MPC APIs Requirement

Endpoint details

Wallet Create Endpoint

1. Wallet creation begins with generating the master key at the root (“m”) using a random mnemonic.

POST /rwcore/api/v1/mpc/wallets/create

Header:

```
X-RW-Device-ID
X-RW-Client-ID - M
X-RW-Request-ID
X-RW-Session-ID - M
Authorization: Bearer ACCESS_TOKEN - M
X-RW-Correlation-ID
X-RW-Forwarded-Proto
X-RW-Forwarded-Port
X-Forwarded-For
User-Agent
Content-Type
Connection
Accept
Host
Date
```

Request:

```
{
  "wallet_id": "<wallet id>", -M
}

Example:
{
  "wallet_id": "44e1d20b-70b7-42c8-a6a4-ef8e9bc667af",
}

```

Success response: 200

```
{
  "result": "success",
  "code": "RW_SUCCESS",
  "msg": "account xpub generated successfully",
  "data": {
     "wallet_id": "<wallet id>",
     "wallet_key": "<shard 3>",
     "xpub_hash": "<master key xpub>"
  }
```

```
}

Example:
{
  "result": "success",
  "code": "RW_SUCCESS",
  "msg": "account xpub generated successfully",
  "data": {
     "wallet_id": "44e1d20b-70b7-42c8-a6a4-ef8e9bc667af",
     "wallet_key": "<shard 3>",
     "xpub_hash": "<master key xpub>"
  }
}

```

Error response:

```
Scenario - 1 - 400
{
  "result": "error",
  "code": "VALIDATION_ERROR",
  "msg": "validation error",
  "errors": [
    {
      "code": "REQUIRED_FIELD_MISSING_ERROR",
      "err_msg": "<field_name> field is required"
    },
    {
      "code": "INVALID_INPUT_ERROR",
      "err_msg": "session id is invalid"
    }
  ]
}

Scenario - 2 - 500
{
  "result": "error",
  "code": "DB_ERROR",
  "msg": "database error",
  "errors": [
    {
      "code": "DB_CONN_ERROR",
      "err_msg": "unable to connect to database"
    }
  ]
}

Scenario - 3 - 412
{
  "result": "error",
  "code": "HEADER_VALIDATION_ERROR",
  "msg": "header validation error",
  "errors": [
    {
      "code": "REQUIRED_HEADER_MISSING_ERROR",
      "err_msg": "<header_name> header is required"
    }
  ]
}

```

```
Scenario - 4 - 401
{
  "result": "error",
  "code": "UNAUTHORIZED_ACCESS_ERROR",
  "msg": "unauthorized access error",
  "errors": [
    {
      "code": "INVALID_TOKEN_ERROR",
      "err_msg": "access token is invalid or expired"
    }
  ]
}

```

Wallet Recovery Endpoint

POST /rwcore/api/v1/mpc/wallets/recovery

Header:

```
X-RW-Device-ID
X-RW-Client-ID - M
X-RW-Request-ID
X-RW-Session-ID - M
Authorization: Bearer ACCESS_TOKEN - M
X-RW-Correlation-ID
X-RW-Forwarded-Proto
X-RW-Forwarded-Port
X-Forwarded-For
User-Agent
Content-Type
Connection
Accept
Host
Date
```

Request:

```
{
  "wallet_id": "<wallet id>", -M
  "xpub_hash": "<master key xpub hash>" -M
}

Example:
{
  "wallet_id": "44e1d20b-70b7-42c8-a6a4-ef8e9bc667af",
  "xpub_hash": "<master key xpub hash>"
}

```

Success response: 200

```
{
  "result": "success",
  "code": "RW_SUCCESS",
  "msg": "account xpub generated successfully",
  "data": {
```

```
    "wallet_id": "<wallet id>",
    "wallet_key": "<shard 3>",
  }
}

Example:
{
  "result": "success",
  "code": "RW_SUCCESS",
  "msg": "account xpub generated successfully",
  "data": {
    "wallet_id": "44e1d20b-70b7-42c8-a6a4-ef8e9bc667af",
    "wallet_key": "<shard 3>",
  }
}

