Configuring Certificates

The service and identity providers can use certificates to make communication more secure.

Configuring Identity Provider Signing Certificates

The identity provider can use a certificate to sign its messages. The certificate can either be loaded from a file or the Certificate Store.

The following is an example of configuring signing certificate loading from a file:

```
"IdentityProviderOptions": {
…
"MetadataSource": "MetadataOptions",
        "Metadata": {
"SigningCertificate": {
                "SourceType": "File",
                "FilePath":
                ".\\secure_files\\SigningCertificate.cer"
            }
}
}
```

This configuration example specifies the following parameters:

- **SourceType:** The source type for certificate loading. It can be either File or CertificateStore.
- **FilePath:** The path to load the certificate from. The path is relative to the execution path of the application.

The following is an example of configuring signing certificate loading from the Certificate Store:

```
"IdentityProviderOptions": {
…
"MetadataSource": "MetadataOptions",
        "Metadata": {
"SigningCertificate": {
                "SourceType": "CertificateStore",
                "StoreLocation": "LocalMachine",
                "StoreName": "My",
                "FindType": "FindBySubjectDistinguishedName",
                "FindValue": "CN=CertificateSubject",
                "ValidOnly": true
            }
}
}
```

This configuration example specifies the following parameters:

- **SourceType:** The source type for certificate loading can be either File or CertificateStore (required if SigningCertificate is configured).
- **StoreLocation:** The location of the store to search for the certificate (required if SourceType has CertificateStore value). There is no default value for the property. Possible values from the System.Security.Cryptography.X509Certificates.StoreLocation enumeration are CurrentUser and LocalMachine.
- **StoreName:** The name of the certificate store to search for the certificate (required if SourceType has CertificateStore value). There is no default value for the property.

Possible values from the System.Security.Cryptography.X509Certificates.StoreName enumeration are AddressBook, AuthRoot, CertificateAuthority, Disallowed, My, Root, TrustedPeople, and TrustedPublisher.

Important

It is recommended that the identity provider's certificate be kept in the "Other People" store, which is specified by the AddressBook enumeration value.

- **FindType:** The value type from the findValue property that will be used to find the certificate (required if SourceType has CertificateStore value). There is no default value for the property. The following values from the System.Security.Cryptography.X509Certificates.X509FindType enumeration are supported: FindByThumbprint, FindBySubjectName, FindBySubjectDistinguishedName, FindByIssuerName, FindByIssuerDistinguishedName, FindBySerialNumber, FindByTemplateName, FindByApplicationPolicy, FindByCertificatePolicy, FindByExtension, FindByKeyUsage, and FindBySubjectKeyIdentifier.

Important

For security, the FindBySerialNumber enumeration value is recommended.

- **FindValue:** The search term (string) to find the certificate (required if SourceType has CertificateStore value). There is no default value for the property.
- **ValidOnly**: Value (true or false) indicating that the certificate that will be loaded must be valid (the default is false).

Important

The certificate is validated on expiration, correct signature, trusted root certificate, and other parameters from the base certificates chain policy. Refer to base policy errors: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/ns-wincrypt-cert_chain_policy_status#members.