

A low-angle shot of a modern building with a glass facade and metal structural elements. A large pink rectangular overlay covers the lower half of the image, containing white text and the T-Mobile logo.

# **BUSINESSMAIL X.400**

## **WEBCONFIG WEB SERVICE V1.5**



**ERLEBEN, WAS VERBINDET.**

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### User information (1)

- In the past the Web Interface (WebConfig) only provided an interactive method to manage the properties and trading relations of your X.400 account and filter rules for Internet messages. You had to log into WebConfig using a browser and to do the necessary changes (create, modify or delete entries) within the GUI manually.
- With the RESTful WebConfig Web Service also an application is now able to do those changes using http(s) requests.
- The user data must be uploaded using a CSV structure within the content and will be provided using a CSV structure.
- When using a group administrator account it is possible to centralized manage all assigned accounts also via Web Service.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### User information (2)

- It is necessary to use a personalized client certificate to log into the WebConfig Web Service.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### New features in WebConfig V1.5

- Support of multiple entries when adding EDI Identifier, Relations (EDI, AS2, MessageGate, SMTP MTA) and SMTP filter to database within one task

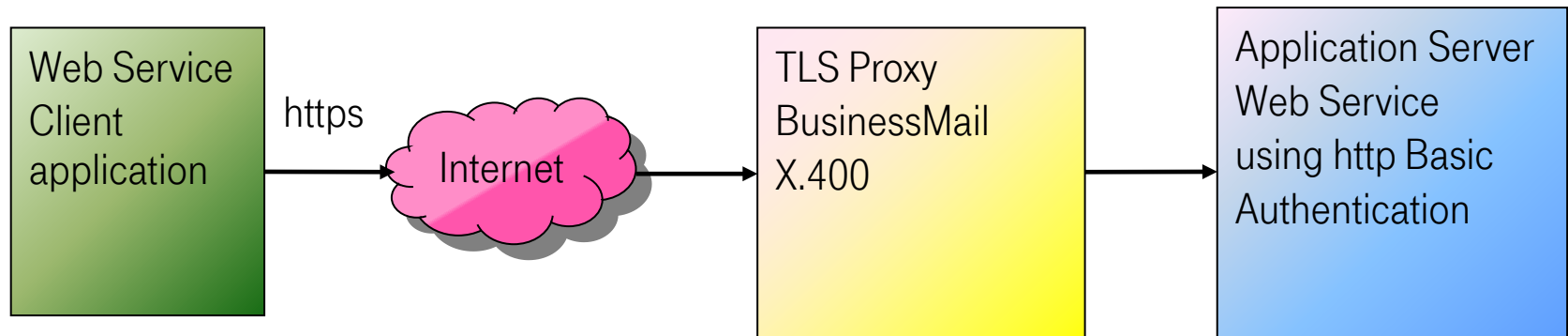


ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

Secure Communication via Internet (https)



Address [https://webconfig.viat.de/webconfig/ConfigData/<User ID>\\*/<Type>](https://webconfig.viat.de/webconfig/ConfigData/<User ID>*/<Type>)

\* User-ID (5 digest)



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### Features

- Transfer of structured data using CSV structure
- View/list (GET), create (POST) and delete (DELETE) EDI Identifier and trading partner within central EDI function (AS2, MessageGate File Interface)
- View (GET) and modify (PUT) the properties of AS2, MessageGate File Interface and SMTP MTA
- View/list (GET), create (POST), modify (PUT) and delete (DELETE) the trading relations within AS2, MessageGate File Interface and SMTP MTA
- View (GET), create (POST), modify (PUT) and delete (DELETE) the filter rules of SMTP Gateway within the different mailbox types (not SMTP MTA!)



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### General information (1)

Erleben, was verbindet.

BusinessMail X.400 :: WebConfig ipmtester (49637)

**WebConfig Management :: Downloads**

WebConfig client certificate is used to automatically authenticate user

Define the export password

Create and download

The following user data is ready for download	MIME Type
X400Adresse.csv	application/csv
SMTPFilter.csv	application/csv

- To use the WebConfig Web Service you must log into your WebConfig account and to download a personalized certificate with private key (\*.p12) and to import this into your Web Service client.
- You may create and download more than one certificate. But if one of those certificate has been compromised the helpdesk has to revoke all certificates.



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### General information (2)

Erleben, was verbindet.

BusinessMail X.400 :: WebConfig ipmtester (49637)

### WebConfig Management :: Modify Login Data

☐ Authenticate user automatically

**Authenticate user automatically**

It will be possible to automatically authenticate user via client certificate. Please download the certificate/key and import it into web browser. You have to use this key whenever you want to log into your WebConfig account. If you would like to use login via username/ password you have to contact the BusinessMail X.400 Helpdesk.

Find the necessary certificate in [Download area](#)

**Modify Password**

Old Password

New Password

Repeat Password

Ok Cancel

- After download and successfully import of the client certificate with private key into your browser and your Web service client you must enable the login via personalized client certificate (option “Authenticate user automatically”).



