User Guide | PUBLIC
Document Version: 1.0 – 2021-06-10

Working with SAP Business One API Gateway

THE BEST RUN SAP

Content

1 About SAP Business One API Gateway

SAP Business One API Gateway is a service that provides a unified service endpoint for you to access business data through an API call from a source system outside SAP Business One system. You can use this API Gateway to log in to the SAP Business One system once and then have access to the business data in all the service units which can then consumed on other user interfaces. In addition, you can access the crystal reports through an API call to the Reporting Service as third-party app developers.

This service is available for software developers for API and application.

> **i** Note
>
> The API Gateway is available for both SAP Business One and SAP Business One, version for SAP HANA.
>
> You can install API Gateway Service using SAP Business One Setup Wizard. For more information, see the Adminsitrator's Guides for SAP Business One and SAP Business One, version for SAP HANA.
>
> You should also install:
>
> - SAP Business One analytics powered by SAP HANA (for SAP Business One, version for SAP HANA). or
> - Job Service (for SAP Business One), and bind Service Unit in the SLD control center.
>   When you use reporting service related APIs, we highly recommend that you install Job Service on another server from other Server Tools components. Otherwise, it may cause Server Tools out of memory, because Server Tools is a 32-bit Java application, put too much component in the same java application will easily cause the out of memory issue.

