Hyperledger Avalon

0.5.0.dev1

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hyperledger Avalon Documentation

Introduction

- README. Overview of Avalon and its source code
- FAQ. Frequently-asked questions with answers about Avalon
  - Glossary
  - Videos
- Avalon Proposal. Avalon Proposal, initial members, motivation, and proposed solutions (2019)
- "Ecosystem Support for EEA Trusted Compute Specification v1.0 Improves Blockchain Privacy and Scalability". Introductory blog by Michael Reed (2019)
- Introduction to Hyperledger Avalon video (20:24)

Community

- Project Wiki
- RocketChat
- Email list
- JIRA feature & bug tracking

Tutorial

- Workload Application Tutorial
- Example Applications

Source Code

- Avalon source code repository, https://github.com/hyperledger/avalon
- Building source code
- Example Avalon applications
- Contributing source code

SDK Reference Manual

The Avalon SDK Reference Manual is generated with Doxygen. To generate the Reference Manual, type the following:

```
cd $TCF_HOME/docs # this directory
sudo apt-get update
sudo apt-get install -y make doxygen texlive-full graphviz
make
```

Documentation generated will be here:

- `$TCF_HOME/docs/refman/html/`HTML documentation
- `$TCF_HOME/docs/refman/man/man3`man page documentation
- `$TCF_HOME/docs/refman/latex`LaTeX documentation
- `$TCF_HOME/docs/refman/refman.pdf`PDF documentation (generated from LaTex)

Reference

- *Hyperledger Avalon Architecture Overview*. Overview of Avalon architecture by Eugene Yarmosh (2020)
- *Off-Chain Trusted Compute Specification* defined by Enterprise Ethereum Alliance (EEA) Task Force
- Cryptography. Cryptographic primitives used, libraries used, and implementation

Chapter 2

BUILD

Building the common libraries

Make sure you have environment variables `SGX_SDK` and `SGX_SSL` defined (see ../../../BUILD.md "BUILD.md") and then run

```
mkdir build
cd build
cmake .. -G "Unix Makefiles" && make
```

Chapter 3

README

Common Crypto library documentation

This `common/cpp/crypto/` directory contains cryptographic code used by untrusted and trusted (Intel SGX Enclave) code.

This code is written in C++, but a Python wrapper is also available (see ../README.md "../README.md")

Avalon applications are free to use third-party cryptographic implementations (such as what a programming language binding may provide) or the cryptographic interfaces provided here.

Software Components Required

OpenSSL 1.1 library and Intel SGX OpenSSL library built from OpenSSL 1.1:

- `https://www.openssl.org/`
- `https://github.com/intel/intel-sgx-ssl`

Cryptographic Primitives Used

Primitive	Algorithm	Keysize	Comments
Digital signature	ECDSA-SECP256K1	256	(1) (2)
Asymmetric encryption	RSA-OAEP	3072	(1)
Authenticated encryption	AES-GCM	256	96b IV, 128b tag
Digest	SHA-256	256	(2)
Digest	KECCACK	256	(2) Differs from SHA-3

(1) Not PQ resistant

(2) Blockchain legacy algorithm

Cryptographic Primitive Usage

- **SHA-256** Computing digests of the work order request and response
- **KECCAK-256** Computing digests of the work order request and response or Ethereum raw transactions Packet bytes
- **AES-GCM-256** Encrypts data items within work order request and response. It also used to encrypt a request digest and custom data encryption keys
- **RSA-OAEP-3072** Encrypt symmetric data encryption keys
- **ECSDA-SECP256K1** Signs work order response digest and worker's encryption RSA-OAEP public key

Implementation of Cryptographic Elements

Cryptographic elements include cryptographic keys, signature, ciphertexts, plaintexts, hashes, and random bitstrings.

Element	Implementation	Representation	Serialize/Deserialize function?
ECDSA public key	C++ class	Custom object	Yes, PEM encoding and 65-byte Bitcoin Hex format
ECDSA private key	C++ class	Custom object	Yes, PEM encoding
ECDSA signature	C++ string	DER binary	No, user defined
RSA public key	C++ class	Custom object	Yes, PEM encoding
RSA private key	C++ class	Custom object	Yes, PEM encoding
RSA ciphertext	C++ string	raw binary	No, user defined
RSA plaintext	C++ string	raw binary	No, user defined
AES-GCM key	C++ string	raw binary	No, user defined
AES-GCM iv	C++ string	raw binary	No, user defined
AES-GCM ciphertext+tag	C++ string	raw binary	No, user defined
AES-GCM plaintext	C++ string	raw binary	No, user defined
SHA-256 digest	C++ string	raw binary	No, user defined
Random bitstring	C++ string	raw binary	No, user defined

Security notes

- **AES-GCM** When using of AES-GCM inside Intel SGX enclaves to preserve confidentiality and integrity of data to be stored outside of the Intel SGX enclaves a different unique or random 12-byte IV must be used for each encrypted message. At most 2^32 distinct IVs can be used until the key needs to be regenerated for security. This limitation can possibly be mitigated in the future by using alternatives to AES-GCM like AES-GCM SIV.

Chapter 4

README

Purpose of Common

The common directory contains source code shared by untrusted and trusted (Intel SGX Enclave) code.

Dependencies:

1. OpenSSL 1.1 https://www.openssl.org/
2. Intel SGX OpenSSL library built from OpenSSL 1.1 https://github.com/intel/intel-sgx-ssl
3. Intel SGX SDK https://software.intel.com/en-us/sgx-sdk/download

Source Directories

Dir	Content
`crypto/`	*.cpp,*.h for OpenSSL based crypto functions. For more information, see crypto/README.md
`.`	*.cpp,*.h error handling and common types
`packages/base64/`	*.cpp,*.h of Renee Nyffinger base64 encoding/decoding
`packages/parson/`	*.cpp,*.h of Parson JSON encoding/decoding

Python Wrapper

The Python SWIG wrapper exports the functions and classes defined in crypto.h, tcf_error.h and types.h. Several classes and functions are renamed. Check `common/python/crypto_utils/crypto/crypto.i` for details.

Chapter 5

README

Purpose of Common

The common directory contains source code shared by trusted (Intel SGX Enclave) code and different workloads(↩Example workloads).

Dependencies:

1. SGX SDK `https://software.intel.com/en-us/sgx-sdk/download`
2. SGX OpenSSL library built from OpenSSL 1.1 `https://github.com/intel/intel-sgx-ssl`

Source Directories

Dir Content

`sgx/iohandler/` *.cpp,*.h files are custom iohandlers which help workloads to execute IO operations from the Intel SGX enclave

`sgx/workload/` work_order_data.cpp,work_order_data.h files are wrapper files for work order data workload↩_processor.cpp, workload_processor.h are workload processor which overrides function exposed by work order interface and also facilitates auto registration of workloads

Chapter 6

Solidity Connector Test Process

1. You have two choices for building Avalon: Docker-based build (recommended) or standalone build.

   - For standalone builds, follow instructions in the "Standalone based Build" section of the the `build document`. Then continue with the next step, step 2.
   - For Docker-based builds, follow instructions in the "Docker-based Build and Execution" section of the the `build document` through step 4 (Docker container shell). Then continue with step 8, below.

2. (Standalone builds only) If needed, update the Ethereum account and direct registry contract information in `sdk/avalon_sdk/tcf_connector.toml`

3. (Standalone builds only) Install Python 3.6.8 if not currently installed. Determine your Python version with `python3 --version` . If it is not installed, install it as follows:

   “‘bash wget `https://www.python.org/ftp/python/3.6.8/Python-3.6.8.tgz` tar -xvf Python-3.6.8.tgz cd Python-3.6.8 ./configure make sudo make install make sure “‘

4. (Standalone builds only) Install the Solidity compiler to compile Solidity contracts from Python: “‘bash pip3 install –upgrade py-solc-x python3 -m solcx.install v0.5.15 “‘

5. (Standalone builds only) To run smart contracts using a Ropsten network account, first install the MetaMask Chrome plugin to your Chrome web browser and create an account in the Ropsten network

6. (Standalone builds only) After creating an account, make sure to add fake ether to the account using:

   - `https://faucet.metamask.io/`
   - `https://blog.bankex.org/how-to-buy-ethereum-using-metamask-ccea0703daec`

7. Install web3.py, which is an Ethereum Python client that interacts with the Ethereum network. For more information about web3.py, see `https://web3py.readthedocs.io/en/stable/quickstart.html`

   “‘bash pip install web3 “‘

8. Run `cd $TCF_HOME/examples/common/python/connectors/ethereum`

9. Fill in your Ropsten testnet address in `eth_account` in `sdk/avalon_sdk/tcf_connector.toml`

10. Deploy solidity contracts to Ropsten network using `eth_cli.py`

    “‘bash ./eth_cli.py “‘

    The above command will display the contract instance address for `direct_registry_contract_address` and `worker_registry_contract_address`

11. Fill in your your contract addresses `direct_registry_contract_address` and `worker_registry_contract_address` in `sdk/avalon_sdk/tcf_connector.toml`

12. Test the DirectRegistry and WorkerRegistry contracts with: "‘bash cd $TCF_HOME/examples/common/python/connectors/ethere _tests python3 test_ethereum_worker_registry_impl.py python3 test_ethereum_worker_registry_list_impl.py "‘

13. Test echo client with direct mode using Ropsten test network. "‘bash cd $TCF_HOME/examples/apps/echo/client ./echo_client.py -m "Hello world" "‘

Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 8

Hierarchical Index

8.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 9

Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 10

File Index

10.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 11

Namespace Documentation

11.1 tcf::crypto::skenc Namespace Reference

Functions

- ByteArray GenerateKey ()
- ByteArray GenerateIV (const std::string &IVstring=std::string(""))
- ByteArray EncryptMessage (const ByteArray &key, const ByteArray &iv, const ByteArray &message)
- ByteArray EncryptMessage (const ByteArray &key, const ByteArray &message)
- ByteArray DecryptMessage (const ByteArray &key, const ByteArray &iv, const ByteArray &message)
- ByteArray DecryptMessage (const ByteArray &key, const ByteArray &message)

11.1.1 Detailed Description

Authenticated encryption.

11.1.2 Function Documentation

11.1.2.1 DecryptMessage() [1/2]

```
ByteArray tcf::crypto::skenc::DecryptMessage (
            const ByteArray & key,
            const ByteArray & iv,
            const ByteArray & message )
```

Throws RuntimeError, ValueError, CryptoError (message authentication failure).

11.1.2.2 DecryptMessage() [2/2]

```
ByteArray tcf::crypto::skenc::DecryptMessage (
            const ByteArray & key,
            const ByteArray & message )
```

Throws RuntimeError, ValueError, CryptoError (message authentication failure). Expects IV prepended to message ciphertext.

11.1.2.3 EncryptMessage() [1/2]

```
ByteArray tcf::crypto::skenc::EncryptMessage (
            const ByteArray & key,
            const ByteArray & iv,
            const ByteArray & message )
```

Throws RuntimeError, ValueError.

11.1.2.4 EncryptMessage() [2/2]

```
ByteArray tcf::crypto::skenc::EncryptMessage (
            const ByteArray & key,
            const ByteArray & message )
```

Uses random IV prepended the returned ciphertext. Throws RuntimeError, ValueError.

11.1.2.5 GenerateIV()

```
ByteArray tcf::crypto::skenc::GenerateIV (
            const std::string & IVstring = std::string("") )
```

Throws RuntimeError.

11.1.2.6 GenerateKey()

```
ByteArray tcf::crypto::skenc::GenerateKey ( )
```

ByteArray here is used to encapsulate raw binary data and does not apply/assume any encoding. Throws Runtime↩ Error.

Chapter 12

Class Documentation

12.1 avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient Class Reference

Public Member Functions

- def __init__ (self, config_file=None, config=None)
- def get_worker_registry_list_instance (self)
- def get_worker_registry_instance (self)
- def get_work_order_instance (self)
- def get_work_order_receipt_instance (self)

12.1.1 Detailed Description

```
This is class for the direct JSON RPC API client.
It is used in the direct model.
1. Worker registry list interacts with the blockchain; it is optional.
2. Worker registry interacts with the JSON RPC listener.
3. Work order interacts with the JSON RPC listener.
4. Work order receipt interacts with the JSON RPC listener.
```

12.1.2 Constructor & Destructor Documentation

12.1.2.1 __init__()

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.__init__ (
              self,
              config_file = None,
              config = None )
```

```
Parameters:
config_file Optional configuration file path as a string
config      Optional dictionary loaded from config_file

Either one of config_file or config needs to be passed.
If both are passed, then config takes precedence.
```

12.1.3 Member Function Documentation

12.1.3.1 get_work_order_instance()

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_work_order_↩
instance (
              self )
```

```
Return the worker's instance of the work order.
The work order interacts with the JSON RPC listener.
```

12.1.3.2 get_work_order_receipt_instance()

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_work_order_↩
receipt_instance (
              self )
```

```
Return the worker's instance of the work order receipt.
The work order receipt interacts with the JSON RPC listener.
```

12.1.3.3 get_worker_registry_instance()

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_worker_registry_↩
instance (
              self )
```

```
Return the worker's instance of the registry.
The registry interacts with the JSON RPC listener.
```

12.1.3.4 get_worker_registry_list_instance()

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_worker_registry_↩
list_instance (
              self )
```

```
Return the worker's instance of the registry list.
This list is optional and interacts with the blockchain.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/avalon_direct_client.py

12.2 avalon_sdk.connector.blockchains.ethereum.ethereum_listener.BlockchainInterface Class Reference

Public Member Functions

- def **__init__** (self, config)
- def newListener (self, contract, event, fromBlock='latest')

12.2.1 Detailed Description

```
Ethereum blockchain interface to event processor.
```

12.2.2 Member Function Documentation

12.2.2.1 newListener()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.BlockchainInterface.new↩
Listener (
              self,
              contract,
              event,
              fromBlock = 'latest' )
```

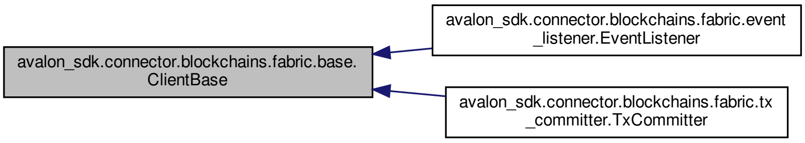
```
Create a filter to get events from latest block by default.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/ethereum_listener.py

12.3 avalon_sdk.connector.blockchains.fabric.base.ClientBase Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.fabric.base.ClientBase:


avalon_sdk.connector.blockchains.fabric.event_listener.EventListener
avalon_sdk.connector.blockchains.fabric.base.ClientBase
avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter

Public Member Functions

- def **__init__** (self, profile, channel_name, org_name, peer_name, user_name)
- def **channel_name** (self)
- def **channel** (self)
- def **org_name** (self)
- def **peer_name** (self)
- def **user_name** (self)
- def **user** (self)

Public Attributes

- **client**

12.3.1 Detailed Description

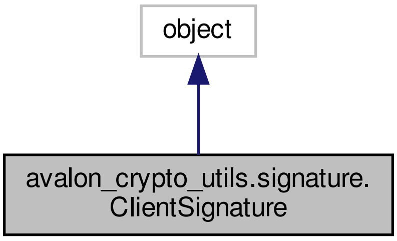
```
Base class for a Hyperledger Fabric client.
```

The documentation for this class was generated from the following file:

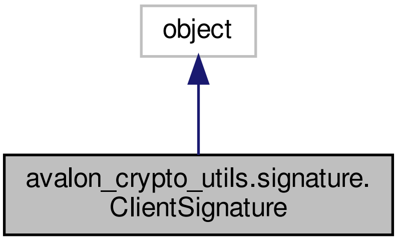
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/base.py

12.4 avalon_crypto_utils.signature.ClientSignature Class Reference

Inheritance diagram for avalon_crypto_utils.signature.ClientSignature:



Collaboration diagram for avalon_crypto_utils.signature.ClientSignature:



Public Member Functions

- def **__init__** (self)
- def calculate_datahash (self, data_objects)
- def generate_signature (self, hash, private_key)
- def generate_client_signature (self, input_json_str, worker, private_key, session_key, session_iv, encrypted_session_key, data_key=None, data_iv=None)
- def verify_signature (self, input_json, verification_key)
- def verify_update_receipt_signature (self, input_json)
- def verify_create_receipt_signature (self, input_json)
- def calculate_request_hash (self, input_json)

Public Attributes

- **private_key**
- **public_key**
- **param_pool**
- **tcs_worker**

12.4.1 Detailed Description

```
Class to perform hash calculation, signature generation and verification
```

12.4.2 Member Function Documentation

12.4.2.1 calculate_datahash()

```
def avalon_crypto_utils.signature.ClientSignature.calculate_datahash (
              self,
              data_objects )
```

```
Function to calculate a hash value of the array concatenating dataHash,
data, encryptedDataEncryptionKey, iv for each item in the
inData/outData array
Parameters:
    - data_objects is each item in inData or outData part of workorder
      request as per Trusted Compute EEA API 6.1.7
      Work Order Data Formats
```

12.4.2.2 calculate_request_hash()

```
def avalon_crypto_utils.signature.ClientSignature.calculate_request_hash (
              self,
              input_json )
```

```
Function to create the work order reuest hash
as defined in EEA spec 6.1.8.1
Parameters:
    - input_json is dictionary contains work order request payload
      as define EEA spec 6.1.1
Returns hash of work order request as string
```

12.4.2.3 generate_client_signature()

```
def avalon_crypto_utils.signature.ClientSignature.generate_client_signature (
              self,
              input_json_str,
              worker,
              private_key,
              session_key,
              session_iv,
              encrypted_session_key,
              data_key = None,
              data_iv = None )
```

```
Function to generate client signature
Parameters:
    - input_json_str is requester Work Order Request payload in a
      JSON-RPC based format defined 6.1.1 Work Order Request Payload
    - worker is a worker object to store all the common details of
      worker as per Trusted Compute EEA API 8.1
      Common Data for All Worker Types
    - private_key is Client private key
    - session_key is one time session key generated by the participant
      submitting the work order.
    - session_iv is an initialization vector if required by the
      data encryption algorithm (encryptedSessionKey).
      The default is all zeros.
    - data_key is a one time key generated by participant used to
      encrypt work order indata
    - data_iv is an initialization vector used along with data_key.
      Default is all zeros.
    - encrypted_session_key is a encrypted version of session_key.
Returns a tuple containing signature and status
```

12.4.2.4 generate_signature()

```
def avalon_crypto_utils.signature.ClientSignature.generate_signature (
              self,
              hash,
              private_key )
```

```
Function to generate signature object
Parameters:
    - hash is the combined array of all hashes calculated on the
      message
    - private_key is Client private key
Returns tuple(status, signature)
```

12.4.2.5 verify_create_receipt_signature()

```
def avalon_crypto_utils.signature.ClientSignature.verify_create_receipt_signature (
              self,
              input_json )
```

```
Function to verify the signature of work order receipt create
Parameters:
    - input_json is dictionary contains request payload of
      WorkOrderReceiptRetrieve API as define EEA spec 7.2.2
Returns enum type SignatureStatus
```

12.4.2.6 verify_signature()

```
def avalon_crypto_utils.signature.ClientSignature.verify_signature (
              self,
              input_json,
              verification_key )
```

```
Function to verify the signature received from the enclave
Parameters:
    - input_json is dictionary contains payload returned by the
      Worker Service in response to successful workorder submit request
      as per Trusted Compute EEA API 6.1.2 Work Order Result Payload
    - verification_key is ECDSA/SECP256K1 public key used to verify
      signatures created by the Enclave.
Returns enum type SignatureStatus
```

12.4.2.7 verify_update_receipt_signature()

```
def avalon_crypto_utils.signature.ClientSignature.verify_update_receipt_signature (
              self,
              input_json )
```

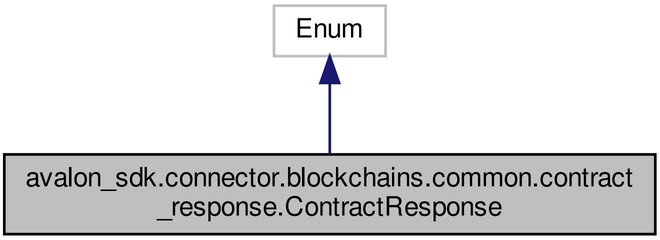
```
Function to verify the signature of work order receipt update
Parameters:
    - input_json is dictionary contains payload returned by the
      WorkOrderReceiptUpdateRetrieve API as define EEA spec 7.2.7
Returns enum type SignatureStatus
```

The documentation for this class was generated from the following file:

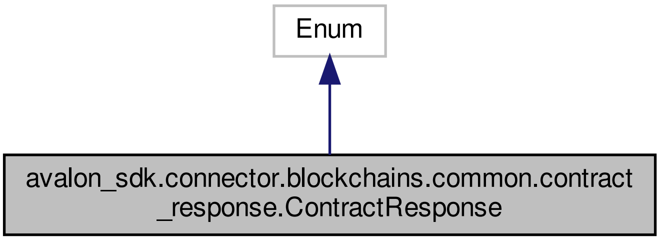
- /home/dano/git/avalon/common/crypto_utils/avalon_crypto_utils/signature.py

12.5 avalon_sdk.connector.blockchains.common.contract_response.ContractResponse Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.common.contract_response.ContractResponse:


Enum
avalon_sdk.connector.blockchains.common.contract_response.ContractResponse

Collaboration diagram for avalon_sdk.connector.blockchains.common.contract_response.ContractResponse:


Enum
avalon_sdk.connector.blockchains.common.contract_response.ContractResponse

Static Public Attributes

- int **SUCCESS** = 0
- int **ERROR** = -1

12.5.1 Detailed Description

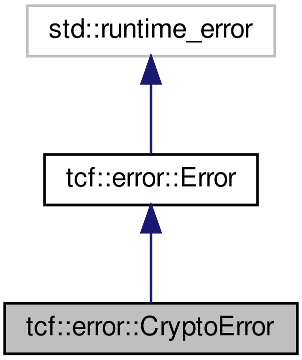
```
Contract response values.
```

The documentation for this class was generated from the following file:

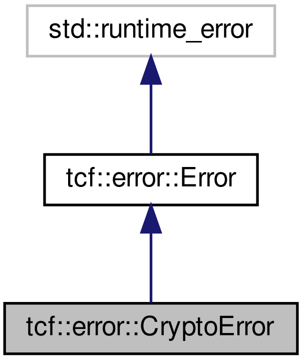
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/common/contract_response.py

12.6 tcf::error::CryptoError Class Reference

Inheritance diagram for tcf::error::CryptoError:



Collaboration diagram for tcf::error::CryptoError:



Public Member Functions

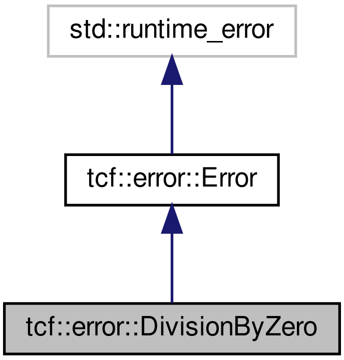
- **CryptoError** (const std::string &msg)

The documentation for this class was generated from the following file:

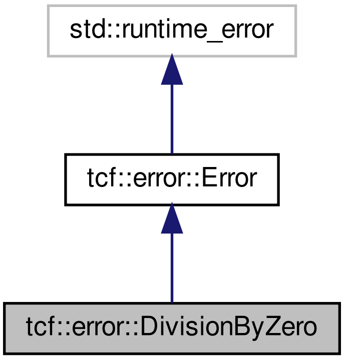
- /home/dano/git/avalon/common/cpp/error.h

12.7 tcf::error::DivisionByZero Class Reference

Inheritance diagram for tcf::error::DivisionByZero:



Collaboration diagram for tcf::error::DivisionByZero:



Public Member Functions

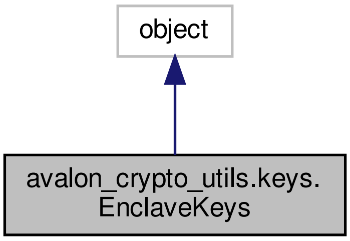
- **DivisionByZero** (const std::string &msg)

The documentation for this class was generated from the following file:

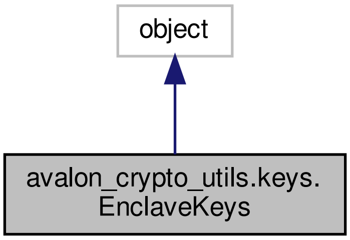
- /home/dano/git/avalon/common/cpp/error.h

12.8 avalon_crypto_utils.keys.EnclaveKeys Class Reference

Inheritance diagram for avalon_crypto_utils.keys.EnclaveKeys:



Collaboration diagram for avalon_crypto_utils.keys.EnclaveKeys:



Public Member Functions

- def __init__ (self, verifying_key, encryption_key)
- def **identity** (self)
- def **hashed_identity** (self)

12.8.1 Detailed Description

```
Wrapper for managing the enclave's keys, the verifying_key is an
ECDSA public key used to verify enclave signatures, the
encryption_key is an RSA public key for encrypting message to the
enclave.
```

12.8.2 Constructor & Destructor Documentation

12.8.2.1 __init__()

```
def avalon_crypto_utils.keys.EnclaveKeys.__init__ (
              self,
              verifying_key,
              encryption_key )
```

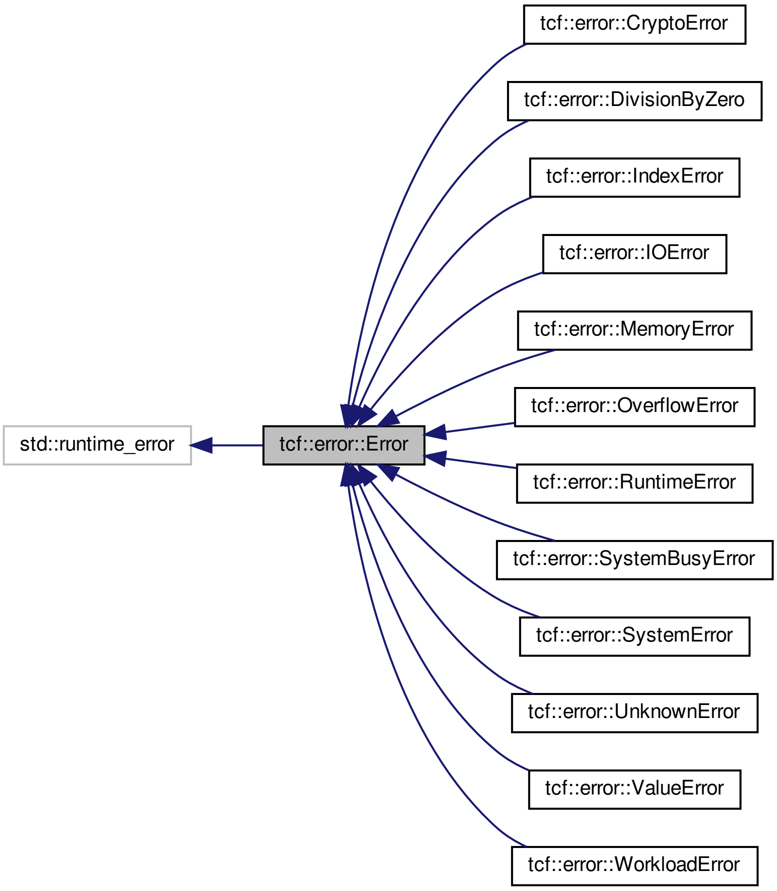
```
Initialize the object

:param verifying_key: PEM encoded ECDSA verifying key
:param encryption_key: PEM encoded RSA encryption key
```

The documentation for this class was generated from the following file:

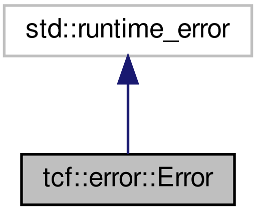
- /home/dano/git/avalon/common/crypto_utils/avalon_crypto_utils/keys.py

12.9 tcf::error::Error Class Reference

Inheritance diagram for tcf::error::Error:



Collaboration diagram for tcf::error::Error:



Public Member Functions

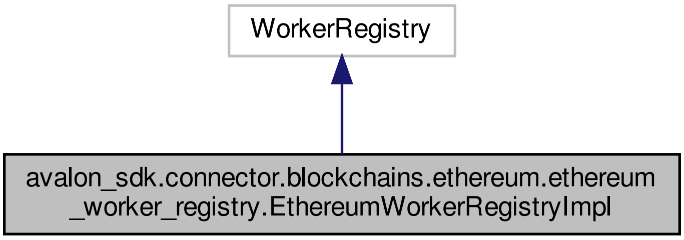
- **Error** (tcf_err_t in_error, const std::string &msg)
- tcf_err_t **error_code** ()

The documentation for this class was generated from the following file:

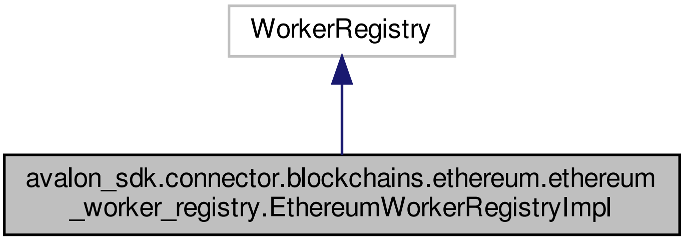
- /home/dano/git/avalon/common/cpp/error.h

12.10 avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.Ethereum↩ WorkerRegistryImpl Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorker↩ RegistryImpl:


WorkerRegistry
avalon_sdk.connector.blockchains.ethereum.ethereum
_worker_registry.EthereumWorkerRegistryImpl

Collaboration diagram for avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.Ethereum↩ WorkerRegistryImpl:


WorkerRegistry
avalon_sdk.connector.blockchains.ethereum.ethereum
_worker_registry.EthereumWorkerRegistryImpl

Public Member Functions

- def __init__ (self, config)
- def worker_lookup (self, worker_type, org_id, application_id, id=None)
- def worker_retrieve (self, worker_id, id=None)
- def worker_lookup_next (self, worker_type, org_id, application_id, lookup_tag)
- def worker_register (self, worker_id, worker_type, organization_id, application_type_ids, details)
- def worker_update (self, worker_id, details)
- def worker_set_status (self, worker_id, status)

12.10.1 Detailed Description

```
This class is sets and gets worker-related information to and from
the Ethereum blockchain.
Detailed method descriptions are available in the WorkerRegistry
interfaces.
```

12.10.2 Constructor & Destructor Documentation

12.10.2.1 __init__()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↩
Impl.__init__ (
              self,
              config )
```

```
Parameters:
config    Dictionary containing Ethereum-specific parameters
```

12.10.3 Member Function Documentation

12.10.3.1 worker_lookup()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↩
Impl.worker_lookup (
              self,
              worker_type,
              org_id,
              application_id,
              id = None )
```

```
Lookup a worker identified by worker_type, org_id, and application_id.
All fields are optional and, if present, condition should match for
all fields. If none are passed it should return all workers.

If the list is too large to fit into a single response (the maximum
number of entries in a single response is implementation specific),
the smart contract should return the first batch of the results
and provide a lookupTag that can be used by the caller to
retrieve the next batch by calling worker_lookup_next.

Parameters:
worker_type    Optional characteristic of workers for which you may
       wish to search
org_id         Optional organization ID that can be used to search
       for one or more workers that belong to this
       organization
application_id Optional application type ID that is supported by
       the worker
id             Optional JSON RPC request ID

Returns:
Tuple containing workers count, lookup tag, and list of
worker IDs:
total_count Total number of entries matching a specified
    lookup criteria. If this number is larger than the
    size of the IDs array, the caller should use
    lookupTag to call worker_lookup_next to retrieve
    the rest of the IDs
lookup_tag  Optional parameter. If it is returned, it means
    that there are more matching worker IDs, which can then
    be retrieved by calling function worker_lookup_next
    with this tag as an input parameter
ids         Array of the worker IDs that match the input parameters

On error returns None.
```

12.10.3.2 worker_lookup_next()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistryImpl.worker_lookup_next (
              self,
              worker_type,
              org_id,
              application_id,
              lookup_tag )
```

```
Retrieve additional worker lookup results after calling worker_lookup.