ERLEBEN, WAS VERBINDET.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### General information (3)

- While accessing the WebConfig Web Service you must use a personalized client certificate.
- Within the URL of Web Service “https://webconfig.viat.de/webconfig/ConfigData/<User-ID>/<value>” the User-ID (5 digests) is optional, because the service will extract the User-ID out of the subject of the certificate. The User-ID is mandatory if a group administrator wants to access one of his assigned accounts.
- The following values will be used for the request GET, if there is no other Accept-Header definition.
  - Format text/csv-c (alternative Accept:text/csv-s)
  - Character set ISO-Latin1 (at present no alternative)
  - Language en-US (alternative Accept-Language:de-DE)

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### General information (4)

- For the http(s) requests POST, PUT or DELETE it is mandatory to use the same CSV structure as provided in the result of the GET request. This also includes the “header line” listing the different field identifier. When using non default values for this CSV structure you must add valid header information (Content-Type: text/csv-s and/ or Content-Language:de-DE). Those values will also be used as “Accept:” in the result of this request.
- To delete an entry via the http(s) request DELETE its data (parameter values) must be uploaded in a CSV structure within this request as a “confirmation”. This data will be checked against the entry in the database to avoid an unintended deletion. We recommend to request a GET for this entry to receive the current data and after that request the deletion of this entry.
- The values for each parameter/ field identifier are equivalent to that in the German or English Graphical User Interface of WebConfig resp. in the CSV files you will find in menu item WebConfig Management :: Downloads.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### General information (5)

- In case of a negative request response apart the http error code additional information will be sent in the content (text or HTML code with Content-Type: text/html).



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Identifier API Reference

Method	Request	Description
List all	GET */ConfigData/<User ID>**/EDI Output CSV structure with header line	Show all EDI Identifier
Create	POST */ConfigData/<User ID>**/EDI Input CSV structure with header line, Output CSV structure with header line	Create EDI Identifier(s)
Delete	DELETE */ConfigData/<User ID>**/EDI Input CSV structure with header line, Output CSV structure with header line	Delete EDI trading partner (only, if there is no correspondent EDI trading partner)

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Trading Partner API Reference

Method	Request	Description
List all	GET */ConfigData/<User ID>**/EDI_TP Output CSV structure with header line	Show all EDI trading partner
List User-ID	GET */ConfigData/<User ID>**/EDI_TP/<User ID> Output CSV structure with header line	Show all EDI trading partner of partner's User ID
Create	POST */ConfigData/<User ID>**/EDI_TP Input CSV structure with header line, Output CSV structure with header line (Be aware, if there are several entries for partner's EDI Identifier, the CSV structure will show all)	Create EDI trading partner(s)
Delete	DELETE */ConfigData/<User ID>**/EDI_TP/<User ID> Input CSV structure with header line, Output CSV structure with header line	Delete EDI trading partner

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Trading Partner (1)

- This description is valid for all accounts with enabled central EDI function (AS2, MessageGate File Interface).
- For the management of EDI Identifier and trading partner you have the methods GET (Show/List), POST (Create) and DELETE (Delete). The method PUT (Modify) is not available.
- The http(s) request GET to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/EDI” without any additional header information will provide a CSV structure, where a comma is the separator, the field identifiers and the values will be in English, and the character set is ISO8859-1. This content shows all Identifier entries configured for this account. When using the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/EDI\_TP” the content will provide the data of all trading partner configured for this account.
- A successfully processed http(s) request GET will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Trading Partner (2)

- The entry of an EDI Identifier may have values for the following field identifiers (en-US) :  
"EDIUser ID", "EDI ID", "EDI Qualifier", "Partnertype", "Comment"
- The entry of an EDI partner may have values for the following field identifiers (en-US) :  
"EDIUser ID", "EDI ID", "EDI Qualifier", "Partner EDI ID", "Partner EDI Qualifier", "EDI Testflag", "X.400 Content Type", "Bodypart", "Comment", "User ID", "Country (c)", "ADMD (a)", "PRMD (p)", "X.121", "UA ID", "Commonname", "Given name", "Surname", "Generation", "Initials", "Organization", "Org Unit 1", "Org Unit 2", "Org Unit 3", "Org Unit 4", "Type1", "Value1", "Type2", "Value2", "Type3", "Value3", "Type4", "Value4"
- While adding the User ID of an X.400 partner to the URL the content of the response will show all entries configured for this User ID (.../EDI\_TP/<User ID>). Be aware about the specialty of EDI TP, that there might be several entries where only the configured (own) EDI Identifier in conjunction with partner's EDI Identifier and if applicable the EDI test flag will make the entry unique.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Trading Partner (3)

- The http(s) request POST to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/ EDI” and a content with CSV structure including the header line would create a new EDI trading partner entry or several within one task.
- Apart valid values for the field identifiers "EDIUser ID", "EDI ID" und "EDI Qualifier" at minimum the "Partnertype" (open, close) must be defined in the CSV structure. A successfully creation of this entry will be confirmed with “201 == Created” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request”, “404 == Not Found”, “409 == Conflict”, “413 == Payload Too Large” e.g., will be provided without data.
- If the request includes several identifier entries, the data of each successful action will be provided within a CSV structure (including header line) and the http content will also show an acknowledgment for each action (e.g., <p>New EDI Account created: TEST111:14 </p> or <p>EDI Account already exists. ILN Id: AS2TESTEDI</p>). In case of a syntax error (e.g., missing delimiter) an I/O error will be displayed, and additional information will be available.





# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Trading Partner (4)

- The http(s) request POST to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/EDI\_TP” and a content with CSV structure, a header line and valid values for the field identifiers ( EDI Identifier ...) would create a new EDI trading partner entry or several within one task.
- Apart valid values for the field identifiers "EDIUser ID", "EDI ID" und "EDI Qualifier" at minimum the "Partner EDI ID", the "X.400 Content Type", the type of "Bodypart" and either the partner's "User ID" or the partner's valid X.400 address must be defined in the CSV structure. A successfully creation of this entry will be confirmed with “201 == Created” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request”, “404 == Not Found”, “409 == Conflict”, “413 == Payload Too Large” e.g., will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Trading Partner (5)

- If the request includes several partner entries, the data of each successful action will be provided within a CSV structure (including header line) and the http content will also show an acknowledgment for each action (e.g., `<p>EDI Account does not exist. TP ID: PDMEDITEST2</p>` or `<p>New EDI Relation created: TEST112:65-376237676:65</p>`). In case of a syntax error (e.g., missing delimiter) an I/O error will be displayed, and additional information will be available.
- To delete an EDI Identifier entry (to modify an entry or the finally deletion) the content of the http(s) request DELETE send to URL “`https://webconfig.viat.de/webconfig/ConfigData/<User ID>/EDI`” must have a CSV structure with valid data of that entry. It is recommended to download this data using the http(s) request GET before initiating this deletion. The deletion of the EDI Identifier will be refused if there are any correspondent EDI trading partner entries.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### EDI Trading Partner (6)

- To delete an EDI trading partner entry (to modify an entry or the finally deletion) the content of the http(s) request DELETE send to URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/EDI\_TP/<User ID>” must have a CSV structure with valid data of that entry. It is recommended to download this data using the http(s) request GET before initiating this deletion.
- A successfully deletion of an entry will be confirmed with “200 == OK” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway API Reference (1)

Method	Request	Description
List Properties	GET */ConfigData/<User ID>**/AS2X400 Output CSV structure with header line	Show AS2 properties
List all	GET */ConfigData/<User ID>**/AS2X400_PS Output CSV structure with header line	List all AS2 relations
List User-ID	GET */ConfigData/<User ID>**/AS2X400_PS/<User ID> Output CSV structure with header line	Show AS2 relation of this partner's User ID
Create Relation	POST */ConfigData/<User ID>**/AS2X400_PS Input CSV structure with header line, Output CSV structure with header line	Create AS2 relation(s)

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway API Reference (2)

Method	Request	Description
Modify Properties	PUT */ConfigData/<User ID>**/AS2X400 Input CSV structure with header line, Output CSV structure with header line	Modify AS2 properties
Modify Relation	PUT */ConfigData/<User ID>**/AS2X400_PS/<User ID> Input CSV structure with header line, Output CSV structure with header line	Modify AS2 relation
Delete Relation	DELETE */ConfigData/<User ID>**/AS2X400_PS/<User ID> Input CSV structure with header line, Output CSV structure with header line	Delete AS2 relation

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway (1)

- For the management of the AS2 properties you have the methods GET (Show) and PUT (Modify) and for the management of the AS2 relation entries the methods GET (Show/List ), POST (Create), PUT (Modify) and DELETE (Delete).
- The http(s) request GET to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/AS2X400” without any additional header information will provide a CSV structure, where a comma is the separator, the field identifiers and the values will be in English, and the character set is ISO8859-1. The content will show the properties of the account. When using the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/AS2X400\_PS” the content will provide the data of all AS2 relations configured for this account.
- A successfully processed http(s) request GET will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway (2)

