

# ACS Rest API

## Table of Contents

SPECIFICATIONS .....	1
STEP 1 – INSOMNIA INSTALLATION (for tests) .....	3
STEP 2 – START INSOMNIA .....	3
SWAGGER UI .....	4
CREATE VOUCHER .....	4
CREATE VOUCHER WITH DELIVERY FROM ACS POINT .....	8
MULTIPART VOUCHERS .....	9
PRINT VOUCHER.....	9
DELETE VOUCHER.....	11
CONTENT OF CYPRUS SHIPMENTS .....	12
ISSUE PICKUP LIST .....	12
PRINT PICKUP LIST .....	14
PICKUP_LIST_DISPLAY_VOUCHER .....	15
GET PICKUP LISTS .....	15
ACS_COD_BENEFICIARY_INFO .....	16
TRACKING SUMMARY .....	17
TRACKING DETAILS .....	19
REFERENCE KEY FOR ACS VOUCHER CODE .....	20
PRICE CALCULATION .....	21
ADDRESS VALIDATION .....	22
FIND AREA BY ZIPCODE.....	23
ACS STATION CODE BASED ON ZIPCODE .....	24
ACS STATIONS .....	25
GENERATE CLIENT CODE .....	26

## SPECIFICATIONS

ACS Web Services is **REST API**, and this document describes the general methodology of the calls, the parameters, and the responses.

The server always receives a **POST** request and answers in **JSON** format.

For each customer there must be an API KEY which is given by ACS.

The server always returns **200 OK** unless the API Key is wrong, or it is not sent at all by the client and returns **403 Forbidden** and **406 Not Acceptable** in case the client exceeds the default maximum number of simultaneous calls per second (default 10 calls).

In all the requests there is a parameter called **ACSAlias** and the procedure for the call. It is the minimum required parameter and the POST format of JSON is

```
{ACSAlias: '<MyInterfaceToCallName>'}
```

In order the request to be success:

1. The customer must have the ACS API KEY.
2. The customer must register this API KEY in the header of the HTTPS request named ACSApiKey.
3. The specific API KEY has rights to the call process i.e., ACSAlias.
4. If there is an IP lock, the IP being called to be whitelisted.

The root URL for all the requests is the same

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

There are the following controllers in the above URL

**A) ACSAutoRestHelp**

1. **GET** with the **ACSAlias** ('<MyInterfaceToCallName>') and returns a description of the process in text.
2. **POST** with the **ACSAlias** ({ACSAlias: '<MyInterfaceToCallName>'}) and returns the **JSON** which is required.

**B) ACSAutoRest**

**POST** with the **ACSAlias** and the required parameters in JSON (as the A2 return to us) and returns as JSON the results of the execution of the specific process according to the parameters we inserted.

The **JSON** is like

```
{
  "ACSAlias": "<MyInterfaceToCallName>",
  "ACSInputParameters": {
    "InParam-1": Param_1_Value,
    "InParam-2": Param_2_Value,
    .
    .
    "InParam-n": Param_n_Value
  }
}
```

All the responses are of the form:

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    <ACS OUTPUT RESPONSE DATA>
  }
}
```

Where:

**ACSExecution\_HasError** → **false** successful execution, **true** unsuccessful execution

**ACSExecutionErrorMessage** → has a description of the execution problem and has a value different from the void only if **ACSExecution\_HasError** is true (details are shown below in section errors).

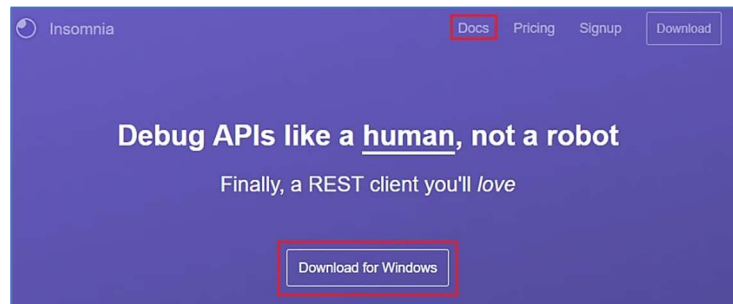
**ACSOutputResponse** has the results of execution and has values only when **ACSExecution\_HasError** → **false**.

**ACSOutputResponse** can contain simple arguments with values or even entire records that are repeated in lists.

If there is a return argument called **Return\_Data**, it can be deserialized to **.NET DataSet**.

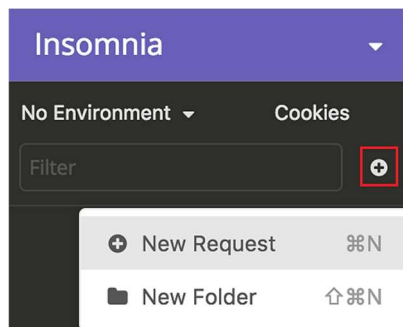
## STEP 1 – INSOMNIA INSTALLATION (for tests)

For **tests only** it is recommended to use the Insomnia Rest Client that can be downloaded from <https://insomnia.rest> . From the home screen select download for windows (at top right choose docs for installation instructions for various operating systems).



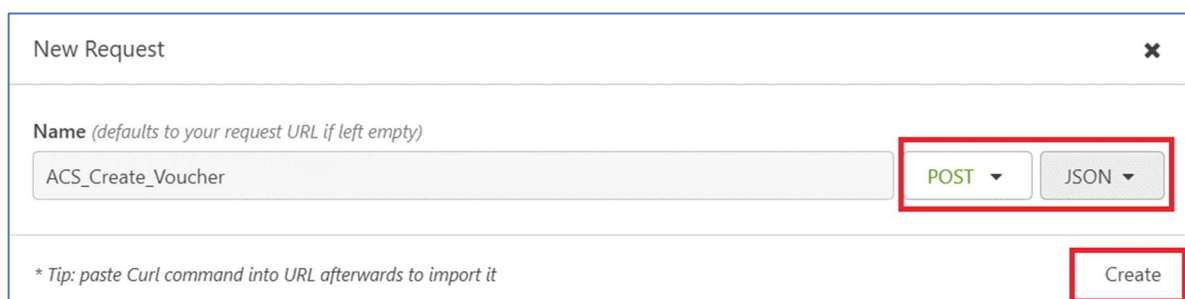
## STEP 2 – START INSOMNIA

To create the first request, click the + icon at the top of the sidebar and select **New Request**.

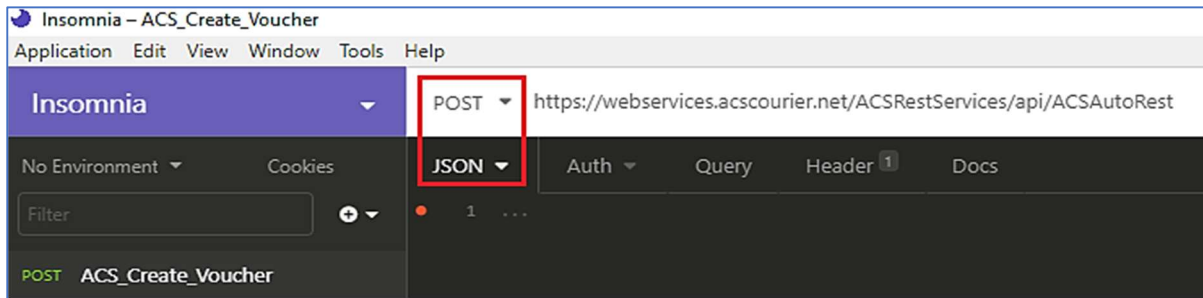


Then, and depending on the tool we will deal with, we will be asked to give a name. In our example we give the name **ACS\_Create\_Voucher**.

Then we select the desired HTTPS method (for ACS tools we always choose GET). Finally, we run **Create**.



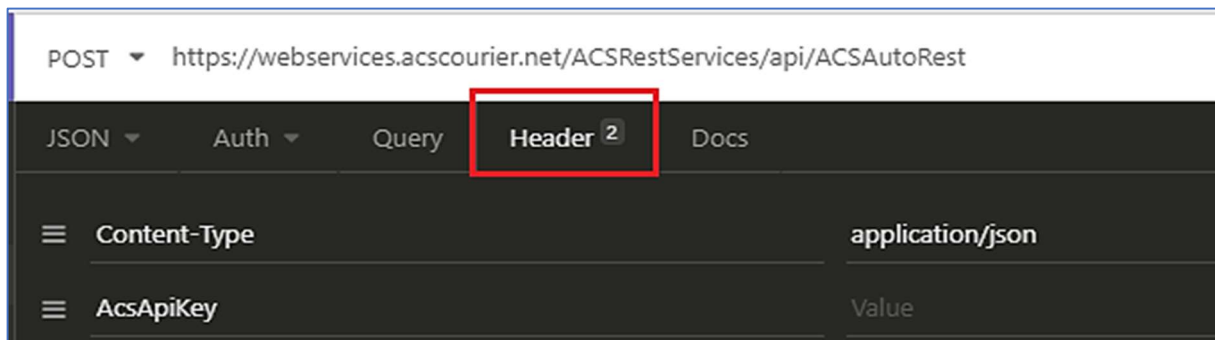
Once the request has been created, we have the screen:



We select the **POST** method, as body **JSON** and define the URL

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

On the **HEADER** tab, we see that the content type already exists. In the new header we enter the wording **AcsApiKey** and in the value field the key given by ACS. We are not dealing with other tabs (Auth, Query, Docs).



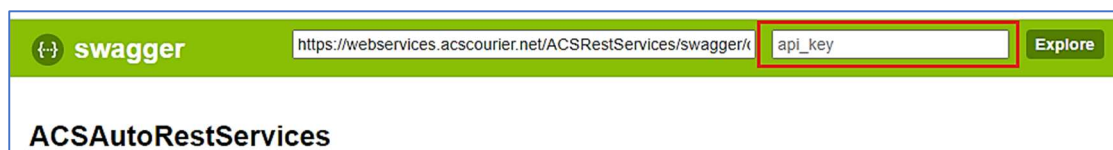
## SWAGGER UI

Swagger UI has been implemented for the rest web services of our company.

The url is the well-known webservices followed by / swagger / i.e.

<https://webservices.acscourier.net/ACSRestServices/swagger/>

To perform a method with the "Try It Out" button you must insert the required ApiKey on the top right.



## CREATE VOUCHER

The method creates the ACS vouchers for dispatches within Greece or Cyprus (Xpress or with EC - ACS CYPRUS ECONOMY). **It does not support creating vouchers to other countries.**

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body :

