BigInteger	是	

### 3.1.10.3.7 冻结

#### 3.1.10.3.7.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 解冻所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.7.2 合约事件

➤ EnterBlacklist(sender,ddcid)

#### 3.1.10.3.7.3 数据结构

字段名	字段	类型	必传	备注
签名者	sender	String	是	签名者账户所对应的账户地址
DDCID	ddcid	BigInteger	是	

### 3.1.10.3.8 解冻

#### 3.1.10.3.8.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 解冻所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.8.2 合约事件

➤ ExitBlacklist(sender,ddcid)

### 3.1.10.3.8.3 数据结构

字段名	字段	类型	必传	备注
签名者	sender	String	是	签名者账户所对应的账户地址
DDCID	ddcid	BigInteger	是	

### 3.1.10.3.9 销毁

#### 3.1.10.3.9.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 销毁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.9.2 合约事件

➤ Transfer(from,address(0),ddcid)

#### 3.1.10.3.9.3 数据结构

字段名	字段	类型	必传	备注
拥有者	from	String	是	DDC 拥有者所对应的账户地址
DDCID	ddcid	BigInteger	是	

### 3.1.10.3.10 URI 设置

#### 3.1.10.3.10.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 资源标识符设置所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.10.2 合约事件

➤ SetURI(uint256 indexed ddcid,string ddcURI)

### 3.1.10.3.10.3 数据结构

字段名	字段	类型	必传	备注
DDC 唯一标识	ddcId	BigInteger	是	
DDC 资源标识符	ddcURI	String	是	

### 3.1.10.3.11 跨链锁定

#### 3.1.10.3.11.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 跨链锁定所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.11.2 合约事件

➤ Locklist(address indexed operator,uint256 indexed ddcId)

#### 3.1.10.3.11.3 数据结构

字段名	字段	类型	必传	备注
操作者	operator	String	是	
DDC 唯一标识	ddcId	BigInteger	是	

### 3.1.10.3.12 跨链解锁

#### 3.1.10.3.12.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 跨链解锁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.12.2 合约事件

➤ UnLocklist(address indexed operator,uint256 indexed ddcId)

### 3.1.10.3.12.3 数据结构

字段名	字段	类型	必传	备注
操作者	operator	String	是	
DDC 唯一标识	ddcid	BigInteger	是	

### 3.1.10.3.13 元交易生成

#### 3.1.10.3.13.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 元交易生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.13.2 合约事件

➤ MetaTransfer(operator,address(0),to,ddcid)

#### 3.1.10.3.13.3 数据结构

字段名	字段	类型	必传	备注
操作人	operator	String	是	
接收账户地址	to	String	是	
DDCID	ddcid	BigInteger	是	

### 3.1.10.3.14 元交易安全生成

#### 3.1.10.3.14.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 元交易安全生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.14.2 合约事件

➤ MetaTransfer(operator,address(0),to,ddcid)

### 3.1.10.3.14.3 数据结构

字段名	字段	类型	必传	备注
操作人	operator	String	是	
接收账户地址	to	String	是	
DDCID	ddcid	BigInteger	是	

### 3.1.10.3.15 元交易批量生成

#### 3.1.10.3.15.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 元交易批量生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.3.15.2 合约事件

➤ MetaTransferBatch(operator,address(0),to,ddclds)

#### 3.1.10.3.15.3 数据结构

字段名	字段	类型	必传	备注
操作人	operator	String	是	
接收账户地址	to	String	是	
DDCID 列表	ddclds	List<BigInteger>	是	

### 3.1.10.3.16 元交易批量安全生成

#### 3.1.10.3.16.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 元交易批量安全生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。



### 3.1.10.3.16.2 合约事件

➤ MetaTransferBatch(operator,address(0),to,ddclds)

### 3.1.10.3.16.3 数据结构

字段名	字段	类型	必传	备注
操作人	operator	String	是	
接收账户地址	to	String	是	
DDCID 列表	ddclds	List<BigInteger>	是	

## 3.1.10.3.17 元交易转移

### 3.1.10.3.17.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 元交易转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.3.17.2 合约事件