- The AS2 properties may have values for the following field identifiers (en-US) :  
"AS2User ID", "AS2 ID", "URL", "Email Address", "Inactive", "Enable AS2 Bypass", "Duplicate Check",  
"Enable End-to-End Security", "Compress", "Sign", "Encrypt", "Request MDN", "Sign MDN", "MDN Transfer",  
"Send requested asynchronous MDN", "Receipt Notification requested in X.400 messages should be",  
"Message Expiration", "X.400 Content Type", "Bodypart", "Encode binary data", "Comment", "Send Timeout  
Message", "Send Timeout MDN", "Receive Timeout MDN", "Max. number of retries"
- The entry of an AS2 relation may have values for the following field identifiers (en-US) :  
"AS2User ID", "AS2 ID", "Enable End-to-End Security", "Altern. URL", "Compress", "Sign", "Encrypt",  
"Request MDN", "Sign MDN", "MDN Transfer", "User ID", "Country (c)", "ADMD (a)", "PRMD (p)", "X.121", "UA  
ID", "Commonname", "Given name", "Surname", "Generation", "Initials", "Organization", "Org Unit 1", "Org  
Unit 2", "Org Unit 3", "Org Unit 4", "Type1", "Value1", "Type2", "Value2", "Type3", "Value3", "Type4", "Value4",  
"Send requested asynchronous MDN", "Receipt Notification requested in X.400 messages should be",  
"Message Expiration", "X.400 Content Type", "Bodypart", "Encode binary data", "Comment"

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway (3)

- While adding the User ID of an X.400 partner to the URL, the content will show the data configured for this User ID (.../AS2X400\_PS/<User ID>).
- The http(s) request POST to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/AS2X400\_PS” and a content with CSV structure, a header line and valid values for the field identifiers would create a new AS2 relation entry or several within one task.
- Apart a valid value for the field identifier "AS2User ID", at minimum the "AS2 ID" assigned to the partner, valid values for "Enable End-to-End Security", "Altern. URL", "Comment", "Compress", "Sign", "Encrypt", "Request MDN", "Sign MDN", "MDN Transfer", "Send requested asynchronous MDN", "Receipt Notification requested in X.400 messages should be", "Message Expiration", "X.400 Content Type", "Bodypart", "Encode binary data" and either the partner's "User ID" or the partner's valid X.400 address have to be defined in the CSV structure.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway (4)

- A successful creation of this relation entry (response to http(s) request POST) will be confirmed with “201 == Created” and the data stored in the database will be provided in the content of the response within a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request”, “404 == Not Found”, “409 == Conflict”, “413 == Payload Too Large” e.g., will be provided without data.
- If the request includes several relation entries, the data of each successful action will be provided within a CSV structure (including header line) and the http content will also show an acknowledgment for each action (e.g., 

New AS2 Relation created ID#1;10400#59; cn#1;erster tester#59; g#1;erster#59; s#1;tester#59; o#1;test-ag#59; n-id#1;2010400#59; a#1;VIAT-TEST#59; c#1;DE

AS2 ID of X.400 Partner: TEST111

 or 

Invalid data in Session. X400 User Id: 1041

). In case of a syntax error (e.g., missing delimiter) an I/O error will be displayed, and additional information will be available.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway (5)

- Be aware that the creation of the AS2 relation assigned to the central EDI function (assigning the AS2 ID) is only possible in GUI of WebConfig but not within Web Service.
- With the http(s) request PUT to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/AS2X400” and a content with CSV structure, a header line and valid values for the field identifiers would modify the AS2 properties. In opposite to GUI of WebConfig this would not modify existing AS2 relations.
- The http(s) request PUT to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/AS2X400\_PS/<User ID>” (<User ID> is own User ID resp. partner’s User ID) and a content with CSV structure, a header line and valid values for the field identifiers would modify the AS2 relation.
- Be aware that the http(s) request PUT only modifies the values of the parameter in the AS2 relation, but it would not modify the partner’s User ID or the X.400 address.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### AS2 Gateway (6)

- A successfully processed http(s) request PUT will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.
- To delete an AS2 relation entry the content of the http(s) request DELETE send to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/ AS2X400\_PS/<User ID>” must have a CSV structure with valid data of that entry. It is recommended to download this data using the http(s) request GET before initiating this deletion.
- A successfully deletion of this entry will be confirmed with “200 == OK” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### MessageGate File Interface API Reference (1)

Method	Request	Description
List Properties	GET */ConfigData/<User ID>**/MessageGate Output CSV structure with header line	Show MessageGate properties
List all relations	GET */ConfigData/<User ID>**/MessageGate_PS Output CSV structure with header line	Show all MessageGate relations
List User-ID	GET */ConfigData/<User ID>**/MessageGate_PS/<User ID> Output CSV structure with header line	Show MessageGate relation of this partner's User ID
Create relation	POST */ConfigData/<User ID>**/MessageGate_PS Input CSV structure with header line, Output CSV structure with header line	Create MessageGate relation(s)

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### MessageGate File Interface API Reference (2)

Method	Request	Description
Modify properties	PUT */ConfigData/<User ID>**/MessageGate Input CSV structure with header line, Output CSV structure with header line	Modify MessageGate properties
Modify relation	PUT */ConfigData/<User ID>**/MessageGate_PS/<User ID> Input CSV structure with header line, Output CSV structure with header line	Modify MessageGate relation
Delete relation	DELETE */ConfigData/<User ID>**/MessageGate_PS/<User ID> Input CSV structure with header line, Output CSV structure with header line	Delete MessageGate relation