```
{
  "ACSAlias": "ACS_Create_Voucher"
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response :

Request	Response (the shipping code in yellow)
<pre>{   "ACSAlias": "ACS_Create_Voucher",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Pickup_Date": '2019-01-10',     "Sender": 'ESHOP',     "Recipient_Name": 'TEST RECIPIENT',     "Recipient_Address": 'P. RALLI',     "Recipient_Address_Number": 45,     "Recipient_Zipcode": 17778,     "Recipient_Region": 'TAVROS',     "Recipient_Phone": 2115005000,     "Recipient_Cell_Phone": 6999999999,     "Recipient_Floor": null,     "Recipient_Company_Name": null,     "Recipient_Country": 'GR',     "Acs_Station_Destination": null,     "Acs_Station_Branch_Destination": 1,     "Billing_Code": '2A0999999',     "Charge_Type": 2,     "Cost_Center_Code": null,     "Item_Quantity": 1,     "Weight": 0.5,     "Dimension_X_In_Cm": null,     "Dimension_Y_in_Cm": null,     "Dimension_Z_in_Cm": null,     "Cod_Ammount": 50.5,     "Cod_Payment_Way": 0,     "Acs_Delivery_Products": 'COD',     "Insurance_Ammount": null,     "Delivery_Notes": null,     "Appointment_Until_Time": null,     "Recipient_Email": null,     "Reference_Key1": null,     "Reference_Key2": null,     "With_Return_Voucher": null,     "Content_Type_ID": null,     "Language": 'EN'   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "Voucher_No": " 7227889174",         "Voucher_No_Return": null,         "Error_Message": ""       }     ],     "ACSTableOutput": {}   } }</pre>