Parameters:
worker_type         Characteristic of Workers for which you may wish
            to search
org_id              Organization ID to which a Worker belongs
application_id      Optional application type ID that is
            supported by the worker
lookup_tag          is returned by a previous call to either this
            function or to worker_lookup
id                  Optional Optional JSON RPC request ID


Returns:
Tuple containing the following:
total_count    Total number of entries matching this lookup
       criteria.  If this number is larger than the number
       of IDs returned so far, the caller should use
       lookupTag to call worker_lookup_next to retrieve
       the rest of the IDs
new_lookup_tag Optional parameter. If it is returned, it
       means that there are more matching worker IDs that
       can be retrieved by calling this function again with
       this tag as an input parameter
ids            Array of the worker IDs that match the input parameters

On error returns None.
```

12.10.3.3 worker_register()

def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↩ Impl.worker_register (
self,
worker_id,
worker_type,
organization_id,
application_type_ids,
details)

```
Register a new worker with details of the worker.

Parameters:
worker_id       Worker ID value. E.g., an Ethereum address or
        a value derived from the worker's DID
worker_type     Type of Worker. Currently defined types are:
        * "TEE-SGX": an Intel SGX Trusted Execution
          Environment
        * "MPC": Multi-Party Compute
        * "ZK": Zero-Knowledge
organization_id Optional parameter representing the
        organization that hosts the Worker,
        e.g. a bank in the consortium or
        anonymous entity
application_ids Optional parameter that defines
        application types supported by the Worker
details         Detailed information about the worker in
        JSON RPC format as defined in
https://entethalliance.github.io/trusted-computing/spec.html
#common-data-for-all-worker-types

Returns:
Transaction receipt if registration succeeds.
None if registration does not succeed.
```

12.10.3.4 worker_retrieve()

def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↩ Impl.worker_retrieve (
self,
worker_id,
id = None)

```
Retrieve the worker identified by worker ID.

Parameters:
worker_id  Worker ID of the registry whose details are requested
id         Optional JSON RPC request ID

Returns:
Tuple containing worker status (defined in worker_set_status),
worker type, organization ID, list of application IDs, and worker
details (JSON RPC string).

On error returns None.
```

12.10.3.5 worker_set_status()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↩
Impl.worker_set_status (
              self,
              worker_id,
              status )
```

```
Set the worker status identified by worker ID.

Parameters:
worker_id Worker ID value. E.g., an Ethereum address or
  a value derived from the worker's DID
status    Worker status. The currently defined values are:
  1 - worker is active
  2 - worker is temporarily "off-line"
  3 - worker is decommissioned
  4 - worker is compromised

Returns:
Transaction receipt if registration succeeds.
None if registration does not succeed.
```

12.10.3.6 worker_update()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↩
Impl.worker_update (
              self,
              worker_id,
              details )
```

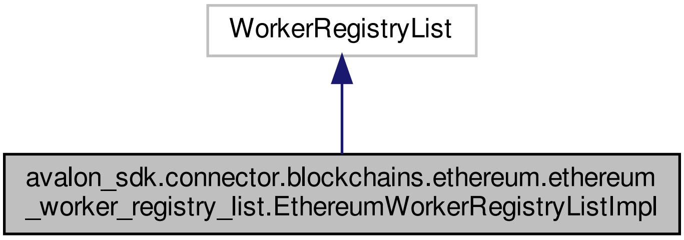
```
Update a worker with details data.

Parameters:
worker_id  Worker ID value. E.g., an Ethereum address or
   a value derived from the worker's DID
details    Detailed information about the worker in JSON format

Returns:
Transaction receipt if registration succeeds.
None if registration does not succeed.
```

The documentation for this class was generated from the following file:

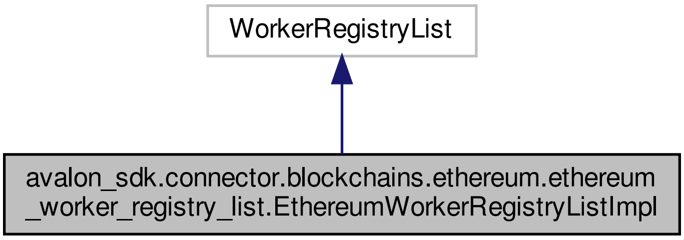
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/ethereum_worker_registry.py

12.11 avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.↩ EthereumWorkerRegistryListImpl Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.Ethereum↩ WorkerRegistryListImpl:


WorkerRegistryList
avalon_sdk.connector.blockchains.ethereum.ethereum
_worker_registry_list.EthereumWorkerRegistryListImpl

Collaboration diagram for avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.Ethereum↩ WorkerRegistryListImpl:


WorkerRegistryList
avalon_sdk.connector.blockchains.ethereum.ethereum
_worker_registry_list.EthereumWorkerRegistryListImpl

Public Member Functions

- def **__init__** (self, config)
- def registry_lookup (self, app_type_id=None)
- def registry_retrieve (self, org_id)
- def registry_lookup_next (self, app_type_id, lookup_tag)
- def registry_add (self, org_id, uri, sc_addr, app_type_ids)
- def registry_update (self, org_id, uri, sc_addr, app_type_ids)
- def registry_set_status (self, org_id, status)

12.11.1 Detailed Description

```
This class provide APIs to read/write registry entries of workers,
which is stored in the Ethereum blockchain.
```

12.11.2 Member Function Documentation

12.11.2.1 registry_add()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorker↩
RegistryListImpl.registry_add (
              self,
              org_id,
              uri,
              sc_addr,
              app_type_ids )
```

```
Add a new registry.

Parameters:
org_id       bytes[] identifies organization that hosts the
     registry, e.g. a bank in the consortium or an
     anonymous entity
uri          String defines a URI for this registry that
     supports the Off-Chain Worker Registry
     JSON RPC API.
sc_addr      bytes[] defines an Ethereum address that
     runs the Worker Registry Smart Contract API
     smart contract for this registry
app_type_ids []bytes[] is an optional parameter that defines
     application types supported by the worker
     managed by the registry

Returns:
Transaction receipt on success or None on error.
```

12.11.2.2 registry_lookup()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorker↩
RegistryListImpl.registry_lookup (
              self,
              app_type_id = None )
```

```
Registry Lookup identified by application type ID.

Parameters:
app_type_id  Application type ID to lookup in the registry

Returns:
Returns tuple containing totalCount, lookupTag, ids on success:
totalCount Total number of entries matching a specified
   lookup criteria.  If this number is larger than the size
   of the IDs array, the caller should use the lookupTag to
   call workerLookUpNext to retrieve the rest of the IDs
lookupTag  Optional parameter. If it is returned, it means that
   there are more matching registry IDs that can be retrieved
   by calling the function registry_lookup_next with this tag
   as an input parameter
ids        Array of the registry organization IDs that match the
   input parameters

Returns None on error.
```

12.11.2.3 registry_lookup_next()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorker←↩
RegistryListImpl.registry_lookup_next (
              self,
              app_type_id,
              lookup_tag )
```

```
Get additional registry lookup results.
This function is called to retrieve additional results of the
Registry lookup initiated by the registry_lookUp call.

Parameters:
app_type_id    Application type that has to be
       supported by the workers retrieved
lookup_tag     Returned by a previous call to either this
       function or to registry_lookup

Returns:
Outputs tuple on success containing the following:
total_count    Total number of entries matching the lookup
       criteria. If this number is larger than the number
       of IDs returned so far, the caller should use
       lookup_tag to call registry_lookup_next to
       the rest of the ids
new_lookup_tag Optional parameter. If it is returned, it means
       that there are more matching registry IDs that
       can be retrieved by calling this function again
       with this tag as an input parameter
ids            Array of the registry IDs that match the input
       parameters

Returns None on error.
```

12.11.2.4 registry_retrieve()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorker←↩
RegistryListImpl.registry_retrieve (
              self,
              org_id )
```

```
Retrieving Registry Information identified by organization ID.

Parameters:
org_id     Organization ID to lookup

Returns:
Tuple containing following on success:
uri                  string defining a URI for this registry that
             supports the Off-Chain Worker Registry JSON
             RPC API. It will be None for the proxy model
sc_addr              Ethereum address for worker registry
             smart contract address
application_type_ids List of application ids(array of byte[])
status               Status of the registry

Returns None on error.
```

12.11.2.5 registry_set_status()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorker↩
RegistryListImpl.registry_set_status (
              self,
              org_id,
              status )
```

```
Set registry status.

Parameters:
org_id  bytes[] identifies organization that hosts
the registry
status  Defines registry status to set.
The currently defined values are:
1 - the registry is active
2 - the registry is temporarily "off-line"
3 - the registry is decommissioned

Returns:
Transaction receipt on success or None on error.
```

12.11.2.6 registry_update()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorker↩
RegistryListImpl.registry_update (
              self,
              org_id,
              uri,
              sc_addr,
              app_type_ids )
```

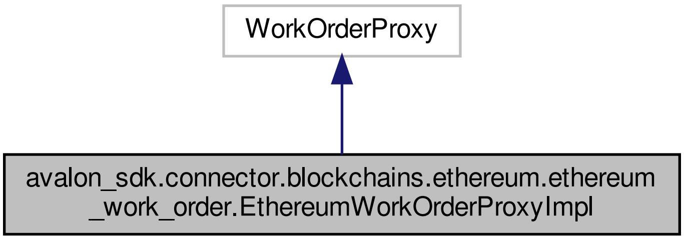
```
Update a registry.

Parameters:
org_id              bytes[] identifies organization that hosts the
             registry, e.g. a bank in the consortium or
             an anonymous entity
uri                 string defines a URI for this registry that
             supports the Off-Chain Worker Registry
             JSON RPC API
sc_addr             bytes[] defines an Ethereum address that
             runs a Worker Registry Smart Contract API
             smart contract for this registry
app_type_ids        []bytes[] is an optional parameter that defines
             application types supported by the worker
             managed by the registry

Returns:
Transaction receipt on success or None on error.
```

The documentation for this class was generated from the following file:

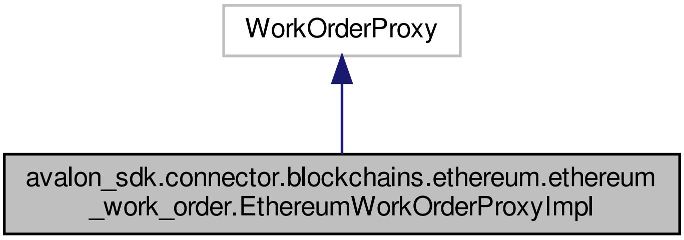
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/ethereum_worker_registry_list.py

12.12 avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl:


WorkOrderProxy
avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl

Collaboration diagram for avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl:


WorkOrderProxy
avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl

Public Member Functions

- def **__init__** (self, config)
- def work_order_submit (self, work_order_id, worker_id, requester_id, work_order_request, id=None)
- def work_order_complete (self, work_order_id, work_order_response)
- def work_order_get_result (self, work_order_id, id=None)
- def encryption_key_retrieve (self, worker_id, last_used_key_nonce, tag, requester_id, signature_nonce=None, signature=None, id=None)
- def encryption_key_start (self, tag, id=None)
- def encryption_key_set (self, worker_id, encryption_key, encryption_nonce, tag, signature, id=None)
- def encryption_key_get (self, worker_id, requester_id, last_used_key_nonce=None, tag=None, signature_nonce=None, signature=None, id=None)

12.12.1 Detailed Description

```
This class is meant to write work order-related data to the Ethereum
blockchain.
Detailed method descriptions are available in the interfaces.
```

12.12.2 Member Function Documentation

12.12.2.1 encryption_key_get()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↩
Impl.encryption_key_get (
              self,
              worker_id,
              requester_id,
              last_used_key_nonce = None,
              tag = None,
              signature_nonce = None,
              signature = None,
              id = None )
```

```
Get Encryption Key Request Payload.
Not supported for Ethereum.
```

12.12.2.2 encryption_key_retrieve()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↩
Impl.encryption_key_retrieve (
              self,
              worker_id,
              last_used_key_nonce,
              tag,
              requester_id,
              signature_nonce = None,
              signature = None,
              id = None )
```

```
Get Encryption Key Request Payload.
Not supported for Ethereum.
```

12.12.2.3 encryption_key_set()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy←↩
Impl.encryption_key_set (
              self,
              worker_id,
              encryption_key,
              encryption_nonce,
              tag,
              signature,
              id = None )
```

```
Set Encryption Key Request Payload.
Not supported for Ethereum.
```

12.12.2.4 encryption_key_start()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy←↩
Impl.encryption_key_start (
              self,
              tag,
              id = None )
```

```
Inform the Worker that it should start
encryption key generation for this requester.
Not supported for Ethereum.
```

12.12.2.5 work_order_complete()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy←↩
Impl.work_order_complete (
              self,
              work_order_id,
              work_order_response )
```

```
This function is called by the Worker Service to
complete a work order successfully or in error.
This API is for the proxy model.

Parameters:
work_order_id       Unique ID to identify the work order request
work_order_response Work order response data in a string

Returns:
errorCode           0 on success or non-zero on error.
```

12.12.2.6 work_order_get_result()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↩
Impl.work_order_get_result (
              self,
              work_order_id,
              id = None )
```

```
Query blockchain to get a work order result.
This function starts an event handler for handling the
workOrderCompleted event from the Ethereum blockchain.

Parameters:
work_order_id Work Order ID that was sent in the
      corresponding work_order_submit request
id            Optional JSON RPC request ID

Returns:
Tuple containing work order status, worker id, work order request,
work order response, and error code.
None on error.
```

12.12.2.7 work_order_submit()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↩
Impl.work_order_submit (
              self,
              work_order_id,
              worker_id,
              requester_id,
              work_order_request,
              id = None )
```

```
Submit work order request to the Ethereum block chain.

Parameters:
work_order_id      Unique ID of the work order request
worker_id          Identifier for the worker
requester_id       Unique id to identify the requester
work_order_request JSON RPC string work order request.
           Complete definition at work_order.py and
           defined in EEA specification 6.1.1.
id                 Optional JSON RPC request ID

Returns:
0 on success and non-zero on error.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/ethereum_work_order.py

12.13 avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper Class Reference

Public Member Functions

- def **__init__** (self, config)
- def compile_source_file (self, file_path)
- def deploy_contract (self, contract_interface)
- def sign_execute_raw_transaction (self, tx_dict)
- def execute_unsigned_transaction (self, tx_dict)
- def execute_transaction (self, tx_dict)
- def get_chain_id (self)
- def get_gas_limit (self)
- def get_gas_price (self)
- def get_account_address (self)
- def get_contract_instance (self, contract_file_name, contract_address)
- def get_contract_instance_from_json (self, json_file_name, contract_address)
- def get_txn_nonce (self)
- def get_transaction_params (self)
- def get_bytes_from_hex (self, hex_str)

12.13.1 Detailed Description

```
Ethereum wrapper class to interact with the Ethereum blockchain to
deploy compile contract code, deploy contract code,
and execute contract code.
```

12.13.2 Member Function Documentation

12.13.2.1 compile_source_file()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.compile_↩
source_file (
              self,
              file_path )
```

```
Compile a Solidity contract file and returns contract instance object.

Parameters:
file_path    Path to Solidity contract file

Returns:
Solidity contract instance object.
```

12.13.2.2 deploy_contract()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.deploy_contract
(
              self,
              contract_interface )
```

```
Deploys a Solidity contract to an Ethereum network identified by
chain_id.

Parameters:
contract_interace  Solidity contract interface

Returns:
Solidity contract address.
```

12.13.2.3 execute_transaction()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.execute_↩
transaction (
              self,
              tx_dict )
```

```
Wrapper function to choose appropriate function to execute a
transaction based on provider (Ropsten vs other).

Parameters:
tx_dict     Transaction to execute
```

12.13.2.4 execute_unsigned_transaction()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.execute_↩
unsigned_transaction (
              self,
              tx_dict )
```

```
Send a transaction to be executed only with the account address,
and wait for receipts.

Parameters:
tx_dict     Unsigned transaction to execute

Returns:
Transaction receipt on success or None on error.
```

12.13.2.5 get_account_address()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_account_↩
address (
              self )
```

```
Retrieve account address.
```

12.13.2.6 get_bytes_from_hex()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_bytes_↩
from_hex (
              self,
              hex_str )
```

```
Convert a hex string to bytes.
```

12.13.2.7 get_chain_id()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_chain_id (
              self )
```

```
Retrieve chain ID.
```

12.13.2.8 get_contract_instance()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_contract_↩
instance (
              self,
              contract_file_name,
              contract_address )
```

```
This function returns two contract instances.
The first is meant for committing transactions or reading from
a blockchain.
The second one is specifically meant for event listening.

Parameters:
contract_file_name  Contract filename
contract_address    Ethereum contract address

Returns:
Two contract instances as explained above.
```

12.13.2.9 get_contract_instance_from_json()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_contract_←
instance_from_json (
              self,
              json_file_name,
              contract_address )
```

```
Return two contract instances from a JSON file.
The first is meant for committing transactions or reading from
a blockchain.
The second one is specifically meant for event listening.

Parameters:
json_file_name    JSON filename
contract_address  Ethereum contract address

Returns:
Two contract instances as explained above.
```

12.13.2.10 get_gas_limit()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_gas_limit (
              self )
```

```
Retrieve gas limit.
```

12.13.2.11 get_gas_price()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_gas_price (
              self )
```

```
Retrieve gas price.
```

12.13.2.12 get_transaction_params()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_transaction←
_params (
              self )
```

```
Construct a dictionary with required parameters
to submit the transaction.
Return dict containing chain id, gas, gas limit, and nonce.
```

12.13.2.13 get_txn_nonce()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_txn_nonce (
              self )
```

```
Return a transaction nonce. Derived from the transaction address.
```

12.13.2.14 sign_execute_raw_transaction()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.sign_execute_↩
raw_transaction (
              self,
              tx_dict )
```

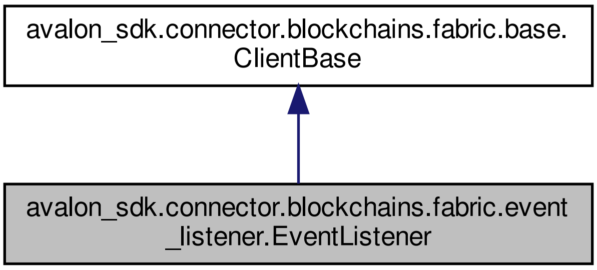
```
Sign the raw transaction with a private key, send it,
and wait for receipts.

Parameters:
tx_dict     Raw transaction to sign


Returns:
Transaction receipt on success or None on error.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/ethereum_wrapper.py

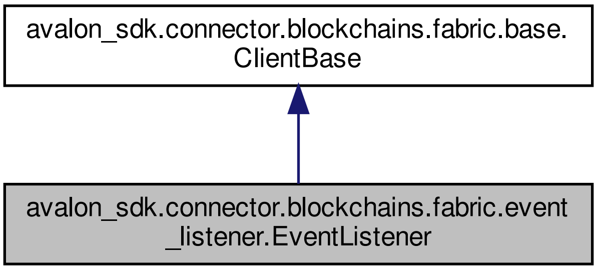
12.14 avalon_sdk.connector.blockchains.fabric.event_listener.EventListener Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.fabric.event_listener.EventListener:



avalon_sdk.connector.blockchains.fabric.base.
ClientBase
avalon_sdk.connector.blockchains.fabric.event
_listener.EventListener

Collaboration diagram for avalon_sdk.connector.blockchains.fabric.event_listener.EventListener:


avalon_sdk.connector.blockchains.fabric.base.
ClientBase
avalon_sdk.connector.blockchains.fabric.event
_listener.EventListener

Public Member Functions

- def **__init__** (self, profile, channel_name, org_name, peer_name, user_name)
- def **handler** (self)
- def **handler** (self, handler)
- def **chaincode** (self)
- def **chaincode** (self, chaincode)
- def **event** (self)
- def **event** (self, event)
- def **config** (self)
- def config (self, config)
- def start_event_handling (self)
- def stop_event_handling (self, seconds=0)
- def get_single_event (self)

Additional Inherited Members

12.14.1 Detailed Description

```
Utility class to listen to Fabric block chain events.
```

12.14.2 Member Function Documentation

12.14.2.1 config()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.config (
              self,
              config )
```

```
Open configuration file named config.
```

12.14.2.2 get_single_event()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.get_single_event (
              self )
```

```
Start event listener and listen for particular event.
Once we got the particular event then unregister and
close the event listenerhub.
```

12.14.2.3 start_event_handling()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.start_event_handling
(
              self )
```

```
Start event listener and listen for events forever
Only way to stop the event listener by calling
stop_event_listener()
```

12.14.2.4 stop_event_handling()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.stop_event_handling (
              self,
              seconds = 0 )
```

```
Stop event listener.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/event_listener.py

12.15 avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor Class Reference

Public Member Functions

- def **__init__** (self, config)
- def listener (self, event_filter)
- def handler (self, callback, kargs, kwargs)
- def sync_handler (self, check_event_callback=None, kargs, kwargs)
- def start (self, event_filter, callback, kargs, kwargs)
- def get_event_synchronously (self, event_filter, callback, kargs, kwargs)
- def stop (self)

Public Attributes

- **queue**
- **listeners**
- **handlers**

12.15.1 Detailed Description

```
This class provides an event processor to capture events
then send these events to event listeners.
```

12.15.2 Member Function Documentation

12.15.2.1 get_event_synchronously()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.get_event_↩
synchronously (
              self,
              event_filter,
              callback,
              kargs,
              kwargs )
```

```
Get a single event synchronously using the event_filter
provided.

Returns an event received for the event_filter used.
```

12.15.2.2 handler()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.handler (
              self,
              callback,
              kargs,
              kwargs )
```

```
Start event handler to handle events.
```

12.15.2.3 listener()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.listener (
              self,
              event_filter )
```

```
Listen to new events since the last poll on this filter.
Although this method uses events, it is not fully asynchronous.
```

12.15.2.4 start()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.start (
              self,
              event_filter,
              callback,
              kargs,
              kwargs )
```

```
Start event processor in an infinite loop.
```

12.15.2.5 stop()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.stop (
              self )
```

```
Stop the event processor that was started with start().
```

12.15.2.6 sync_handler()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.sync_handler (
              self,
              check_event_callback = None,
              kargs,
              kwargs )
```

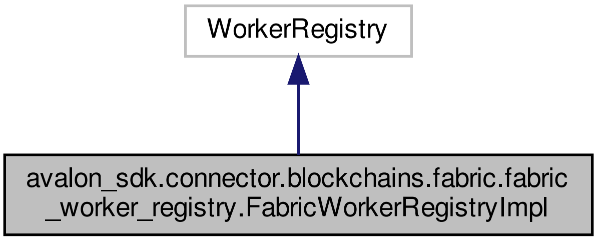
```
Start a synchronous event handler to handle an event.
```

The documentation for this class was generated from the following file:

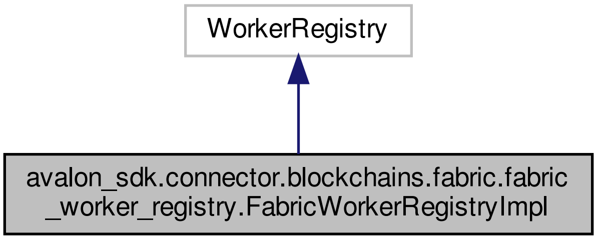
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/ethereum_listener.py

12.16 avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorker←↩ RegistryImpl Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistry←↩ Impl:


WorkerRegistry
avalon_sdk.connector.blockchains.fabric.fabric
_worker_registry.FabricWorkerRegistryImpl

Collaboration diagram for avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistry←↩ Impl:


WorkerRegistry
avalon_sdk.connector.blockchains.fabric.fabric
_worker_registry.FabricWorkerRegistryImpl

Public Member Functions

- def __init__ (self, config)
- def worker_lookup (self, worker_type=None, org_id=None, application_id=None, id=None)
- def worker_retrieve (self, worker_id, id=None)
- def worker_lookup_next (self, worker_type, org_id, application_id, lookup_tag, id=None)
- def worker_register (self, worker_id, worker_type, org_id, application_ids, details, id=None)
- def worker_set_status (self, worker_id, status, id=None)
- def worker_update (self, worker_id, details, id=None)

Public Attributes

- **CHAIN_CODE**

12.16.1 Detailed Description

```
This class provide worker APIs which interact with the
Hyperledger Fabric blockchain.
Detailed method descriptions are available in the WorkerRegistry
interface.
```

12.16.2 Constructor & Destructor Documentation

12.16.2.1 __init__()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
__init__ (
              self,
              config )
```

```
Parameters:
config    Dictionary containing Fabric-specific parameters
```

12.16.3 Member Function Documentation

12.16.3.1 worker_lookup()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
worker_lookup (
              self,
              worker_type = None,
              org_id = None,
              application_id = None,
              id = None )
```

```
Lookup a worker identified worker_type, org_id, and application_id.
All fields are optional and, if present, condition should match for
all fields. If none are passed it should return all workers.

If the list is too large to fit into a single response (the maximum
number of entries in a single response is implementation specific),
the smart contract should return the first batch of the results
and provide a lookup_tag that can be used by the caller to
retrieve the next batch by calling worker_lookup_next.

Parameters:
worker_type         Optional characteristic of workers for which
            you may wish to search
org_id              Optional organization ID to which a worker belongs
application_id      Optional application type ID that is
            supported by the worker
id                  Optional JSON RPC request ID

Returns:
```

```
Tuple containing workers count, lookup tag, and list of
worker IDs:
total_count Total number of entries matching a specified
    lookup criteria. If this number is larger than the
    size of the IDs array, the caller should use
    lookupTag to call worker_lookup_next to retrieve
    the rest of the IDs
lookup_tag  Optional parameter. If it is returned, it means
    that there are more matching worker IDs, which can then
    be retrieved by calling function worker_lookup_next
    with this tag as an input parameter
ids         Array of the worker IDs that match the input parameters

On error returns None.
```

12.16.3.2 worker_lookup_next()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↩
worker_lookup_next (
              self,
              worker_type,
              org_id,
              application_id,
              lookup_tag,
              id = None )
```

```
Retrieve additional worker lookup results after calling worker_lookup.

Parameters:
worker_type         Characteristic of Workers for which you may wish
            to search.
org_id              Organization ID to which a worker belongs
application_id      Optional application type ID that is
            supported by the worker
lookup_tag          is returned by a previous call to either this
            function or to worker_lookup
id                  Optional Optional JSON RPC request ID

Returns:
Tuple containing the following:
total_count    Total number of entries matching this lookup
       criteria.  If this number is larger than the number
       of IDs returned so far, the caller should use
       lookupTag to call worker_lookup_next to retrieve
       the rest of the IDs
new_lookup_tag Optional parameter. If it is returned, it
       means that there are more matching worker IDs that
       can be retrieved by calling this function again with
       this tag as an input parameter
ids            Array of the worker IDs that match the input parameters

On error returns None.
```

12.16.3.3 worker_register()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.←
worker_register (
              self,
              worker_id,
              worker_type,
              org_id,
              application_ids,
              details,
              id = None )
```

```
Register a new worker with details of the worker.