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### MessageGate File Interface (1)

- For the management of the properties of a MessageGate File Interface account you have the methods GET (Show) and PUT (Modify) and for the management of the relation entries the methods GET (Show/List ), POST (Create), PUT (Modify) and DELETE (Delete).
- The http(s) request GET to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/ MessageGate” without any additional header information will provide a CSV structure, where a comma is the separator, the field identifiers and the values will be in English, and the character set is ISO8859-1. The content will show the properties of the account. To receive the data of all relations configured for this account use the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/ MessageGate\_PS”.
- A successfully processed http(s) request GET will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### MessageGate File Interface (2)

- The MessageGate properties may have values for the following field identifiers (en-US) :  
"MGUser ID", "When sending X.400 Messages map a requested Notification into", "Receipt Notification requested in X.400 messages should be", "Message Expiration", "X.400 Content Type", "Bodypart", "Encode binary data", "Comment"
- The entry of a MessageGate relation may have values for the following field identifiers (en-US) :  
"MGUser ID", "When sending X.400 Messages map a requested Notification into", "Receipt Notification requested in X.400 messages should be", "Message Expiration", "X.400 Content Type", "Bodypart", "Encode binary data", "Comment", "User ID", "Country (c)", "ADMD (a)", "PRMD (p)", "X.121", "UA ID", "Commonname", "Given name", "Surname", "Generation", "Initials", "Organization", "Org Unit 1", "Org Unit 2", "Org Unit 3", "Org Unit 4", "Type1", "Value1", "Type2", "Value2", "Type3", "Value3", "Type4", "Value4"
- While adding the User ID of an X.400 partner to the URL the content will show the data of the relation configured for this User ID (.../MessageGate\_PS/<User ID>).



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### MessageGate File Interface (3)

- The http(s) request POST to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/MessageGate\_PS” and a content with CSV structure, a header line and valid values for the field identifiers would create a new MessageGate relation entry or several within one task.
- Apart a valid value for the field identifier "MGUser ID", at minimum the valid values for "When sending X.400 Messages map a requested Notification into", "Receipt Notification requested in X.400 messages should be", "Message Expiration", "X.400 Content Type", "Bodypart", "Encode binary data " and either the partner's "User ID" or the partner's valid X.400 address must be defined in the CSV structure.
- A successfully creation (response to http(s) request POST) of this entry will be confirmed with “201 == Created” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request”, “404 == Not Found”, “409 == Conflict”, “413 == Payload Too Large” e.g., will be provided without data.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### MessageGate File Interface (4)

- If the request includes several relation entries, the data of each successful action will be provided within a CSV structure (including header line) and the http content will also show an acknowledgment for each action (e.g.<p>New MessageGate Relation created with ID#61;10400#59; cn#61;erster tester#59; g#61;erster#59; s#61;tester#59; o#61;test-ag#59; n-id#61;2010400#59; a#61;VIAT-TEST#59; c#61;DE</p> or <p>X.400 Address does not exist. X400 User Id: 49640</p>). In case of a syntax error (e.g., missing delimiter) an I/O error will be displayed, and additional information will be available.
- The http(s) request PUT to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/MessageGate” and a content with CSV structure, a header line and valid values for the field identifiers would modify the MessageGate properties .
- The http(s) request PUT to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/MessageGate\_PS/<User ID>” (<User ID> is own User ID resp. partner’s User ID) and a content with CSV structure, a header line and valid values for the field identifiers would modify the MessageGate relation.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### MessageGate File Interface (5)

- Be aware that the http(s) request PUT only modifies the values of the parameter in the MessageGate relation, but it would not modify the partner's User ID or the X.400 address.
- A successfully processed http(s) request PUT will be confirmed with "200 == OK". Otherwise, a negative acknowledgement "400 == Bad Request" or "404 == Not Found" will be provided without data.
- To delete a MessageGate relation entry the content of the http(s) request DELETE send to the URL "https://webconfig.viat.de/webconfig/ConfigData/<User ID>/ MessageGate \_PS/<User ID>" must have a CSV structure with valid data of that entry stored. It is recommended to download this data using the http(s) request GET before initiating this deletion.
- A successfully deletion of this entry will be confirmed with "200 == OK" and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement "400 == Bad Request" or "404 == Not Found" will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA API Reference (1)

Method	Request	Description
List Properties	GET */ConfigData/<User ID>**/MessageGatePMDF Output CSV structure with header line	Show SMTP MTA properties
List all relations	GET */ConfigData/<User ID>**/MessageGatePMDF_PS Output CSV structure with header line	Show all SMTP MTA relations
List User-ID	GET */ConfigData/<User ID>**/MessageGatePMDF_PS/<User ID> Output CSV structure with header line	Show SMTP MTA relation of this partner's User ID
Create relation	POST */ConfigData/<User ID>**/MessageGatePMDF_PS Input CSV structure with header line, Output CSV structure with header line	Create SMTP MTA relation(s)