### Explanation of parameters

"Company_ID"	unique code given by ACS
"Company_Password"	unique code given by ACS
"User_ID"	unique code given by ACS
"User_Password"	unique code given by ACS

"Pickup_Date"	the date of receipt of the shipments by the courier. Register as "YYYY-MM-DD"
"Sender"	The Sender's name
"Recipient_Name"	The Recipient's Name
"Recipient_Address"	The Recipient's Address <b>without the area/region</b>
"Recipient_Address_Number"	The Recipient's Address Number
"Recipient_Zipcode"	The Recipient's zip code
"Recipient_Region"	The Recipient's region/ area
"Recipient_Phone"	The Recipient's phone number
"Recipient_Cell_Phone"	The Recipient's mobile number
"Recipient_Floor"	The Recipient's floor
"Recipient_Company_Name"	The brand name of the recipient company (if any)
"Recipient_Country"	the country of destination. For Greece = GR, For Cyprus = CY
"Acs_Station_Destination"	the ACS store that will perform the delivery ...value null
"Acs_Station_Branch_Destination"	acceptable values "0" or "1"
"Billing_Code"	The ACS credit code
"Charge_Type"	For sender's charge "2", for recipient's "4"
"Cost_Center_Code"	The Cost Center Code (if any)
"Item_Quantity"	if >1 multiple shipment
"Weight"	min value 0,5 kg.
"Dimension_X_In_Cm"	Dimension_X_In_Cm
"Dimension_Y_in_Cm"	Dimension_X_In_Cm
"Dimension_Z_in_Cm"	Dimension_Z_in_Cm
"Cod_Ammount"	If there is no value, then null
"Cod_Payment_Way"	For cash, the value "0", for check "1". If no COD, then null
"Acs_Delivery_Products"	<p>Additional products with codes (<b>*They may incur extra costs</b>):</p> <p>INS Insurance Shipment*</p> <p>SAT: Delivery Saturday *</p> <p>MDV Morning Delivery*</p> <p>TDD 2-hour Delivery Appointment*</p> <p>COD: COD*</p> <p>RDO: Return Documents*</p> <p>REM: Remote Area*</p> <p>PRO: Protocol*</p> <p>REC: Reception Delivery (store pick up)</p> <p>CEC Cyprus Economy</p> <p><b>P2P point to point</b></p>

	D2P      door to point P2D      point to door Reds are only for shipments to Cyprus
"Insurance_Amount"	Max value €3000 otherwise null
"Delivery_Notes"	Free text
"Appointment_Until_Time"	2 hours' time frame (HH:MM). Register as "hh:mm". <b>It may incur extra cost</b>
"Recipient_Email"	The Recipient's email
"Reference_Key1"	The Sender's Reference Key 1 (if any)
"Reference_Key2"	The Sender's Reference Key 2 (if any)
"With_Return_Voucher"	'0' without return voucher or "1" for return voucher
"Language"	For Greek null or GR otherwise EN
"Content_Type_ID": null	Mandatory if the country of destination is Cyprus CY

#### Notes:

1. To finalize the printed vouchers, it is essential to create and print the day receipt list (Issue\_Pickup\_List in the documentation). If the pickup list is not created, the vouchers' barcodes will not be recognized.
2. Not all phones are required, either fixed or mobile. It is preferable to have the mobile phone to be able to communicate automatically via sms-viber with the recipient, for example, after an unattended note.
3. Decimal in the fields **Cod Amount - Weight - Insurance Amount** with .
4. Amounts in the fields **Cod\_Ammount - Weight - Insurance\_Ammount - Cod\_Payment\_Way - Charge\_Type - Item\_Quantity** without quotes (single or double).
5. If in **Item\_Quantity**, the value > 1 then multiple vouchers will be generated. To export the numbers (if we want it), the **Multipart\_Vouchers** method described in this documentation needs to be used.
6. If the 3 dimensions are entered, the tool does not perform a volumetric calculation. The voucher always shows the weight of the **Weight** field.
7. For security amounts >€3000, there needs to be a business insurance business agreement.
8. **With\_Return\_Voucher** does not refer to any merchant return policy (e.g., replacements) but to the Document Return Service, in which the courier delivers and at the same time takes something back. Print and the Return voucher is only available in thermal prints (see **Print Voucher** section in the documentation).
9. If we want to register more than 1 product, we register them with a comma, e.g., '**COD, SAT**' or '**SAT, COD**', i.e., the order of entry does not play a role.
10. If the shipment is to be delivered to ACS Smart Point, you need to fill in the field **"Acs\_Station\_Destination"** the code of the area where the desired Smart Point is located e.g., ΑΔ and in the field **"Acs\_Station\_Branch\_Destination"** the code of the Smart Point e.g. 401. This information is provided by the **"Acs\_Stations"** method (see page 26).