Parameters:
worker_id       Worker ID value. E.g., a Fabric address
worker_type     Type of Worker. Currently defined types are:
        * "TEE-SGX": an Intel SGX Trusted Execution
          Environment
        * "MPC": Multi-Party Compute
        * "ZK": Zero-Knowledge
org_id          Optional parameter representing the
        organization that hosts the Worker,
        e.g. a bank in the consortium or
        anonymous entity
application_ids Optional parameter that defines
        application types supported by the Worker
details         Detailed information about the worker in
        JSON RPC format as defined in
https://entethalliance.github.io/trusted-computing/spec.html
#common-data-for-all-worker-types
id              Optional Optional JSON RPC request ID

Returns:
ContractResponse.SUCCESS on success or
ContractResponse.ERROR on error.
```

12.16.3.4 worker_retrieve()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.←
worker_retrieve (
              self,
              worker_id,
              id = None )
```

```
Retrieve the worker identified by worker ID.

Parameters:
worker_id  Worker ID of the registry whose details are requested
id         Optional Optional JSON RPC request ID

Returns:
Tuple containing worker status (defined in worker_set_status),
worker type, organization ID, list of application IDs, and worker
details (JSON RPC string).

On error returns None.
```

12.16.3.5 worker_set_status()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.←↩
worker_set_status (
              self,
              worker_id,
              status,
              id = None )
```

```
Set the registry status identified by worker ID

Parameters:
worker_id Worker ID value. E.g., a Fabric address
status    Worker status. The currently defined values are:
  1 - worker is active
  2 - worker is temporarily "off-line"
  3 - worker is decommissioned
  4 - worker is compromised
id        Optional Optional JSON RPC request ID

Returns:
ContractResponse.SUCCESS on success
or ContractResponse.ERROR on error.
```

12.16.3.6 worker_update()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.←↩
worker_update (
              self,
              worker_id,
              details,
              id = None )
```

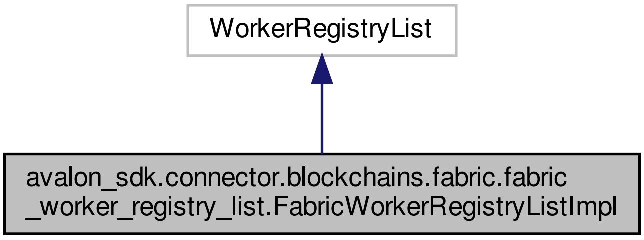
```
Update a worker with details data.

Parameters:
worker_id  Worker ID, e.g. a Fabric address
details    Detailed information about the worker in JSON format
id         Optional Optional JSON RPC request ID

Returns:
ContractResponse.SUCCESS on success
or ContractResponse.ERROR on error.
```

The documentation for this class was generated from the following file:

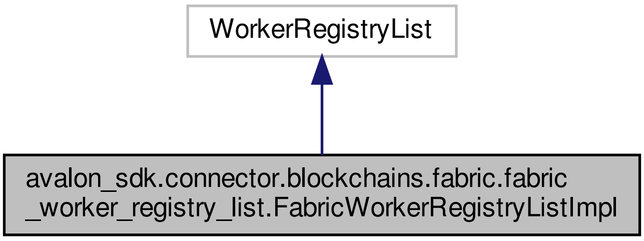
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/fabric_worker_registry.py

12.17 avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl:


WorkerRegistryList
avalon_sdk.connector.blockchains.fabric.fabric
_worker_registry_list.FabricWorkerRegistryListImpl

Collaboration diagram for avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl:


WorkerRegistryList
avalon_sdk.connector.blockchains.fabric.fabric
_worker_registry_list.FabricWorkerRegistryListImpl

Public Member Functions

- def __init__ (self, config)
- def registry_lookup (self, app_type_id=None)
- def registry_retrieve (self, org_id)
- def registry_lookup_next (self, app_type_id, lookup_tag)
- def registry_add (self, org_id, uri, sc_addr, app_type_ids)
- def registry_update (self, org_id, uri, sc_addr, app_type_ids)
- def registry_set_status (self, org_id, status)

Public Attributes

- **CHAIN_CODE**

12.17.1 Detailed Description

```
This class provide APIs to read/write registry entries of workers,
which is stored in the Hyperledger Fabric blockchain.
```

12.17.2 Constructor & Destructor Documentation

12.17.2.1 __init__()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry↩
ListImpl.__init__ (
              self,
              config )
```

```
Parameters:
config    Dictionary containing Fabric-specific parameters.
```

12.17.3 Member Function Documentation

12.17.3.1 registry_add()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry↩
ListImpl.registry_add (
              self,
              org_id,
              uri,
              sc_addr,
              app_type_ids )
```

```
Add a new registry.

Parameters:
org_id       bytes[] identifies organization that hosts the
     registry, e.g. a bank in the consortium or an
     anonymous entity
uri          String defining a URI for this registry that
     supports the Off-Chain Worker Registry
     JSON RPC API
sc_addr      bytes[] defines a Fabric chain code name that
     runs the Worker Registry Smart Contract API
     smart contract for this registry
app_type_ids []bytes[] is an optional parameter that defines
     application types supported by the worker
     managed by the registry

Returns:
Transaction receipt on success or None on error.
```

12.17.3.2 registry_lookup()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry↩
ListImpl.registry_lookup (
              self,
              app_type_id = None )
```

```
Registry Lookup identified by application type ID

Parameters:
app_type_id  Application type ID to lookup in the registry

Returns:
Tuple containing totalCount, lookupTag, and ids on success:
totalCount Total number of entries matching a specified lookup
   criteria. If this number is larger than the size of the
   ids array, the caller should use the lookupTag to call
   registry_lookup_next to retrieve the rest of the IDs
lookupTag  Optional parameter. If it is returned, it means that
   there are more matching registry IDs that can be
   retrieved by calling the function registry_lookup_next
   with this tag as an input parameter.
ids        Array of the registry organization ids that match the
   input parameters.

Returns None on error.
```

12.17.3.3 registry_lookup_next()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry↩
ListImpl.registry_lookup_next (
              self,
              app_type_id,
              lookup_tag )
```

```
Get additional registry lookup results.
This function is called to retrieve additional results of the
Registry lookup initiated by the registry_lookup call.

Parameters:
app_type_id    Application type ID that has to be
       supported by the workers retrieved
lookup_tag     Returned by a previous call to either this function
       or to registry_lookup

Returns:
Outputs a tuple on success containing the following:
total_count    Total number of entries matching the lookup
       criteria. If this number is larger than the number
       of IDs returned so far, the caller should use
       lookup_tag to call registry_lookup_next to
       retrieve the rest of the IDs
new_lookup_tag is an optional parameter. If it is returned, it means
       that there are more matching registry IDs that can be
       retrieved by calling this function again with this tag
       as an input parameter
ids            Array of the registry IDs that match the input
       parameters

Returns None on error.
```

12.17.3.4 registry_retrieve()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl.registry_retrieve (
              self,
              org_id )
```

```
Retrieve registry information identified by the organization ID.

Parameters:
org_id                Organization ID to lookup

Returns:
Tuple containing following on success:
uri                  String defines a URI for this registry that
             supports the Off-Chain Worker Registry JSON RPC
             API. It will be None for the proxy model
sc_addr              Fabric address for worker registry
             smart contract address
application_type_ids List of application ids (array of byte[])
status               Status of the registry

Returns None on error.
```

12.17.3.5 registry_set_status()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl.registry_set_status (
              self,
              org_id,
              status )
```

```
Set registry status.

Parameters:
org_id  bytes[] identifies organization that hosts the
registry, e.g. a bank in the consortium or an
anonymous entity
status  Defines the registry status to set.
The currently defined values are:
1 - the registry is active
2 - the registry is temporarily "off-line"
3 - the registry is decommissioned

Returns:
Transaction receipt on success or None on error.
```

12.17.3.6 registry_update()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry←
ListImpl.registry_update (
              self,
              org_id,
              uri,
              sc_addr,
              app_type_ids )
```

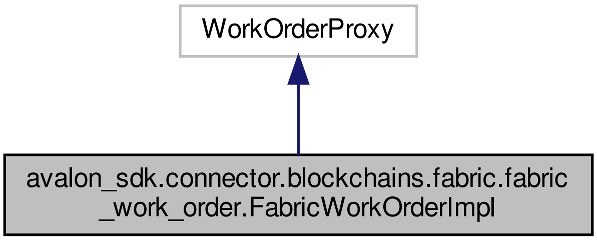
```
Update a registry.

Parameters:
org_id       bytes[] identifies organization that hosts the
     registry, e.g. a bank in the consortium or an
     anonymous entity
uri          string that defines a URI for this registry that
     supports the Off-Chain Worker Registry
     JSON RPC API
sc_addr      bytes[] defines a Fabric chain code name that
     runs the Worker Registry Smart Contract API
     smart contract for this registry
app_type_ids []bytes[] is an optional parameter that defines
     application types supported by the worker
     managed by the registry

Returns:
Transaction receipt on success or None on error.
```

The documentation for this class was generated from the following file:

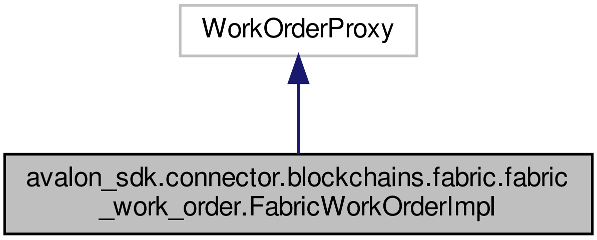
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/fabric_worker_registry_list.py

12.18 avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl:


WorkOrderProxy
avalon_sdk.connector.blockchains.fabric.fabric
_work_order.FabricWorkOrderImpl

Collaboration diagram for avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl:


WorkOrderProxy
avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl

Public Member Functions

- def __init__ (self, config)
- def work_order_submit (self, work_order_id, worker_id, requester_id, work_order_request, id=None)
- def work_order_get_result (self, work_order_id, id=None)
- def work_order_complete (self, work_order_id, work_order_response)
- def encryption_key_start (self, tag)
- def encryption_key_get (self, worker_id, requester_id, last_used_key_nonce=None, tag=None, signature_←
nonce=None, signature=None)
- def encryption_key_set (self, worker_id, encryption_key, encryption_nonce, tag, signature)
- def get_work_order_submitted_event_handler (self, handler_func)
- def get_work_order_completed_event_handler (self, handler_func)

Public Attributes

- **CHAIN_CODE**
- **WORK_ORDER_SUBMITTED_EVENT_NAME**
- **WORK_ORDER_COMPLETED_EVENT_NAME**

12.18.1 Detailed Description

```
This class provides work order management APIs which interact with the
Fabric blockchain. Detail method descriptions are
available in WorkOrder interface.
```

12.18.2 Constructor & Destructor Documentation

12.18.2.1 __init__()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.__init__ (
              self,
              config )
```

```
Parameters:
config    Dictionary containing Fabric-specific parameters
```

12.18.3 Member Function Documentation

12.18.3.1 encryption_key_get()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.encryption↩
_key_get (
              self,
              worker_id,
              requester_id,
              last_used_key_nonce = None,
              tag = None,
              signature_nonce = None,
              signature = None )
```

```
Get worker's key from Fabric blockchain.
Not supported for Fabric.
```

12.18.3.2 encryption_key_set()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.encryption↩
_key_set (
              self,
              worker_id,
              encryption_key,
              encryption_nonce,
              tag,
              signature )
```

```
Set worker's encryption key.
Not supported for Fabric.
```

12.18.3.3 encryption_key_start()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.encryption←↩
_key_start (
              self,
              tag )
```

```
Initiate setting the encryption key of the worker.
Not supported for Fabric.
```

12.18.3.4 get_work_order_completed_event_handler()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.get_work_←↩
order_completed_event_handler (
              self,
              handler_func )
```

```
Start event handler loop for a workOrderCompleted event.

Parameters:
handler_func Callback function name as a string
```

12.18.3.5 get_work_order_submitted_event_handler()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.get_work_←↩
order_submitted_event_handler (
              self,
              handler_func )
```

```
Start event handler loop for a workOrderSubmitted event.

Parameters:
handler_func  Callback function name as a string

Returns:
Event handler object.
```

12.18.3.6 work_order_complete()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.work_order↩
_complete (
              self,
              work_order_id,
              work_order_response )
```

```
This function is called by the Worker Service to
complete a work order successfully or in error.
This API is for the proxy model.

Parameters:
work_order_id       Unique ID to identify the work order request
work_order_response Work order response data in a string

Returns:
errorCode           0 on success or non-zero on error.
```

12.18.3.7 work_order_get_result()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.work_order↩
_get_result (
              self,
              work_order_id,
              id = None )
```

```
Query blockchain to get work order result.

Parameters:
work_order_id Work Order ID that was sent in the
      corresponding work_order_submit request
id            Optional JSON RPC request ID

Returns:
Tuple containing work order status, worker id, work order request,
work order response, and error code.
None on error.
```

12.18.3.8 work_order_submit()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.work_order↩
_submit (
              self,
              work_order_id,
              worker_id,
              requester_id,
              work_order_request,
              id = None )
```

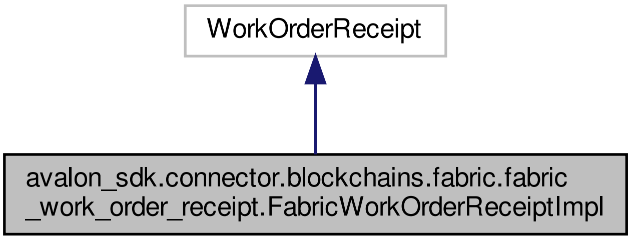
```
Submit work order request to the Fabric block chain.

Parameters:
work_order_id      Unique ID of the work order request
worker_id          Identifier for the worker
requester_id       Unique id to identify the requester
work_order_request JSON RPC string work order request.
           Complete definition at work_order.py and
           defined in EEA specification 6.1.1
id                 Optional JSON RPC request ID

Returns:
0 on success and non-zero on error.
```

The documentation for this class was generated from the following file:

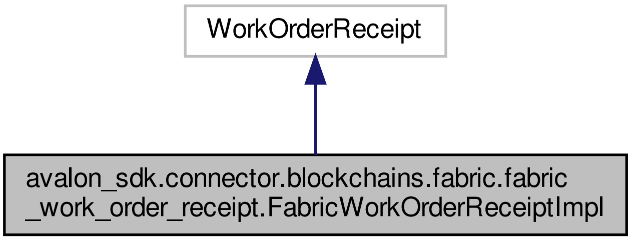
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/fabric_work_order.py

12.19 avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWork↩OrderReceiptImpl Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrder↩ReceiptImpl:


WorkOrderReceipt
avalon_sdk.connector.blockchains.fabric.fabric
_work_order_receipt.FabricWorkOrderReceiptImpl

Collaboration diagram for avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrder↩ReceiptImpl:


WorkOrderReceipt
avalon_sdk.connector.blockchains.fabric.fabric
_work_order_receipt.FabricWorkOrderReceiptImpl

Public Member Functions

- def __init__ (self, config)
- def work_order_receipt_create (self, work_order_id, worker_id, worker_service_id, requester_id, receipt_create_status, work_order_request_hash)
- def work_order_receipt_update (self, work_order_id, updater_id, update_type, update_data, update_signature=None, signature_rules=None)
- def work_order_receipt_retrieve (self, work_order_id)
- def work_order_receipt_update_retrieve (self, work_order_id, updater_id, update_index)
- def work_order_receipt_lookup (self, worker_service_id, worker_id, requester_id, receipt_status)
- def work_order_receipt_lookup_next (self, worker_service_id, worker_id, requester_id, receipt_status, last_lookup_tag)

Public Attributes

- **CHAIN_CODE**

12.19.1 Detailed Description

```
This class provides work order receipt management APIs
which interact with the Fabric blockchain.
Detailed method descriptions are available in the
WorkOrderReceipt interface.
```

12.19.2 Constructor & Destructor Documentation

12.19.2.1 __init__()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceiptImpl.__init__ (
              self,
              config )
```

```
Parameters:
config    Dict containing Fabric-specific parameters.
```

12.19.3 Member Function Documentation

12.19.3.1 work_order_receipt_create()

def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←↩ Impl.work_order_receipt_create (
self,
work_order_id,
worker_id,
worker_service_id,
requester_id,
receipt_create_status,
work_order_request_hash)

```
Create work order receipt in the Fabric block chain.

Parameters:
work_order_id           ID of the Work Order
worker_id               Worker id that should execute the Work Order
worker_service_id       ID of the Worker Service that
                hosts the Worker
requester_id            ID of the requester
receipt_create_status   Initial receipt status defined
                in EEA spec 7.1.1
work_order_request_hash Hash value of the work order request as
                defined in EEA spec 6.7

Returns:
0 on success and -1 on error.
```

12.19.3.2 work_order_receipt_lookup()

def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←↩ Impl.work_order_receipt_lookup (
self,
worker_service_id,
worker_id,
requester_id,
receipt_status)

```
Lookup a work order receipt.

Parameters:
worker_service_id Worker Service ID whose receipts will be
          retrieved
worker_id         Worker Id whose receipts are requested
requester_id      ID of the entity requesting receipts
receipt_status    Defines the status of the receipts retrieve
id                Optional JSON RPC request ID

Returns:
Tuple containing total count, last_lookup_tag, and
list of work order IDs, on success:
total_count     Total number of receipts matching the lookup criteria.
        If this number is bigger than the size of the ids
        array, the caller should use a lookup_tag to call
        work_order_receipt_lookup_next() to retrieve the rest
        of the receipt IDs.
last_lookup_tag Optional lookup_tag when the receipts exceed the ids
        array size
ids             Array of work order receipt ids that match the input

On error, returns -1.
```

12.19.3.3 work_order_receipt_lookup_next()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←
Impl.work_order_receipt_lookup_next (
              self,
              worker_service_id,
              worker_id,
              requester_id,
              receipt_status,
              last_lookup_tag )
```

```
Retrieve subsequent work order receipts after calling
work_order_receipt_lookup().

Parameters:
worker_service_id Worker Service ID
worker_id         Worker ID value derived from the worker’s DID
requester_id      Requester ID
last_lookup_tag   One of the output parameters for function
          work_order_receipt_lookup()
id                Optional JSON RPC request ID

Returns:
On success, return a tuple containing total count, look up tag, and
list of work order IDs:
total_count       Total number of receipts matching the lookup
          criteria
lookup_tag        Optional parameter. If it is returned, it means
          that there are more matching receipts that can be
          retrieved by calling this function again and with
          this tag as an input parameter.
ids               Array of the Work Order receipt IDs that match the
          input criteria from the corresponding call to
          work_order_receipt_lookup().
Return -1 on error.
```

12.19.3.4 work_order_receipt_retrieve()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←
Impl.work_order_receipt_retrieve (
              self,
              work_order_id )
```

```
Retrieve a Work Order Receipt.

Parameters:
work_order_id ID of the Work Order to be retrieved
id        Optional JSON RPC request ID

Returns:
worker_service_id, requester_id, work_order_id, receipt_create_status,
and work_order_request_hash, as defined in work_order_receipt_create().
Return -1 on error.
```

12.19.3.5 work_order_receipt_update()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←↩
Impl.work_order_receipt_update (
              self,
              work_order_id,
              updater_id,
              update_type,
              update_data,
              update_signature = None,
              signature_rules = None )
```

```
Update a Work Order Receipt.

Parameters:
work_order_id    Work Order ID that was sent in the
         corresponding work_order_submit request
updater_id       ID of the updating entity. It is optional if it
         is the same as the transaction sender address
update_type      Type of the Work Order update that defines
         how the update should be handled
update_data      Update-specific data that depends on the
         updater type defined in EEA spec 7.1.2
update_signature Optional signature of concatenated
         work_order_id, update_type, and update_data
signature_rules  Defines hashing and signing algorithms,
         that are separated by forward slash ’/’

Returns:
0 on success, -1 on error.
```

12.19.3.6 work_order_receipt_update_retrieve()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←↩
Impl.work_order_receipt_update_retrieve (
              self,
              work_order_id,
              updater_id,
              update_index )
```

```
Retrieve an update to a work order receipt.

Parameters:
work_order_id Work Order ID that was sent in the
      corresponding work_order_submit request
updater_id    ID of the updating entity. Ignored if null
update_index  Index of the update to retrieve
      Value "0xFFFFFFFF" is reserved to retrieve the
      last received update

Returns:
On success, return updater_id, update_type, update_data,
update_signature, signature_rules, as defined in
work_order_receipt_update(), and update_count.
On error, return -1.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/fabric_work_order_receipt.py

12.20 avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper Class Reference

Public Member Functions

- def __init__ (self, config)
- def invoke_chaincode (self, chaincode_name, method_name, params)
- def get_event_handler (self, event_name, chain_code, handler_func)

12.20.1 Detailed Description

```
Fabric wrapper class to interact with Fabric blockchain.
It provides wrapper functions to invoke and query chain code.
```

12.20.2 Constructor & Destructor Documentation

12.20.2.1 __init__()

```
def avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper.__init__ (
              self,
              config )
```

```
Constructor to initialize wrapper with required parameter.

Parameters:
config    Dictionary containing parameters for Fabric.
  These parameters are read from a .toml file
```

12.20.3 Member Function Documentation

12.20.3.1 get_event_handler()

```
def avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper.get_event_handler (
              self,
              event_name,
              chain_code,
              handler_func )
```

```
Create event handler object.

Parameters:
event_name   String to identify the event name
chain_code   Chain code name as string
handler_func Callback function name

Returns:
Event object
```

12.20.3.2 invoke_chaincode()

```
def avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper.invoke_chaincode (
              self,
              chaincode_name,
              method_name,
              params )
```

```
This is wrapper method to invoke chain code.

Parameters:
chaincode_name Name of the chain code
method_name    Chain code method name
params         List of arguments to method

Returns:
If the call to chain code query, then it
returns the payload of the chain code response
on success or None on error.
If the call is invoking chain code, then it
returns ContractResponse.SUCCESS on success
and ContractResponse.ERROR on failure.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/fabric_wrapper.py

12.21 FileIoExecutor Class Reference

Public Member Functions

- void **SetIoHandlerId** (uint32_t handler_id)
- void **SetFileName** (std::string file_name)
- uint32_t GetIoHandlerId (const char ∗handlerName)
- size_t GetMaxFileSize ()
- size_t GetMaxIoResultSize ()
- uint32_t FileOpen (uint8_t ∗result, size_t result_size)
- uint32_t FileClose (uint8_t ∗result, size_t result_size)
- uint32_t FileRead (uint8_t ∗result, size_t result_size, uint8_t ∗out_buf, size_t out_buf_size)
- uint32_t FileWrite (uint8_t ∗result, size_t result_size, const uint8_t ∗in_buf, size_t in_buf_size)
- uint32_t FileTell (uint8_t ∗result, size_t result_size, uint8_t ∗out_buf, size_t out_buf_size)
- uint32_t FileSeek (size_t position, uint8_t ∗result, size_t result_size)
- uint32_t FileDelete (uint8_t ∗result, size_t result_size)

12.21.1 Member Function Documentation

12.21.1.1 FileClose()

```
uint32_t FileIoExecutor::FileClose (
             uint8_t * result,
             size_t result_size )
```

Closes given file and updates status in the result buffer.

Parameters

result	Status of file close operation (0 is success, non-0 is failure)
result_size	Maximum size of the result buffer in bytes

Returns

Status of operation (0 on success, non-0 on failure)

12.21.1.2 FileDelete()

```
uint32_t FileIoExecutor::FileDelete (
            uint8_t * result,
            size_t result_size )
```

Deletes the file whose name is stored in the FileIoExecutor instance.

Parameters

result	Status of file delete operation (0 is success, non-0 is failure)
result_size	Maximum size of the result buffer in bytes

Returns

Status of operation (0 on success, non-0 on failure)

12.21.1.3 FileOpen()

```
uint32_t FileIoExecutor::FileOpen (
            uint8_t * result,
            size_t result_size )
```

Opens given file and updates status in the result buffer.

Parameters

result	Status of file open operation (0 is success, non-0 is failure)
result_size	Maximum size of the result buffer in bytes

Returns

Status of operation (0 on success, non-0 on failure)

12.21.1.4 FileRead()

```
uint32_t FileIoExecutor::FileRead (
            uint8_t * result,
            size_t result_size,
            uint8_t * out_buf,
            size_t out_buf_size )
```

Reads given file, stores content in out buffer and updates status in result buffer.

Parameters

result	Status of file read operation (0 is success, non-0 is failure)
result_size	Maximum size of the result buffer in bytes
out_buf	Buffer to hold file content
out_buf_size	Maximum size of out_buf to contain the file contents in bytes

Returns

Status of operation (0 on success, non-0 on failure)

12.21.1.5 FileSeek()

```
uint32_t FileIoExecutor::FileSeek (
            size_t position,
            uint8_t * result,
            size_t result_size )
```

Moves the file position the file to the given position and updates the status in result buffer.

Parameters

position	Byte offset of new file position
result	Status of file seek operation (0 is success, non-0 is failure)
result_size	Maximum size of the result buffer in bytes

Returns

Status of operation (0 on success, non-0 on failure)

12.21.1.6 FileTell()

```
uint32_t FileIoExecutor::FileTell (
            uint8_t * result,
            size_t result_size,
            uint8_t * out_buf,
            size_t out_buf_size )
```

Gets the current position of the file, stores it in buffer out_buf, and updates status in result buffer.

Parameters

result	status of file tell operation (0 is success, non-0 is failure)
result_size	Maximum size of the result buffer in bytes
out_buf	Buffer to hold file position
out_buf_size	Maximum size of out_buf to contain the file position in bytes

Returns

Status of operation (0 on success, non-0 on failure)

12.21.1.7 FileWrite()

```
uint32_t FileIoExecutor::FileWrite (
            uint8_t * result,
            size_t result_size,
            const uint8_t * in_buf,
            size_t in_buf_size )
```

Writes given file with content in input buffer and updates status in result buffer.

Parameters

result	Status of file write operation (0 is success, non-0 is failure)
result_size	Maximum size of the result buffer in bytes
in_buf	Buffer with content to be written to the file
in_buf_size	Maximum size of in_buf to write to the file in bytes

Returns

Status of operation (0 on success, non-0 on failure)

12.21.1.8 GetIoHandlerId()

```
uint32_t FileIoExecutor::GetIoHandlerId (
            const char * handler_name )
```

Get the I/O handler ID corresponding to IoHandler handler_name.

Parameters

handlerName	Name of handler

Returns

I/O handler ID. That is, 1 for handler "tcf-base-file-io"
0 on error

12.21.1.9 GetMaxFileSize()

```
size_t FileIoExecutor::GetMaxFileSize ( )
```

Get the maximum size of the buffer used for file I/O.

Returns

Maximum buffer size in bytes

12.21.1.10 GetMaxIoResultSize()

```
size_t FileIoExecutor::GetMaxIoResultSize ( )
```

Get the maximum size of the result buffer used to store the I/O status.

Returns

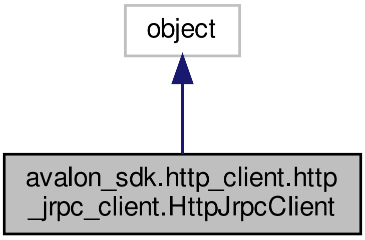
Maximum result buffer size in bytes

The documentation for this class was generated from the following files:

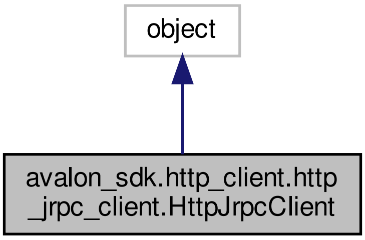
- /home/dano/git/avalon/common/sgx_workload/iohandler/file_io.h
- /home/dano/git/avalon/common/sgx_workload/iohandler/file_io.cpp

12.22 avalon_sdk.http_client.http_jrpc_client.HttpJrpcClient Class Reference

Inheritance diagram for avalon_sdk.http_client.http_jrpc_client.HttpJrpcClient:



Collaboration diagram for avalon_sdk.http_client.http_jrpc_client.HttpJrpcClient:



Public Member Functions

- def **__init__** (self, url)

Public Attributes

- **ServiceURL**
- **ProxyHandler**

12.22.1 Detailed Description

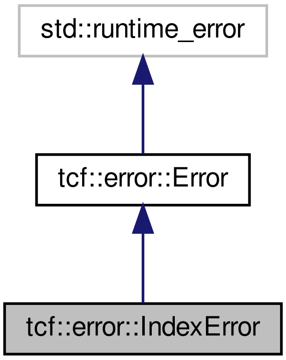
```
Class to handle HTTP JSON RPC communication by the client.
```

The documentation for this class was generated from the following file:

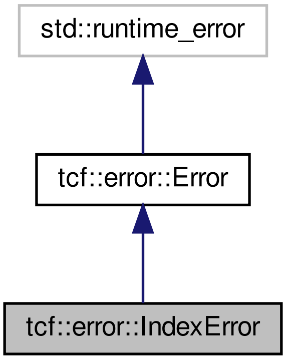
- /home/dano/git/avalon/sdk/avalon_sdk/http_client/http_jrpc_client.py

12.23 tcf::error::IndexError Class Reference

Inheritance diagram for tcf::error::IndexError:



Collaboration diagram for tcf::error::IndexError:



Public Member Functions

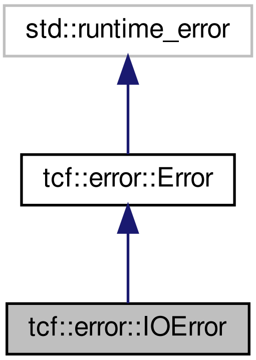
- **IndexError** (const std::string &msg)

The documentation for this class was generated from the following file:

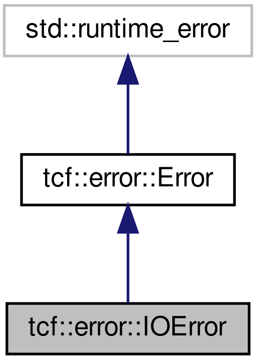
- /home/dano/git/avalon/common/cpp/error.h

12.24 tcf::error::IOError Class Reference

Inheritance diagram for tcf::error::IOError:



Collaboration diagram for tcf::error::IOError:



Public Member Functions

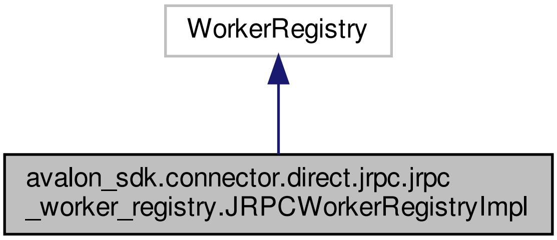
- **IOError** (const std::string &msg)

The documentation for this class was generated from the following file:

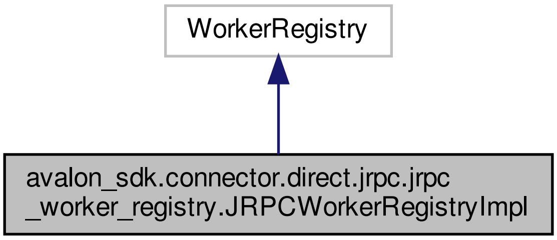
- /home/dano/git/avalon/common/cpp/error.h

12.25 avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl Class Reference

Inheritance diagram for avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl:


WorkerRegistry
avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl

Collaboration diagram for avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl:


WorkerRegistry
avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl

Public Member Functions

- def **__init__** (self, config)
- def worker_retrieve (self, worker_id, id=None)
- def worker_lookup (self, worker_type=None, organization_id=None, application_type_id=None, id=None)
- def worker_lookup_next (self, lookup_tag, worker_type=None, organization_id=None, application_type_id=None, id=None)
- def worker_register (self, worker_id, worker_type, org_id, application_type_ids, details, id=None)
- def worker_update (self, worker_id, details, id=None)
- def worker_set_status (self, worker_id, status, id=None)

12.25.1 Detailed Description

```
This class is to read the worker registry to get the more details
of worker.
```

12.25.2 Member Function Documentation

12.25.2.1 worker_lookup()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_lookup
(
              self,
              worker_type = None,
              organization_id = None,
              application_type_id = None,
              id = None )
```

```
Worker lookup based on worker type, organization ID,
and application ID.
All fields are optional and, if present, condition should match for
all fields. If none are passed it should return all workers.

Parameters:
worker_type         Optional characteristic of Workers for which you
            may wish to search. Currently defined types are:
            * "TEE-SGX": an Intel SGX Trusted Execution
              Environment
            * "MPC": Multi-Party Compute
            * "ZK": Zero-Knowledge
organization_id     Optional parameter representing the
            organization that hosts the Worker,
            e.g. a bank in the consortium or
            anonymous entity
application_type_id Optional application type that has to be supported
            by the worker
id                  Optional Optional JSON RPC request ID


Returns:
JRPC response containing number of workers,
lookup tag, and list of worker IDs.
```

12.25.2.2 worker_lookup_next()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_↩
lookup_next (
              self,
              lookup_tag,
              worker_type = None,
              organization_id = None,
              application_type_id = None,
              id = None )
```

```
Retrieve subsequent Worker lookup results based on worker type,
organization ID, and application ID.
Similar to workerLookUp with additional parameter lookup_tag.