```

Error response:

```
Scenario - 1 - 400
{
  "result": "error",
  "code": "VALIDATION_ERROR",
  "msg": "validation error",
  "errors": [
    {
      "code": "REQUIRED_FIELD_MISSING_ERROR",
      "err_msg": "<field_name> field is required"
    },
    {
      "code": "INVALID_INPUT_ERROR",
      "err_msg": "session id is invalid"
    }
  ]
}

Scenario - 2 - 500
{
  "result": "error",
  "code": "DB_ERROR",
  "msg": "database error",
  "errors": [
    {
      "code": "DB_CONN_ERROR",
      "err_msg": "unable to connect to database"
    }
  ]
}

Scenario - 3 - 412
{
  "result": "error",
  "code": "HEADER_VALIDATION_ERROR",
  "msg": "header validation error",
  "errors": [
    {
      "code": "REQUIRED_HEADER_MISSING_ERROR",
      "err_msg": "<header_name> header is required"
    }
  ]
```

```
}

Scenario - 4 - 401
{
  "result": "error",
  "code": "UNAUTHORIZED_ACCESS_ERROR",
  "msg": "unauthorized access error",
  "errors": [
    {
      "code": "INVALID_TOKEN_ERROR",
      "err_msg": "access token is invalid or expired"
    }
  ]
}

```

Account XPUB generation Endpoint

1. Using all shards, generate the account xpub for each account based on the HD path provided in the API request.

POST /rwcore/api/v1/mpc/wallets/{wallet_id}/accounts/xpub

Header:

```
X-RW-Device-ID
X-RW-Client-ID - M
X-RW-Request-ID
X-RW-Session-ID - M
Authorization: Bearer ACCESS_TOKEN - M
X-RW-Correlation-ID
X-RW-Forwarded-Proto
X-RW-Forwarded-Port
X-Forwarded-For
User-Agent
Content-Type
Connection
Accept
Host
Date
```

Parameters:

```
wallet_id: path parameter, wallet id belongs to the user and created at the time
          of wallet creation step
```

Request:

```
{
  "wallet_key": "<shard 3>", -M
  "accounts": [ -M
    {
      "account_id": "<account id>", -M
      "path": "<full HD path of account>" -M
    },
```

```
    {
      "account_id": "<account id>",
      "path": "<full HD path of account>"
    }
  ]
}

Example:
{
  "wallet_key": "<shard 3>",
  "accounts": [
    {
      "account_id": "8d022687-2a29-40ec-bd17-6831e495e6f5",
      "path": "m/44'/236'/0'"
    },
    {
      "account_id": "52e1d20b-70b7-42c8-a6a4-ef8e9bc667af",
      "path": "m/44'/0'/0'"
    }
    {
      "account_id": "c05cc7e5-e041-4196-90a6-aabc2335a16c",
      "path": "m/44'/60'/0'"
    }
  ]
}

```

Success response: 200

```
{
  "result": "success",
  "code": "RW_SUCCESS",
  "msg": "account xpub generated successfully",
  "data": {
     "accounts": [
      {
        "account_id": "<account id>",
        "path": "<full HD path of account>",
        "xpub": "<xpub generated for the account using HD path>"
      },
      {
        "account_id": "<account id>",
        "path": "<full HD path of account>",
        "xpub": "<xpub generated for the account using HD path>"
      }
    ]
  }
}

Example:
{
  "result": "success",
  "code": "RW_SUCCESS",
  "msg": "account xpub generated successfully",
  "data": {
     "accounts": [
      {
        "account_id": "8d022687-2a29-40ec-bd17-6831e495e6f5",
        "path": "m/44'/236'/0'",
```

```
        "xpub":
"xpub67qXPFNYeFhsct27oJDmerxuU8rjdWzKRRi5tStp8JaWZaEqhDz8SMjQvcFqBr8ZT
AL1U3YwGUmzpQd4SS4LqfcwUoc6ztfYgZLHgGXtjhZ"
      },
      {
        "account_id": "52e1d20b-70b7-42c8-a6a4-ef8e9bc667af",
        "path": "m/44'/0'/0'",
        "xpub":
"xpub67qXPFNYeFhsct27oJDmerxuU8rjdWzKRRi5tStp8JaWZaEqhDz8SMjQvcFqBr8ZT
AL1U3YwGUmzpQd4SS4LqfcwUoc6ztfYgZLHgGXtjhZ"
      }
      {
        "account_id": "c05cc7e5-e041-4196-90a6-aabc2335a16c",
        "path": "m/44'/60'/0'",
        "xpub":
"xpub67qXPFNYeFhsct27oJDmerxuU8rjdWzKRRi5tStp8JaWZaEqhDz8SMjQvcFqBr8ZT
AL1U3YwGUmzpQd4SS4LqfcwUoc6ztfYgZLHgGXtjhZ"
      }
    ]
  }
}


```

Error response:

```
Scenario - 1 - 400
{
  "result": "error",
  "code": "VALIDATION_ERROR",
  "msg": "validation error",
  "errors": [
    {
      "code": "REQUIRED_FIELD_MISSING_ERROR",
      "err_msg": "<field_name> field is required"
    },
    {
      "code": "INVALID_INPUT_ERROR",
      "err_msg": "session id is invalid"
    },
    {
      "code": "INVALID_INPUT_ERROR",
      "err_msg": "wallet id is invalid"
    }
  ]
}

Scenario - 2 - 500
{
  "result": "error",
  "code": "DB_ERROR",
  "msg": "database error",
  "errors": [
    {
      "code": "DB_CONN_ERROR",
      "err_msg": "unable to connect to database"
    }
  ]
}

