Configuration file

Theconfiguration file is a JSON file that must be specified whenstarting Tessera .

Configuration items can beoverridden from the command line .

Example configuration file

{ "useWhiteList": "boolean", "jdbc": { "url": "String", "username": "String", "password": "String", "autoCreateTables": "boolean" }, "serverConfigs": [{ "app": "ENCLAVE", // Defines us using a remote enclave, leave out if using built-in enclave "serverAddress": "http://localhost:9081", //Where to find the remote enclave "communicationType": "REST" }, { "app": "ThirdParty", "serverAddress": "http://localhost:9081", "bindingAddress": "String - url with port e.g. http://127.0.0.1:9081", "communicationType": "REST", "cors": { "allowedMethods": ["GET", "POST", "PUT", "DELETE", "OPTIONS", "HEAD"], "allowedOrigins": ["http://localhost:63342"], "allowedHeaders": ["content-type"], "allowCredentials": true } }, { "app": "Q2T", "serverAddress": "unix:/tmp/tm.ipc", "communicationType": "REST" }, { "app": "P2P", "serverAddress": "http://localhost:9001", "bindingAddress": "String - url with port e.g. http://127.0.0.1:9001", "sslConfig": { "tls": "enum STRICT,OFF", "generateKeyStoreIfNotExisted": "boolean", "sslConfigType": "Enumeration: SERVER_ONLY, CLIENT_ONLY, SERVER_AND_CLIENT", "serverKeyStore": "Path", "serverTlsKeyPath": "Path", "serverTlsCertificatePath": "Path", "serverKeyStorePassword": "String", "serverTrustStore": "Path", "serverTrustCertificates": ["Path..."], "serverTrustStorePassword": "String", "serverTrustMode": "Enumeration: CA, TOFU, WHITELIST, CA_OR_TOFU, NONE", "clientKeyStore": "Path", "clientTlsKeyPath": "Path", "clientTlsCertificatePath": "Path", "clientKeyStorePassword": "String", "clientTrustStore": "Path", "clientTrustCertificates": ["Path..."], "clientTrustStorePassword": "String", "clientTrustMode": "Enumeration: CA, TOFU, WHITELIST, CA_OR_TOFU, NONE", "knownClientsFile": "Path", "knownServersFile": "Path" }, "communicationType": "REST", "properties": { "partyInfoInterval": "Long", "enclaveKeySyncInterval": "Long", "syncInterval": "Long", "resendWaitTime": "Long" } }], "peer": [{ "url": "url e.g. http://127.0.0.1:9000/" }], "keys": { "passwordFile": "Path", "keyVaultConfigs": [{ "keyVaultType": "Enumeration: AZURE, HASHICORP, AWS", "properties": "Map[string]string" }], "keyData": [{ "config": { "data": { "aopts": { "variant": "Enum : id,d or i", "memory": "int", "iterations": "int", "parallelism": "int" }, "bytes": "String", "snonce": "String", "asalt": "String", "sbox": "String", "password": "String" }, "type": "Enum: argon2sbox or unlocked. If unlocked is defined then config data is required. " }, "privateKey": "String", "privateKeyPath": "Path", "azureVaultPrivateKeyId": "String", "azureVaultPrivateKeyVersion": "String", "publicKey": "String", "publicKeyPath": "Path", "azureVaultPublicKeyId": "String", "azureVaultPublicKeyVersion": "String", "hashicorpVaultSecretEngineName": "String", "hashicorpVaultSecretName": "String", "hashicorpVaultSecretVersion": "Integer (defaults to 0 (latest) if not set)", "hashicorpVaultPrivateKeyId": "String", "hashicorpVaultPublicKeyId": "String" }] }, "alwaysSendTo": ["String..."], "bootstrapNode": false, "unixSocketFile": "Path", "features": { "enableRemoteKeyValidation": false, "enablePrivacyEnhancements": false }, "encryptor": { "type": "Enumeration: NACL, EC", "properties": { "symmetricCipher": "String (defaults to AES/GCM/NoPadding if type = EC)", "ellipticCurve": "String (defaults to secp256r1 if type = EC)", "nonceLength": "String (defaults to 24 if type = EC)", "sharedKeyLength": "String (defaults to 32 if type = EC)" } } }

Configuration items

mode

Set themode toorion to use Tessera as the privacy manager when usingHyperledger Besu in non-GoQuorum mode .Enabling this mode changes Tessera's behavior. This property is optional.

useWhiteList

Use theuseWhiteList field to restrict connections to Tessera to specified peers. If set totrue , then only nodes listed in thepeer list are allowed to connect.

jdbc

Use thejdbc property to connect to the database. You can also specify an external database. Any valid JDBC URL can be specified.