Parameters:
lookup_tag          Used to lookup subsequent results after calling
```

```
            worker_lookup
worker_type         Optional characteristic of Workers for which you
            may wish to search. Currently defined types are:
            * "TEE-SGX": an Intel SGX Trusted Execution
              Environment
            * "MPC": Multi-Party Compute
            * "ZK": Zero-Knowledge
organization_id     Optional parameter representing the
            organization that hosts the Worker,
            e.g. a bank in the consortium or
            anonymous entity
application_type_id Optional application type that has to be supported
            by the worker
id                  Optional Optional JSON RPC request ID

Returns:
JRPC response containing number of workers,
lookup tag, and list of worker IDs.
```

12.25.2.3 worker_register()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_↩
register (
              self,
              worker_id,
              worker_type,
              org_id,
              application_type_ids,
              details,
              id = None )
```

```
Adds worker details to registry

Parameters:
worker_id            Worker ID value derived from the worker’s DID
worker_type          Type of Worker. Currently defined types are:
             * "TEE-SGX": an Intel SGX Trusted Execution
               Environment
             * "MPC": Multi-Party Compute
             * "ZK": Zero-Knowledge
org_id               Organization that hosts the Worker,
             e.g. a bank in the consortium or
             anonymous entity
application_type_ids Application types supported by the worker
id                   Optional JSON RPC request ID

Returns:
JRPC response with worker registry status.
```

12.25.2.4 worker_retrieve()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_↩
retrieve (
              self,
              worker_id,
              id = None )
```

```
Retrieve the worker identified by worker ID.

Parameters:
worker_id Worker ID value derived from the worker's DID
id        Optional Optional JSON RPC request ID

Returns:
JRPC response containing:
organization ID, application ID, worker status,
and worker details.
```

12.25.2.5 worker_set_status()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_set_↩
status (
              self,
              worker_id,
              status,
              id = None )
```

```
Set the worker status to active, offline,
decommissioned, or compromised state.

Parameters:
worker_id  Worker ID value derived from the worker's DID
status     Worker status value to set
id         Optional JSON RPC request ID

Returns:
JRPC response with status.
```

12.25.2.6 worker_update()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_update
(
              self,
              worker_id,
              details,
              id = None )
```

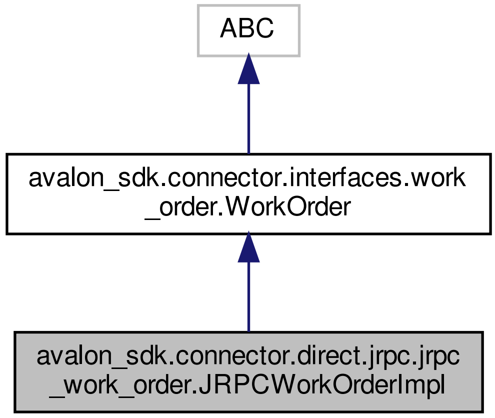
```
Update worker with new information.

Parameters:
worker_id Worker ID value derived from the worker's DID
details   Detailed information about the worker in
  JSON RPC format as defined in
id        Optional JSON RPC request ID

Returns:
JRPC response with update status.
```

The documentation for this class was generated from the following file:

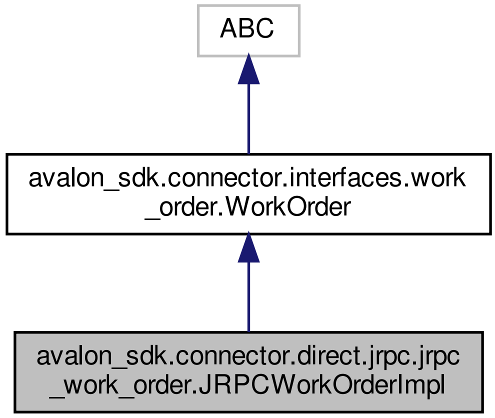
- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/jrpc_worker_registry.py

12.26 avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl Class Reference

Inheritance diagram for avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl:


ABC
avalon_sdk.connector.interfaces.work_order.WorkOrder
avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl

Collaboration diagram for avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl:


ABC
avalon_sdk.connector.interfaces.work_order.WorkOrder
avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl

Public Member Functions

- def **__init__** (self, config)
- def work_order_submit (self, work_order_id, worker_id, requester_id, work_order_request, id=None)

- def work_order_get_result_nonblocking (self, work_order_id, id=None)
- def work_order_get_result (self, work_order_id, id=None)
- def encryption_key_get (self, worker_id, requester_id, last_used_key_nonce=None, tag=None, signature_↩ nonce=None, signature=None, id=None)
- def encryption_key_set (self, worker_id, encryption_key, encryption_nonce, tag, signature_nonce, signature, id=None)

12.26.1 Detailed Description

```
This class is for to manage to the work orders from client side.
```

12.26.2 Member Function Documentation

12.26.2.1 encryption_key_get()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.encryption_key_get (
              self,
              worker_id,
              requester_id,
              last_used_key_nonce = None,
              tag = None,
              signature_nonce = None,
              signature = None,
              id = None )
```

```
API to receive a worker's key.

Parameters:
worker_id           Worker ID of the worker whose encryption key
            is requested
last_used_key_nonce Optional nonce associated with the last retrieved
            key. If it is provided, the key retrieved should
            be newer than this one.
            Otherwise any key can be retrieved
tag                 Tag that should be associated with the returned
            key, e.g. the requester ID. This is an optional
            parameter. If it is not provided, requester_id is
            used as a key
requester_id        ID of the requester that plans to use
            the returned key to submit one or more work orders
            using this key
signature_nonce     Optional nonce associated with the signature and
            is used only if signature below is also provided
signature           Optional signature of worker_id,
            last_used_key_nonce, tag, and signature_nonce.
id                  Optional JSON RPC request ID
```

12.26.2.2 encryption_key_set()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.encryption_key_set (
              self,
              worker_id,
              encryption_key,
              encryption_nonce,
              tag,
              signature_nonce,
              signature,
              id = None )
```

```
API called by a Worker or Worker Service to receive a Worker's key.

Parameters:
worker_id        ID of the worker to set an encryption key
encryption_key   Encryption key to set
encryption_nonce Nonce associated with the key
tag              Tag that should be associated with the returned key,
         e.g. requester ID.
signature_nonce  Nonce associated with the signature
signature        Signature generated by the worker on the worker_id,
         tag and encryption_nonce
id               Optional JSON RPC request ID

Returns:
JRPC response with the result of the operation.
```

12.26.2.3 work_order_get_result()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.work_order_get_result (
              self,
              work_order_id,
              id = None )
```

```
Get the work order result in a blocking way until it gets a
result or error.

Parameters:
work_order_id     Work order ID
id                Optional JSON RPC request ID

Returns:
JSON RPC response of dictionary type
```

12.26.2.4 work_order_get_result_nonblocking()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.work_order_get_result↩
_nonblocking (
              self,
              work_order_id,
              id = None )
```

```
Get the work order result in non-blocking way.

Parameters:
work_order_id     Work order ID
id                Optional JSON RPC request ID

Returns:
JSON RPC response of dictionary type
```

12.26.2.5 work_order_submit()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.work_order_submit (
              self,
              work_order_id,
              worker_id,
              requester_id,
              work_order_request,
              id = None )
```

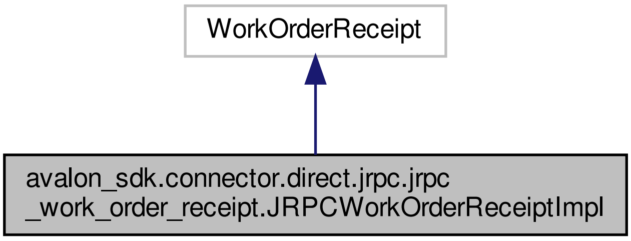
```
Submit a work order request to an Avalon listener.

Parameters:
work_order_id     Work order ID
worker_id         Worker ID value derived from the worker's DID
requester_id      Requester ID
work_order_request Work order request in JSON RPC string format
id                Optional JSON RPC request ID
```

The documentation for this class was generated from the following file:

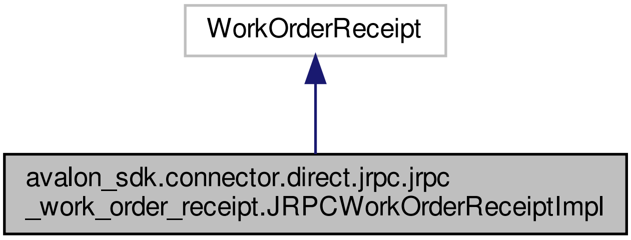
- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/jrpc_work_order.py

12.27 avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl Class Reference

Inheritance diagram for avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl:


WorkOrderReceipt
avalon_sdk.connector.direct.jrpc.jrpc
_work_order_receipt.JRPCWorkOrderReceiptImpl

Collaboration diagram for avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl:


WorkOrderReceipt
avalon_sdk.connector.direct.jrpc.jrpc
_work_order_receipt.JRPCWorkOrderReceiptImpl

Public Member Functions

- def **__init__** (self, config)
- def work_order_receipt_create (self, work_order_id, worker_service_id, worker_id, requester_id, receipt_create_status, work_order_request_hash, requester_nonce, requester_signature, signature_rules, receipt_verification_key, id=None)
- def work_order_receipt_update (self, work_order_id, updater_id, update_type, update_data, update_signature, signature_rules, id=None)
- def work_order_receipt_retrieve (self, work_order_id, id=None)
- def work_order_receipt_update_retrieve (self, work_order_id, updater_id, update_index, id=None)
- def work_order_receipt_lookup (self, worker_service_id=None, worker_id=None, requester_id=None, receipt_status=None, id=None)
- def work_order_receipt_lookup_next (self, last_lookup_tag, worker_service_id=None, worker_id=None, requester_id=None, receipt_status=None, id=None)

12.27.1 Detailed Description

```
This class is an implementation of WorkOrderReceiptInterface
to manage work order receipts from the client side.
```

12.27.2 Member Function Documentation

12.27.2.1 work_order_receipt_create()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_order_receipt_create (
              self,
              work_order_id,
              worker_service_id,
              worker_id,
              requester_id,
              receipt_create_status,
              work_order_request_hash,
              requester_nonce,
              requester_signature,
              signature_rules,
              receipt_verification_key,
              id = None )
```

```
Create a Work Order Receipt JSON RPC request and submit to an
Avalon listener.

Parameters:
work_order_id            Work order ID
worker_service_id        Worker service ID
worker_id                Worker ID value derived from the worker's DID
requester_id             Requester ID
receipt_create_status    Receipt creation status
work_order_request_hash  Work order request hash value
requester_nonce          Requester generated nonce
requester_signature      Signature generated by the requester
signature_rules          Defines hashing and signing algorithms;
                 separated by forward slash '/'
receipt_verification_key Receipt verification key
id                       Optional JSON RPC request ID
```

12.27.2.2 work_order_receipt_lookup()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_↩
order_receipt_lookup (
              self,
              worker_service_id = None,
              worker_id = None,
              requester_id = None,
              receipt_status = None,
              id = None )
```

```
Work Order Receipt Lookup
All fields are optional and, if present, condition should match for
all fields. If none are passed it should return all
work order receipts.

Parameters:
worker_service_id        Optional worker service ID to lookup
worker_id                Optional worker ID value derived from
                 the worker's DID
requester_id             Optional requester ID to lookup
receipt_status           Optional receipt status
id                       Optional JSON RPC request ID
```

12.27.2.3 work_order_receipt_lookup_next()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_←
order_receipt_lookup_next (
              self,
              last_lookup_tag,
              worker_service_id = None,
              worker_id = None,
              requester_id = None,
              receipt_status = None,
              id = None )
```

```
Work Order Receipt Lookup Next.
Call to retrieve subsequent results after calling
work_order_receipt_lookup or

Parameters:
last_lookup_tag          Last lookup tag returned by
                 work_order_receipt_lookup
worker_service_id        Optional worker service ID to lookup
worker_id                Optional worker ID value derived from
                 the worker's DID
requester_id             Optional requester ID to lookup
receipt_status           Optional receipt status
id                       Optional JSON RPC request ID
```

12.27.2.4 work_order_receipt_retrieve()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_←
order_receipt_retrieve (
              self,
              work_order_id,
              id = None )
```

```
Retrieve a work order receipt JSON RPC request and submit to an
Avalon listener.

Parameters:
work_order_id Work order ID
id            Optional Optional JSON RPC request ID
```

12.27.2.5 work_order_receipt_update()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_←
order_receipt_update (
              self,
              work_order_id,
              updater_id,
              update_type,
              update_data,
              update_signature,
              signature_rules,
              id = None )
```

```
Update a Work Order Receipt JSON RPC request and submit an
Avalon listener.

Parameters:
work_order_id    Work Order ID
updater_id       Updater ID
update_type      Updater type
update_data      Receipt update data
update_signature Signature of the update
signature_rules  Defines hashing and signing algorithms;
         separated by forward slash '/'
id               Optional JSON RPC request ID
```

12.27.2.6 work_order_receipt_update_retrieve()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_↩
order_receipt_update_retrieve (
              self,
              work_order_id,
              updater_id,
              update_index,
              id = None )
```

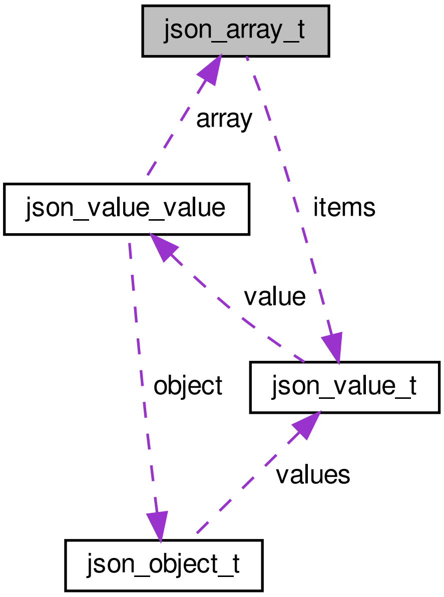
```
Retrieve a work order receipt update JSON RPC request and submit to an
Avalon listener.

Parameters:
work_order_id Work order ID
id            Optional Optional JSON RPC request ID
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/jrpc_work_order_receipt.py

12.28 json_array_t Struct Reference

Collaboration diagram for json_array_t:



Public Attributes

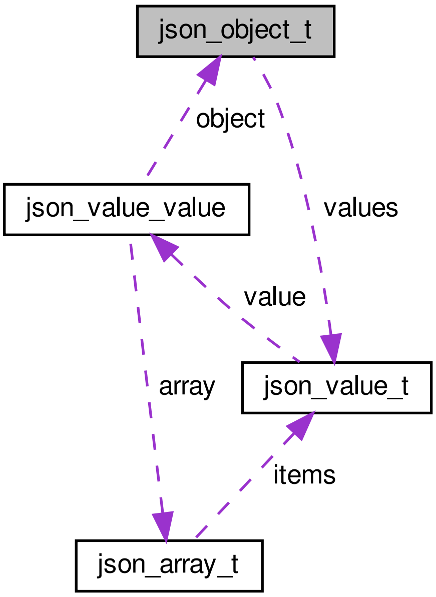
- JSON_Value ∗∗ **items**
- size_t **count**
- size_t **capacity**

The documentation for this struct was generated from the following file:

- /home/dano/git/avalon/common/cpp/packages/parson/parson.cpp

12.29 json_object_t Struct Reference

Collaboration diagram for json_object_t:



Public Attributes

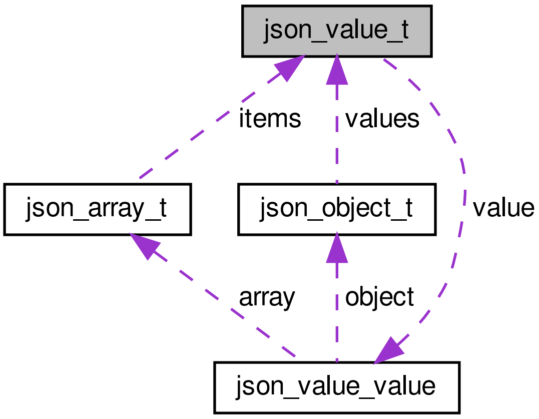
- char ∗∗ **names**
- JSON_Value ∗∗ **values**
- size_t **count**
- size_t **capacity**

The documentation for this struct was generated from the following file:

- /home/dano/git/avalon/common/cpp/packages/parson/parson.cpp

12.30 json_value_t Struct Reference

Collaboration diagram for json_value_t:



Public Attributes

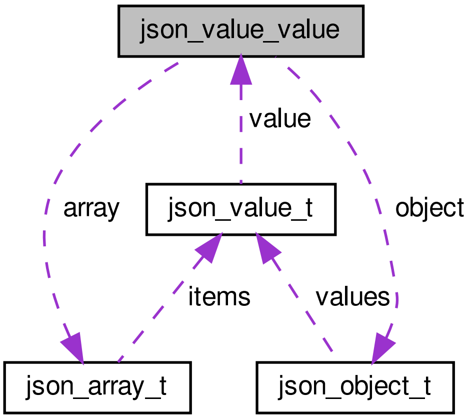
- JSON_Value_Type **type**
- JSON_Value_Value **value**

The documentation for this struct was generated from the following file:

- /home/dano/git/avalon/common/cpp/packages/parson/parson.cpp

12.31 json_value_value Union Reference

Collaboration diagram for json_value_value:



Public Attributes

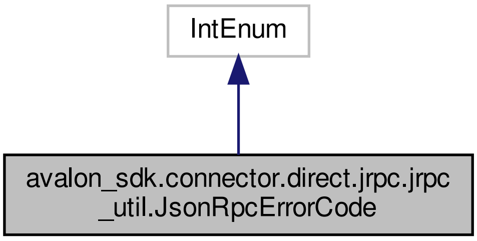
- char ∗ **string**
- double **number**
- JSON_Object ∗ **object**
- JSON_Array ∗ **array**
- int **boolean**
- int **null**

The documentation for this union was generated from the following file:

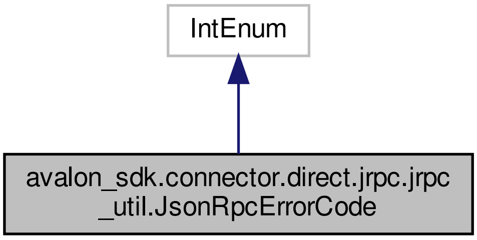
- /home/dano/git/avalon/common/cpp/packages/parson/parson.cpp

12.32 avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode Class Reference

Inheritance diagram for avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode:



Collaboration diagram for avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode:



Static Public Attributes

- int **SUCCESS** = 0
- int **UNKNOWN_ERROR** = 1
- int **INVALID_PARAMETER** = 2
- int **ACCESS_DENIED** = 3
- int **INVALID_SIGNATURE** = 4
- int **NO_LOOKUP_RESULTS** = 5
- int **UNSUPPORTED_MODE** = 6

12.32.1 Detailed Description

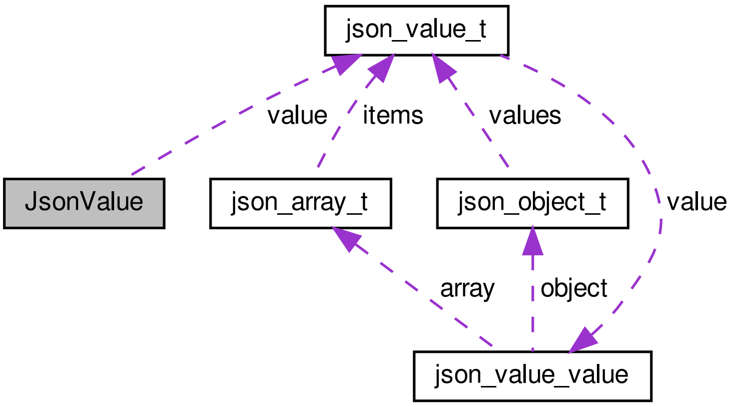
```
JSON RPC error code values:
0 - SUCCESS
1 - UNKNOWN_ERROR
2 - INVALID_PARAMETER format or value
3 - ACCESS_DENIED
4 - INVALID_SIGNATURE
5 - NO_LOOKUP_RESULTS no more lookup results remaining
6 - UNSUPPORTED_MODE (e.g. synchronous, asynchronous, poll,
    or notification)
-32768 to -32000 - reserved for pre-defined errors in the JSON RPC spec.

From EEA spec 4.1.1.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/jrpc_util.py

12.33 JsonValue Class Reference

Collaboration diagram for JsonValue:



Public Member Functions

- **JsonValue** (JSON_Value ∗value=nullptr)
- **operator JSON_Value** ∗ ()
- **operator const JSON_Value** ∗ () const

Public Attributes

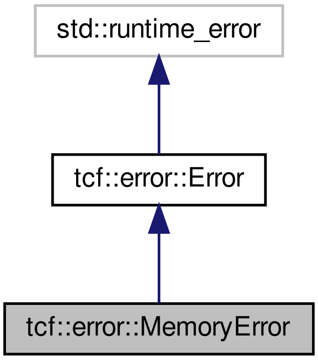
- JSON_Value ∗ **value**

The documentation for this class was generated from the following file:

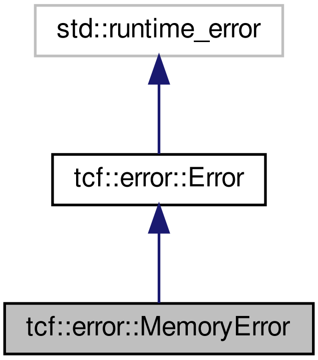
- /home/dano/git/avalon/common/cpp/jsonvalue.h

12.34 tcf::error::MemoryError Class Reference

Inheritance diagram for tcf::error::MemoryError:



Collaboration diagram for tcf::error::MemoryError:



Public Member Functions

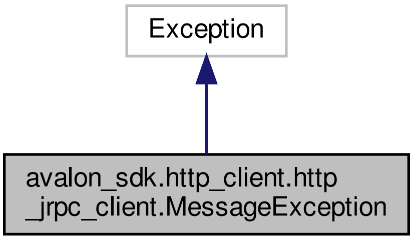
- **MemoryError** (const std::string &msg)

The documentation for this class was generated from the following file:

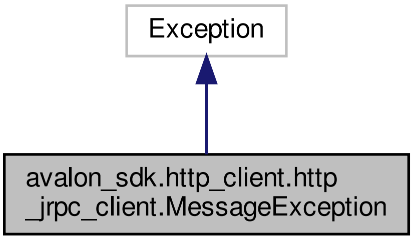
- /home/dano/git/avalon/common/cpp/error.h

12.35 avalon_sdk.http_client.http_jrpc_client.MessageException Class Reference

Inheritance diagram for avalon_sdk.http_client.http_jrpc_client.MessageException:



Collaboration diagram for avalon_sdk.http_client.http_jrpc_client.MessageException:



12.35.1 Detailed Description

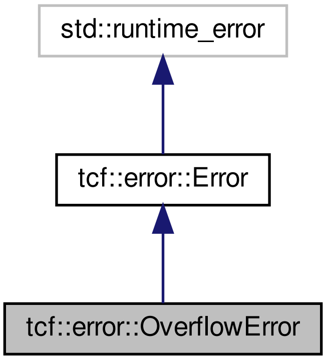
```
A class to capture communication exceptions when communicating with
services.
```

The documentation for this class was generated from the following file:

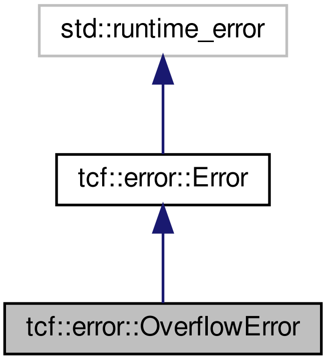
- /home/dano/git/avalon/sdk/avalon_sdk/http_client/http_jrpc_client.py

12.36 tcf::error::OverflowError Class Reference

Inheritance diagram for tcf::error::OverflowError:



Collaboration diagram for tcf::error::OverflowError:



Public Member Functions

- **OverflowError** (const std::string &msg)

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/common/cpp/error.h

12.37 tcf::crypto::pkenc::PrivateKey Class Reference

Public Member Functions

- **PrivateKey** (const PrivateKey &privateKey)
- **PrivateKey** (PrivateKey &&privateKey)
- **PrivateKey** (const std::string &encoded)
- PrivateKey & **operator=** (const PrivateKey &privateKey)
- void **Deserialize** (const std::string &encoded)
- void **Generate** ()
- PublicKey **GetPublicKey** () const
- std::string **Serialize** () const
- ByteArray **DecryptMessage** (const ByteArray &ct) const

The documentation for this class was generated from the following files:

- /home/dano/git/avalon/common/cpp/crypto/pkenc_private_key.h
- /home/dano/git/avalon/common/cpp/crypto/pkenc_private_key.cpp

12.38 tcf::crypto::sig::PrivateKey Class Reference

Public Member Functions

- **PrivateKey** (const PrivateKey &privateKey)
- **PrivateKey** (PrivateKey &&privateKey)
- **PrivateKey** (const std::string &encoded)
- PrivateKey & **operator=** (const PrivateKey &privateKey)
- void **Deserialize** (const std::string &encoded)
- void **Generate** ()
- PublicKey **GetPublicKey** () const
- std::string **Serialize** () const
- ByteArray **SignMessage** (const ByteArray &hashMessage) const

The documentation for this class was generated from the following files:

- /home/dano/git/avalon/common/cpp/crypto/sig_private_key.h
- /home/dano/git/avalon/common/cpp/crypto/sig_private_key.cpp

12.39 tcf::crypto::sig::PublicKey Class Reference

Public Member Functions

- **PublicKey** (const PublicKey &publicKey)
- **PublicKey** (PublicKey &&publicKey)
- **PublicKey** (const PrivateKey &privateKey)
- **PublicKey** (const std::string &encoded)
- PublicKey & **operator=** (const PublicKey &publicKey)
- void **Deserialize** (const std::string &encoded)
- std::string **Serialize** () const
- std::string **SerializeXYToHex** () const
- void **DeserializeXYFromHex** (const std::string &hexXY)
- int **VerifySignature** (const ByteArray &hashMessage, const ByteArray &signature) const

The documentation for this class was generated from the following files:

- /home/dano/git/avalon/common/cpp/crypto/sig_public_key.h
- /home/dano/git/avalon/common/cpp/crypto/sig_public_key.cpp

12.40 tcf::crypto::pkenc::PublicKey Class Reference

Public Member Functions

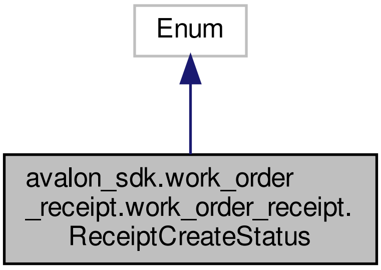
- **PublicKey** (const PublicKey &publicKey)
- **PublicKey** (PublicKey &&publicKey)
- **PublicKey** (const PrivateKey &privateKey)
- **PublicKey** (const std::string &encoded)
- PublicKey & **operator=** (const PublicKey &publicKey)
- void **Deserialize** (const std::string &encoded)
- std::string **Serialize** () const
- ByteArray **EncryptMessage** (const ByteArray &message) const

The documentation for this class was generated from the following files:

