

# Integration Microservice for AS2 free trial Setup

Contact:  
Phillip Torbecke  
info@integrationmicroservice.com

## 1. create app service (web app)

- linux container
  - container type: single container
  - registry source: private registry
  - server url: <https://integrationmicroservicedemo.azurecr.io>
  - username: integrationmicroservicedemo
  - password: 1OCJ7zfRetK3dg0ODtIz8W9YjTEC7hzAuZ0hYslbT4+ACRA903NW
  - image/tag: as2:demo
  - Continuous deployment: yes/no
- We recommend to flag this setting to no once the environment is in production.
- go to settings-identity - set identity on to allow managed identity access

The container registry integrationmicroservicedemo.azurecr.io is exclusively used for free trial.  
If you purchase a license, we can host a dedicated container for your company or we can push the docker image to your own container registry.

## 2. create keyvault or use existing one

2.1 create access policy with "secret reader" and "certificate reader" permission for newly created app service identity from step 1

2.2 create certificates and secrets:

- 2.2.1 create certificate with your own private certificate
- you can import pfx pfx file or create new certificate in your keyvault/certificates
  - make sure your certificate is having X.509 Key Usage Flags for Digital Signature and Key Encipherment
  - this private certificate is the as2 identity of your company/endpoint
  - the certificate name is referenced in appsetting LocalPrivateCertificate (4.1)
- 2.2.2 create secret named "host--functionKey--default"
- this function key is the client secret used to authenticate as2 send method
  - max length of host--functionKey--default is 128 characters
  - stop and start the app service after change (restart will not do the trick)
  - When calling the AS2Send method, the function key for authentication can be send as "code" at the end of your AS2Send URL seen in 5. or in http header x-function-key.
- 2.2.3 create partner public keys by creating a secret for each partner
- content type needs to contain "as2 public key"
  - public key needs to be uploaded as base64 encoded string

## 3. create storage account or use existing

3.1 create blob container for as2 metadata:

- this container will be used to store ingoing as2 messages with their metadata
- the metadata blob name will start with your partners as2-from header, followed by as2-message-id, followed by guid
- blob created for each ingoing as2 message will contain the following information
  - as2 headers
  - encryption algorithm used
  - signature algorithm used
  - result of signature validation
  - information about the public key used to validate signature

3.2 create blob container for as2 payload:

- this container will be used to store the payload of ingoing as2 messages as utf-8 encoded blob
- the payload blob name will start with your partners as2-from header, followed by as2-message-id, followed by guid

3.3 create blob container for asynchronous mdn:

- the container will be used to store your asynchronously received mdn messages
- only in use when you request an asynchronous mdn from your partner by setting header "asyncMdn"
- the mdn blob name will start with your partners as2-from header, followed by as2-message-id, followed by guid

## 4. connect app service with storage and keyvault

4.1 create app-settings in your new as2 appservice:

- update the configuration with the following information:
  - set keyvault uri in appsetting AzureWebJobsSecretStorageKeyVaultUri (see 2.)
  - set appsetting AzureWebJobsSecretStorageType = true
  - set your identity in appsetting LocalPrivateCertificate (see 2.2.1)
  - set storage connection string in appsetting AzureWebJobsStorage (see 3.)
  - set metadata container name in appsetting MetadataContainer (see 3.1)
  - set payload container name in appsetting PayloadContainer (see 3.2)
  - set asyncMdn container name in appsetting AsyncMdnContainer (see 3.3)
- an example of the settings you need to add can be found under 6.

## 5. after resource setup you can use the attached postman project to test your setup

- postman call http get [https://\[youras2appservicename\].azurewebsites.net/api/ip](https://[youras2appservicename].azurewebsites.net/api/ip) to see outbound ip address of as2 Service
- configure AppInsights to see detailed tracing and statistics
- postman roundtrip with own certificates to check sending and receiving of as2 messages
- the postman project will be attached as a json file

POST  Send

Params Authorization Headers (20) Body Scripts Settings Cookies

Key	Value	Description
<input checked="" type="checkbox"/> recipientCertName	[your partner public key]	partner public key stored under secrets in keyvau...
<input checked="" type="checkbox"/> uri	https://[yourappservicename].azurewebsites.net/api/AS2Receive	as2 url of your partner
<input checked="" type="checkbox"/> from	yourAs2Name	as2 header
<input checked="" type="checkbox"/> to	partnerAs2Name	as2 header
<input checked="" type="checkbox"/> useragent	integrationmicroservice.com	info about as2 client
<input checked="" type="checkbox"/> subject	test subject	as2 message subject
<input checked="" type="checkbox"/> encryptionAlgorithm	3DES	encryption alorithm [3DES, RC2, AES256]
<input checked="" type="checkbox"/> signingAlgorithm	SHA1	signature algorithm [MD5, SHA1, SHA2, SHA256, ...
<input type="checkbox"/> timeoutMs	60000	optional: timeout in milliseconds, default 3 minutes
<input type="checkbox"/> messageId	test subject	optional: as2 messageid, default will generategui...
<input type="checkbox"/> signMdN	SHA1	optional: preferred response mdn signature algori...
<input type="checkbox"/> asyncMdn	https://[yourappservicename].t.azurewebsites.net/api/AsyncMdnRe...	optional: http url for async mdn ↵ ...

6. additional app service settings used for as2 service

```
{
  {
    "name": "DOCKER_ENABLE_CI",
    "value": "true",
    "slotSetting": false
  },
  {
    "name": "DOCKER_REGISTRY_SERVER_PASSWORD",
    "value": "1OCJ7zfRetK3dg00DtIz8W9YjTBC7hzAuZ0hYs1bT4+ACRA903NW",
    "slotSetting": false
  },
  {
    "name": "DOCKER_REGISTRY_SERVER_URL",
    "value": "https://integrationmicroservicedemo.azurecr.io",
    "slotSetting": false
  },
  {
    "name": "DOCKER_REGISTRY_SERVER_USERNAME",
    "value": "integrationmicroservicedemo",
    "slotSetting": false
  },
  {
    "name": "AzureWebJobsSecretStorageKeyVaultUri",
    "value": "https://microservice-kv-test.vault.azure.net/",
    "slotSetting": false
  },
  {
    "name": "AzureWebJobsSecretStorageType",
    "value": "keyvault",
    "slotSetting": false
  },
  {
    "name": "AzureWebJobsStorage",
    "value": "DefaultEndpointsProtocol=https;AccountName=todo;AccountKey=8[secret];EndpointSuffix=core.windows.net",
    "slotSetting": false
  },
  {
    "name": "LocalPrivateCertificate",
    "value": "[your private certificate name]",
    "slotSetting": false
  },
  {
    "name": "MetadataContainer",
    "value": "as2-metadata",
    "slotSetting": false
  },
  {
    "name": "PayloadContainer",
    "value": "as2-payload",
    "slotSetting": false
  },
  {
    "name": "AsyncMdnContainer",
    "value": "async-mdn",
    "slotSetting": false
  },
  {
    "name": "Tracking",
    "value": "True",
    "slotSetting": false
  }
}
```