#### Error messages (Can be changed in the future without notice):

- **Invalid pick-up date** - when the pick-up date is wrong, for example past date.
- **Pickup date is not allowed on Sunday or national holiday** - when the pick-up date is Sunday or holiday
- **The recipient's name cannot be empty** - if the recipient's name is empty.
- **The recipient's address cannot be empty** - if there is no delivery address.

- **Unacceptable zip code or country** - if the zip code is incorrect or does not match the Recipient\_Country field (e.g., there is GR, but the zip code is Cypriot or vice versa).
- **Unacceptable ACS Destination Value** - if the initial delivery letters are not correct (if this information is entered in the Acs\_Station\_Destination field) and / or if the value in the Acs\_Station\_Branch\_Destination field is incorrect.
- **Not more than 99 pieces per shipment are supported** - if the pieces are of great value by mistake.
- **Unacceptable weight value (0.5-999)** - if by mistake the weight is too large.
- **Unacceptable shipping charge value** - if the value in the Charge\_Type field is not 2 or 4.
- **Unacceptable payment method for COD** - if the method of payment for the COD is not 0 for cash or 1 for check.
- **Cannot find the COD** - if there is a COD amount and payment method but there is no COD service in the Acs\_Delivery\_Products field.
- **No valid credit code** - if the Billing Code is incorrect.
- **The Acs-SmartPoint destination must have 1 mobile phone** - if the delivery point is Smart Point and no mobile phone is registered in the Recipient\_Cellphone field.
- **For an Acs-SmartPoint destination with cash on delivery (COD) there must be a recipient email** - if the delivery point is Smart Point locker and no email is registered in the Recipient\_Email field.
- **Acs-SmartPoint destination can not be combined with multiple shipment (items>1)** - if the delivery point is Smart Point and the Item\_Quantity is > 1.
- **The product "RVO" is only combined with return voucher (with\_return = 1)** - if the RVO is entered in the Acs\_Delivery\_Products field but there is no '1' in the With\_Return\_Voucher field.
- **You cannot create Cyprus Economy (EC) shipments with the billing code: 2XXXXXXX** – if the billing code is not for the product EC- Cyprus Economy.
- **You cannot combine these products together** - if the registered products cannot be combined e.g., SAR, REC.
- **The product 5Σ is not supported for this destination.** the product SAT (Saturday delivery) is not supported for that destination.
- If time is entered in the field **"Appointment\_Until\_Time"** then in the shipment will be automatically added the product of the Morning delivery (if the registered time until 10:00) or 2-hour definite delivery (if the registered time > 10:00).
- **For shipments from Greece to Cyprus the shipment Content\_Type\_ID field must have a valid value.** If the country of destination is Cyprus (CY) and no content code has been entered.

## CREATE VOUCHER WITH DELIVERY FROM ACS POINT

The consignee may wish to collect his shipment from an ACS Point. Such are the central stores of ACS as well as all ACS Smartpoints with Parcel Locker or without Parcel Locker.

To find the data of all these ACS Points, call the **"Acs\_Stations"** method (see information on p. 25).

## CREATE SHIPMENT WITH DELIVERY FROM ACS STORE

We rely on the **"ACS\_Create\_Voucher"** method above and in the **"Acs\_Delivery\_Products"** field we enter the value **"REC"**.

