Configure servers for Tessera API

You can configure theservers for the Tessera API in the Tesseraconfiguration file .

Specify the servers to be started as a list inserverConfigs .

Server configuration "serverConfigs": [ ]

Server addresses

The server configuration has two address entries:

- serverAddress
-
    - Address of the server. This can be specified as an IP address or a DNS name.
- bindingAddress
-
    - (optional) Endpoint to use for the binding. Specify to bind to an internal IP while advertising an external IP usingserverAddress
- .

Each server is individually configured and can advertise overHTTP ,HTTPS , or aUnix Socket .

You can alsoconfigure CORS for theThirdParty server type.

HTTP server configuration

- Syntax
- ThirdParty example

```
{ "app": "", "serverAddress": "http://[host]:[port]/[path]", "communicationType": "REST" } { "app": "ThirdParty", "serverAddress": "http://localhost:9081", "communicationType": "REST" }
```

HTTPS server configuration

- Syntax
- P2P example

```
{ "app": "", "serverAddress": "https://[host]:[port]/[path]", "communicationType" : "REST", "sslConfig": {} } { "app": "P2P", "serverAddress": "http://localhost:9001", "sslConfig": { "tls": "enum STRICT,OFF", "generateKeyStoreIfNotExisted": "boolean", "serverKeyStore": "Path", "serverTlsKeyPath": "Path", "serverTlsCertificatePath": "Path", "serverKeyStorePassword": "String", "serverTrustStore": "Path", "serverTrustCertificates": ["Path"], "serverTrustStorePassword": "String", "serverTrustMode": "TOFU", "clientKeyStore": "Path", "clientTlsKeyPath": "Path", "clientTlsCertificatePath": "Path", "clientKeyStorePassword": "String", "clientTrustStore": "Path", "clientTrustCertificates": ["Path"], "clientTrustStorePassword": "String", "clientTrustMode": "TOFU", "knownClientsFile": "Path", "knownServersFile": "Path" }, "communicationType": "REST", "properties": { "partyInfoInterval": "Long", "enclaveKeySyncInterval": "Long", "syncInterval": "Long", "resendWaitTime": "Long" } }
```

Unix socket server configuration

- Syntax
- Q2T example

```
{ "app": "", "serverAddress": "unix://[path]", "communicationType": "REST" } { "app": "Q2T", "serverAddress": "unix:/tmp/tm.ipc", "communicationType": "REST" }
```

Configure CORS

TheThirdParty server type supportsconfiguring CORS to control access to resources.

ThirdParty CORS configuration { "app": "ThirdParty", "serverAddress": "http://localhost:9081", "communicationType": "REST", "cors": { "allowedMethods": ["GET", "POST", "PUT", "DELETE", "OPTIONS", "HEAD"], "allowedOrigins": ["http://localhost:63342"], "allowedHeaders": ["content-type"], "allowCredentials": true } } Edit this page Last updatedonNov 29, 2023 byJoshua Fernandes Previous Configure HashiCorp Vault Next Peer discovery