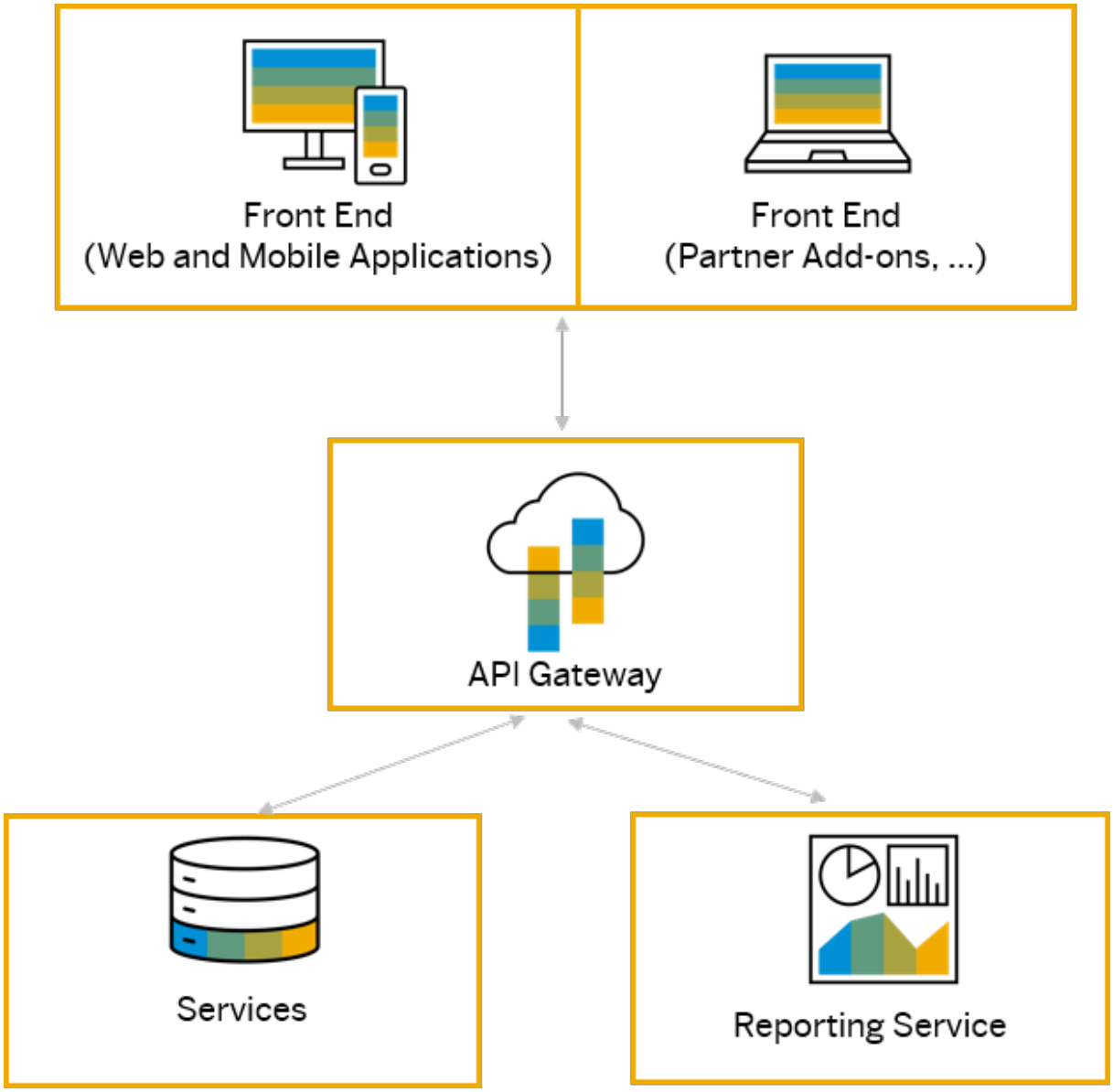
Features

The following features are available for the API Gateway:

- Powerful components that support multiple service units.
- Support for binding with one system landscape directory (SLD) service which is used across all the service units.
- Single and unified endpoint URL based on the service unit and the logged-in company.
- Simple one-time authentication mechanism.
- Support for APIs of Reporting as a Service.

For more information, see Using SAP Business One API Gateway [page 6].

Architecture Overview



This diagram illustrates how the APIs you build in the API Gateway provide you with a unified and integrated experience that is developer-friendly. It's a single endpoint of authentication. Acting as an API front end, the API Gateway accepts API calls for the accessing and controlling to the data, business logic, or functionality from your back-end services. For example, you can access the crystal reports in the Reporting Service from your front-end systems, such as web clients, mobile applications, partner add-ons or other third-party front end.

Related Information

Authentication [page 6]

Access to the Report as a Service [page 8]
Setting up Server-Side Certificates [page 17]

2 Using SAP Business One API Gateway

With the API Gateway, you can do the following operations:

- Authentication [page 6]
  The authentication service is based on the SAML protocol and can be consumed on other interfaces or API calls via SLD. It's a one-time authentication that requires users to log in to the SAP Business One system only once, and then all the API services are available while the log in session is active. Therefore, it provides a unified endpoint for accessing all APIs across the SAP Business One landscape.
  For security purposes, you can authenticate the upstream servers by generating key stores, importing certificates into those trusted key stores and then specifying the directory and password of the key store. For more information, see Setting up Server-Side Certificates [page 17].
- Access to the Report as a Service [page 8]
  You can access the crystal reports with the information that is exchanged by the sender and receiver involved. For example, you can carry out the following operations with crystal reports:
  - Get the crystal report list
  - Get report parameters for a specific report
  - Export crystal report and generate a PDF version of the report

Related Information

About SAP Business One API Gateway [page 3]
Setting up Server-Side Certificates [page 17]

2.1 Authentication

The sessions are sent in the request as a payload. The following sessions are available:

Log in Session

Resource	Description
Operation	Login
HTTP Method	Post
URL	`https://<Server Name/IP>:<Port>/login`

Resource	Description
Headers	Content-Type: application/json
Parameters	Company Name (Mandatory) User Name (Mandatory) Password (Mandatory) DB Instances (Optional)
Payload	Example `{ "CompanyDB": "10COR0809H", "UserName": "manager", "DBInstance": "C200@10.58.114.200:30013", "Password": "manager" }`
Response	Example `{ "Version": "0.0.1", "SessionTimeout": 30 }`
Error Handling	Unknown user name or password. Or DB Instance is required.