If there is also a cash on delivery, we enter **"REC, COD"** or **"COD, REC"** ...i.e.. the order doesn't matter.

## CREATE SHIPMENT WITH DELIVERY FROM ACS SMARTPOINT

To create a shipment to a Smartpoint with or without a parcel locker, what is needed is in the 2 fields **"Acs\_Station\_Destination"** & **"Acs\_Station\_Branch\_Destination"** of the **"ACS\_Create\_Voucher"** method to enter the code of the desired Smartpoint by the recipient (you can find the code from the **"Acs\_Stations"** method )

Example

**"Acs\_Station\_Destination": 'AK',**

**"Acs\_Station\_Branch\_Destination": 502,**



## MULTIPART VOUCHERS

If the **Item\_Quantity** of the Create Voucher is **>1** then there will be a multiple shipment. The multiple shipment consists of 1 main voucher & the sub vouchers. It is used when the contents of one shipment do not fit into 1 parcel and must be divided.

In this case all the items must be delivered at the same time. The sender is charged as 1 shipment.

To see these numbers (if you want) the method needs to be used called **Multipart\_Vouchers**.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```
{
  "ACSAlias": "ACS_Get_Multipart_Vouchers"
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_Get_Multipart_Vouchers",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": 'EN',     "Main_Voucher_No": 7227890681   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "Error_Message": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "MultiPart_Voucher_No": "8052453426"         }       ]     }   } }</pre>

In our example we have the main Voucher **7227890681** in which we have set Item\_Quantity = 2 in the Create Voucher. The method gives us the voucher code of the accompanying voucher **8052453426**.

## PRINT VOUCHER

Used to create and print vouchers in PDF format **depending on the printer that the customer has (laser or thermal)**.

**Vouchers must be printed before the Pickup list is printed, otherwise it is not possible to print them.**

demo request

```
{
  "ACSAlias": "ACS_Print_Voucher",
  "ACSInputParameters": {
```

```

"Company_ID": 'demo',
"Company_Password": 'demo',
"User_ID": 'demo',
"User_Password": 'demo',
"Voucher_No": 'XXXXXXXXXX',
"Print_Type": 2,
"Start_Position": 1
}
}

```

To deserialize the json in the ACSObjectOutput you get as a response, each line returned is a dictionary with key the voucher and value the byte array that the pdf has.

#### Notes:

- For laser printing (A4 page) in the field "Print\_Type" the value is 2 while for thermal the value is 1.
- The "Start\_Position" field specifies the position of the label on page A4. Acceptable values 1,2 or 3. It only makes sense in the case of laser Printing (Print\_Type = 2).
- Up to 10 vouchers can be printed simultaneously by entering the vouchers id in the "Voucher\_No" field separated by a comma.
- If the shipment has a return voucher, in create voucher the field With\_Return\_Voucher = 1, in the field Voucher\_No only the main voucher is registered but it is returned to the print and the main and the return.
- If in Create Voucher in the Item\_Quantity field the value was > 1 then in the print will appear the main voucher and the accompanying ones.

#### Sample Vouchers

- Laser label (A4 page with 3 labels per page)