➤ MetaTransfer(operator,from, to, ddclid)

### 3.1.10.3.17.3 数据结构

字段名	字段	类型	必传	备注
操作人	operator	String	是	
拥有账户地址	from	String	是	
接收账户地址	to	String	是	
DDCID	ddclid	BigInteger	是	

3.1.10.3.18 元交易安全转移

3.1.10.3.18.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 元交易安全转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.3.18.2 合约事件

➤ MetaTransfer(operator,from, to, ddclid)

3.1.10.3.18.3 数据结构

字段名	字段	类型	必传	备注
操作人	operator	String	是	
拥有账户地址	from	String	是	
接收账户地址	to	String	是	
DDCID	ddclid	BigInteger	是	

3.1.10.3.19 元交易销毁

3.1.10.3.19.1 功能说明

用于对 BSN-DDC-721 业务合约进行 DDC 元交易销毁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.3.19.2 合约事件

➤ MetaTransfer(operator,from,address(0),ddclid)

3.1.10.3.19.3 数据结构

字段名	字段	类型	必传	备注
-----	----	----	----	----

操作人	operator	String	是	
拥有者	from	String	是	DDC 拥有者所对应的账户地址
DDCID	ddcid	BigInteger	是	

### 3.1.10.4 1155 数据

#### 3.1.10.4.1 安全生成

##### 3.1.10.4.1.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 安全生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

##### 3.1.10.4.1.2 合约事件

➤ TransferSingle(operator,address(0),to,ddcid,amount)

##### 3.1.10.4.1.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
接收账户地址	to	String	是	
DDCID	ddcid	BigInteger	是	
数量	amount	BigInteger	是	

#### 3.1.10.4.2 批量安全生成

##### 3.1.10.4.2.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 批量安全生成所产生

的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.4.2.2 合约事件

➤ TransferBatch(operator,address(0),to,ddclds,amounts)

#### 3.1.10.4.2.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
接收账户地址	to	String	是	
DDC 集合	ddcs	Map<BigInteger, BigInteger>	是	Key: ddclid Value:amount

### 3.1.10.4.3 安全转移

#### 3.1.10.4.3.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 安全转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.4.3.2 合约事件

➤ TransferSingle(operator,from,to,ddclid,amount)

#### 3.1.10.4.3.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	

接收账户地址	to	String	是	
DDCID	ddcid	BigInteger	是	
数量	amount	BigInteger	是	

#### 3.1.10.4.4 批量安全转移

##### 3.1.10.4.4.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 批量安全转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

##### 3.1.10.4.4.2 合约事件

➤ TransferBatch(operator,from,to,ddclds,amounts)

##### 3.1.10.4.4.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	
接收账户地址	to	String	是	
DDC 集合	ddcs	Map<BigInteger, BigInteger>	是	Key: ddcld Value:amount

#### 3.1.10.4.5 冻结

##### 3.1.10.4.5.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 冻结所产生的交易回

执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.4.5.2 合约事件

➤ EnterBlacklist(sender,ddcid)

3.1.10.4.5.3 数据结构

字段名	字段	类型	必传	备注
签名者	sender	String	是	签名者账户所对应的账户地址
DDCID	ddcid	BigInteger	是	

3.1.10.4.6 解冻

3.1.10.4.6.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 解冻所产生的交易回  
执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.4.6.2 合约事件

➤ ExitBlacklist(sender,ddcid)

3.1.10.4.6.3 数据结构

字段名	字段	类型	必传	备注
签名者	sender	String	是	签名者账户所对应的账户地址
DDCID	ddcid	BigInteger	是	

3.1.10.4.7 销毁

3.1.10.4.7.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 销毁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.4.7.2 合约事件

➤ TransferSingle(operator,from,address(0),ddcId,amount)

3.1.10.4.7.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	
DDCID	ddcId	BigInteger	是	
数量	amount	BigInteger	是	

3.1.10.4.8 批量销毁

3.1.10.4.8.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 批量销毁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