The log in payload `https://<Server Name/IP>:<Port>/Login`. In the payload, mandatory parameters include company name, user name and password. By default, DB instance is optional and is only required when the system tells you to enter the required DB instance.

Different API services are then available for you during the time when the log in session is active:

- Access to the Report as a Service [page 8]

Log out Session

Resource	Description
Operation	Log out
HTTP Method	Post
URL	`https://<Server Name/IP>:<Port>/logout`
Headers	Content-Type: application/json

Resource	Description

Response	Example
	{
	"code": 200,
	"message": {
	"lang": "en-us",
	"value": "User logged out."
	}
	}

If the user is not logged into the SAP Business One systems, the API gateway will redirect the user to the URL. When the session is logged out, you have no authorization to access the services.

Request to Report Service When Logged out

Resource	Description
Operation	Request Transaction
HTTP Method	Post
URL	`https://<Server Name/IP>:<Port>/login`
Headers	Content-Type: application/json
Response	401 Unauthorized

2.2 Access to the Report as a Service

After you have been authenticated in the API gateway, you can send the requests to the API Gateway to access the Crystal Reports. You then get a response from the API Gateway that contains the report code ID, the root name, the name of the report and the root GUID.

> **Note**
>
> Only authorized users can retrieve the reports. To check or determine the required authorizations for a user, go to the *Report Layout API* field of the *Authorizations* window (*Administration* > *System Initialization* > *Authorizations* > *General Authorizations* > *Report Layout API*). There are two options: *Full Authorization* or *No Authorization*. A superuser has full authorization. A normal user by default has no authorization and can be set full authorization or no authorization by a superuser.

Get Authorized Report List

Here is an example of getting the crystal report list.

Resource	Description
Operation	Get a list of crystal reports
HTTP Method	Get
URL	`https://<Server Name/IP>:<Port>/rs/v1/LoadAuthorizedCRList`
Headers	Content-Type: application/json
Parameter	Report Code ID Root Name Report Name Root GUID

Resource	Description

Response	Example
	{
	"result": "Success",
	"resultSet": [
	{
	"code": "RCRI0001",
	"root_name": "#1000004#8",
	"name": "Bericht Bestandsbewertungsmethode",
	"root_guid": "63"
	},
	{
	"code": "RCRI0002",
	"root_name": "#1000004#7",
	"name": "Zahlungsaufträge nach Geschäftspartner",
	"root_guid": "54"
	},
	{
	"code": "RCRI0003",
	"root_name": "#1000004#7",
	"name": "Zahlungsaufträge nach Zahlungslauf",
	"root_guid": "54"
	},
	{
	"code": "RCRI0004",
	"root_name": "#1000004#4",
	"name": "Jahresumsatzanalyse (nach Quartal)",
	"root_guid": "44"
	},
	{
	"code": "RCRI0005",
	"root_name": "#1000004#2",
	"name": "Offene Belege - Kunde",
	"root_guid": "28"
	},
	{
	"code": "RCRI0006",
	"root_name": "#1000004#8",
	"name": "Analyse zur Umschlagshäufigkeit",
	"root_guid": "63"
	},
	{
	"code": "RCRI0007",
	"root_name": "#1000004#4",
	"name": "Monatlicher Kundenstatus",
	"root_guid": "44"
	}
	]
	}

Get Report RCRI0001 Parameter

After you have a list of all the relevant crystal reports, you can send the request to get the details of the report. You will then receive a response from the API Gateway that contains the parameter details, such as the item description, the value and the database table where the report parameters are stored.

Here is an example of getting the crystal report with the RCRI0001 parameter.