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA API Reference (2)

Method	Request	Description
Modify properties	PUT */ConfigData/<User ID>**/MessageGatePMDF Input CSV structure with header line, Output CSV structure with header line	Modify SMTP MTA properties
Modify relation	PUT */ConfigData/<User ID>**/MessageGatePMDF/<User ID> Input CSV structure with header line, Output CSV structure with header line	Modify SMTP MTA relation
Delete relation	DELETE */ConfigData/<User ID>**/MessageGatePMDF/<User ID> Input CSV structure with header line, Output CSV structure with header line	Delete SMTP MTA relation

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA (1)

- For the management of the properties of a SMTP (MTA) account you have the methods GET (Show) and PUT (Modify) and for the management of the relation entries the methods GET (Show/List ), POST (Create), PUT (Modify) and DELETE (Delete).
- The http(s) request GET to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/MessageGatePMDF” without any additional header information will provide a CSV structure, where a comma is the separator, the field identifiers and the values will be in English, and the character set is ISO8859-1. The content will show the properties of the account.  
To receive the data of all the Message-GatePMDF relations configured for this account use the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/ MessageGatePMDF\_PS”.
- A successfully processed http(s) request GET will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA (2)

- The SMTP (MTA) properties may have values for the following field identifiers (en-US) :  
"MGPMDFUser ID", "If MDN requested in SMTP message request a", "If DN/NDN requested in X.400 message", "If RN/NRN requested in X.400 message", "Message Expiration", "X.400 Content Type", "Bodypart", "Mapping into SMTP address", "Encode binary data", "Comment"
- The entry of a SMTP (MTA) relation may have values for the following field identifiers (en-US) :  
"MGPMDFUser ID", "If MDN requested in SMTP message request a", "If DN/NDN requested in X.400 message", "If RN/NRN requested in X.400 message", "Message Expiration", "X.400 Content Type", "Bodypart", "Mapping into SMTP address", "Comment", "User ID", "Country (c)", "ADMD (a)", "PRMD (p)", "X.121", "UA ID", "Commonname", "Given name", "Surname", "Generation", "Initials", "Organization", "Org Unit 1", "Org Unit 2", "Org Unit 3", "Org Unit 4", "Type1", "Value1", "Type2", "Value2", "Type3", "Value3", "Type4", "Value4"

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA (3)

- While adding the User ID of an X.400 partner to the URL the content will show the data of the relation configured for this User ID (.../MessageGatePMDF\_PS/<User ID>).
- The http(s) request POST to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/MessageGatePMDF\_PS” and a content with CSV structure, a header line and valid values for the field identifiers would create a new SMTP (MTA) relation entry or several within one task.
- Apart a valid value for the field identifier "MGPMDFUser ID", at minimum the valid values for "If MDN requested in SMTP message request a", "If DN/NDN requested in X.400 message", "If RN/NRN requested in X.400 message", "Message Expiration", "X.400 Content Type", "Bodypart", "Mapping into SMTP address" and either the partner's "User ID" or the partner's valid X.400 address must be defined in the CSV structure.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA (4)

- A successful creation of this entry (response to http(s) request POST) will be confirmed with “201 == Created” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request”, “404 == Not Found”, “409 == Conflict”, “413 == Payload Too Large” e.g., will be provided without data.
- If the request includes several relation entries, the data of each successful action will be provided within a CSV structure (including header line) and the http content will also show an acknowledgment for each action (e.g., 

New MessageGate SMTP Relation created with ID#61;10400#59; cn#61;erster tester#59; g#61;erster#59; s#61;tester#59; o#61;test-ag#59; n-id#61;2010400#59; a#61;VIAT-TEST#59; c#61;DE

 or 

X.400 Address does not exist. X400 User Id: 60036

). In case of a syntax error (e.g., missing delimiter) an I/O error will be displayed, and additional information will be available.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA (5)

- The http(s) request PUT to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/MessageGatePMDF” and a content with CSV structure, a header line and valid values for the field identifiers would modify the SMTP (MTA) properties .
- The http(s) request PUT to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/MessageGatePMDF\_PS/<User ID>” (<User ID> is own User ID resp. partner’s User ID) and a content with CSV structure, a header line and valid values for the field identifiers would modify this SMTP (MTA) relation.
- Be aware that the http(s) request PUT only modifies the values of the parameter in the SMTP (MTA) relation, but it would not modify the partner’s User ID or the X.400 address.
- A successfully processed http(s) request PUT will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP MTA (6)

- To delete a SMTP (MTA) relation entry the content of the http(s) request DELETE send to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/ MessageGate \_PS/<User ID>” must have a CSV structure with valid data of that stored entry. It is recommended to download this data using the http(s) request GET before initiating this deletion.
- A successfully deletion of this entry will be confirmed with “200 == OK” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP Filter API Reference (1)