3.1.10.4.8.2 合约事件

➤ TransferBatch(operator,from,address(0),ddcIds,amounts)

3.1.10.4.8.3 数据结构

字段名	字段	类型	必传	备注
-----	----	----	----	----

签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	
DDC 集合	ddcs	Map<BigInteger, BigInteger>	是	Key: ddclId Value: amount

### 3.1.10.4.9 URI 设置

#### 3.1.10.4.9.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 资源标识符设置所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.4.9.2 合约事件

- SetURI(address indexed owner,uint256 indexed ddclId,string ddcURI)

#### 3.1.10.4.9.3 数据结构

字段名	字段	类型	必传	备注
拥有者	owner	String	是	
DDC 唯一标识	ddclId	BigInteger	是	DDC 唯一标识
DDC 资源标识符	ddcURI	String	是	

### 3.1.10.4.10 跨链锁定

#### 3.1.10.4.10.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 跨链锁定所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。



### 3.1.10.4.10.2 合约事件

- Locklist(address indexed operator,uint256 indexed ddclid)

### 3.1.10.4.10.3 数据结构

字段名	字段	类型	必传	备注
操作者	operator	String	是	
DDC 唯一标识	ddclid	BigInteger	是	

## 3.1.10.4.11 跨链解锁

### 3.1.10.4.11.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 跨链解锁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.4.11.2 合约事件

- UnLocklist(address indexed operator,uint256 indexed ddclid)

### 3.1.10.4.11.3 数据结构

字段名	字段	类型	必传	备注
操作者	operator	String	是	
DDC 唯一标识	ddclid	BigInteger	是	

## 3.1.10.4.12 元交易安全生成

### 3.1.10.4.12.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 元交易安全生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.4.12.2 合约事件

➤ MetaTransferSingle(operator,address(0),to,ddcld,amount)

### 3.1.10.4.12.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
接收账户地址	to	String	是	
DDCID	ddcld	BigInteger	是	
数量	amount	BigInteger	是	

## 3.1.10.4.13 元交易批量安全生成

### 3.1.10.4.13.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 元交易批量安全生成所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.4.13.2 合约事件

➤ MetaTransferBatch(operator,address(0),to,ddclds,amounts)

### 3.1.10.4.13.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
接收账户地址	to	String	是	
DDC 集	ddcs	Map<BigInteger,	是	Key: ddcld

合		BigInteger>		Value:amount
---	--	-------------	--	--------------

### 3.1.10.4.14 元交易安全转移

#### 3.1.10.4.14.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 元交易安全转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.4.14.2 合约事件

➤ MetaTransferSingle(operator,from,to,ddcId,amount)

#### 3.1.10.4.14.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	
接收账户地址	to	String	是	
DDCID	ddcId	BigInteger	是	
数量	amount	BigInteger	是	

### 3.1.10.4.15 元交易批量安全转移

#### 3.1.10.4.15.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 元交易批量安全转移所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.4.15.2 合约事件

➤ MetaTransferBatch(operator,from,to,ddclids,amounts)

### 3.1.10.4.15.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	
接收账户地址	to	String	是	
DDC 集合	ddcs	Map<BigInteger, BigInteger>	是	Key: ddclId Value:amount

## 3.1.10.4.16 元交易销毁

### 3.1.10.4.16.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 元交易销毁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.4.16.2 合约事件

➤ MetaTransferSingle(operator,from,address(0),ddclId,amount)

### 3.1.10.4.16.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	

DDCID	ddcid	BigInteger	是	
数量	amount	BigInteger	是	

### 3.1.10.4.17 元交易批量销毁

#### 3.1.10.4.17.1 功能说明

用于对 BSN-DDC-1155 业务合约进行 DDC 元交易批量销毁所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

#### 3.1.10.4.17.2 合约事件

➤ MetaTransferBatch(operator,from,address(0),ddclds,amounts)

#### 3.1.10.4.17.3 数据结构

字段名	字段	类型	必传	备注
签名者	operator	String	是	签名者账户所对应的账户地址
拥有账户地址	from	String	是	
DDC 集合	ddcs	Map<BigInteger, BigInteger>	是	Key:ddcid Value:amount