<b>ACS</b> Τα πάντα, παντού.		<b>ΑΡΙΘ. ΑΠΟΣΤΟΛΗΣ</b> <b>7400272003</b>		ΕΔΡΑ: ΑΣΚΛΗΠΙΟΥ 25 Τ.Κ. 14568 ΚΡΗΝΟΤΕΡΙ ΑΤΤΙΚΗΣ ΑΦΜ: 094360202		DEMO ACS CONNECT ΤΚ: 14565 ΔΟΥ: ΦΑΕ ΑΘΗΝΩΝ	
<b>ΑΠΟΣΤΟΛΕΑΣ</b> ΟΝΟΜΑ: TEST ΔΙΕΥ/ΝΣΗ-ΥΠ/ΜΑ: ΑΣΚΛΗΠΙΟΥ 25 ΤΚ / ΠΟΛΗ: 14565 / ΑΘΗΝΑ ΤΗΛ. / ΚΙΝ.: 8190000				ΚΩΔ. ΠΕΛΑΤΗ: 2ΑΘ999999 ΗΜΕΡΟΜΗΝΙΑ: 14/06/2021 ΒΑΡΟΣ (ΚΓ): 1,00 ΤΕΜΑΧΙΑ: 1 ΠΟΣΟ ΑΝΤ/ΛΗΣ: ΠΑΡΑΔΟΣΗ ΜΕΤΑΞΥ: ΚΥΡΙΟ ΑΝΩ: ΟΔΗΓΙΕΣ:			
<b>ΠΑΡΑΛΗΠΤΗΣ</b> ΟΝΟΜΑ: TEST RECIPIENT ΕΤΑΙΡΙΑ / ΟΡΟΦΟΣ: P.RALLI 45 ΟΔΟΣ/ΑΡ.: 17778 / ΤΑΥΡΟΣ ΤΚ / ΠΟΛΗ: 2108190000 ΤΗΛ. / ΚΙΝ.: 2108190000				ΠΑΡΑΤΗΡΗΣΕΙΣ:			
ACS Connect ACS H.A.E.E. - Ασκληπιού 25 Τ.Κ. 14568 Κρηνοτέρη Αττικής ΑΦΜ: 094360202 ΔΟΥ: Φ.Α.Ε. ΑΘΗΝΩΝ, Τηλ: 210-8190000, Fax: 210-8190061				ΕΕΤΤ Α.Μ. : 99-122 Εδωξέ & Γενική Άδεια Ταχυδρομικών Υπηρεσιών			
				<b>ACS</b> WebBusinessTools Τα πάντα, παντού.		<b>ΠΡΟΟΡΙΣΜΟΣ</b>  <b>BT</b>	

\*Οι χρονικές προτιμήσεις παράδοσης δεν δεσμεύουν την ACS εάν δεν συνδυάζονται με την αντίστοιχη επιλογή υπηρεσίας.

- Thermal label (roll)

Αρ. Αποστολής/AWB Nr <b>7227889174</b>	Προέλευση/Origin ΑΘ	Τεμ./Pieces 1	Βάρος/Weight 0,50	Κωδ.Πελ./Cust Code 2ΑΘ9999999	Ημερομηνία/Date 10/01/2019	Courier	Ώρα/Time 13:43	Όνομα Ν/Υ/PC ACSIIS19
<b>ΑΠΟΣΤΟΛΕΑΣ / SENDER</b> Ονομα: ESHOP Διεύθυνση: ΑΣΚΛΗΠΙΟΥ 25 Τ.Κ. 14565 Πόλη: ΑΘΗΝΑ Τηλ. 8190000 Επάγγελμα: ΤΑΧΥΜΕΤΑΦΟΡΕΣ Δ.Ο.Υ. ΜΕΓΑΛΩΝ ΕΠΙ Profession: ΕΓΓΡΑΦΩΝ & ΔΕΜΑ Α.Φ.Μ. 094360202				<b>ΛΟΓΙΣΤΟΤΗΤΑ/ADD.INFO</b> Χρέωση/Debit: ΑΠΟΣΤΟΛΕΑ Προϊόντα/Products: ΑΝ Αντ/λη/С.ο.д <input checked="" type="checkbox"/> Αγορά/Buy <input type="checkbox"/> Ποσό/Total: 50,00 € ΜΕΤΡΗΤΑ Παράδοση Έως/Delivery Until: ΑΝΩ Επιστροφή: Κύριο ΑΝΩ: <b>ΠΑΡΑΤΗΡΗΣΕΙΣ/NOTES</b>				
<b>ΠΑΡΑΛΗΠΤΗΣ / RECEIVER</b> Ονομα: TEST RECIPIENT Εταιρία/Όροφος: / Διεύθυνση: P. RALLI 45 Πόλη: ΤΑΥΡΟΣ Τ.Κ. 2108190000 Τηλ. 2115005000 Κιν. Cell. 17778				<b>ACS</b> Τα πάντα, παντού. www.acscourier.gr <b>Προορισμός Destination</b>  <b>BT</b>				
ACS A.E.E. - Έδρα Ασκληπιού 25 Τ.Κ. 14568 Κρηνοτέρη Αττικής ΑΦΜ: 094360202 ΔΟΥ: Φ.Α.Ε. ΑΘΗΝΩΝ, Τηλ: 210-8190000, Fax: 210-8190								

\* Ισχύουν οι γενικοί όροι ανάληψης μεταφοράς της ACS και τυχόν υφιστάμενου συμφωνητικού μεταξύ ACS και πελάτη.