Field Required Description url Required JDBC URL of the database. username Required Database username. password Required Database password. You can alsoencrypt the password using Jasypt. autoCreateTables Optional Automatically generates the required database tables. Iffalse , then users must manually create the required tables using thesupplied DDLs . Defaults tofalse .

serverConfigs

Use theserverConfigs property to configure the following servers:

- ENCLAVE
- P2P
- Q2T
- ThirdParty

Each server can also be configured to:

- Secure communication usingTLS
- .
- Store API metrics in anInfluxDB
- .

ENCLAVE

Defines an optionalremote enclave . Leave out if using alocal enclave .

Field Required Description app Required Type of server being configured. Set toENCLAVE . serverAddress RequiredServer address . bindingAddress Optional Specify a bind to an internal IP while advertising an external IP usingserverAddress . communicationType Required Type of server communication. OnlyREST is currently supported. influxConfig Optional Configure the server to use InfluxDB. sslConfig Optional Secure communication with TLS .

P2P

The peer-to-peer (P2P)server is used to perform discovery and send and receive encrypted payloads.

Field Required Description app Required Type of server being configured. Set toP2P . serverAddress RequiredServer address . bindingAddress Optional Specify a bind to an internal IP while advertising an external IP usingserverAddress . communicationType Required Type of server communication. OnlyREST is currently supported. influxConfig Optional Configure the server to use InfluxDB. sslConfig Optional Secure communication with TLS .

Q2T

The Quorum-to-Tessera (Q2T)server is used to check if the Tessera node is running, and to send and receive private transactions.

Field Required Description app Required Type of server being configured. Set toQ2T . serverAddress RequiredServer address . bindingAddress Optional Specify a bind to an internal IP while advertising an external IP usingserverAddress . communicationType Required Type of server communication. OnlyREST is currently supported. influxConfig Optional Configure the server to use InfluxDB. sslConfig Optional Secure communication with TLS .

ThirdParty

Tessera uses theThirdParty server to store encrypted payloads for external applications.

Field Required Description app Required Type of server being configured. Set toThirdParty . serverAddress RequiredServer address . bindingAddress Optional Specify a bind to an internal IP while advertising an external IP usingserverAddress . communicationType Required Type of server communication. OnlyREST is currently supported. cors Optional Configure CORS to control access to resources outside the domain. influxConfig OptionalConfigure the server to use InfluxDB. sslConfig Optional Secure communication with TLS .

influxConfig

Configure an InfuxDBserver to record metrics.

Field Required Description serverAddress Required InfluxDB server address. dbName Required InfluxDB database name.

pushIntervalInSecs Required How often, in seconds, Tessera pushes metrics to the database. sslConfig Optional Configure one-way TLS . If TLS is enabled, clients can validate the identity of the InfluxDB server.

sslConfig

Field Required Description tls Required Setting toSTRICT enables TLS . Setting toOFF disables TLS. generateKeyStoreIfNotExisted Optional Tessera checks whether files exist in theserverKeyStore andclientKeyStore paths. If the files don't exist, new key stores are generated in theserverKeyStore andclientKeyStore paths. sslConfigType Optional TLS configuration type based on server configuration, options areSERVER_ONLY ,CLIENT_ONLY ,SERVER_AND_CLIENT serverKeyStore Optional Path to server key store. serverKeyStorePassword Optional Password required forserverKeyStore . serverTlsKeyPath Optional File containing the private key for the server TLS certificate. serverTlsCertificatePath Optional File containing the server TLS certificate. serverTrustStore Optional Path to the server truststore. serverTrustStorePassword Optional Password for the server trust store. serverTrustCertificates Optional Array of trust store certificates ifserverTrustStore is undefined. serverTrustMode Required Trust mode for the server, options areTOFU ,WHITELIST ,CA ,CA_OR_TOFU , andNONE . clientKeyStore Optional Path to clientkey store . clientKeyStorePassword Optional Password for the client key store. clientTlsKeyPath Optional Path to client TLS key. clientTlsCertificatePath Optional Path to client TLS certificate. clientTrustStore Optional Path to client trust store. clientTrustStorePassword Optional Password for the client trust store. clientTrustCertificates Optional Array of trust store certificates ifclientTrustStore is undefined. clientTrustMode Required Trust mode for the client, options areTOFU ,WHITELIST ,CA ,CA_OR_TOFU , andNONE . knownClientsFile Optional Known clients file for the server. This contains the fingerprints of public keys of other nodes that are allowed to connect to this node. knownServersFile Optional Known servers file for the client. This contains the fingerprints of public keys of other nodes that this node has encountered. environmentVariablePrefix Optional Prefix to uniquely identify environment variables for this server SSL configuration.

cors

Configurecross-origin resource sharing (CORS) to control access to resources outside the domain.