### 3.1.10.5 跨链应用

#### 3.1.10.5.1 基础数据设置

##### 3.1.10.5.1.1 功能说明

用于对跨链应用合约进行基础数据设置所产生的交易回执中的事件进行解析，并组装成所对应的数据结构。

### 3.1.10.5.1.2 合约事件

- SetBaseData(address indexed operator,address indexed eccmpAddress,uint64 fromChainID)

### 3.1.10.5.1.3 数据结构

字段名	字段	类型	必传	备注
操作者	operator	String	是	
跨链管理代理合约	eccmpAddress	String	是	
原链链 ID	fromChainID	BigInteger	是	

## 3.1.10.5.2 DDC 跨链流转

### 3.1.10.5.2.1 功能说明

用于对跨链应用合约进行 DDC 跨链流转所产生的交易回执中的事件进行解析，并组装成所对应的数据结构

### 3.1.10.5.2.2 合约事件

- CrossChainTransfer(address indexed operator, uint256 crossChainId,DDCType ddcType, address signer,address to,uint256 ddclId,string ddcURI,uint256 amount,uint64 fromChainID,uint64 toChainID,string fromCCAddr,string toCCAddr,uint256 crossChainFee)

### 3.1.10.5.2.3 数据结构

字段名	字段	类型	必传	备注
操作者	operator	String	是	
跨链 ID	crossChainID	BigInteger	否	

DDC 类型	ddcType	DDCType	是	
目标链签名者账户	signer	String	是	
目标链接收者账户	to	String	是	
DDC 唯一标识	ddcId	BigInteger	是	
DDC 资源标识符	ddcURI	String	是	
数量	amount	BigInteger	否	
原链链 ID	fromChainID	BigInteger	是	
目标链链 ID	toChainID	BigInteger	是	
原链应用合约地址	fromCCAddr	String	是	
目标链应用合约地址	toCCAddr	String	是	
跨链业务费	crossChainFee	BigInteger	是	

### 3.1.10.5.3 DDC 跨链通知

#### 3.1.10.5.3.1 功能说明

用于对跨链应用合约进行 DDC 跨链通知所产生的交易回执中的事件进行解析，并组装成所对应的数据结构

#### 3.1.10.5.3.2 合约事件

- UpdateCrossChainStatus(address indexed operator,uint256 crossChainID,State state,string remark)

### 3.1.10.5.3.3 数据结构

字段名	字段	类型	必传	备注
操作者	operator	String	是	
跨链 ID	crossChainID	BigInteger	是	
状态	state	State	是	1: 跨链成功 2: 跨链失败
备注	remark	String	是	

## 4. 附录

## 4.1 区块信息示例

#### 4.1.1 泰安链

```
{
  "dbHash":
    "0x0000000000000000000000000000000000000000000000000000000000000000",
  "extraData": [],
  "gasLimit": "0x0",
  "gasUsed": "0x0",
  "hash":
    "0xfa639d1454362a8cdfcab1ca1948a5defaf7048b28f67e80780ab1e24e8f8c59",
  "logsBloom":
    "0x0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "0000000000000000000000000000000000000000000000000000000000000000",
    "00",
  "number": "0x1",
  "parentHash":
    "0x249f59e00beac8424a7821c4750fdd70c128f4ce795afbab53f345e9fce95d1a"
```



```
    "receiptsRoot":
      "0x69a04fa6073e4fc0947bac7ee6990e788d1e2c5ec0fe6c2436d0892e7f3c09d2"
    ,
    "sealer": "0x0",
    "sealerList": [

      "4ca3a91a4937355dba6a2e5fe76141479a1fc44e9caa86750092dab64e0b8382f6
      b8476749c2d2de414350a54491620d38813d2a1442f524e36e3d9946109c4d"
    ],
    "signatureList": [
      {
        "index": "0x0",
        "signature":
          "0x4602135870d9a4846e2536d4a48e831825a5d95768dd0d4f08544a0bd4c2af
          41242dec1751a05c07d7572027f8d6ac1625c48145beb004e2dce8b7ce9e2bb73
          d00"
      }
    ],
    "stateRoot":
      "0x0000000000000000000000000000000000000000000000000000000000000000
      00",
    "timestamp": "0x175ac38cf10",
    "transactions": [
      {
        "blockHash":
          "0xfa639d1454362a8cdfcab1ca1948a5defaf7048b28f67e80780ab1e24e8f8c59",
        "blockLimit": "0x100",
        "blockNumber": "0x1",
        "chainId": "0x1",
        "extraData": "0x",
        "from": "0x57c7be32cbfb3bfed4fddc87efcc735b4e945fb3",
        "gas": "0x2faf080",
        "gasPrice": "0xa",
        "groupId": "0x1",
        "hash":
          "0x3961fac263d8e640b148ddcfafd71d2069e93a006abc937c32fb16cfa96e661d"
      }
    ],
    "input":
      "0x4ed3885e00000000000000000000000000000000000000000000000000000000
      0000000020000000000000000000000000000000000000000000000000000000
      000000000a464953434f2042434f530000000000000000000000000000000000
      000000000",
```