Resource	Description
Operation	Get Parameter Details
HTTP Method	Get
URL	`https://<Server Name/IP>:<Port/rs/v1/LoadCR?DocCode=RCRI0001`
Headers	Content-Type: application/json
Parameters	Company Name (Mandatory)

Resource	Description
Response	Example


```
{
"error": false,
 "resultSet": [
 {
 "parameterType": "ReportParameter",
 "values": [],
 "defaultValuesDescription": {},
 "length": "131070",
 "isOptionalPrompt": "false",
 "description": "Artikel",
 "allowNullValue": "false",
 "isShownOnPanel": "true",
 "type": "xsd:string",
 "editMask": "",
 "minimumValue": "",
 "initialValues": [],
 "isEditableOnPanel": "true",
 "valueRangeKind": "Discrete",
 "allowMultiValue": "false",
 "name": "Title_Items@Title",
 "defaultValues": [],
 "currentvalues": [],
 "maximumValue": "",
 "allowCustomCurrentValues": "true"
 },
 {
 "parameterType": "ReportParameter",
 "values": [],
 "defaultValuesDescription": {},
 "length": "131070",
 "isOptionalPrompt": "true",
 "description": "Artikelnr.",
 "allowNullValue": "false",
 "isShownOnPanel": "true",
 "type": "xsd:string",
 "editMask": "",
 "minimumValue": "",
 "initialValues": [],
 "isEditableOnPanel": "true",
 "valueRangeKind": "Range",
 "allowMultiValue": "false",
 "name": "Item@Select * from OITM",
 "defaultValues": [],
 "currentvalues": [],
 "maximumValue": "",
 "allowCustomCurrentValues": "true"
 },
 {
 "parameterType": "ReportParameter",
 "values": [],
 "defaultValuesDescription": {},
 "length": "131070",
 "isOptionalPrompt": "true",
 "description": "Artikelgruppe:",
 "allowNullValue": "false",
 "isShownOnPanel": "true",
 "type": "xsd:string",
 "editMask": "",
 "minimumValue": "",
```

Resource	Description


```
 "initialValues": [],
 "isEditableOnPanel": "true",
 "valueRangeKind": "Discrete",
 "allowMultiValue": "false",
 "name": "Item Group@select \"ItmsGrpCod\",
\"ItmsGrpNam\" from OITB",
 "defaultValues": [],
 "currentvalues": [],
 "maximumValue": "",
 "allowCustomCurrentValues": "true"
 },
 {
 "parameterType": "ReportParameter",
 "values": [],
 "defaultValuesDescription": {},
 "length": "131070",
 "isOptionalPrompt": "false",
 "description": "Sep_item@separator eingeben:",
 "allowNullValue": "false",
 "isShownOnPanel": "true",
 "type": "xsd:string",
 "editMask": "",
 "minimumValue": "",
 "initialValues": [],
 "isEditableOnPanel": "true",
 "valueRangeKind": "Discrete",
 "allowMultiValue": "false",
 "name": "Sep_item@separator",
 "defaultValues": [],
 "currentvalues": [],
 "maximumValue": "",
 "allowCustomCurrentValues": "true"
 },
 {
 "parameterType": "ReportParameter",
 "values": [],
 "defaultValuesDescription": {},
 "length": "131070",
 "isOptionalPrompt": "false",
 "description": "Lager",
 "allowNullValue": "false",
 "isShownOnPanel": "true",
 "type": "xsd:string",
 "editMask": "",
 "minimumValue": "",
 "initialValues": [],
 "isEditableOnPanel": "true",
 "valueRangeKind": "Discrete",
 "allowMultiValue": "false",
 "name": "Title_Warehouse@title",
 "defaultValues": [],
 "currentvalues": [],
 "maximumValue": "",
 "allowCustomCurrentValues": "true"
 },
 {
 "parameterType": "ReportParameter",
 "values": [],
 "defaultValuesDescription": {},
 "length": "131070",
 "isOptionalPrompt": "true",
 "description": "Lagercode:",
```