Method	Request	Description
List status/ filter entries	GET */ConfigData/<User ID>** /SMTPFilter Output CSV structure with header line	Show SMTP filter status and entries
Create filter entry	POST */ConfigData/<User ID>** /SMTPFilter Input CSV structure with header line, Output CSV structure with header line	Create SMTP filter entry/entries
Modify status	PUT */ConfigData/<User ID>** /SMTPFilter Input CSV structure with header line, Output CSV structure with header line	Modify SMTP filter status

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP Filter API Reference (2)

Method	Request	Description
Delete filter entry	DELETE */ConfigData/<User ID>** /SMTPFilter Input CSV structure with header line, Output CSV structure with header line	Delete SMTP filter entry

\* <https://webconfig.viat.de/webconfig>

\*\* optional



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP Gateway Filter (1)

- This description is valid for all X.400 (mailbox) accounts (Classic X.400 Mailbox, AS2, MessageGate File Interface), but not for domains accounts used for SMTP MTA.
- For the management of the SMTP Gateway filters you have the methods GET (Show/List ), POST (Create), PUT (Modify) and DELETE (Delete). SMTP filter entries will be only used if the status of the filter functionality is set to “partly disabled”.
- The SMTP filter entries are no real relations so there are some specialties to be aware. The method PUT (Modify) is not available for the management of the SMTP filter entries. This method is only used to modify the status of the filter functionality.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP Gateway Filter (2)

- The http(s) request GET to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/SMTPFilter” without any additional header information will provide a CSV structure, where a comma is the separator, the field identifiers and the values will be in English, and the character set is ISO8859-1. The content will show the status of the SMTP filter functionality and all SMTP filter entries.
- The entry of a SMTP Gateway Filter may have values for the following field identifiers (en-US) :  
"SMTPUser ID","Filter Status","Valid SMTP Originator"
- The first line below the header line will show the status of the filter functionality (enabled, disabled, partly disabled). Only in that case that the status is "partly disabled" the filter entries showed in the next lines will be used.
- A successfully processed http(s) request GET will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP Gateway Filter (3)

- The http(s) request POST to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/SMTPFilter” and a content with CSV structure, a header line and valid values for the field identifiers would create a new SMTP filter entry or several within one task.
- Apart a valid value for the field identifier "SMTPUser ID", the valid value for "Valid SMTP Originator" must be defined in the CSV structure. This might be a complete Email address (including @) or only the name of an Internet Domain (without @).
- A successfully creation of this entry (response to http(s) request POST) will be confirmed with “201 == Created” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request”, “404 == Not Found”, “409 == Conflict”, “413 == Payload Too Large” e.g., will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP Gateway Filter (4)

- If the request includes several filter entries, the data of each successful action will be provided within a CSV structure (including header line) and the http content will also show an acknowledgment for each action (e.g., <p>Add new SMTP Originator Address: telekom.de</p> or <p>SMTP Originator address already exists. SMTP Originator: loosen.net</p>). In case of a syntax error (e.g., missing delimiter) an I/O error will be displayed, and additional information will be available.
- The http(s) request PUT to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/SMTPFilter” and a content with CSV structure, a header line and a valid value (enabled, disabled, partly disabled) would modify the status of SMTP filter. Only in that case that the status is "partly disabled" the filter entries showed in the next lines will be used.
- A successfully processed http(s) request PUT will be confirmed with “200 == OK”. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.



# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### SMTP Gateway Filter (5)

- To delete a SMTP filter entry the content of the http(s) request DELETE send to the URL “https://webconfig.viat.de/webconfig/ConfigData/<User ID>/SMTPFilter” must have a CSV structure with a valid address.
- A successfully deletion of this entry will be confirmed with “200 == OK” and the data stored in the database will be provided in the content of the response in a CSV structure. Otherwise, a negative acknowledgement “400 == Bad Request” or “404 == Not Found” will be provided without data.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### Examples using CURL (1)

Here are some examples how to use the command line-based module “Curl” \* to call the Web Service. To reduce the length of a line the (optional) own User ID is not printed. The Web Service would use the User ID sent in certificate to identify the mailbox.

#### All requests using default values (format csv-c and language en-US)

```
curl https://webconfig.viat.de/webconfig/ConfigData/EDI -cert EDItester_cert.pem -key EDItester_key.pem > GET_EDIBOX_ALL_EDI_ID.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/EDI -X POST -cert EDItester_cert.pem -key EDItester_key.pem -T NEW_EDIBOX_EDI_ID.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/EDI -X DELETE -cert EDItester_cert.pem -key EDItester_key.pem -T DEL_EDIBOX_EDI_ID.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/EDI_TP -cert EDItester_cert.pem -key EDItester_key.pem > GET_EDIBOX_TP.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/EDI_TP/xxxxx -cert EDItester_cert.pem -key EDItester_key.pem > GET_EDIBOX_TP_xxxxx.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/EDI_TP -X POST -cert EDItester_cert.pem -key EDItester_key.pem -T NEW_EDIBOX_TP_<User ID>_.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/EDI_TP/xxxxx -X DELETE -cert EDItester_cert.pem -key EDItester_key.pem -T DEL_EDIBOX_TP_xxxxx.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400 -cert as2tester_cert.pem -key as2tester_key.pem > GET_AS2_Base.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400_PS -cert as2tester_cert.pem -key as2tester_key.pem > GET_AS2_PS.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400_PS/xxxxx -cert as2tester_cert.pem -key as2tester_key.pem > GET_AS2_PS_xxxxx.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400_PS -X POST -cert as2tester_cert.pem -key as2tester_key.pem -T NEW_AS2_PS_xxxxx.txt -v
```