[illegible]

```
000a000000000000000000000000000000000000000000000000000000000000  
0020000000000000000000000000000000000000000000000000000000000000  
4000000000000000000000000000000000000000000000000000000000000008  
0000000000000000000000000000000000000000000000000000000000000005  
6464632d3200000000000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000000000000000564  
64632d3100000000000000000000000000000000000000000000000000000000",  
    "nonce": "0x1a",  
    "to": "0x45bcf2855649fb116c4623f8f32091476933fe3",  
    "transactionIndex": "0x0",  
    "value": "0x0",  
    "type": "0x0",  
    "v": "0x2b89",  
    "r":  
"0x6d0cb65392d3857e0ccb82f7c372b628074ec2d74bfde2230057cce97429aad"  
,"  
    "s":  
"0x13cef22abd8d4e81eb875430706cd9cc3513900854d5474cd7a08b5751e1af97"  
"}  
],  
    "transactionsRoot":  
"0xd7cd21ddebd2710404c7f075f5b66c3d626aad5819c757dac5b50e99007ba196"  
1",  
    "uncles": []  
}  
  
}
```

### 4.1.3 文昌链

[illegible]



```
"contractAddress":  
    "0x0000000000000000000000000000000000000000000000000000000000000000",  
    "from": "0xcdce60801c0a2e6bb534322c32ae528b9dec8d2",  
    "gasUsed": "0x1fb8d",  
    "input":  
        "0xb602109a00000000000000000000000000000000000000000000000000000000  
00000000080000000000000000000000000000000000000000000000000000000000000000  
00000000c00000000000000000000000000000000000000000000000000000000000000000  
00000001000000000000000000000000000000000000000000000000000000000000000000  
00000001400000000000000000000000000000000000000000000000000000000000000000  
0000000203078313030303030303030303030303030303030303030303030303030303030303  
03030303030000000000000000000000000000000000000000000000000000000000000000  
00000000832303139303733300000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000000000000000000000000000  
00000000026161000000000000000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000000000000000000000000000  
00000000026262000000000000000000000000000000000000000000000000000000000000  
0000000000",  
    "logs": [ ],  
    "logsBloom":  
        "0x0000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
00",  
    "output":  
        "0x0000000000000000000000000000000000000000000000000000000000000000  
00",  
  
    "root":"0x38723a2e5e8a17aa7950dc008209944e898f69a7bd10a23c839d341e9  
35fd5ca",  
    "status": "0x0",  
    "to": "0x15538acd403ac1b2ff09083c70d04856b8c0bdfd",  
    "transactionHash":  
        "0x708b5781b62166bd86e543217be6cd954fd815fd192b9a124ee9327580df8f3  
f",  
    "transactionIndex": "0x0"  
}
```

### 4.2.2 武汉链

[illegible]

[illegible]



### 4.2.3 文昌链

[illegible]

[illegible]