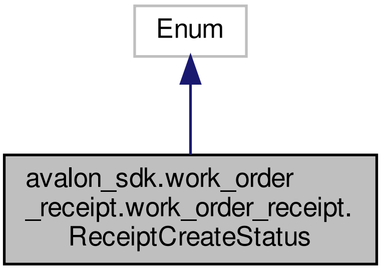
- /home/dano/git/avalon/common/cpp/crypto/pkenc_public_key.h
- /home/dano/git/avalon/common/cpp/crypto/pkenc_public_key.cpp

12.41 avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus Class Reference

Inheritance diagram for avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus:


Enum
avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus

Collaboration diagram for avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus:


Enum
avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus

Static Public Attributes

- int **PENDING** = 0
- int **COMPLETED** = 1
- int **PROCESSED** = 2
- int **FAILED** = 3
- int **REJECTED** = 4

12.41.1 Detailed Description

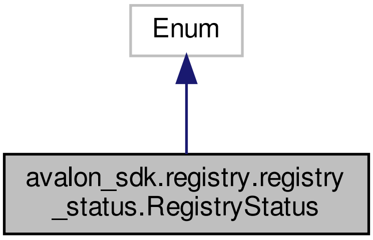
```
Receipt creation status values:
0 - PENDING. The work order is waiting to be processed by the worker
1 - COMPLETED. The worker processed the Work Order and no more worker
    updates are expected
2 - PROCESSED. The worker processed the Work Order, but additional worker
    updates are expected, e.g. oracle notifications
3 - FAILED. The Work Order processing failed, e.g. by the worker service
    because of an invalid workerId
4 - REJECTED. The Work Order is rejected by the smart contract,
    e.g. invalid workerServiceId
5 to 254 - reserved
255 - indicates any status
>255 - application-specific values

Defined in EEA spec 7.1.
```

The documentation for this class was generated from the following file:

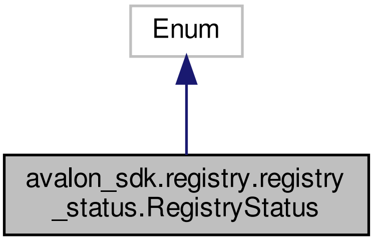
- /home/dano/git/avalon/sdk/avalon_sdk/work_order_receipt/work_order_receipt.py

12.42 avalon_sdk.registry.registry_status.RegistryStatus Class Reference

Inheritance diagram for avalon_sdk.registry.registry_status.RegistryStatus:



Collaboration diagram for avalon_sdk.registry.registry_status.RegistryStatus:



Static Public Attributes

- int **ACTIVE** = 1
- int **OFF_LINE** = 2
- int **DECOMMISSIONED** = 3

12.42.1 Detailed Description

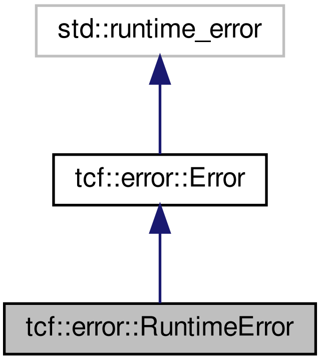
```
Worker registry status values:
1 - registry is ACTIVE
2 - registry is temporarily OFF_LINE
3 - registry is DECOMMISSIONED

From EEA spec 5.2.
```

The documentation for this class was generated from the following file:

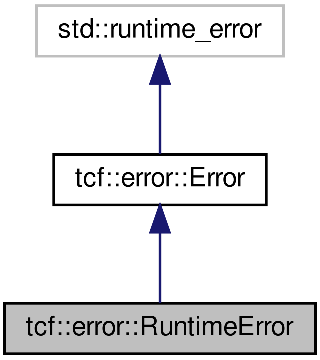
- /home/dano/git/avalon/sdk/avalon_sdk/registry/registry_status.py

12.43 tcf::error::RuntimeError Class Reference

Inheritance diagram for tcf::error::RuntimeError:



Collaboration diagram for tcf::error::RuntimeError:



Public Member Functions

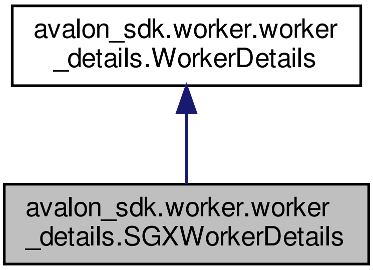
- **RuntimeError** (const std::string &msg)

The documentation for this class was generated from the following file:

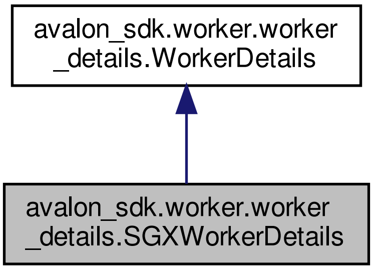
- /home/dano/git/avalon/common/cpp/error.h

12.44 avalon_sdk.worker.worker_details.SGXWorkerDetails Class Reference

Inheritance diagram for avalon_sdk.worker.worker_details.SGXWorkerDetails:



Collaboration diagram for avalon_sdk.worker.worker_details.SGXWorkerDetails:



Public Member Functions

- def **__init__** (self)
- def load_worker (self, worker_data)

Public Attributes

- **verification_key**
- **extended_measurements**
- **proof_data_type**
- **proof_data**
- **encryption_key**
- **encryption_key_nonce**
- **encryption_key_signature**

- **enclave_certificate**
- **worker_id**
- **hashing_algorithm**
- **signing_algorithm**
- **key_encryption_algorithm**
- **data_encryption_algorithm**

12.44.1 Detailed Description

```
Contains Intel SGX TEE worker type data.
```

12.44.2 Member Function Documentation

12.44.2.1 load_worker()

```
def avalon_sdk.worker.worker_details.SGXWorkerDetails.load_worker (
              self,
              worker_data )
```

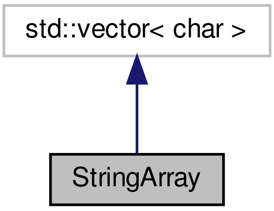
```
Load member variables of this class
based on worker-retrieved details.

Parameters:
    worker_data Worker Data to load into the class
```

The documentation for this class was generated from the following file:

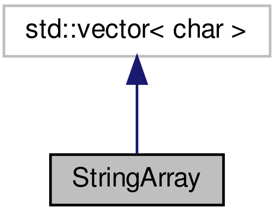
- /home/dano/git/avalon/sdk/avalon_sdk/worker/worker_details.py

12.45 StringArray Class Reference

Inheritance diagram for StringArray:



Collaboration diagram for StringArray:



Public Member Functions

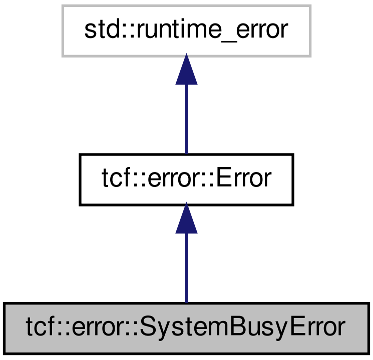
- **StringArray** (const std::string &value)
- **StringArray** (const size_t size)
- void **assign** (const std::string &value)
- std::string **str** ()

The documentation for this class was generated from the following file:

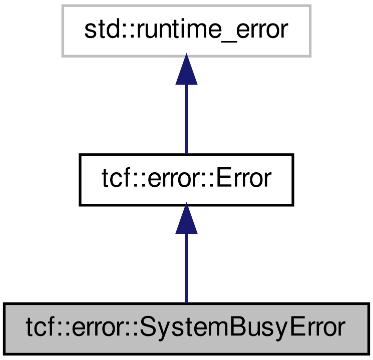
- /home/dano/git/avalon/common/cpp/types.h

12.46 tcf::error::SystemBusyError Class Reference

Inheritance diagram for tcf::error::SystemBusyError:



Collaboration diagram for tcf::error::SystemBusyError:



Public Member Functions

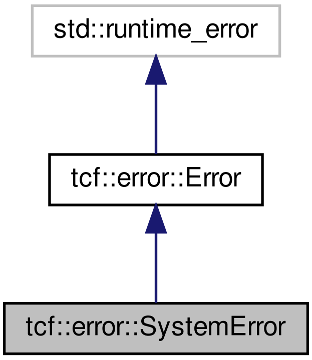
- **SystemBusyError** (const std::string &msg)

The documentation for this class was generated from the following file:

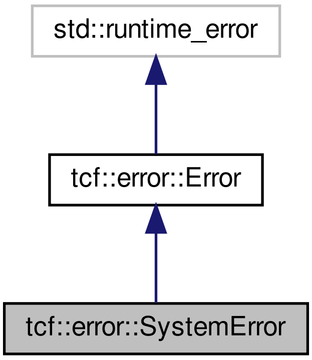
- /home/dano/git/avalon/common/cpp/error.h

12.47 tcf::error::SystemError Class Reference

Inheritance diagram for tcf::error::SystemError:



Collaboration diagram for tcf::error::SystemError:



Public Member Functions

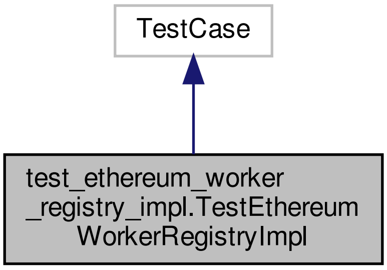
- **SystemError** (const std::string &msg)

The documentation for this class was generated from the following file:

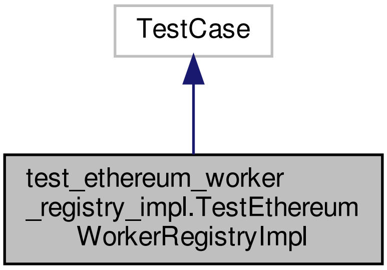
- /home/dano/git/avalon/common/cpp/error.h

12.48 test_ethereum_worker_registry_impl.TestEthereumWorkerRegistryImpl Class Reference

Inheritance diagram for test_ethereum_worker_registry_impl.TestEthereumWorkerRegistryImpl:



Collaboration diagram for test_ethereum_worker_registry_impl.TestEthereumWorkerRegistryImpl:



Public Member Functions

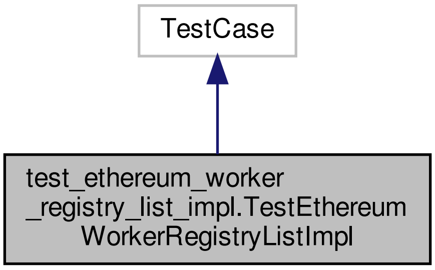
- def **__init__** (self, config_file)
- def **test_worker_register** (self)
- def **test_worker_set_status** (self)
- def **test_worker_update** (self)
- def **test_worker_lookup** (self)
- def **test_worker_retrieve** (self)
- def **test_worker_lookup_next** (self)

The documentation for this class was generated from the following file:

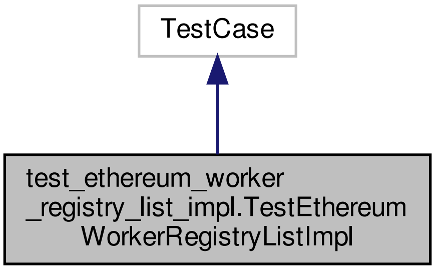
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/unit_tests/test_ethereum_worker↩ _registry_impl.py

12.49 test_ethereum_worker_registry_list_impl.TestEthereumWorkerRegistryListImpl Class Reference

Inheritance diagram for test_ethereum_worker_registry_list_impl.TestEthereumWorkerRegistryListImpl:



Collaboration diagram for test_ethereum_worker_registry_list_impl.TestEthereumWorkerRegistryListImpl:



Public Member Functions

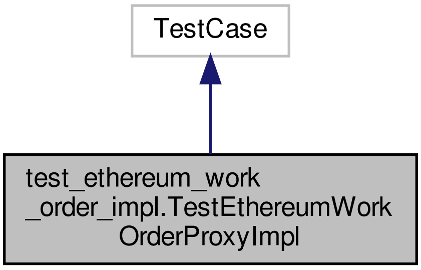
- def **__init__** (self, config_file)
- def **test_registry_add** (self)
- def **test_registry_update** (self)
- def **test_registry_set_status** (self)
- def **test_registry_lookup** (self)
- def **test_registry_retrieve** (self)
- def **test_registry_lookup_next** (self)

The documentation for this class was generated from the following file:

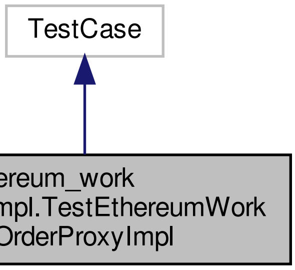
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/unit_tests/test_ethereum_worker↩ _registry_list_impl.py

12.50 test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl Class Reference

Inheritance diagram for test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl:



Collaboration diagram for test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl:


TestCase
test_ethereum_work _order_impl.TestEthereumWork OrderProxyImpl

Public Member Functions

- def **__init__** (self, config_file)
- def **test_work_order_submit_positive** (self)
- def **test_work_order_submit_mismatch** (self)
- def **test_work_order_get_result** (self)
- def test_work_order_complete (self)
- def test_work_order_complete_error (self)
- def test_is_wo_id_in_event_positive (self)
- def test_is_wo_id_in_event_wo_id_not_matched (self)
- def test_is_wo_id_in_event_error_result (self)
- def **test_is_wo_id_in_event_no_wo_id** (self)
- def **test_is_valid_work_order_json** (self)

12.50.1 Member Function Documentation

12.50.1.1 test_is_wo_id_in_event_error_result()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_is_wo_id_in_event_↩
error_result (
              self )
```

```
This case mocks an event and verifies the wo_id_in_event function
for a positive result. The event has an error response from
work order execution.
```

12.50.1.2 test_is_wo_id_in_event_positive()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_is_wo_id_in_event_↩
positive (
              self )
```

```
This case mocks an event and verifies the wo_id_in_event function
for a positive result.
```

12.50.1.3 test_is_wo_id_in_event_wo_id_not_matched()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_is_wo_id_in_event_wo_↩
id_not_matched (
              self )
```

```
This case mocks an event and verifies the wo_id_in_event function
for a negative result. The wo_id does not match.
```

12.50.1.4 test_work_order_complete()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_work_order_complete (
              self )
```

```
This function verifies if work order complete function
succeeds when the in work order execution is done.
```

12.50.1.5 test_work_order_complete_error()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_work_order_complete_↩
error (
              self )
```

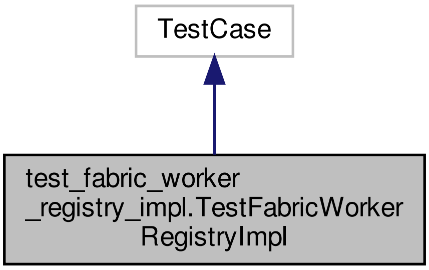
```
This function verifies if work order complete function
succeeds when there is an error in work order execution.
```

The documentation for this class was generated from the following file:

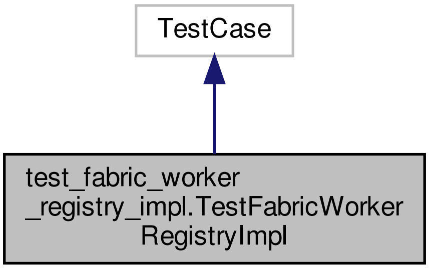
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/unit_tests/test_ethereum_work_↩
order_impl.py

12.51 test_fabric_worker_registry_impl.TestFabricWorkerRegistryImpl Class Reference

Inheritance diagram for test_fabric_worker_registry_impl.TestFabricWorkerRegistryImpl:


TestCase
test_fabric_worker _registry_impl.TestFabricWorker RegistryImpl

Collaboration diagram for test_fabric_worker_registry_impl.TestFabricWorkerRegistryImpl:


TestCase
test_fabric_worker _registry_impl.TestFabricWorker RegistryImpl

Public Member Functions

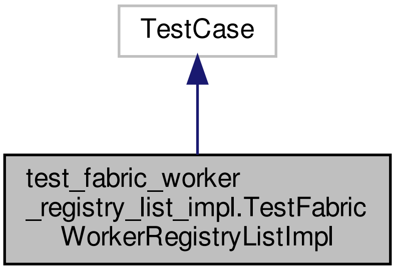
- def **__init__** (self, config_file)
- def **test_worker_register** (self)
- def **test_worker_set_status** (self)
- def **test_worker_update** (self)
- def **test_worker_lookup** (self)
- def **test_worker_retrieve** (self)
- def **test_worker_lookup_next** (self)

The documentation for this class was generated from the following file:

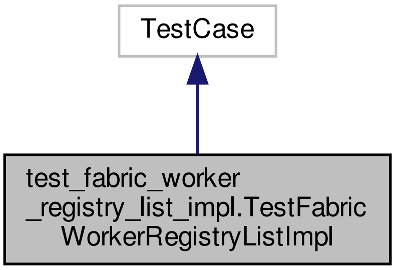
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/unit_tests/test_fabric_worker_registry←↩ _impl.py

12.52 test_fabric_worker_registry_list_impl.TestFabricWorkerRegistryListImpl Class Reference

Inheritance diagram for test_fabric_worker_registry_list_impl.TestFabricWorkerRegistryListImpl:



Collaboration diagram for test_fabric_worker_registry_list_impl.TestFabricWorkerRegistryListImpl:



Public Member Functions

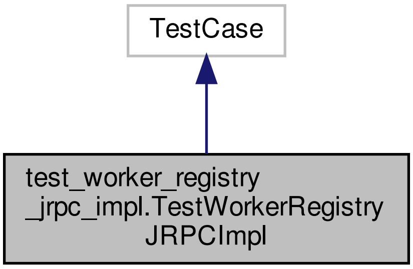
- def **__init__** (self, config_file)
- def **test_registry_add** (self)
- def **test_registry_update** (self)
- def **test_registry_set_status** (self)
- def **test_registry_lookup** (self)
- def **test_registry_retrieve** (self)
- def **test_registry_lookup_next** (self)

The documentation for this class was generated from the following file:

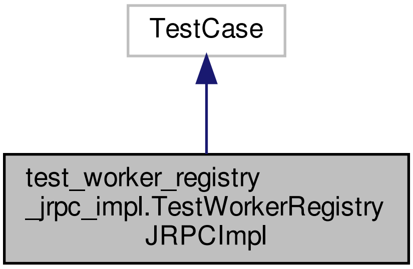
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/unit_tests/test_fabric_worker_registry↩ _list_impl.py

12.53 test_worker_registry_jrpc_impl.TestWorkerRegistryJRPCImpl Class Reference

Inheritance diagram for test_worker_registry_jrpc_impl.TestWorkerRegistryJRPCImpl:



Collaboration diagram for test_worker_registry_jrpc_impl.TestWorkerRegistryJRPCImpl:



Public Member Functions

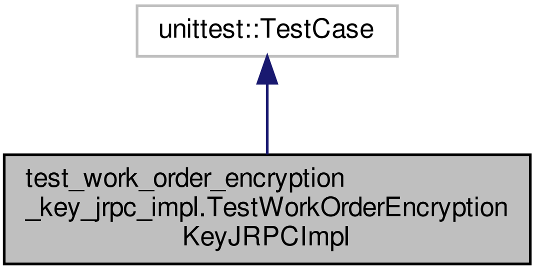
- def **__init__** (self, config_file)
- def **test_worker_register** (self)
- def **test_worker_update** (self)
- def **test_worker_set_status** (self)
- def **test_worker_retrieve** (self)
- def **test_worker_lookup** (self)
- def **test_worker_lookup_next** (self)

The documentation for this class was generated from the following file:

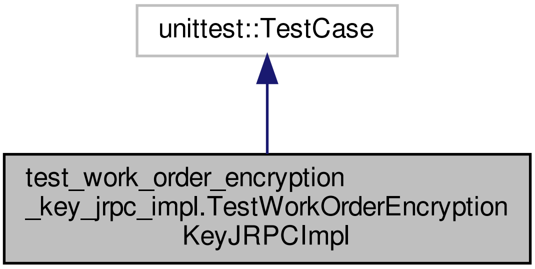
- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/unit_tests/test_worker_registry_jrpc_impl.py

12.54 test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl Class Reference

Inheritance diagram for test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl:


unittest::TestCase
test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl

Collaboration diagram for test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl:


unittest::TestCase
test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl

Public Member Functions

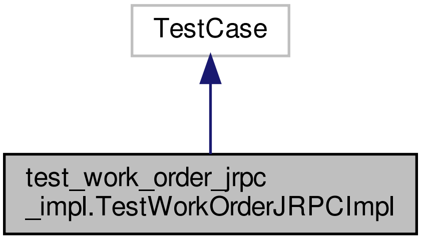
- def **__init__** (self, config_file)
- def **test_encryption_key_get** (self)
- def **test_encryption_key_set** (self)

The documentation for this class was generated from the following file:

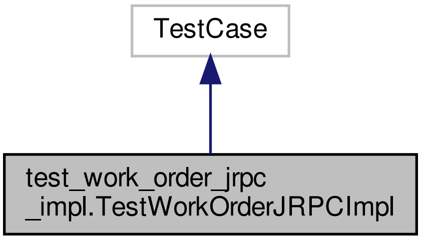
- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/unit_tests/test_work_order_encryption_key_jrpc_impl.py

12.55 test_work_order_jrpc_impl.TestWorkOrderJRPCImpl Class Reference

Inheritance diagram for test_work_order_jrpc_impl.TestWorkOrderJRPCImpl:



Collaboration diagram for test_work_order_jrpc_impl.TestWorkOrderJRPCImpl:



Public Member Functions

- def **__init__** (self, config_file)
- def **test_work_order_submit** (self)
- def **test_work_order_get_result** (self)

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/unit_tests/test_work_order_jrpc_impl.py

12.56 tcf::utility::Timer Class Reference

Public Member Functions

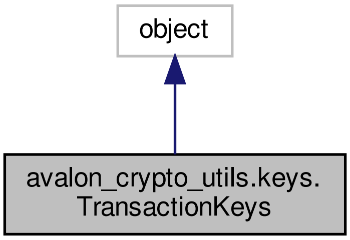
- **Timer** (const std::string &key)

The documentation for this class was generated from the following file:

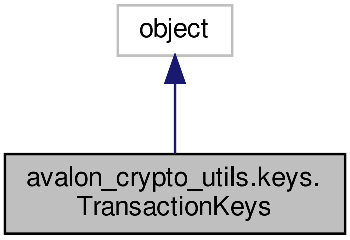
- /home/dano/git/avalon/common/cpp/timer.h

12.57 avalon_crypto_utils.keys.TransactionKeys Class Reference

Inheritance diagram for avalon_crypto_utils.keys.TransactionKeys:



Collaboration diagram for avalon_crypto_utils.keys.TransactionKeys:



Public Member Functions

- def **__init__** (self)
- def **hashed_identity** (self)
- def **txn_private** (self)
- def **txn_public** (self)

Public Attributes

- **private_key**
- **public_key**

12.57.1 Detailed Description

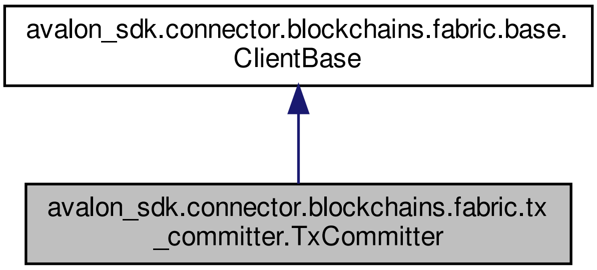
```
Wrapper for managing Avalon transaction keys
```

The documentation for this class was generated from the following file:

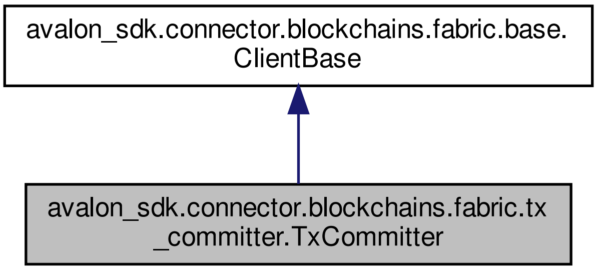
- /home/dano/git/avalon/common/crypto_utils/avalon_crypto_utils/keys.py

12.58 avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter Class Reference

Inheritance diagram for avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter:


avalon_sdk.connector.blockchains.fabric.base.
ClientBase
avalon_sdk.connector.blockchains.fabric.tx
_committer.TxCommitter

Collaboration diagram for avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter:


avalon_sdk.connector.blockchains.fabric.base.
ClientBase
avalon_sdk.connector.blockchains.fabric.tx
_committer.TxCommitter

Public Member Functions

- def cc_invoke (self, args, cc_name, fcn, cc_version, queryonly=False)
- def cc_query (self, args, cc_name, fcn)

Additional Inherited Members

12.58.1 Detailed Description

```
Utility class to invoke Fabric chain code and query
chain code.
```

12.58.2 Member Function Documentation

12.58.2.1 cc_invoke()

```
def avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter.cc_invoke (
              self,
              args,
              cc_name,
              fcn,
              cc_version,
              queryonly = False )
```

```
Invoke a chaincode method.

Parameters:
args       JSON RPC serialized data used as the
   sole parameter to invoke the chaincode
cc_name    chaincode name
fcn        chaincode function name to be invoked
cc_version chaincode version to be used
queryonly  If the invocation does not result in ledger change,
   queryonly should be set to True.
   If the invocation does result in ledger change, it should
   be set to False.
```

12.58.2.2 cc_query()

```
def avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter.cc_query (
              self,
              args,
              cc_name,
              fcn )
```

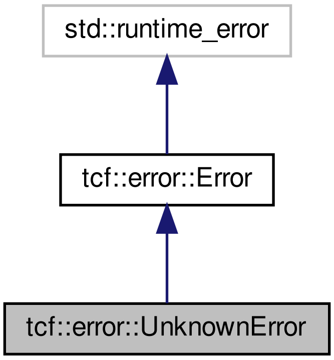
```
Invoke a chaincode query method. If there is no query method from the
chaincode, then this will fail.

Parameters:
args     Array of the strings used as the parameters to query method
cc_name  Chaincode name
fcn      Chaincode function name
```

The documentation for this class was generated from the following file:

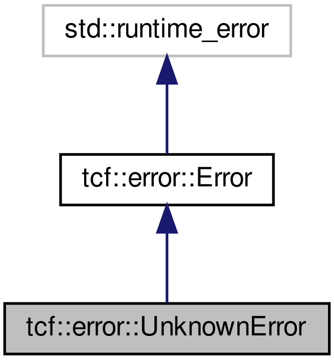
- /home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/fabric/tx_committer.py

12.59 tcf::error::UnknownError Class Reference

Inheritance diagram for tcf::error::UnknownError:



Collaboration diagram for tcf::error::UnknownError:



Public Member Functions

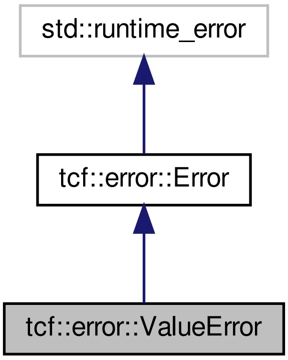
- **UnknownError** (const std::string &msg)

The documentation for this class was generated from the following file:

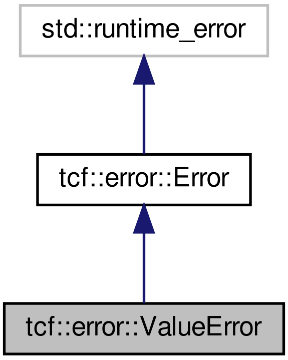
- /home/dano/git/avalon/common/cpp/error.h

12.60 tcf::error::ValueError Class Reference

Inheritance diagram for tcf::error::ValueError:



Collaboration diagram for tcf::error::ValueError:



Public Member Functions

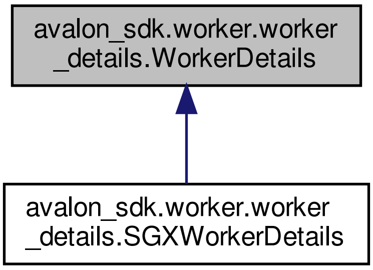
- **ValueError** (const std::string &msg)

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/common/cpp/error.h

12.61 avalon_sdk.worker.worker_details.WorkerDetails Class Reference

Inheritance diagram for avalon_sdk.worker.worker_details.WorkerDetails:



Public Member Functions

- def __init__ (self)
- def validate_worker_details (self, details)

Public Attributes

- **work_order_sync_uri**
- **work_order_async_uri**
- **work_order_pull_uri**
- **work_order_notify_ri**
- **receipt_invocation_uri**
- **work_oder_invocation_address**
- **receipt_invocation_address**
- **from_address**
- **hashing_algorithm**
- **signing_algorithm**
- **key_encryption_algorithm**
- **data_encryption_algorithm**
- **work_order_payload_formats**

12.61.1 Detailed Description

```
Class to store the worker details
```

12.61.2 Constructor & Destructor Documentation

12.61.2.1 __init__()

```
def avalon_sdk.worker.worker_details.WorkerDetails.__init__ (
              self )
```

```
Set the member variables of this class with default
values as per the EEA Spec.
```

12.61.3 Member Function Documentation

12.61.3.1 validate_worker_details()

```
def avalon_sdk.worker.worker_details.WorkerDetails.validate_worker_details (
              self,
              details )
```

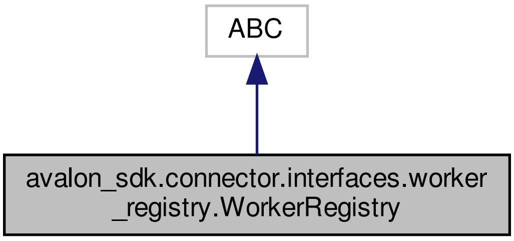
```
Validate the details field of a worker.