(-v provides additional information about the session; an alternative would be -i)

\* <https://curl.haxx.se/download.html>



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### Examples using CURL (2)

```
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400 -X PUT --cert as2tester_cert.pem --key as2tester_key.pem -T MOD_AS2_Base.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400_PS/xxxxx -X PUT --cert as2tester_cert.pem --key as2tester_key.pem -T MOD_AS2_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400_PS/xxxxx -X DELETE --cert as2tester_cert.pem --key as2tester_key.pem -T DEL_AS2_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGate --cert MGatetester_cert.pem --key MGatetester_key.pem > GET_MGate_Base.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGate_PS --cert MGatetester_cert.pem --key MGatetester_key.pem > GET_MGate_PS.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGate_PS/xxxxx --cert MGatetester_cert.pem --key MGatetester_key.pem > GET_MGate_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGate_PS -X POST --cert MGatetester_cert.pem --key MGatetester_key.pem -T NEW_MGate_PS_<User ID>_mod.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGate -X PUT --cert MGatetester_cert.pem --key MGatetester_key.pem -T MOD_MGate_Base.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGate_PS/xxxxx -X PUT --cert MGatetester_cert.pem --key MGatetester_key.pem -T MOD_MGate_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGate_PS/xxxxx -X DELETE --cert MGatetester_cert.pem --key MGatetester_key.pem -T DEL_MGate_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGatePMDF --cert PMDF_cert.pem --key PMDF_key.pem > GET_PMDf_Base.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGatePMDF_PS --cert PMDF_cert.pem --key PMDF_key.pem > GET_PMDf_PS.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGatePMDF_PS/xxxxx --cert PMDF_cert.pem --key PMDF_key.pem > GET_PMDf_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGatePMDF_PS -X POST --cert PMDF_Service_cert.pem --key PMDF_Service_key.pem -T NEW_PMDf_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGatePMDF -X PUT --cert PMDF_cert.pem --key PMDF_key.pem -T MOD_PMDf_Base.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGatePMDF_PS/xxxxx -X PUT --cert PMDF_cert.pem --key PMDF_key.pem -T MOD_PMDf_PS_XXXXX.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/MessageGatePMDF_PS/xxxxx -X DELETE --cert PMDF_cert.pem --key PMDF_key.pem -T DEL_PMDf_PS_XXXXX.txt -v
```

(- v provides additional information about the session; an alternative would be -i)



ERLEBEN, WAS VERBINDET.

# BUSINESSMAIL X.400

## WEBCONFIG WEB SERVICE

### Examples using CURL (3)

```
curl https://webconfig.viat.de/webconfig/ConfigData/SMTPFilter -cert MBtester_cert.pem -key MBtester_key.pem > GET_MBOX_SMTPFilter.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/SMTPFilter -X POST -cert MBtester_cert.pem -key MBtester_key.pem -T NEW_MBOX_SMTPFilter.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/SMTPFilter -X DELETE -cert MBtester_cert.pem -key MBtester_key.pem -T DEL_MBOX_SMTPFilter.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/SMTPFilter -X PUT -cert MBtester_cert.pem -key MBtester_key.pem -T MOD_MBOX_SMTPFilter_Status.txt -v
```

#### Requests using non default values (format csv-s and language de-DE)

```
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400 -H "accept:text/csv-s" -H "accept-language:de-DE" -cert as2tester_cert.pem -key as2tester_key.pem > GET_AS2_Base.txt -v
curl https://webconfig.viat.de/webconfig/ConfigData/AS2X400 -X PUT -H "content-type:text/csv-s" -H "content-language:de-DE" -cert as2tester_cert.pem -key as2tester_key.pem -T MOD_AS2_Base.txt -v
```

#### Request directly using the PKCS12 certificate file (not necessary to split it for example using OpenSSL into a certificate and key file)

```
curl https://webconfig.viat.de/webconfig/ConfigData/EDI -cert editester.p12:<export password> -cert-type p12 > GET_EDIBOX_ALL_EDI_ID.txt -v
```

(-v provides additional information about the session; an alternative would be -i)





**BUSINESSMAIL X.400**  
**THANK YOU!**



**ERLEBEN, WAS VERBINDET.**