}

```
{  
    "blockHash":  
        "0xfa639d1454362a8cdfcab1ca1948a5defaf7048b28f67e80780ab1e24e8f8c59",  
    "blockLimit": "0x100",  
    "blockNumber": "0x1",  
    "chainId": "0x1",  
    "extraData": "0x",  
    "from": "0x57c7be32cbfb3bfed4fdde87efcc735b4e945fb3",  
    "gas": "0x2faf080",  
    "gasPrice": "0xa",  
    "groupId": "0x1",  
    "hash":  
        "0x3961fac263d8e640b148ddcfafd71d2069e93a006abc937c32fb16cfa96e661d"  
    ,  
    "input":  
        "0x4ed3885e00000000000000000000000000000000000000000000000000000000  
0000000020000000000000000000000000000000000000000000000000000000000000  
000000000a464953434f2042434f530000000000000000000000000000000000000000  
0000000000",  
    "nonce":  
        "0x3eb675ec791c2d19858c91d0046821c27d815e2e9c15160491220500000296  
8",  
    "signature": {  
        "r":  
            "0x9edf7c0cb63645442aff11323916d51ec5440de979950747c0189f338afdcefd",  
        "s":  
            "0x2f3473184513c6a3516e066ea98b7cfb55a79481c9db98e658dd016c37f03dcf  
",  
        "signature":  
            "0x9edf7c0cb63645442aff11323916d51ec5440de979950747c0189f338afdcefd2  
f3473184513c6a3516e066ea98b7cfb55a79481c9db98e658dd016c37f03dcf00",  
        "v": "0x0"  
    },  
    "to": "0x8c17cf316c1063ab6c89df875e96c9f0f5b2f744",  
    "transactionIndex": "0x0",
```

```
"value": "0x0"
}
```

### 4.3.2 武汉链

[illegible]

### 4.3.3 文昌链

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "result": {
    "blockHash":
```

[illegible]

## 4.4 离线生成账户

### 4.4.1 功能介绍

平台方或终端用户可以通过此方法生成离线账户

### 4.4.2 API 定义

- 方法定义：Account createAccount();
- 调用者：平台方、终端用户；
- 核心逻辑：无
- 输入参数：

字段名	字段	类型	必传	备注

➤ 输出参数：

字段名	字段	类型	必传	备注
账户地址	address	String	是	
公钥	publicKey	String	是	
私钥	privateKey	String	是	
助记词	mnemonic	String	是	

## 4.5 接入 Url 设置

### 4.5.1 功能介绍

平台方或终端用户可以通过此方法对接入的网关 URL 地址进行设置

### 4.5.2 API 定义

➤ 方法定义：Boolean setGatewayUrl(String gatewayUrl);

➤ 调用者：平台方、终端用户；

➤ 核心逻辑：无

➤ 输入参数：

字段名	字段	类型	必传	备注
网关 URL	gatewayUrl	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
设置结果		Boolean	是	

## 4.6 接入 Key 设置

### 4.6.1 功能介绍

平台方或终端用户可以通过此方法对接入的 Key 进行设置，注：apiKey  
在设置的时候 KEY 用“x-api-key”，值根据实际情况填写。

### 4.6.2 API 定义

- 方法定义：Boolean setGatewayApiKey(String apiKey);
- 调用者：平台方、终端用户；
- 核心逻辑：无
- 输入参数：

字段名	字段	类型	必传	备注
接入 KEY	apiKey	String	是	

- 输出参数：

字段名	字段	类型	必传	备注
设置结果		Boolean	是	

## 4.7 GAS 查询

### 4.7.1 功能介绍

平台方或终端用户可以通过此方法查询链账户所对应的 GAS 费余额。

### 4.7.2 API 定义

- 方法定义：BigInteger BalanceOfGas(String account);

➤ 调用者：平台方、终端用户；

➤ 核心逻辑：

1. 检查 account 为标准 address 地址，并且不能为 0 地址；

➤ 输入参数：

字段名	字段	类型	必传	备注
链账户	account	String	是	

➤ 输出参数：

字段名	字段	类型	必传	备注
GAS 余额		BigInteger	是	