Resource	Description
	"allowNullValue": "false", "isShownOnPanel": "true", "type": "xsd:string", "editMask": "", "minimumValue": "", "initialValues": [], "isEditableOnPanel": "true", "valueRangeKind": "Discrete", "allowMultiValue": "true", "name": "warehouse@Select * from OWHS", "defaultValues": [], "currentvalues": [], "maximumValue": "", "allowCustomCurrentValues": "true" }] }

Export to PDF

After you receive the details of the report, you can export the report from the system and generate a PDF version. Based on the response in the previous step, you must then put the requests into a payload before receiving a response from the API Gateway that contains all the parameters in the previous step.

Here is an example of exporting the crystal report with the RCRI0001 parameter.

Resource	Description
Operation	Export Report to PDF File
HTTP Method	Post
URL	`https://<Server Name/IP>:<Port>/rs/v1/ExportPDFData?DocCode=RCRI0001`
Headers	Content-Type: application/json
Parameters	Name Type Value

Resource	Description
Payload	**Example** `[ { "name": "Item@Select * from OITM", "type": "xsd:string", "value": [ [ "I1","I1" ] ] }, { "name": "Item Group@select \"ItmsGrpCod\", \"ItmsGrpNam\" from OITB", "type": "xsd:string", "value": [ [ "100" ] ] }, { "name": "warehouse@Select * from OWHS", "type": "xsd:string", "value": [ [ "01" ] ] } ]`
Response	**Example** `JVBERi0xLjcgCiXi48/ TIAo3IDAgb2JqCjw8Ci9Db250ZW50cyBbIDggMCBSICBdIAov UGFyZW50 IDUgMCBSIAovUmVzb3VyY2VzIDYgMCBSIAovVHlwZSAvUGFnZ Qo+PgplbmRvYmoKNiAwIG9iago8 PAovRm9udCA8PAovdHRmMCAxMSAwIFIgCi90dGYxIDE3IDAgU iAKL3R0ZjIgMjMgMCBSIAovdHRm MyAyOSAwIFIgCj4+Ci9Qcm9jU2V0IDMzIDAgUiAKPj4KZW5kb 2JqCjggMCBvYmoKPDwKL0ZpbHRl ciBbIC9GbGF0ZURlY29kZSBdCi9MZW5ndGggNjc0Cj4+CnN0c mVhbQp4nHVUTW/bMAy9fw2B/bMy9+1fw2B2s ...`

Note

The API gateway responses with base 64 data. You need to manually convert the format to PDF. To do so, proceed as follows:

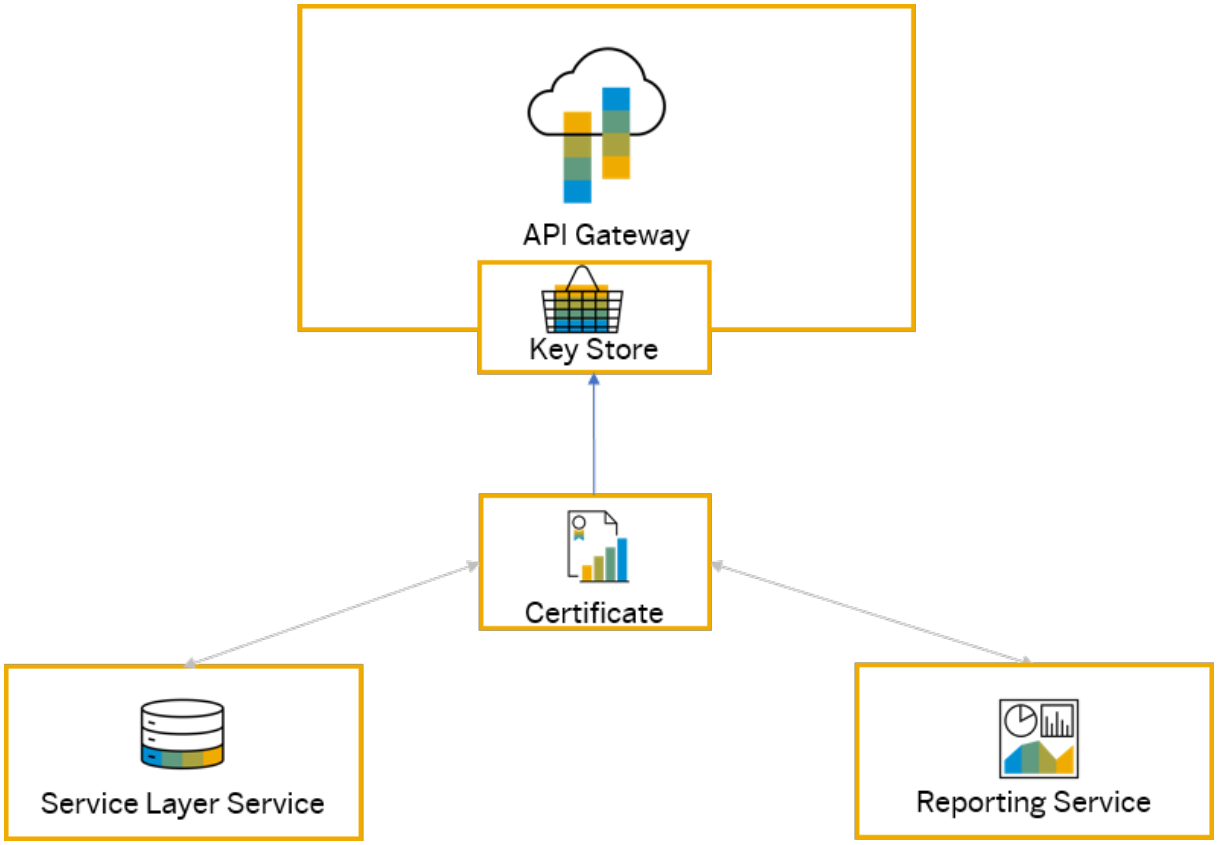
1. Open the Base 64 Converter at https://base64converter.com
2. Copy the system response and paste the base 64 data.

3. Convert it into PDF and download it.

3 Setting up Server-Side Certificates

You can use the API gateway to be authenticated to access the upstream servers. The server-side certificates are retrieved from the back-end server and then imported to the key stores in the API gateway. This allows you to verify HTTP requests from the API gateway to your back-end system and it ensures that only the authorized requests can have access to back-end services.

Certification Process



This diagram illustrates the certification process in the API gateway. The server-side certificates are first retrieved and then imported to the key stores in the API gateway. It provides the opportunity for you to decide about the authentication process to access the upstream servers, such as the crystal reporting service, by specifying the key store, in which all the necessary server certificates of upstream servers have been imported correctly.

To set up the server-side certificates in the API gateway, you need to use a key store to import all the certificates of the upstream servers and then set the system environment variables accordingly. Depending on the scenario, you can perform the following steps:

1. Generate Keys and Key Stores [page 18]
2. Optional Step: Export Certificates [page 19]
   For servers like the Reporting Service, this additional step is needed. Before you import certificates to the key store of the API gateway, you need to export the certificates from the key store of the upstream server.
3. Import Certificates as Trusted Certificates [page 20]
4. Set up System Environment Variables for the API Gateway [page 21]

Related Information

About SAP Business One API Gateway [page 3]
Using SAP Business One API Gateway [page 6]

3.1 Generate Keys and Key Stores

Context

In order for the certificates from the upstream servers to be imported into the API gateway, you need to create a key store to hold the public and private keys.

You can refer to the following steps as an example of how to generate keys and key stores. Depending on your business requirements, you can create your own.