!!! important

CORS is only supported with theThirdParty server type.

Field Required Description allowedMethods Optional List of methods to allow. Options areGET ,POST ,PUT ,DELETE ,OPTIONS , andHEAD . If not included, all methods are allowed. allowedOrigins Optional List of comma-separated origin domain URLs for CORS validation. Each entry in the list can contain the “*” (wildcard) character to match any sequence of characters. For example,*localhost matcheshttp://localhost orhttps://localhost . allowedHeaders Optional List of allowed headers. If not included, the requestAccess-Control-Request-Headers are copied into the response asAccess-Control-Allow-Headers . allowCredentials Optional The value for the Access-Control-Allow-Credentials response header. The default istrue .

peer

List of Tessera node URLs used to discover other nodes.

keys

Configure access to yourkeys .

Field Required Description passwordFile Optional Path to the password file. keyVaultConfigs Optional Configuration details of the vault being used . keyData Required Details to access the private and public key pair.

keyVaultConfigs

Configuration details for the vault used.

Field Required Description keyVaultType Optional Type of vault. Options areHASHICORP ,AWS , andAZURE . properties Optional Properties to accessAWS Secrets Manager ,Azure Key Vault , orHashiCorp Vault .

keyData

Configuration details toaccess the private key and public key .

Field Required Description config Optional Configuration details for theprotected orunprotected inline key pairs. privateKey Optional Private key in plain text. privateKeyPath Optional Path to the private key file . publicKey Optional Public key in plain text. publicKeyPath Optional Path to the public key file . awsSecretsManagerPublicKeyId Optional ID of the public key secret inAWS Secrets Manager . awsSecretsManagerPrivateKeyId Optional ID of the private key secret inAWS Secrets Manager . azureVaultPrivateKeyId Optional ID of the private key secret inAzure Key Vault . azureVaultPrivateKeyVersion Optional Version of the private key to access inAzure Key Vault . azureVaultPublicKeyId Optional ID of the public key secret inAzure Key Vault . azureVaultPublicKeyVersion Optional Version of the private key to access inAzure Key Vault . hashicorpVaultSecretEngineName Optional Name of theHashiCorp Vault secrets engine. hashicorpVaultSecretName Optional Name of the secret in theHashiCorp Vault secrets engine. hashicorpVaultSecretVersion Optional Version of the secret in theHashiCorp Vault secrets engine. hashicorpVaultPrivateKeyId Optional ID of the private key secret inHashiCorp Vault . hashicorpVaultPublicKeyId Optional ID of the public key secret inHashiCorp Vault .

alwaysSendTo

Comma-separated list of public keys to include as recipients for every transaction sent through the node. This allows you to configure a node that is sent a copy of every transaction, even if it isn't specified as a party to the transaction.

This can be used, for example, to send a copy of every transaction to a node for audit purposes.

bootstrapNode

If set totrue , the node functions as abootstrap for other nodes.

unixSocketFile

Path to the Unix socket file.

features

Enables additional security and privacy features.

Field Required Description enableRemoteKeyValidation Optional Checks that a remote node owns the public keys being advertised . The default isfalse . enablePrivacyEnhancements Optional Enableprivacy enhancements . The default isfalse . enableMultiplePrivateStates Optional Enablemultiple private states . The default isfalse .

encryptor

Configure Tessera to use alternative curves and symmetric ciphers . If an encryptor configuration is not specified, the defaultNaCl encryptor is used.

Field Description type The encryptor type. Possible values areEC ,NACL , andCUSTOM . The default isNACL . Iftype is set toEC , the followingproperties fields can also be configured:

Field Default Description ellipticCurve secp256r1 The elliptic curve to use. SeeSunEC provider for other options. Depending on the JCE provider you use, there may be additional curves available. symmetricCipher AES/GCM/NoPadding The symmetric cipher to use for encrypting data (GCM is mandatory as an initialization vector is supplied during encryption). nonceLength 24 The nonce length (used as the initialization vector (IV) for symmetric encryption). sharedKeyLength 32 The key length used for symmetric encryption (the key derivation operation always produces 32-byte keys and the encryption algorithm must support it). Edit this page Last updatedonOct 9, 2023 bydependabot[bot]Previous Subcommands Next Bootstrap node configuration