Parameters:
    Details is json formatted string
Returns:
    None on success and error string on failure
```

The documentation for this class was generated from the following file:

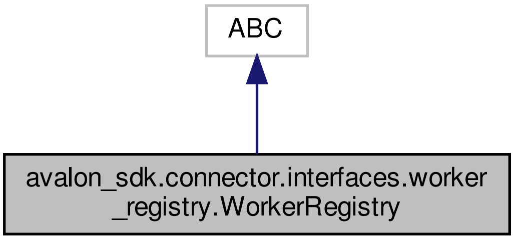
- /home/dano/git/avalon/sdk/avalon_sdk/worker/worker_details.py

12.62 avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry Class Reference

Inheritance diagram for avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry:



Collaboration diagram for avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry:


ABC
avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry

Public Member Functions

- def **__init__** (self)
- def worker_retrieve (self, worker_id, id=None)
- def worker_lookup (self, worker_type, organization_id, application_type_id, id=None)
- def worker_lookup_next (self, worker_type, organization_id, application_type_id, lookup_tag, id=None)
- def worker_register (self, worker_id, worker_type, organization_id, application_type_ids, details, id=None)
- def worker_update (self, worker_id, details, id=None)
- def worker_set_status (self, worker_id, status, id=None)

12.62.1 Detailed Description

```
This class is an abstract base class containing abstract APIs
which can be called from client to manage workers.
```

12.62.2 Member Function Documentation

12.62.2.1 worker_lookup()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_lookup (
              self,
              worker_type,
              organization_id,
              application_type_id,
              id = None )
```

```
Lookup a worker identified worker_type, organization, and
application_id.
All fields are optional and, if present, condition should match for
all fields. If none are passed it should return all workers.

If the list is too large to fit into a single response (the maximum
number of entries in a single response is implementation specific),
the smart contract should return the first batch of the results
and provide a lookupTag that can be used by the caller to
retrieve the next batch by calling worker_lookup_next.

Parameters:
worker_type     Optional characteristic of workers for which you may
        wish to search
organization_id Optional organization ID that can be used to search
        for one or more workers that belong to this
        organization
application_id  Optional application type ID that is supported by
        the worker
id              Optional JSON RPC request ID

Returns:
Tuple containing workers count, lookup tag, and list of
worker IDs:
total_count Total number of entries matching a specified
    lookup criteria. If this number is larger than the
    size of the IDs array, the caller should use
    lookupTag to call worker_lookup_next to retrieve
    the rest of the IDs
lookup_tag  Optional parameter. If it is returned, it means
    that there are more matching worker IDs, which can then
    be retrieved by calling function worker_lookup_next
    with this tag as an input parameter
ids         Array of the worker IDs that match the input parameters

On error returns None.
```

12.62.2.2 worker_lookup_next()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_lookup_next (
              self,
              worker_type,
              organization_id,
              application_type_id,
              lookup_tag,
              id = None )
```

```
Retrieve additional worker lookup results after calling worker_lookup.

Parameters:
worker_type         Characteristic of Workers for which you may wish
            to search.
organization_id     Organization ID to which a Worker belongs
application_id      Optional application type ID that is
            supported by the worker
lookup_tag          is returned by a previous call to either this
            function or to worker_lookup
id                  Optional Optional JSON RPC request ID


Returns:
Tuple containing the following:
total_count    Total number of entries matching this lookup
       criteria.  If this number is larger than the number
       of IDs returned so far, the caller should use
       lookupTag to call worker_lookup_next to retrieve
       the rest of the IDs
new_lookup_tag Optional parameter. If it is returned, it
       means that there are more matching worker IDs that
       can be retrieved by calling this function again with
       this tag as an input parameter
ids            Array of the worker IDs that match the input parameters

On error returns None.
```

12.62.2.3 worker_register()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_register (
              self,
              worker_id,
              worker_type,
              organization_id,
              application_type_ids,
              details,
              id = None )
```

```
Register a new worker with details of the worker.

Parameters:
worker_id       Worker ID value. E.g., a Fabric address
        or Ethereum DID
worker_type     Type of Worker. Currently defined types are:
        * "TEE-SGX": an Intel SGX Trusted Execution
          Environment
        * "MPC": Multi-Party Compute
        * "ZK": Zero-Knowledge
organization_id Optional parameter representing the
        organization that hosts the Worker,
        e.g. a bank in the consortium or
        anonymous entity
application_ids Optional parameter that defines
        application types supported by the Worker
details         Detailed information about the worker in
        JSON RPC format as defined in
https://entethalliance.github.io/trusted-computing/spec.html
#common-data-for-all-worker-types
id              Optional Optional JSON RPC request ID

Returns:
ContractResponse.SUCCESS on success or
ContractResponse.ERROR on error.
```

12.62.2.4 worker_retrieve()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_retrieve (
              self,
              worker_id,
              id = None )
```

```
Retrieve worker identified by worker ID.

Parameters:
worker_id  Worker ID of the registry whose details are requested
id         Optional JSON RPC request ID

Returns:
Tuple containing worker status (defined in worker_set_status),
worker type, organization ID, list of application IDs, and worker
details (JSON RPC string).

On error returns None.
```

12.62.2.5 worker_set_status()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_set_status (
              self,
              worker_id,
              status,
              id = None )
```

```
Set the registry status identified by worker ID

Parameters:
worker_id Worker ID value. E.g., a Fabric address
  or Ethereum DID
status    Worker status. The currently defined values are:
  1 - worker is active
  2 - worker is temporarily "off-line"
  3 - worker is decommissioned
  4 - worker is compromised
id        Optional Optional JSON RPC request ID

Returns:
ContractResponse.SUCCESS on success
or ContractResponse.ERROR on error.
```

12.62.2.6 worker_update()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_update (
              self,
              worker_id,
              details,
              id = None )
```

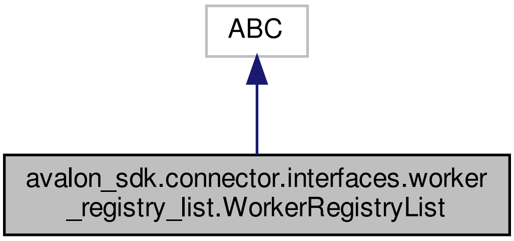
```
Update a worker with details data.

Parameters:
worker_id  Worker ID
details    Detailed information about the worker in JSON format
id         Optional Optional JSON RPC request ID

Returns:
ContractResponse.SUCCESS on success
or ContractResponse.ERROR on error.
```

The documentation for this class was generated from the following file:

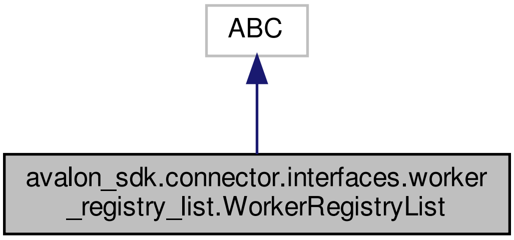
- /home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/worker_registry.py

12.63 avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList Class Reference

Inheritance diagram for avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList:


ABC
avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList

Collaboration diagram for avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList:


ABC
avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList

Public Member Functions

- def **__init__** (self)
- def registry_retrieve (self, organization_id)
- def registry_lookup (self, application_type_id)
- def registry_lookup_next (self, application_type_id, lookup_tag)
- def registry_add (self, organization_id, uri, sc_addr, application_type_ids)
- def registry_update (self, organization_id, uri, sc_addr, application_type_ids)
- def registry_set_status (self, organization_id, status)

12.63.1 Detailed Description

```
This is an abstract base class to read/write the worker registries,
which can be called by client.
```

12.63.2 Member Function Documentation

12.63.2.1 registry_add()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_add (
              self,
              organization_id,
              uri,
              sc_addr,
              application_type_ids )
```

```
Add a new registry.

Parameters:
organization_id      bytes[] identifies organization that hosts the
             registry, e.g. a bank in the consortium or an
             anonymous entity
uri                  String defines a URI for this registry that
             supports the Off-Chain Worker Registry
             JSON RPC API.
sc_addr              bytes[] defines an smart contract address that
             runs the Worker Registry Smart Contract API
             smart contract for this registry
application_type_ids []bytes[] is an optional parameter that defines
             application types supported by the worker
             managed by the registry

Returns:
Transaction receipt on success or None on error.
```

12.63.2.2 registry_lookup()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_lookup (
              self,
              application_type_id )
```

```
Registry Lookup identified by application type ID

Parameters:
application_type_id  Application type ID to lookup in the registry

Returns:
Tuple containing totalCount, lookupTag, and ids on success:
totalCount Total number of entries matching a specified lookup
   criteria. If this number is larger than the size of the
   ids array, the caller should use the lookupTag to call
   registry_lookup_next to retrieve the rest of the IDs
lookupTag  Optional parameter. If it is returned, it means that
   there are more matching registry IDs that can be
   retrieved by calling the function registry_lookup_next
   with this tag as an input parameter.
ids        Array of the registry organization ids that match the
   input parameters.

Returns None on error.
```

12.63.2.3 registry_lookup_next()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_lookup_↩
next (
              self,
              application_type_id,
              lookup_tag )
```

```
This function is called to retrieve additional results of the
Registry lookup initiated by the registry_lookUp call.

Parameters:
application_type_id    Application type that has to be
               supported by the workers retrieved
lookup_tag             Returned by a previous call to either this
               function or to registry_lookup

Returns:
Outputs tuple on success containing the following:
total_count    Total number of entries matching the lookup
       criteria. If this number is larger than the number
       of IDs returned so far, the caller should use
       lookup_tag to call registry_lookup_next to
       retrieve the rest of the IDs
new_lookup_tag Optional parameter. If it is returned, it means
       that there are more matching registry IDs that
       can be retrieved by calling this function again
       with this tag as an input parameter
ids            Array of the registry IDs that match the input
       parameters

Returns None on error.
```

12.63.2.4 registry_retrieve()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_retrieve
(
              self,
              organization_id )
```

```
Retrieve registry information identified by the organization ID.

Parameters:
organization_id      Organization ID to lookup

Returns:
Tuple containing following on success:
uri                  String defines a URI for this registry that
             supports the Off-Chain Worker Registry JSON RPC
             API. It will be None for the proxy model
sc_addr              smart contract address for worker registry
             smart contract address
application_type_ids List of application ids (array of byte[])
status               Status of the registry

Returns None on error.
```

12.63.2.5 registry_set_status()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_set_↩
status (
              self,
              organization_id,
              status )
```

```
Set registry status.

Parameters:
organization_id bytes[] identifies organization that hosts the
        registry, e.g. a bank in the consortium or an
        anonymous entity
status          Defines the registry status to set.
        The currently defined values are:
        1 - the registry is active
        2 - the registry is temporarily "off-line"
        3 - the registry is decommissioned

Returns:
Transaction receipt on success or None on error.
```

12.63.2.6 registry_update()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_update (
              self,
              organization_id,
              uri,
              sc_addr,
              application_type_ids )
```

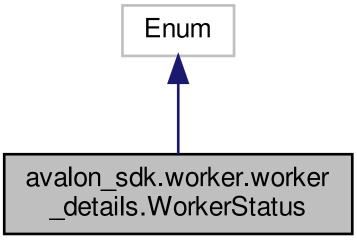
```
Update a registry.

Parameters:
organization_id      bytes[] identifies organization that hosts the
             registry, e.g. a bank in the consortium or an
             anonymous entity
uri                  string defines a URI for this registry that
             supports the Off-Chain Worker Registry
             JSON RPC API
sc_addr              bytes[] defines an smart contract address that
             runs a Worker Registry Smart Contract API
             smart contract for this registry
application_type_ids []bytes[] is an optional parameter that defines
             application types supported by the worker
             managed by the registry

Returns:
Transaction receipt on success or None on error.
```

The documentation for this class was generated from the following file:

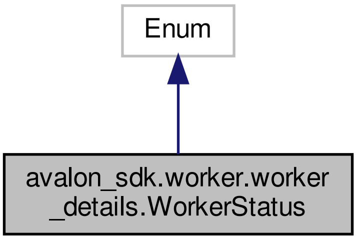
- /home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/worker_registry_list.py

12.64 avalon_sdk.worker.worker_details.WorkerStatus Class Reference

Inheritance diagram for avalon_sdk.worker.worker_details.WorkerStatus:



Collaboration diagram for avalon_sdk.worker.worker_details.WorkerStatus:



Static Public Attributes

- int **ACTIVE** = 1
- int **OFF_LINE** = 2
- int **DECOMMISSIONED** = 3
- int **COMPROMISED** = 4

12.64.1 Detailed Description

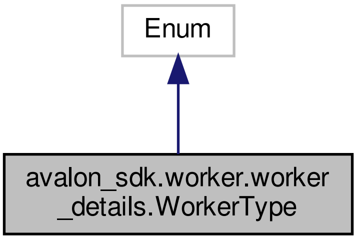
```
Worker status values:
1 - worker is ACTIVE
2 - worker is temporarily OFF_LINE
3 - worker is DECOMMISSIONED
4 - worker is COMPROMISED

From EEA spec 5.2.
```

The documentation for this class was generated from the following file:

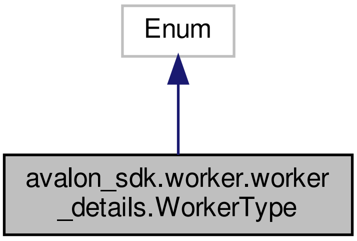
- /home/dano/git/avalon/sdk/avalon_sdk/worker/worker_details.py

12.65 avalon_sdk.worker.worker_details.WorkerType Class Reference

Inheritance diagram for avalon_sdk.worker.worker_details.WorkerType:



Collaboration diagram for avalon_sdk.worker.worker_details.WorkerType:



Static Public Attributes

- int **TEE_SGX** = 1
- int **MPC** = 2
- int **ZK** = 3

12.65.1 Detailed Description

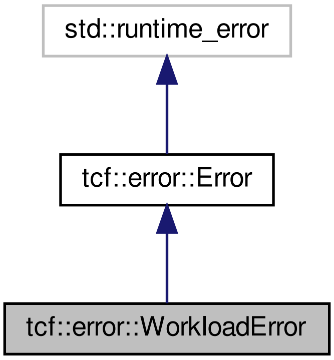
```
Worker types are:
1 = TEE-SGX: Intel SGX Trusted Execution Environment (hardware based)
2 = MPC: Trusted Multi-Party Compute (software/hardware based)
3 = ZK: Zero-knowledge proofs (software based)
```

The documentation for this class was generated from the following file:

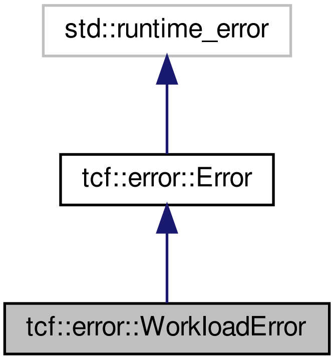
- /home/dano/git/avalon/sdk/avalon_sdk/worker/worker_details.py

12.66 tcf::error::WorkloadError Class Reference

Inheritance diagram for tcf::error::WorkloadError:



Collaboration diagram for tcf::error::WorkloadError:



Public Member Functions

- **WorkloadError** (const std::string &msg)

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/common/cpp/error.h

12.67 WorkloadProcessor Class Reference

```
#include <workload_processor.h>
```

Public Member Functions

- virtual WorkloadProcessor ∗ Clone () const =0
- virtual void ProcessWorkOrder (std::string workload_id, const ByteArray &requester_id, const ByteArray &worker_id, const ByteArray &work_order_id, const std::vector< tcf::WorkOrderData > &in_work_order←_data, std::vector< tcf::WorkOrderData > &out_work_order_data)=0

Static Public Member Functions

- static WorkloadProcessor ∗ CreateWorkloadProcessor (std::string workload_id)
- static WorkloadProcessor ∗ RegisterWorkloadProcessor (std::string workload_id, WorkloadProcessor ∗processor)

Static Public Attributes

- static std::map< std::string, WorkloadProcessor ∗ > workload_processor_table

12.67.1 Detailed Description

Class to register, create, and process a workload.

12.67.2 Member Function Documentation

12.67.2.1 Clone()

```
virtual WorkloadProcessor* WorkloadProcessor::Clone ( ) const  [pure virtual]
```

Clone a WorkloadProcessor

12.67.2.2 CreateWorkloadProcessor()

```
WorkloadProcessor * WorkloadProcessor::CreateWorkloadProcessor (
            std::string workload_id )  [static]
```

Create a WorkloadProcessor

Parameters

workload↩ _id	Workload identifier

Returns

Pointer to WorkloadProcessor

12.67.2.3 ProcessWorkOrder()

```
virtual void WorkloadProcessor::ProcessWorkOrder (
             std::string workload_id,
             const ByteArray & requester_id,
             const ByteArray & worker_id,
             const ByteArray & work_order_id,
             const std::vector< tcf::WorkOrderData > & in_work_order_data,
             std::vector< tcf::WorkOrderData > & out_work_order_data )  [pure virtual]
```

Process the workload.

Parameters

workload_id	Workload identifier string
requester_id	Requester ID to identify who submitted work order
worker_id	Worker ID, a unique string identifying this type of work order processor
work_order_id	Unique work order ID for this type of work order processor
in_work_order_data	Work order data input submitted to the work order processor
out_work_order_data	Work order data returned by the work order processor

12.67.2.4 RegisterWorkloadProcessor()

```
WorkloadProcessor * WorkloadProcessor::RegisterWorkloadProcessor (
             std::string workload_id,
             WorkloadProcessor * processor )  [static]
```

Register a WorkloadProcessor. Used by the workloads to register themselves

Parameters

workload↩ _id	Workload identifier

Returns

Pointer to WorkloadProcessor

12.67.3 Member Data Documentation

12.67.3.1 workload_processor_table

```
std::map< std::string, WorkloadProcessor * > WorkloadProcessor::workload_processor_table  [static]
```

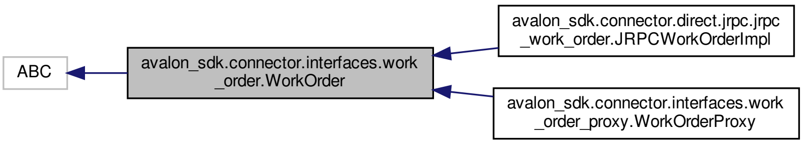
Mapping between workload id and WorkloadProcessor.

The documentation for this class was generated from the following files:

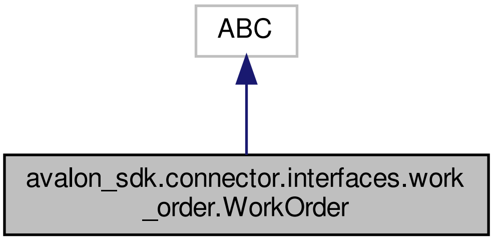
- /home/dano/git/avalon/common/sgx_workload/workload/workload_processor.h
- /home/dano/git/avalon/common/sgx_workload/workload/workload_processor.cpp

12.68 avalon_sdk.connector.interfaces.work_order.WorkOrder Class Reference

Inheritance diagram for avalon_sdk.connector.interfaces.work_order.WorkOrder:


ABC
avalon_sdk.connector.interfaces.work_order.WorkOrder
avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl
avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy

Collaboration diagram for avalon_sdk.connector.interfaces.work_order.WorkOrder:



Public Member Functions

- def **__init__** (self)
- def work_order_submit (self, work_order_id, worker_id, requester_id, work_order_request, id=None)
- def work_order_get_result (self, work_order_id, id=None)
- def encryption_key_get (self, worker_id, requester_id, last_used_key_nonce=None, tag=None, signature_↩ nonce=None, signature=None, id=None)
- def encryption_key_set (self, worker_id, encryption_key, encryption_nonce, tag, signature, id=None)

12.68.1 Detailed Description

```
This class is an abstract base class that contains
abstract APIs to manage work orders.
```

12.68.2 Member Function Documentation

12.68.2.1 encryption_key_get()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.encryption_key_get (
              self,
              worker_id,
              requester_id,
              last_used_key_nonce = None,
              tag = None,
              signature_nonce = None,
              signature = None,
              id = None )
```

```
Get Encryption Key Request Payload.

Inputs
1. worker_id is an id of the worker to retrieve an encryption key for.
2. last_used_key_nonce is an optional nonce associated with the
last key retrieved.
If it is provided, the key retrieved should be newer than this one.
Otherwise any key can be retrieved.
3. tag is tag that should be associated with the returned key,
e.g. requester id. This is an optional parameter. If it is not
provided, requesterId below is used as a key.
4. requester_id is an id of the requester that plans to use the
returned key to submit one or more work orders using this key.
5. signature_nonce is an optional parameter and is used only
if signature below is also provided.
6. signature is an optional signature of workerId,
lastUsedKeyNonce, tag, and signatureNonce.
7. id is json rpc request id, it is optional
Output
1. errorCode is the result of the operation.
   1 - generic error
   2 - operation is not supported
   3 - invalid parameter
   4 - access denied
   5 - not ready, retry later. This is a recoverable error that may
   happen if the requester makes its first request for keys,
   or the requester retrieves keys, faster than the worker
   produces them.
   The requester should retry later.
2. worker_id is an id of the worker that created the encryption key.
3. encryptionKey is an encryption key.
4. encryptionKeyNonce is a nonce associated with the key.
5. tag is tag associated with the key.
6. signature is a signature generated by the worker.
```

12.68.2.2 encryption_key_set()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.encryption_key_set (
              self,
              worker_id,
              encryption_key,
              encryption_nonce,
              tag,
              signature,
              id = None )
```

```
Set encryption key request payload

Parameters:
worker_id        ID of the worker to retrieve an encryption key
encryption_key   Encryption key
encryption_nonce Nonce associated with the key
tag              Tag that should be associated with the returned key,
         e.g. requester id. This is an optional parameter.
         If it is not provided, requesterId below is used
         as a key
signature        Signature generated by the worker on the worker_id,
         tag, and encryption_nonce
id               Optional Optional JSON RPC request ID

Returns:
Error code is the result of the operation.
```

12.68.2.3 work_order_get_result()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.work_order_get_result (
              self,
              work_order_id,
              id = None )
```

```
Query blockchain to get a work order result.

If a Requester receives a response stating that its work order state
is "scheduled" or "processing", it should poll the Worker Service
later to get the result:
1. Poll the Worker Service periodically until the Work Order is
completed successfully or in error
2. Wait for the Work Order Receipt complete event and retrieve a final
result.

Parameters:
work_order_id Work Order ID that was sent in the
      corresponding work_order_submit request
id            Optional JSON RPC request ID

Returns:
Tuple containing work order status, worker id, work order request,
work order response, and error code.
None on error.
```

12.68.2.4 work_order_submit()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.work_order_submit (
              self,
              work_order_id,
              worker_id,
              requester_id,
              work_order_request,
              id = None )
```

```
Submit a work order request.

Parameters:
work_order_id      Unique ID of the work order request
worker_id          Identifier for the worker
requester_id       Unique id to identify the requester
work_order_request JSON RPC string work order request.
           Defined in EEA specification 6.1.1.
id                 Optional JSON RPC request ID

Returns:
errorCode          0 on success and non-zero on error.

work_order_request is a JSON string containing following parameters:
{
    "responseTimeoutMSecs": <integer>,
    "payloadFormat": <string>
    "resultUri": <string>,
    "notifyUri": <string>,
    "workOrderId": <hex string>,
    "workerId": <hex string or DID>,
    "workloadId": <hex string>,
    "requesterId": <hex string>,
    "workerEncryptionKey": <hex string>,
    "dataEncryptionAlgorithm": <string>,
    "encryptedSessionKey": <hex string>,
    "sessionKeyIv": <hex string>,
    "requesterNonce": <hex string>,
    "encryptedRequestHash": <hex string>,
    "requesterSignature": <BASE64 string>,
},

1. responseTimeoutMSecs - is a maximum timeout in milliseconds that
the caller will wait for the response. Setting this timeout to zero
means that the work order is submitted in the asynchronous (resultUri
is present), notify (notifyUri is present), or poll mode (neither
resultUri nor notifyUri is present). In this case, the TCS should
schedule the request for execution and immediately return an error
response with error code set to "scheduled". If the timeout is not
zero, the work order is in synchronous mode. The TCS should wait for
the work order completion before returning the response to the
participant. If the request cannot be completed within the allocated
interval, the work order should be cancelled and a corresponding error
should be returned to the participant.
2. payloadFormat defines how signatures and data items are formatted in
this work order request and corresponding response.
3. resultUri is an optional parameter. If it is specified, the
WorkerService should submit the Work Order result to this URI.
See section Work Order Asynchronous Result.
4. notifyUri is an optional parameter. If it is specified, the
WorkerService should send an event to this URI upon the Work Order
completion.
5. workOrderId is an id assigned to the Work Order by the Requester
and can be registered using the Work Order Receipts API.
6. workerId is a worker id to process the work order,
e.g. an Ethereum address or its DID.
7. workloadId is an id of the workload to be executed by the worker.
It is an optional value if the worker includes a single workload.
8. requesterId is either the Requester’s Ethereum address or its DID.
9. workerEncryptionKey is an optional parameter containing the worker
```

```
encryption key used for this Work Order. It is useful if a Worker
frequently updates its encryption key in the registry and allows some
time overlap in utilizing multiple keys.
We assume here that the ’details’ submitted during the registration of
a worker contain one or more public keys associated with the worker.
10. dataEncryptionAlgorithm is an optional parameter that defines an
algorithm for encrypting the data in this work order. The default is
the first value in the corresponding parameter for the worker
(defined by workerId). See section Common Data for All Worker Types.
11. encryptedSessionKey is a one-time encryption key generated by the
participant submitting the work order. It is sent encrypted with the
worker’s public encryption key. It is used to encrypt
encryptedRequestHash and data item specific data encryption keys.
For the latter see Work Order Data Formats.
12. sessionKeyIv is an initialization vector if required by the data
encryption algorithm (encryptedSessionKey). The default is all zeros.
13. requesterNonce is a random string generated by the participant.
It is used to calculate a hash of this work order request.
14. encryptedRequestHash is a hash of the work order request encrypted
with the key provided in encryptedSessionKey.
15. requesterSignature is an optional parameter. See section
Work Order Signing for the details.
16. inData contains either a JWT of the specified data or an array of
one or more Work Order inputs, e.g. state, message containing input
parameters.
{
    "index": <number>,
    "dataHash": <hex string>,
    "data": <BASE64 string>,
    "encryptedDataEncryptionKey": <hex string>,
    "iv": <hex string>
}
i. index is an index that determines order of the data items for the
hash generation. It also can be used by the worker to identify
different inputs and outputs.
ii. dataHash is an optional hash value of the data. It is only
applicable to inData in the work order request and outData in the
response.
iii. data contains either data inline within the JSON document or a
reference (e.g. URI) to the data. It is up to the worker to determine
how to interpret the data content. This parameter is applicable to
    -> inData in the work order request
    -> outData in the request if it contains a reference for the output
    -> outData in the response
iv. encryptedDataEncryptionKey defines if data are encrypted and what
key to use. It is included only in the work order request as one of the
options below.
If this key is not provided or set to "null" or to "", the data is
encrypted using encryptedSessionKey from the work order request.
If the key value is set to "-", the data item is not encrypted, a.k.a.
sent as clear text.
Otherwise, the data item is sent encrypted with a one-time encryption
key generated by a 3rd party that owns this data item (it may be
different from the work order requester).
encryptedDataEncryptionKey contains this encryption key in double
encrypted format.
First, it is encrypted with the worker’s public encryption key (e.g. by
a 3rd party that owns the data so the requester cannot see the data).
Then the result of the previous encryption above is encrypted with the
key from encryptedSessionKey (by the requester to enforce the work
order integrity).
v. iv is an initialization vector if required by the data encryption
algorithm. The default is all zeros. If the same encryption key is
used to encrypt more than one data item or the hash value of the
work order request, the iv must be a unique random number for every
encryption operation. It is included only in the work order request.
17. outData contains information about what and how the work order
execution results should be delivered. Same as inData
18. id is used for json rpc request
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/work_order.py

12.69 tcf::WorkOrderData Class Reference

```
#include <work_order_data.h>
```

Public Member Functions

- **WorkOrderData** (int in_index, ByteArray data)

Public Attributes

- int **index**
- ByteArray **decrypted_data** = {}

12.69.1 Detailed Description

Wrapper class for work order data submitted to workload processors.

The documentation for this class was generated from the following files:

- /home/dano/git/avalon/common/sgx_workload/workload/work_order_data.h
- /home/dano/git/avalon/common/sgx_workload/workload/work_order_data.cpp

12.70 avalon_sdk.work_order.work_order_params.WorkOrderParams Class Reference

Public Member Functions

- def **__init__** (self, work_order_id, worker_id, workload_id, requester_id, session_key, session_↩ iv, requester_nonce, verifying_key=None, payload_format="JSON-RPC", response_timeout_msecs=6000, result_uri=None, notify_uri=None, worker_encryption_key=None, data_encryption_algorithm=None)
- def set_response_timeout_msecs (self, response_timeout_msecs)
- def set_payload_format (self, payload_format)
- def set_result_uri (self, result_uri)
- def set_notify_uri (self, notify_uri)
- def set_worker_id (self, worker_id)
- def set_work_order_id (self, work_order_id)
- def set_workload_id (self, workload_id)
- def set_requester_id (self, requester_id)
- def set_worker_encryption_key (self, worker_encryption_key)
- def set_data_encryption_algorithm (self, data_encryption_algorithm)
- def set_encrypted_session_key (self, encrypted_session_key)
- def set_session_key_iv (self, session_iv)
- def set_requester_nonce (self, requester_nonce)
- def add_encrypted_request_hash (self)
- def add_requester_signature (self, private_key)
- def set_verifying_key (self, verifying_key)
- def add_in_data (self, data, data_hash=None, encrypted_data_encryption_key=None, data_iv=None)
- def add_out_data (self, data, data_hash=None, encrypted_data_encryption_key=None, data_iv=None)
- def get_params (self)

- def get_in_data (self)
- def get_out_data (self)
- def get_requester_nonce (self)
- def get_worker_id (self)
- def get_workload_id (self)
- def get_requester_id (self)
- def get_session_key_iv (self)
- def get_work_order_id (self)
- def to_jrpc_string (self, id)
- def to_string (self)

Public Attributes

- **params_obj**
- **session_iv**
- **session_key**
- **final_hash**

12.70.1 Member Function Documentation

12.70.1.1 add_encrypted_request_hash()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_encrypted_request_hash (
              self )
```

```
Calculates request hash based on EEA trusted-computing spec 6.1.8.1
and set encryptedRequestHash parameter in the request.
```

12.70.1.2 add_in_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_in_data (
              self,
              data,
              data_hash = None,
              encrypted_data_encryption_key = None,
              data_iv = None )
```

```
Add inData work order parameter.
```

12.70.1.3 add_out_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_out_data (
              self,
              data,
              data_hash = None,
              encrypted_data_encryption_key = None,
              data_iv = None )
```

```
Add outData work order parameter.
```

12.70.1.4 add_requester_signature()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_requester_signature (
              self,
              private_key )
```

```
Calculate the signature of the request
as defined in Off-Chain Trusted Compute EEA spec 6.1.8.3
and set the requesterSignature parameter in the request.
```

12.70.1.5 get_in_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_in_data (
              self )
```

```
Return inData work order parameter.
```

12.70.1.6 get_out_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_out_data (
              self )
```

```
Return outData work order parameter.
```

12.70.1.7 get_params()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_params (
              self )
```

```
Return a copy of work order parameters.
```

12.70.1.8 get_requester_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_requester_id (
              self )
```

```
Return requesterId work order parameter.
```

12.70.1.9 get_requester_nonce()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_requester_nonce (
              self )
```

```
Return requesterNonce work order parameter.
```

12.70.1.10 get_session_key_iv()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_session_key_iv (
              self )
```

```
Return sessionKeyIv work order parameter.
```

12.70.1.11 get_work_order_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_work_order_id (
              self )
```

```
Return workOrderId work order parameter.
```

12.70.1.12 get_worker_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_worker_id (
              self )
```

Return workerId work order parameter.