```

```
Scenario - 3 - 412
{
  "result": "error",
  "code": "HEADER_VALIDATION_ERROR",
  "msg": "header validation error",
  "errors": [
    {
      "code": "REQUIRED_HEADER_MISSING_ERROR",
      "err_msg": "<header_name> header is required"
    }
  ]
}

Scenario - 4 - 401
{
  "result": "error",
  "code": "UNAUTHORIZED_ACCESS_ERROR",
  "msg": "unauthorized access error",
  "errors": [
    {
      "code": "INVALID_TOKEN_ERROR",
      "err_msg": "access token is invalid or expired"
    }
  ]
}

```

Transaction Signature Endpoint

1. The full HD path for the address is derived by combining the provided account path and address path.
2. Derive the private key of addresses dynamically using HD path of address and master key.
3. For the Bitcoin bases asset, a list of UTXOs included as transaction inputs will be provided.
4. For the account based chain, account_chain_details will be provided.

POST /rwcore/api/v1/mpc/wallets/{wallet_id}/transactions/sign

Header:

```
X-RW-Device-ID
X-RW-Client-ID - M
X-RW-Request-ID
X-RW-Session-ID - M
Authorization: Bearer ACCESS_TOKEN - M
X-RW-Correlation-ID
X-RW-Forwarded-Proto
X-RW-Forwarded-Port
X-Forwarded-For
User-Agent
Content-Type
Connection
Accept
Host
Date
```

Parameters:

```
wallet_id: path parameter, wallet id belongs to the user and created at the time
          of wallet creation step
```

Request:

```
{
     "tx_id": "<internal transaction id>" -M (mandatory)
     "tx_data": <unsigned transaction hex>, -M
     "wallet_key": "<shard 3>", -M
     "blockchain_type": "<>" -M //UTXO_BASED, ACCOUNT_BASED
     "network_fee": <>, -M
     "account_path": "<account path>", // m/44'/236'/0'
     "utxos": [  -O <this field will come only for UTXO based currency like BSV, BTC..>
       {
         "tx_hash": "<utxo parent transaction hash>",
         "vout": <utxo output index>,
         "script_pub_key_hex": "<utxo scriptPbKeyHex>",
         "value": <utxo amount>,
         "address_path": "<address path after account - 0/1>"
       }
     ],
     "account_chain_details": { -O //this object comes if blockchain_type value is ACCOUNT_BASED for phase-2
       "address": "<address>",
       "address_path": "<address path after account - 0/1>",
       "chain_id": <chain id> //example chain id for eth is 1
     }
  }
```

Success response: 200

```
{
  "result": "success",
  "code": "RW_SUCCESS",
  "msg": "transation signed successfully",
  "data": {
     "tx_id": "<internal transaction id>" -M
     "tx_data": <signed transaction hex>, -M
  }
}
```

Error response:

```
Scenario - 1 - 400
{
  "result": "error",
  "code": "VALIDATION_ERROR",
  "msg": "validation error",
  "errors": [
    {
      "code": "REQUIRED_FIELD_MISSING_ERROR",
      "err_msg": "<field_name> field is required"
    },
    {
```

```
      "code": "INVALID_INPUT_ERROR",
      "err_msg": "session id is invalid"
    },
    {
      "code": "INVALID_INPUT_ERROR",
      "err_msg": "wallet id is invalid"
    },
    {
      "code": "INVALID_INPUT_ERROR",
      "err_msg": "account id is invalid"
    }
  ]
}

Scenario - 2 - 500
{
  "result": "error",
  "code": "DB_ERROR",
  "msg": "database error",
  "errors": [
    {
      "code": "DB_CONN_ERROR",
      "err_msg": "unable to connect to database"
    }
  ]
}

Scenario - 3 - 412
{
  "result": "error",
  "code": "HEADER_VALIDATION_ERROR",
  "msg": "header validation error",
  "errors": [
    {
      "code": "REQUIRED_HEADER_MISSING_ERROR",
      "err_msg": "<header_name> header is required"
    }
  ]
}

Scenario - 4 - 401
{
  "result": "error",
  "code": "UNAUTHORIZED_ACCESS_ERROR",
  "msg": "unauthorized access error",
  "errors": [
    {
      "code": "INVALID_TOKEN_ERROR",
      "err_msg": "access token is invalid or expired"
    }
  ]
}

```