Procedure

1. Create a key store using the keytool command **keytool -genkey -alias clientKey -keyalg RSA -keystore proxyclient.p12 -validity 3650 -storetype PKCS12**

   Using this command you can generate keys with the following information:
   - The `alias clientKey` is to be used to as the keystore entry containing the keys that will be generated.
   - The keys are based on **RSA algorithm**.
   - The key store name is `proxyclient.p12`.
   - The validity is **3650 days**.
   - The key store type is **PKCS12**.

   The key store password is `keystores`.
2. Remove the private keys that are generated automatically in the key store, using the keytool command **keytool -delete -alias mykey -keystore proxyclient.p12 -storepass keystores**

Using this command you can delete keys with the following information:

- The `alias mykey` is to be used to as the keystore entry containing the keys that will be deleted.
- The key store name is `proxyclient.p12`.
- The key store password is `keystores`.

The key store `proxyclient.p12` is now created in the API gateway.

Related Information

Import Certificates as Trusted Certificates [page 20]
Setting up Server-Side Certificates [page 17]

3.2 Export Certificates

Context

For servers like the Reporting Service, certificates are already stored in the key store of the server. In order for the certificates to be imported to the key store of the API gateway, you must first export the certificate public key.

You can refer to the following steps as an example of how to export certificates from key stores. Depending on your business requirements, you can create your own.

Procedure

1. Export the public key, using the keytool command `keytool -export -trustcacerts -alias serverKey -file server_cert.cer -keystore proxyservice.p12 -storepass keystores`

   Using this command you can export the public keys with the following information:

   - The `alias serverKey` is the private key and is to be used to as the keystore entry containing the keys that will be exported.
   - The certificate file is `server_cert.cer`.
   - The key store name is `proxyclient.p12`.
   - The key store password is `keystores`.
2. Now the certificates are exported. Import the certificates as trusted certificates. For more information, see Import Certificates as Trusted Certificates [page 20]

3.3 Import Certificates as Trusted Certificates

Context

Certificates that are retrieved from the upstream servers are imported as trusted certificates in the API gateway key store.

You can refer to the following steps as an example of how to import certificates to key stores. Depending on your business requirements, you can create your own.

Procedure

1. Import the certificates from the upstream server of service layer, using the keytool command `keytool -import -trustcacerts -alias serverKey -file server.crt -keystore proxyclient.p12 -storepass keystores`

   Using this command you can import certificates as trusted certificates with the following information:
   - The `alias serverKey` is to be used as the keystore entry containing the keys that will be imported.
   - The file is `server.crt`.
   - The key store name is **proxyclient.p12**.
   - The key store password is `keystores`.
2. Check the entries in the key store, using the keytool command `keytool -list -keystore proxyclient.p12 -storepass keystores`.

   You can now use the key store `proxyclient.p12` in the API gateway to authenticate upstream server.

Related Information

Set up System Environment Variables for the API Gateway [page 21]
Setting up Server-Side Certificates [page 17]

3.4 Set up System Environment Variables for the API Gateway

Context

The API gateway uses the following key store and password to authenticate the upstream servers:

- Key store name: **com.sap.b1.ssl.trustStore**
- Key store password: **com.sap.b1.ssl.trustStorePassword**

Depending on the scenario, you can choose to perform one of the following activities to set up the environment variables:

- When the API Gateway service is launched, you can use the JVM parameters to specify the directory and password of the trusted key store.
  java -jar sbo-api-gateway-service.jar -Dcom.sap.b1.ssl.trustStore="C:\Users\UserID\Desktop\Certs\certKeys\client.p12" -Dcom.sap.b1.ssl.trustStorePassword=keystores
- For a Linux/Unix system, choose one of the following two options before you launch the API gateway service:
  - Input values in `.bashrc` for the current user to be authenticated
  - Export the variables before you launch the API gateway service
- For a Windows system, set the system variables under *Properties* > *Advanced System Settings* > *Environment Variables* before you launch the API gateway service.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

www.sap.com/contactsap

THE BEST RUN SAP