12.70.1.13 get_workload_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_workload_id (
              self )
```

Return workloadId work order parameter.

12.70.1.14 set_data_encryption_algorithm()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_data_encryption_algorithm (
              self,
              data_encryption_algorithm )
```

Set dataEncryptionAlgorithm work order parameter.

12.70.1.15 set_encrypted_session_key()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_encrypted_session_key (
              self,
              encrypted_session_key )
```

Set encryptedSessionKey work order parameter.

12.70.1.16 set_notify_uri()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_notify_uri (
              self,
              notify_uri )
```

Set notifyUri work order parameter.

12.70.1.17 set_payload_format()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_payload_format (
              self,
              payload_format )
```

Set payloadFormat work order parameter.

12.70.1.18 set_requester_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_requester_id (
              self,
              requester_id )
```

Set requesterId work order parameter.

12.70.1.19 set_requester_nonce()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_requester_nonce (
              self,
              requester_nonce )
```

Set requesterNonce work order parameter.

12.70.1.20 set_response_timeout_msecs()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_response_timeout_msecs (
              self,
              response_timeout_msecs )
```

Set responseTimeoutMSecs work order parameter.

12.70.1.21 set_result_uri()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_result_uri (
              self,
              result_uri )
```

Set resultUri work order parameter.

12.70.1.22 set_session_key_iv()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_session_key_iv (
              self,
              session_iv )
```

Set sessionKeyIv work order parameter.

12.70.1.23 set_verifying_key()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_verifying_key (
              self,
              verifying_key )
```

Set verifyingKey work order parameter.

12.70.1.24 set_work_order_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_work_order_id (
              self,
              work_order_id )
```

Set workOrderId work order parameter.

12.70.1.25 set_worker_encryption_key()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_worker_encryption_key (
              self,
              worker_encryption_key )
```

Set workerEncryptionKey work order parameter.

12.70.1.26 set_worker_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_worker_id (
              self,
              worker_id )
```

Set workerId work order parameter.

12.70.1.27 set_workload_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_workload_id (
              self,
              workload_id )
```

```
Set workloadId work order parameter.
```

12.70.1.28 to_jrpc_string()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.to_jrpc_string (
              self,
              id )
```

```
Create a JRPC request in string format using
the work order params_obj.

Parameters:
id         JRPC request ID

Returns:
Work order JRPC request as a string.
```

12.70.1.29 to_string()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.to_string (
              self )
```

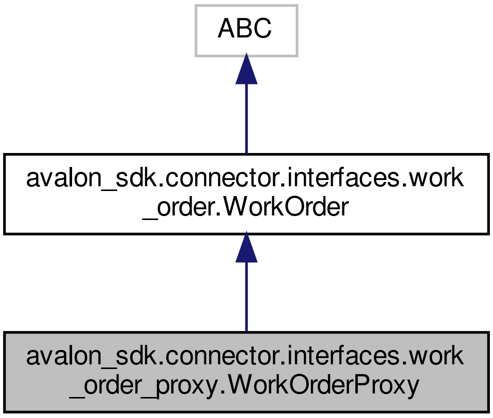
```
Create work order request string.
It is used to submit a work order.

Returns:
Work order request as a string
```

The documentation for this class was generated from the following file:

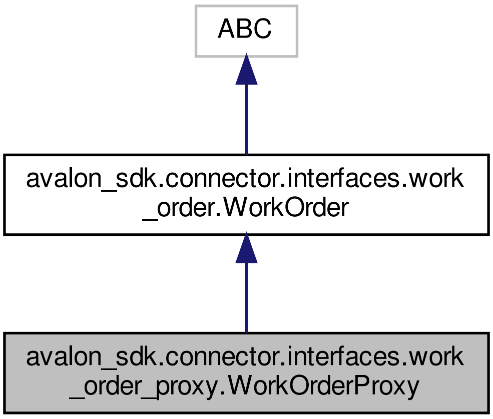
- /home/dano/git/avalon/sdk/avalon_sdk/work_order/work_order_params.py

12.71 avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy Class Reference

Inheritance diagram for avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy:


ABC
avalon_sdk.connector.interfaces.work_order.WorkOrder
avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy

Collaboration diagram for avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy:


ABC
avalon_sdk.connector.interfaces.work_order.WorkOrder
avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy

Public Member Functions

- def **__init__** (self)
- def encryption_key_start (self, tag, id=None)
- def work_order_complete (self, work_order_id, work_order_response)

12.71.1 Detailed Description

```
This class is an abstract base class that contains
abstract APIs to manage work orders.
This interface is going to be used by proxy model.
```

12.71.2 Member Function Documentation

12.71.2.1 encryption_key_start()

```
def avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy.encryption_key_start (
              self,
              tag,
              id = None )
```

```
Inform the Worker that it should start
encryption key generation for this requester.
This API is for the proxy model.

Parameters:
tag       is an optional parameter.
  If it is zero, the transaction sender's address
  is used as a tag
id        Optional JSON RPC request ID

Returns:
0 on success, otherwise an error code.
```

12.71.2.2 work_order_complete()

```
def avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy.work_order_complete (
              self,
              work_order_id,
              work_order_response )
```

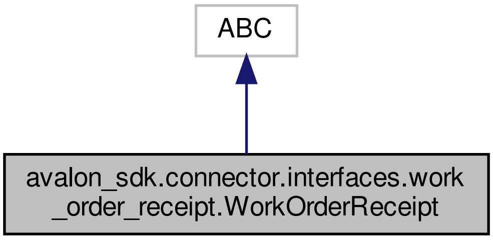
```
This function is called by the Worker Service to
complete a Work Order successfully or in error.
This API is for the proxy model.

Parameters:
work_order_id       Unique ID to identify the work order request
work_order_response Work order response data in a string

Returns:
errorCode           0 on success or non-zero on error.
```

The documentation for this class was generated from the following file:

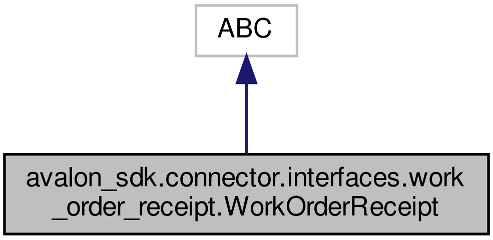
- /home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/work_order_proxy.py

12.72 avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt Class Reference

Inheritance diagram for avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt:


ABC
avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt

Collaboration diagram for avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt:


ABC
avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt

Public Member Functions

- def **__init__** (self)
- def work_order_receipt_create (self, work_order_id, worker_id, worker_service_id, requester_id, receipt_create_status, work_order_request_hash, id=None)
- def work_order_receipt_update (self, work_order_id, updater_id, update_type, update_data, update_signature, signature_rules, id=None)
- def work_order_receipt_retrieve (self, work_order_id, id=None)
- def work_order_receipt_update_retrieve (self, work_order_id, updater_id, update_index, id=None)
- def work_order_receipt_lookup (self, worker_service_id, worker_id, requester_id, receipt_status, id=None)
- def work_order_receipt_lookup_next (self, worker_service_id, worker_id, requester_id, receipt_status, last_lookup_tag, id=None)
- def work_order_receipt_update (self, work_order_id, updater_id, update_type, update_data, update_signature, signature_rules, id=None)

12.72.1 Detailed Description

```
This class is an abstract base class that contains
abstract APIs to manage work order receipts.
```

12.72.2 Member Function Documentation

12.72.2.1 work_order_receipt_create()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↩
create (
              self,
              work_order_id,
              worker_id,
              worker_service_id,
              requester_id,
              receipt_create_status,
              work_order_request_hash,
              id = None )
```

```
Create a work order receipt.

Parameters:
work_order_id           ID of the Work Order
worker_id               Worker id that should execute the Work Order
worker_service_id       ID of the Worker Service that
                hosts the Worker
requester_id            ID of the requester
receipt_create_status   Initial receipt status defined
                in EEA spec 7.1.1
work_order_request_hash Hash value of the work order request as
                defined in EEA spec 6.7.
id                      Optional JSON RPC request ID

Returns:
0 on success, otherwise an error code.

receipt_create_status values are:
0 - "pending". The work order is waiting to be processed by the
    worker
1 - "completed". The worker processed the Work Order and no more
    worker updates are expected
2 - "processed". The worker processed the Work Order, but
    additional worker updates are expected,
    e.g. oracle notifications
3 - "failed". The Work Order processing failed,
    e.g. by the worker service because of invalid workerId
4 - "rejected". The Work Order is rejected by the smart contract,
    e.g. invalid workerServiceId
5 to 254 - are reserved
255      - indicates any status
>255     - application-specific values
```

12.72.2.2 work_order_receipt_lookup()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↩
lookup (
              self,
              worker_service_id,
              worker_id,
              requester_id,
              receipt_status,
              id = None )
```

```
Lookup a work order receipt.

Parameters:
worker_service_id Worker Service ID whose receipts will be
          retrieved
worker_id         Worker Id whose receipts are requested
requester_id      ID of the entity requesting receipts
receipt_status    Defines the status of the receipts retrieved
id                Optional JSON RPC request ID

Returns:
On success, return tuple containing matching count, lookup tag,
and list of work order receipt ids:
total_count   Total number of receipts matching the lookup
      criteria. If this number is bigger than the size
      of the ids array, the caller should use a lookup_tag
      to call work_order_receipt_lookup_next() to retrieve
      the remainder of the receipt IDs
lookup_tag    Optional parameter. If returned, it means that
      there are more matching receipts. They can be retrieved
      by calling work_order_receipt_lookup_next() with
      this tag as input
ids           Array of the Work Order receipt IDs that match the input
```

12.72.2.3 work_order_receipt_lookup_next()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↩
lookup_next (
              self,
              worker_service_id,
              worker_id,
              requester_id,
              receipt_status,
              last_lookup_tag,
              id = None )
```

```
Retrieve subsequent work order receipts after calling
work_order_receipt_lookup().

Parameters:
worker_service_id Worker Service ID
worker_id         Worker ID value derived from the worker's DID
requester_id      Requester ID
last_lookup_tag   One of the output parameters for function
          work_order_receipt_lookup()
id                Optional JSON RPC request ID

Returns:
```

```
On success, return a tuple containing total count, look up tag, and
list of work order IDs:
total_count       Total number of receipts matching the lookup
          criteria
lookup_tag        Optional parameter. If it is returned, it means
          that there are more matching receipts that can be
          retrieved by calling this function again and with
          this tag as an input parameter
ids               Array of the Work Order receipt IDs that match the
          input criteria from the corresponding call to
          work_order_receipt_lookup().
```

12.72.2.4 work_order_receipt_retrieve()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↩
retrieve (
              self,
              work_order_id,
              id = None )
```

```
Retrieve a work order receipt.

Parameters:
work_order_id    ID of the Work Order to be retrieved
id               Optional JSON RPC request ID

Outputs:
On success, return worker_service_id, requester_id, work_order_id,
receipt_create_status, and work_order_request_hash, as defined in
work_order_receipt_create().

receipt_create_status matches the status at the time of the
receipt creation if there has not been any receipt updates
changing its status. Otherwise it matches the status set by
the latest receipt update.
```

12.72.2.5 work_order_receipt_update() [1/2]

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↩
update (
              self,
              work_order_id,
              updater_id,
              update_type,
              update_data,
              update_signature,
              signature_rules,
              id = None )
```

```
Update a Work Order Receipt.
This API is implemented by a work order receipts smart contract and it
can be called by one of the following participants:
- By or on the behalf of the Worker identified during the receipt
  creation, e.g. to notify about the work order completion
- By or on the behalf of other Workers, e.g. to submit an oracle
  notification
- By the Work Order Receipt creator (requester)
- By other participants, e.g. to acknowledge the Work Order results
  in case of multi-party Work Order processing

Parameters::
work_order_id    Work Order ID that was sent in the
         corresponding work_order_submit request
updater_id       ID of the updating entity. It is optional if it
         is the same as the transaction sender address
update_type      Type of the Work Order update that defines
         how the update should be handled.
         If update_type is from 0 to 255, the update sets
         the receipt status to update_type value. Refer to
         Creating a Work Order Receipt. For other values,
         the processing is application-specific
update_data      Update-specific data that depends on the
         updater type defined in EEA spec 7.1.2.
         If the update sets the Work Order Receipt status
         to completed or processed, it is a hash value of
         the Work Order Response. In all other
         cases, update_data are application-specific
update_signature Optional signature of concatenated
         work_order_id, update_type, and update_data.
         It is required only if the updater_id is not the
         same as the transaction sender address. Hashing and
         signing algorithms are defined by signature_rules
signature_rules  Defines hashing and signing algorithms,
         that are separated by forward slash '/'.
         E.g. "SHA-256/RSA-OAEP-4096". Optional parameter
         but required if signing algorithms are different
         from the algorithms defined for the Worker defined
         during receipt creation
id               Optional JSON RPC request ID

Returns:
Zero on success, otherwise an error code.
```

12.72.2.6 work_order_receipt_update() [2/2]

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↩
update (
              self,
              work_order_id,
              updater_id,
              update_type,
              update_data,
              update_signature,
              signature_rules,
              id = None )
```

```
Update a Work Order Receipt.
This API is implemented by a Work Order Receipts smart contract and
it can be called by one of the following participants:
- By or on the behalf of the Worker identified during the receipt
creation, e.g. to notify about the work order completion
- By or on the behalf of other Workers, e.g. to submit an oracle
```

```
notification
- By the Work Order Receipt creator (requester)
- By other participants, e.g. to acknowledge the Work Order results
in case of multi-party Work Order processing

Parameters:
work_order_id    Work Order ID that was sent in the
         corresponding work_order_submit request
updater_id       ID of the updating entity. It is optional if it
         is the same as the transaction sender address
update_type      Type of the Work Order update that defines
         how the update should be handled.
         If update_type is from 0 to 255, the update sets
         the receipt status to update_type value. Refer to
         Creating a Work Order Receipt. For other values,
         the processing is application-specific
update_data      Update-specific data that depends on the
         updater type defined in EEA spec 7.1.2.
         If the update sets the Work Order Receipt status
         to completed or processed, it is a hash value of
         the Work Order Response. In all other
         cases, update_data are application-specific
update_signature Optional signature of concatenated
         work_order_id, update_type, and update_data.
         It is required only if the updater_id is not the
         same as the transaction sender address. Hashing and
         signing algorithms are defined by signature_rules
signature_rules  Defines hashing and signing algorithms,
         that are separated by forward slash '/'.
         E.g. "SHA-256/RSA-OAEP-4096". Optional parameter
         but required if signing algorithms are different
         from the algorithms defined for the Worker defined
         during receipt creation
id               Optional JSON RPC request ID
```

12.72.2.7 work_order_receipt_update_retrieve()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↩
update_retrieve (
              self,
              work_order_id,
              updater_id,
              update_index,
              id = None )
```

```
Retrieving an update to a work order receipt.

Parameters:
work_order_id Work Order ID that was sent in the
      corresponding work_order_submit request
updater_id    ID of the updating entity. Ignored if null
update_index  Index of the update to retrieve
      Value "0xFFFFFFFF" is reserved to retrieve the
      last received update
id            Optional JSON RPC request ID

Returns:
On success, return updater_id, update_type, update_data,
update_signature, signature_rules as defined
work_order_receipt_update(), and update_count.

If updater_id is null, update_count is the total number of
updates for this receipt, otherwise it is the total number
of updates made by updater_id.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/work_order_receipt.py

12.73 avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest Class Reference

Public Member Functions

- def **__init__** (self)
- def create_receipt (self, wo_request, receipt_create_status, signing_key, nonce=None)
- def update_receipt (self, work_order_id, update_type, update_data, signing_key)

Public Attributes

- **sig_obj**
- **SIGNING_ALGORITHM**
- **HASHING_ALGORITHM**

12.73.1 Detailed Description

```
Class to create work order receipt APIs such as
create, update, retrieve, and lookup.
```

12.73.2 Member Function Documentation

12.73.2.1 create_receipt()

```
def avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest.create_receipt (
              self,
              wo_request,
              receipt_create_status,
              signing_key,
              nonce = None )
```

```
Create a work order receipt corresponding to a workorder ID.

Parameters:
wo_request            JSON RPC work order request used to create the
              work order request as defined in EEA spec 6.1.1
receipt_create_status Receipt creation status
signing_key           Private key of the signer
nonce                 Optional random number or monotonic counter

Returns:
JSON RPC request of type dictionary
```

12.73.2.2 update_receipt()

```
def avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest.update_receipt (
              self,
              work_order_id,
              update_type,
              update_data,
              signing_key )
```

```
Update the existing work order receipt with
update_type and update_data.

Parameters:
work_order_id Work order ID whose receipt
      needs to be updated
update_type   Update type. These values correspond to
      receipt status as defined in EEA Spec 7.1.1
update_data   Update-specific data that depends on
      the workOrderStatus
Returns:
JSON RPC work order update receipt request of type dictionary
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/work_order_receipt/work_order_receipt.py

12.74 avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator Class Reference

Public Member Functions

- def __init__ (self)
- def validate_parameters (self, params)
- def validate_data_format (self, data)

12.74.1 Detailed Description

```
WorkOrderRequestValidator validates work order requests
for proper parameter fields and valid data formats.
```

12.74.2 Constructor & Destructor Documentation

12.74.2.1 __init__()

```
def avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator.__init__ (
              self )
```

```
Initialize __param_key_map and __value_key_map,
which are key-value pairs.
The key is the field name and value is a boolean that
indicates whether a field is mandatory (True) or optional (False).
```

12.74.3 Member Function Documentation

12.74.3.1 validate_data_format()

```
def avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator.validate_↩
data_format (
              self,
              data )
```

```
Validate data format of the params data field (inData or outData).

Parameters:
data    Data (inData or OutData)

Returns:
True and empty string on success and
False and string with error message on failure.
```

12.74.3.2 validate_parameters()

```
def avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator.validate_↩
parameters (
              self,
              params )
```

```
Validate params dictionary for existence of
fields and mandatory fields

Parameters:
params    Parameter dictionary to validate

Returns:
True and empty string on success and
False and string with error message on failure.
```

The documentation for this class was generated from the following file:

- /home/dano/git/avalon/sdk/avalon_sdk/work_order/work_order_request_validator.py

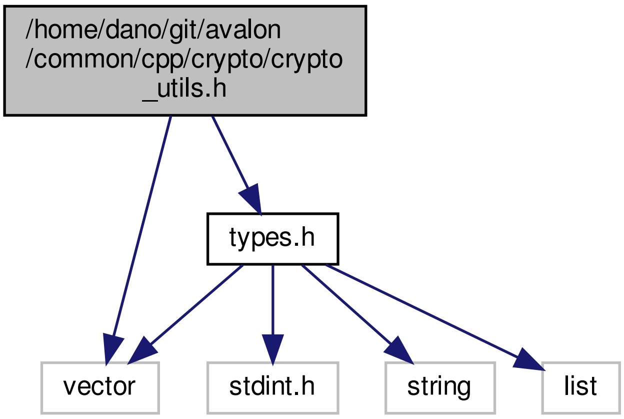
Chapter 13

File Documentation

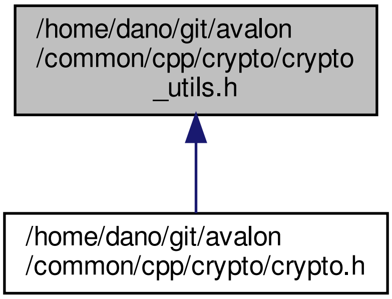
13.1 /home/dano/git/avalon/common/cpp/crypto/crypto_utils.h File Reference

```
#include <vector>
#include "types.h"
```

Include dependency graph for crypto_utils.h:



This graph shows which files directly or indirectly include this file:



Functions

- ByteArray tcf::crypto::ComputeMessageHash (const ByteArray &message)
- ByteArray tcf::crypto::RandomBitString (size_t length)
- int tcf::crypto::EVP_DecodeBlock_wrapper (unsigned char ∗out, int out_len, const unsigned char ∗in, int in_len)
- int tcf::crypto::decode_base64_block (unsigned char ∗decoded_data, const unsigned char ∗base64_data, int num_of_blocks)
- std::string tcf::crypto::CreateHexEncodedEncryptionKey ()
- std::string tcf::crypto::DecryptData (std::string cipher, std::string key)
- std::string tcf::crypto::EncryptData (std::string msg, std::string key)

13.1.1 Detailed Description

Avalon Crypto Utilities: hashing, base 64 conversion, random number generation, key generation, encrypt, and decrypt.

13.1.2 Function Documentation

13.1.2.1 ComputeMessageHash()

```
ByteArray tcf::crypto::ComputeMessageHash (
             const ByteArray & message )
```

SHA256 hashing.

Compute SHA256 hash of message.data(). Returns ByteArray containing raw binary data.

13.1.2.2 CreateHexEncodedEncryptionKey()

```
std::string tcf::crypto::CreateHexEncodedEncryptionKey ( )
```

Create symmetric encryption key and return hex encoded key string.

13.1.2.3 decode_base64_block()

```
int tcf::crypto::decode_base64_block (
             unsigned char * decoded_data,
             const unsigned char * base64_data,
             int num_of_blocks )
```

Decodes specified number of blocks of base64 encoded data.

13.1.2.4 DecryptData()

```
std::string tcf::crypto::DecryptData (
             std::string cipher,
             std::string key )
```

Decrypt cipher using given encryption key and return message.

13.1.2.5 EncryptData()

```
std::string tcf::crypto::EncryptData (
             std::string msg,
             std::string key )
```

Encrypt the message using given encryption key and return cipher.

13.1.2.6 EVP_DecodeBlock_wrapper()

```
int tcf::crypto::EVP_DecodeBlock_wrapper (
             unsigned char * out,
             int out_len,
             const unsigned char * in,
             int in_len )
```

Wrapper function for EVP_DecodeBlock. EVP_DecodeBlock pads its output with \0 if the output length is not a multiple of 3. Check if the base64 string is padded at the end and adjust the output length.

13.1.2.7 RandomBitString()

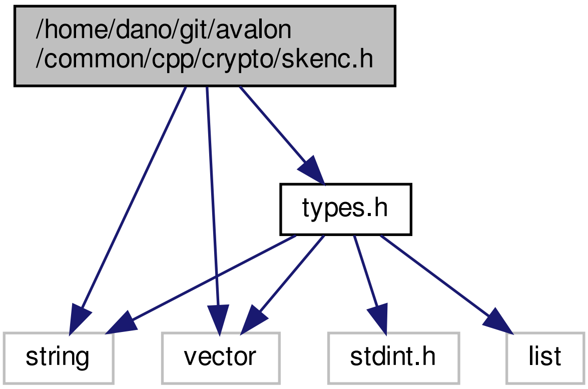
```
ByteArray tcf::crypto::RandomBitString (
             size_t length )
```

Generate a cryptographically strong random bitstring.

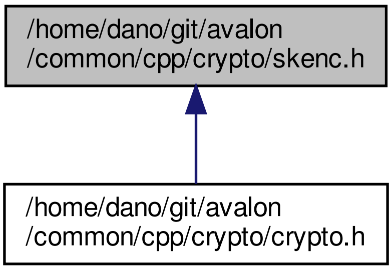
13.2 /home/dano/git/avalon/common/cpp/crypto/skenc.h File Reference

```
#include <string>
#include <vector>
#include "types.h"
```

Include dependency graph for skenc.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- tcf::crypto::skenc

Functions

- ByteArray tcf::crypto::skenc::GenerateKey ()

- ByteArray tcf::crypto::skenc::GenerateIV (const std::string &IVstring=std::string(""))
- ByteArray tcf::crypto::skenc::EncryptMessage (const ByteArray &key, const ByteArray &iv, const ByteArray &message)
- ByteArray tcf::crypto::skenc::EncryptMessage (const ByteArray &key, const ByteArray &message)
- ByteArray tcf::crypto::skenc::DecryptMessage (const ByteArray &key, const ByteArray &iv, const ByteArray &message)
- ByteArray tcf::crypto::skenc::DecryptMessage (const ByteArray &key, const ByteArray &message)

Variables

- const int tcf::crypto::constants::IV_LEN = 12
- const int **tcf::crypto::constants::SYM_KEY_LEN** = 32
- const int **tcf::crypto::constants::TAG_LEN** = 16

13.2.1 Detailed Description

Avalon secret key encryption. Uses AES-GCM 256, which also includes authentication.

13.2.2 Variable Documentation

13.2.2.1 IV_LEN

```
const int tcf::crypto::constants::IV_LEN = 12
```

AES-GCM 256 for authenticated encryption.

13.3 /home/dano/git/avalon/common/cpp/crypto/verify_certificate.h File Reference

Functions

- bool verify_certificate_chain (const char *cert_pem, const char *ca_cert_pem)

13.3.1 Detailed Description

Avalon CA certification verification.

13.3.2 Function Documentation

13.3.2.1 verify_certificate_chain()

```
bool verify_certificate_chain (
            const char * cert_pem,
            const char * ca_cert_pem )
```

Verifies certificate against CA certificate.

13.4 /home/dano/git/avalon/common/cpp/crypto/verify_signature.h File Reference

Functions

- bool verify_signature (const char *cert_pem, const char *msg, unsigned int msg_len, char *signature, unsigned int signature_len)

13.4.1 Detailed Description

Avalon signature verification.

13.4.2 Function Documentation

13.4.2.1 verify_signature()

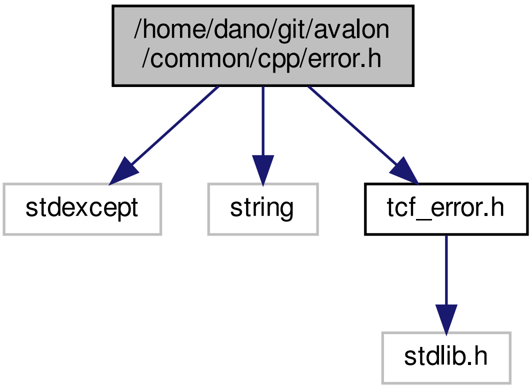
```
bool verify_signature (
            const char * cert_pem,
            const char * msg,
            unsigned int msg_len,
            char * signature,
            unsigned int signature_len )
```

Verifies signature of the message by extracting public key from certificate.

13.5 /home/dano/git/avalon/common/cpp/error.h File Reference

```
#include <stdexcept>
#include <string>
#include "tcf_error.h"
```

Include dependency graph for error.h:



Classes

- class tcf::error::Error
- class tcf::error::CryptoError
- class tcf::error::MemoryError
- class tcf::error::IOError
- class tcf::error::RuntimeError
- class tcf::error::IndexError
- class tcf::error::DivisionByZero
- class tcf::error::OverflowError
- class tcf::error::ValueError
- class tcf::error::SystemError
- class tcf::error::SystemBusyError
- class tcf::error::WorkloadError
- class tcf::error::UnknownError

Functions

- template<typename PointerType >
  void **tcf::error::ThrowIfNull** (const PointerType ptr, const char ∗msg=nullptr)
- template<typename except >
  void **tcf::error::ThrowIf** (bool condition, const char ∗msg)

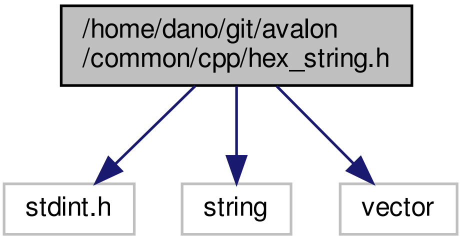
13.5.1 Detailed Description

Avalon error and exception handling functions.

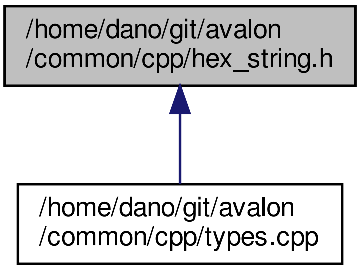
13.6 /home/dano/git/avalon/common/cpp/hex_string.h File Reference

```
#include <stdint.h>
#include <string>
#include <vector>
```

Include dependency graph for hex_string.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define HEX_STRING_SIZE(x) (static_cast<size_t>(((x) ∗ 2)))

Functions

- std::vector< uint8_t > **tcf::HexStringToBinary** (const std::string &inHexString)
- void **tcf::HexStringToBinary** (uint8_t ∗outBinaryData, size_t inBinaryDataLength, const std::string &inHexString)
- std::string **tcf::BinaryToHexString** (const std::vector< uint8_t > &inBinaryData)
- std::string **tcf::BinaryToHexString** (const uint8_t ∗inBinaryData, size_t inBinaryDataLength)

13.6.1 Detailed Description

Avalon hexadecimal string conversion functions.

13.6.2 Macro Definition Documentation

13.6.2.1 HEX_STRING_SIZE

```
#define HEX_STRING_SIZE(
              x ) (static_cast<size_t>(((x) * 2)))
```

This macro calculates the length of the actual data portion of the hex-string encoding of a buffer with x bytes PLUS the additional byte needed for the string terminator.

13.6.3 Function Documentation

13.6.3.1 BinaryToHexString()

```
std::string tcf::BinaryToHexString (
             const std::vector< uint8_t > & inBinaryData )
```

Convert an array of bytes (represented as either a std::vector of bytes or a raw array) to a hex string.

13.6.3.2 HexStringToBinary()

```
std::vector< uint8_t > tcf::HexStringToBinary (
             const std::string & inHexString )
```

Convert a hex string (i.e., a string of characters with values between '0'-'9', 'A'-'F') to an array of bytes.

13.7 /home/dano/git/avalon/common/cpp/json_utils.h File Reference

Functions

- const char ∗ **GetJsonStr** (const JSON_Object ∗json_object, const char ∗name, const char ∗err_msg=NULL)
- void **JsonSetStr** (JSON_Object ∗json, const char ∗name, const char ∗value, const char ∗err)
- void **GetByteArray** (const JSON_Object ∗object, const char ∗name, const char ∗err_msg, ByteArray &dst)
- double **GetJsonNumber** (const JSON_Object ∗object, const char ∗name)
- void **JsonSetNumber** (JSON_Object ∗json, const char ∗name, double value, const char ∗err)

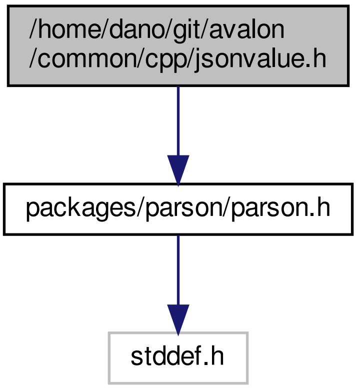
13.7.1 Detailed Description

Avalon JSON utilities.

13.8 /home/dano/git/avalon/common/cpp/jsonvalue.h File Reference

```
#include "packages/parson/parson.h"
```

Include dependency graph for jsonvalue.h:



Classes

- class JsonValue

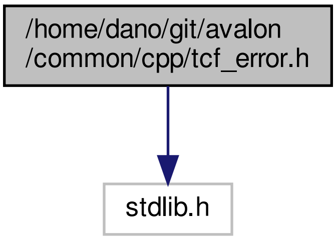
13.8.1 Detailed Description

Avalon JSON object extraction utilities.

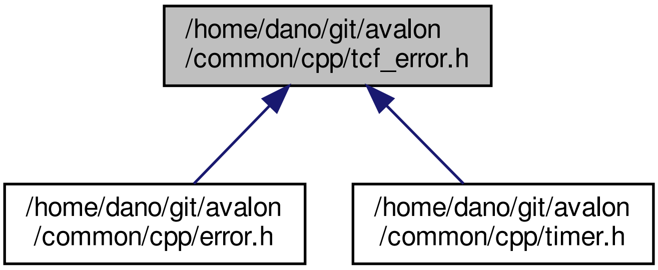
13.9 /home/dano/git/avalon/common/cpp/tcf_error.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for tcf_error.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef void(∗ **tcf_log_t**) (tcf_log_level_t, const char ∗message)

Enumerations

- enum tcf_err_t {
  TCF_SUCCESS = 0, **TCF_ERR_UNKNOWN** = -1, **TCF_ERR_MEMORY** = -2, **TCF_ERR_IO** = -3,
  TCF_ERR_RUNTIME = -4, **TCF_ERR_INDEX** = -5, **TCF_ERR_DIVIDE_BY_ZERO** = -6, **TCF_ERR_OVE↩RFLOW** = -7,
  TCF_ERR_VALUE = -8, **TCF_ERR_SYSTEM** = -9, TCF_ERR_SYSTEM_BUSY = -10, **TCF_ERR_CRYPTO** = -11,
  TCF_ERR_INVALID_WORKLOAD = -12 }
- enum **tcf_log_level_t** {
  TCF_LOG_DEBUG = 0, **TCF_LOG_INFO** = 1, **TCF_LOG_WARNING** = 2, **TCF_LOG_ERROR** = 3,
  TCF_LOG_CRITICAL = 4 }

13.9.1 Detailed Description

Avalon logging levels and error codes.

13.9.2 Enumeration Type Documentation

13.9.2.1 tcf_err_t

```
enum tcf_err_t
```

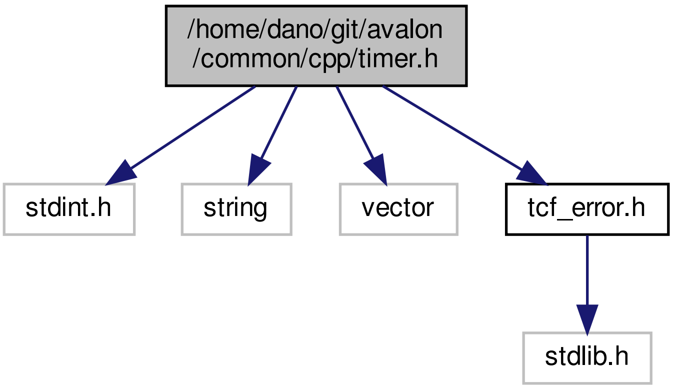
Enumerator

TCF_ERR_SYSTEM_BUSY	Indicates that the system is busy and the operation may be retried again. If retries fail this should be converted to a TCF_ERR_SYSTEM for reporting.
TCF_ERR_INVALID_WORKLOAD	Invalid workload ID

13.10 /home/dano/git/avalon/common/cpp/timer.h File Reference

```
#include <stdint.h>
#include <string>
#include <vector>
#include "tcf_error.h"
```

Include dependency graph for timer.h:



Classes

- class tcf::utility::Timer

Macros

- #define **__TIMEIT__**() {}

Functions

- uint64_t **GetTimer** (void)
- void **Log** (int level, const char ∗fmt,...)

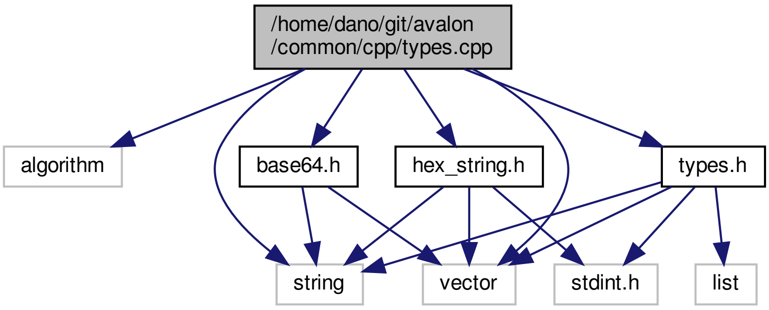
13.10.1 Detailed Description

Avalon timer utilities.

13.11 /home/dano/git/avalon/common/cpp/types.cpp File Reference

```
#include <algorithm>
#include <string>
#include <vector>
#include "types.h"
#include "base64.h"
#include "hex_string.h"
```

Include dependency graph for types.cpp:



Functions

- std::string ByteArrayToString (const ByteArray &inArray)
- StringArray ByteArrayToStringArray (const ByteArray &inArray)
- Base64EncodedString ByteArrayToBase64EncodedString (const ByteArray &buf)
- ByteArray Base64EncodedStringToByteArray (const Base64EncodedString &encoded)
- HexEncodedString ByteArrayToHexEncodedString (const ByteArray &buf)
- ByteArray HexEncodedStringToByteArray (const HexEncodedString &encoded)

13.11.1 Detailed Description

Avalon string utilities, including base 64, hex, and byte array conversion.

13.11.2 Function Documentation

13.11.2.1 Base64EncodedStringToByteArray()

```
ByteArray Base64EncodedStringToByteArray (
             const Base64EncodedString & encoded )
```

Simple conversion from Base64EncodedString to ByteArray.

13.11.2.2 ByteArrayToBase64EncodedString()

```
Base64EncodedString ByteArrayToBase64EncodedString (
             const ByteArray & buf )
```

Simple conversion from ByteArray to Base64EncodedString.

13.11.2.3 ByteArrayToHexEncodedString()

```
HexEncodedString ByteArrayToHexEncodedString (
             const ByteArray & buf )
```

Simple conversion from ByteArray to HexEncodedString.

13.11.2.4 ByteArrayToString()

```
std::string ByteArrayToString (
             const ByteArray & inArray )
```

Simple conversion from ByteArray to std::string

13.11.2.5 ByteArrayToStringArray()

```
StringArray ByteArrayToStringArray (
             const ByteArray & inArray )
```

Conversion from byte array to string array.

13.11.2.6 HexEncodedStringToByteArray()

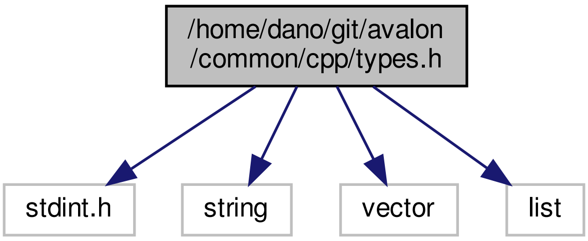
```
ByteArray HexEncodedStringToByteArray (
             const HexEncodedString & encoded )
```

Simple conversion from HexEncodedString to ByteArray. Throws ValueError.

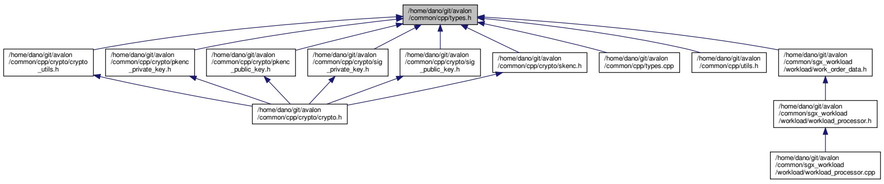
13.12 /home/dano/git/avalon/common/cpp/types.h File Reference

```
#include <stdint.h>
#include <string>
#include <vector>
#include <list>
```

Include dependency graph for types.h:



This graph shows which files directly or indirectly include this file:


/home/dano/git/avalon /common/cpp/types.h
/home/dano/git/avalon /common/cpp/crypto/crypto _utils.h
/home/dano/git/avalon /common/cpp/crypto/pkenc _private_key.h
/home/dano/git/avalon /common/cpp/crypto/pkenc _public_key.h
/home/dano/git/avalon /common/cpp/crypto/sig _private_key.h
/home/dano/git/avalon /common/cpp/crypto/sig _public_key.h
/home/dano/git/avalon /common/cpp/crypto/skenc.h
/home/dano/git/avalon /common/cpp/types.cpp
/home/dano/git/avalon /common/cpp/utils.h
/home/dano/git/avalon /common/sgx_workload /workload/work_order_data.h
/home/dano/git/avalon /common/cpp/crypto/crypto.h
/home/dano/git/avalon /common/sgx_workload /workload/workload_processor.h
/home/dano/git/avalon /common/sgx_workload /workload/workload_processor.cpp

Classes

- class StringArray

Typedefs

- typedef std::vector< uint8_t > **ByteArray**
- typedef std::string **Base64EncodedString**
- typedef std::string **HexEncodedString**

Functions

- std::string ByteArrayToString (const ByteArray &inArray)
- StringArray ByteArrayToStringArray (const ByteArray &inArray)
- Base64EncodedString ByteArrayToBase64EncodedString (const ByteArray &buf)
- ByteArray Base64EncodedStringToByteArray (const Base64EncodedString &encoded)
- HexEncodedString ByteArrayToHexEncodedString (const ByteArray &buf)
- ByteArray HexEncodedStringToByteArray (const HexEncodedString &encoded)

13.12.1 Detailed Description

Avalon string utilities, including base 64, hex, and byte array conversion.

13.12.2 Function Documentation

13.12.2.1 Base64EncodedStringToByteArray()

```
ByteArray Base64EncodedStringToByteArray (
             const Base64EncodedString & encoded )
```

Simple conversion from Base64EncodedString to ByteArray.

13.12.2.2 ByteArrayToBase64EncodedString()

```
Base64EncodedString ByteArrayToBase64EncodedString (
             const ByteArray & buf )
```

Simple conversion from ByteArray to Base64EncodedString.

13.12.2.3 ByteArrayToHexEncodedString()

```
HexEncodedString ByteArrayToHexEncodedString (
             const ByteArray & buf )
```

Simple conversion from ByteArray to HexEncodedString.

13.12.2.4 ByteArrayToString()

```
std::string ByteArrayToString (
             const ByteArray & inArray )
```

Simple conversion from ByteArray to std::string

13.12.2.5 ByteArrayToStringArray()

```
StringArray ByteArrayToStringArray (
            const ByteArray & inArray )
```

Conversion from byte array to string array.

13.12.2.6 HexEncodedStringToByteArray()

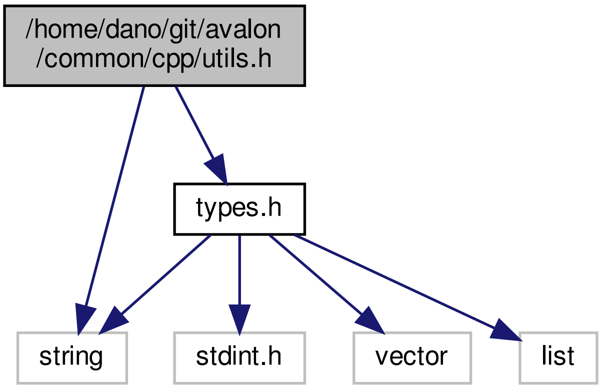
```
ByteArray HexEncodedStringToByteArray (
            const HexEncodedString & encoded )
```

Simple conversion from HexEncodedString to ByteArray. Throws ValueError.

13.13 /home/dano/git/avalon/common/cpp/utils.h File Reference

```
#include <string>
#include "types.h"
```

Include dependency graph for utils.h:



Functions

- ByteArray **StrToByteArray** (std::string str)
- std::string **ByteArrayToStr** (ByteArray ba)

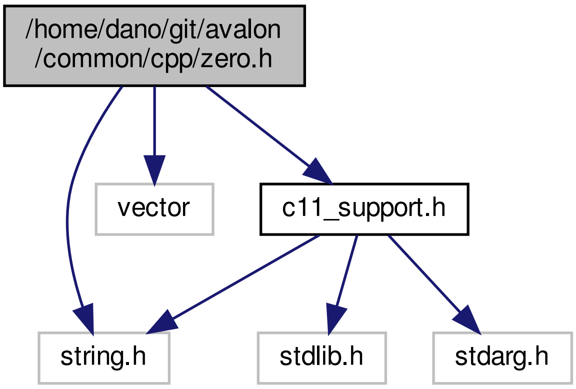
13.13.1 Detailed Description

Avalon ByteArray and String conversion utilities.

13.14 /home/dano/git/avalon/common/cpp/zero.h File Reference

```
#include <string.h>
#include <vector>
#include "c11_support.h"
```

Include dependency graph for zero.h:



Functions

- template<typename T >
  void **Zero** (T &v)
- template<typename T >
  void **ZeroV** (std::vector< T > &v)
- void **Zero** (void ∗v, size_t length)

13.14.1 Detailed Description

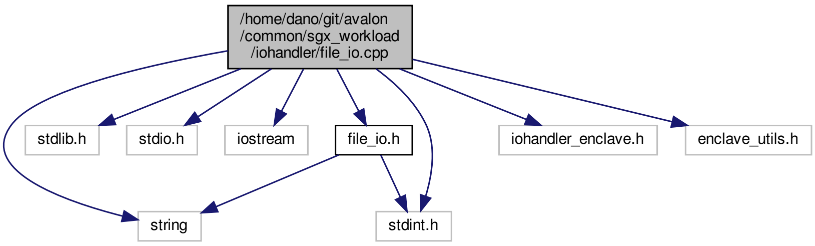
Avalon zero initialization utilities.

13.15 /home/dano/git/avalon/common/sgx_workload/iohandler/file_io.cpp File Reference

```
#include <string>
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <stdint.h>
#include "file_io.h"
#include "iohandler_enclave.h"
```

```
#include "enclave_utils.h"
```

Include dependency graph for file_io.cpp:



Macros

- #define **MAX_FILE_SIZE** 1024
- #define **MAX_IO_RESULT_SIZE** 128

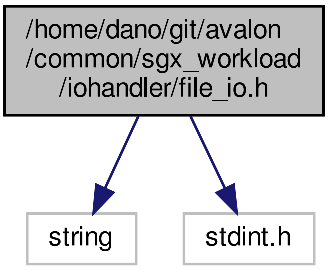
13.15.1 Detailed Description

FileIoExecutor C++ class implementation for Avalon Inside-Out File I/O. To use, #include "file_io.h"

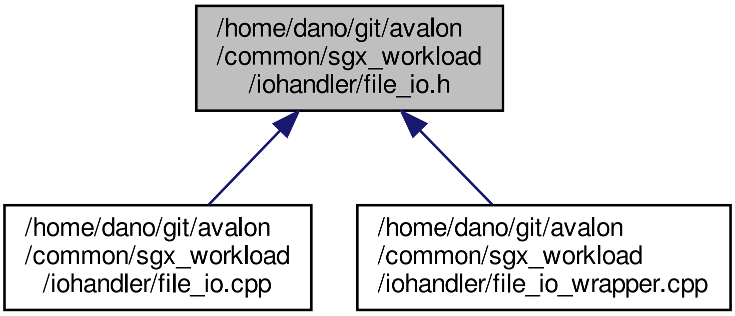
13.16 /home/dano/git/avalon/common/sgx_workload/iohandler/file_io.h File Reference

```
#include <string>
#include <stdint.h>
```

Include dependency graph for file_io.h:



This graph shows which files directly or indirectly include this file:



Classes

- class FileIoExecutor

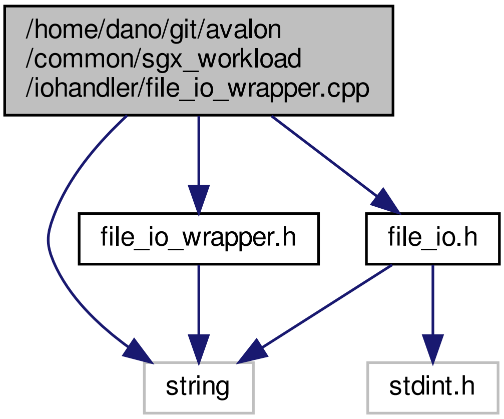
13.16.1 Detailed Description

FileIoExecutor C++ class definitions for Avalon Inside-Out File I/O. To use, #include "file_io.h"

13.17 /home/dano/git/avalon/common/sgx_workload/iohandler/file_io_wrapper.cpp File Reference

```
#include <string>
#include "file_io.h"
#include "file_io_wrapper.h"
```

Include dependency graph for file_io_wrapper.cpp:



Functions

- std::string Read (std::string file_name)
- uint32_t Write (std::string file_name, std::string data)
- uint32_t Delete (std::string file_name)

13.17.1 Detailed Description

C++ non-class wrapper implementation for Avalon Inside-Out File I/O. To use, #include "file_io_wrapper.h"

13.17.2 Function Documentation

13.17.2.1 Delete()

```
uint32_t Delete (
            std::string file_name )
```

Delete a file named file_name. Return the integer status (0 is success, non-0 is failure).

Parameters

file_name	Name of the file to delete

Returns

Status of operation (0 on success, non-0 on failure)

13.17.2.2 Read()

```
std::string Read (
            std::string file_name )
```

Read a file named file_name and return the contents in a string. Return an empty string ("") on failure.

Parameters

file_name	Name of the file to be read

Returns

String containing file contents

13.17.2.3 Write()

```
uint32_t Write (
            std::string file_name,
            std::string data )
```

Write the contents of string data to a file named file_name. Return the integer status (0 is success, non-0 is failure).

Parameters

file_name	Name of the file to write
data	Contents of file to write

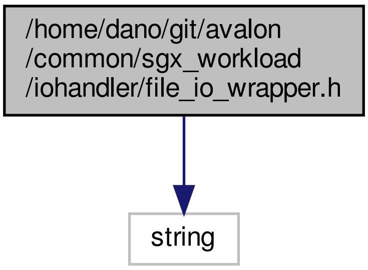
Returns

Status of operation (0 on success, non-0 on failure)

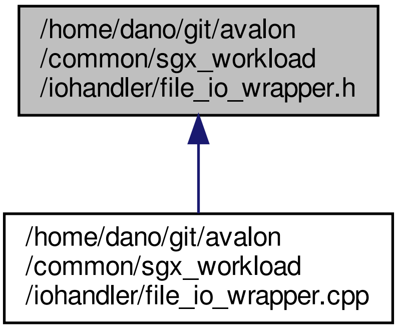
13.18 /home/dano/git/avalon/common/sgx_workload/iohandler/file_io_wrapper.h File Reference

```
#include <string>
```

Include dependency graph for file_io_wrapper.h:



This graph shows which files directly or indirectly include this file:



Functions

- std::string Read (std::string file_name)
- uint32_t Write (std::string file_name, std::string data)
- uint32_t Delete (std::string file_name)

13.18.1 Detailed Description

C++ non-class wrapper definitions for Avalon Inside-Out File I/O. To use, #include "file_io_wrapper.h"

13.18.2 Function Documentation

13.18.2.1 Delete()

```
uint32_t Delete (
            std::string file_name )
```

Delete a file named file_name. Return the integer status (0 is success, non-0 is failure).

Parameters

file_name	Name of the file to delete

Returns

Status of operation (0 on success, non-0 on failure)

13.18.2.2 Read()

```
std::string Read (
            std::string file_name )
```

Read a file named file_name and return the contents in a string. Return an empty string ("") on failure.

Parameters

file_name	Name of the file to be read

Returns

String containing file contents

13.18.2.3 Write()

```
uint32_t Write (
            std::string file_name,
            std::string data )
```

Write the contents of string data to a file named file_name. Return the integer status (0 is success, non-0 is failure).

Parameters

file_name	Name of the file to write
data	Contents of file to write

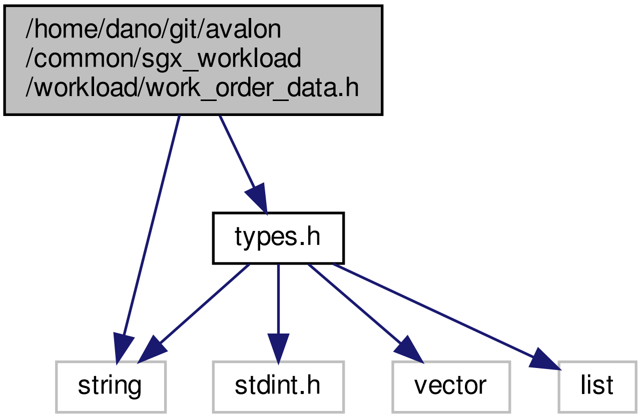
Returns

Status of operation (0 on success, non-0 on failure)

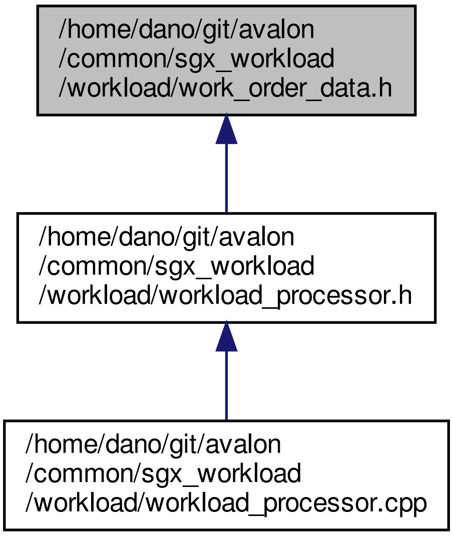
13.19 /home/dano/git/avalon/common/sgx_workload/workload/work_order_data.h File Reference

```
#include <string>
#include "types.h"
```

Include dependency graph for work_order_data.h:



This graph shows which files directly or indirectly include this file:



Classes

- class tcf::WorkOrderData

13.19.1 Detailed Description

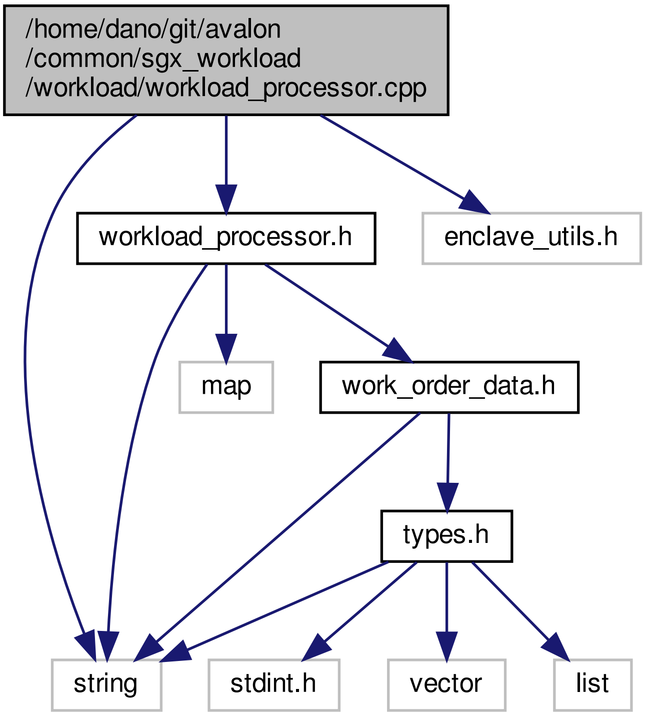
Defines class tcf::WorkOrderData for work order data submitted to workload processors. To use, #include "work←_order_data.h"

13.20 /home/dano/git/avalon/common/sgx_workload/workload/workload_processor.cpp File Reference

```
#include <string>
#include "workload_processor.h"
```

```
#include "enclave_utils.h"
```

Include dependency graph for workload_processor.cpp:



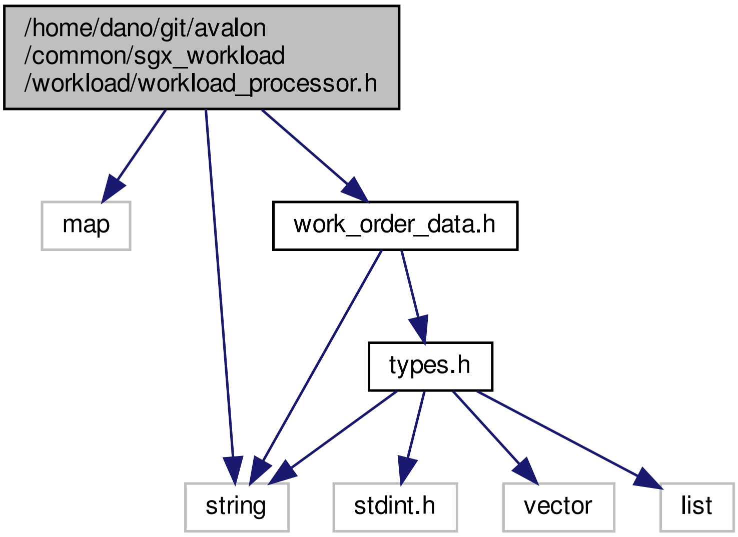
13.20.1 Detailed Description

Implements base class WorkloadProcessor to create an Avalon workload processor.

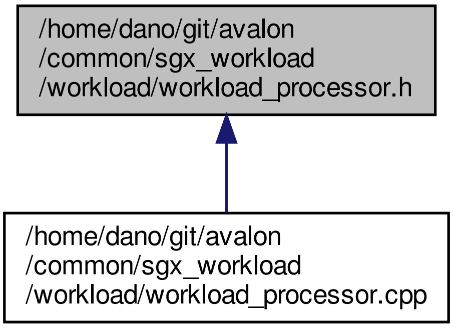
13.21 /home/dano/git/avalon/common/sgx_workload/workload/workload_processor.h File Reference

```
#include <map>
#include <string>
#include "work_order_data.h"
```

Include dependency graph for workload_processor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class WorkloadProcessor

Macros

- #define IMPL_WORKLOAD_PROCESSOR_CLONE(TYPE) WorkloadProcessor∗ Clone() const { return new TYPE(∗this); }
- #define REGISTER_WORKLOAD_PROCESSOR(WORKLOADID_STR, TYPE)

13.21.1 Detailed Description

Defines base class WorkloadProcessor and other definitions to create an Avalon workload processor. To use, #include "workload_processor.h"

13.21.2 Macro Definition Documentation

13.21.2.1 IMPL_WORKLOAD_PROCESSOR_CLONE

```
#define IMPL_WORKLOAD_PROCESSOR_CLONE(
             TYPE ) WorkloadProcessor* Clone() const { return new TYPE(*this); }
```

This macro clones an instance of class WorkloadProcessor for an Avalon worker. Example usage in a .h header file: IMPL_WORKLOAD_PROCESSOR_CLONE(Workload)

Parameters

TYPE	Name of the Workload class

13.21.2.2 REGISTER_WORKLOAD_PROCESSOR

```
#define REGISTER_WORKLOAD_PROCESSOR(
             WORKLOADID_STR,
             TYPE )
```

Value:

```
WorkloadProcessor* TYPE##_myProcessor = \
      WorkloadProcessor::RegisterWorkloadProcessor(
      WORKLOADID_STR, new TYPE());
```

This macro registers a workload processor for a specific application. It associates a string with a workload. This is the same string that is passed in the work order request JSON payload. Example usage in a .cpp source file: REGISTER_WORKLOAD_PROCESSOR(workload_id_string, Workload)

Parameters

WORKLOADID_STR	A string literal or variable identifying the workload type
TYPE	Name of the Workload class

Index