## Notes:

- There are 2 printing options for vouchers. The format can be either laser (page A4 with 3 labels per page) or thermal.
- In the examples where XXXXXXXXX the **Voucher\_No** returns the Create Voucher method.
- The vouchers can be printed 1 to 1 or all together.
  - To print all vouchers at the same time, on the same link we register all the Voucher\_No = XXXXXXXXXX.
  - Multiple XXXXXXXXXX are separated by the character "|".
- The **PrintType** parameter defines the print format: 1 = Thermal, 2 = Laser (default value: 1).
- The **StartFromNumber** parameter define the starting position. The acceptable values are 1,2,3 (it makes sense only for printing on a Laser Printer - PrintType=2).
- If the field **With\_Return\_Voucher** of the create voucher, has value **1** or/ and there is the additional product "RDO", insert as voucherno only the main voucher.
- If the field **Item\_Quantity** of the create voucher has value >1, then the main voucher and the sub vouchers will appear in the printout.
- **Reprint Vouchers:** to reprint the vouchers simply re-call the same print link.

## DELETE VOUCHER

Through this method we can delete a voucher because, for example, we will not send the shipment. **This is possible only if this shipment has not been included in a pickup list** (see **Issue\_Pickup\_List** chapter).

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body :

```
{
  "ACSAlias": "ACS_Delete_Voucher",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

## demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_Delete_Voucher",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": null,     "Voucher_No": 7227889480   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSExecutionResponse": {     "ACSValueOutput": [       {         "Error_Message": null       }     ],     "ACSTableOutput": {}   } }</pre>

## Notes:

- We can delete the vouchers one by one or bulk up to 20 vouchers at the same time.
- For bulk deletion we separate the voucher with comma.

- If the shipment has multiple vouchers, the deletion of the main voucher also deletes the sub vouchers.

Error messages (Can be changed in the future without notice):

- **Cannot delete ACS voucher when the courier pickup list is issued** - If the voucher has been included in a pick-up list, it is final, and deletion is only possible from ACS upon request from the sender.

## CONTENT OF CYPRUS SHIPMENTS

When the country of destination ("Recipient\_Country") in the create voucher method is Cyprus - CY, it is mandatory to register the content of the shipment (concerns a decision of the customs of Cyprus). Failure to describe the content results in delays and fines from the Larnaca customs.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body :

```
{
  "ACSAlias": "ACS_Get_Content_Types",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request

Request
<pre>{   "ACSAlias": "ACS_Get_Content_Types",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": 'GR',   } }</pre>

## ISSUE PICKUP LIST

**The creation of the pickup list is a mandatory step to finalize the vouchers, otherwise the vouchers are not recognized by the ACS systems.**

Through this method we receive the **PickupList\_No** of the pickup list through which we will print it.

**This manifest should be created at the end of production and printing of the labels** and is given to the courier at the time of receiving the shipments.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```
{  
  "ACSAlias": "ACS_Issue_Pickup_List",  
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_Issue_Pickup_List",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": 'GR',     "Pickup_Date": '2019-01-11',     "MyData": null,   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSExecutionResponse": {     "ACSValueOutput": [       {         "PickupList_No": "7227889830",         "Unprinted_Found": 0,         "Error_Message": ""       }     ],     "ACSTableOutput": {       "Table_Data": []     }   } }</pre>

In our example for the date 29-01-11 we have 1 pickup list with a mass code **7227889830**.

#### Notes:

- In the Pickup\_Date field we enter the date of issue of the list. **This date must be the same as the date the vouchers were created.**
- In the field **MyData** if the value is = 0 finalize the daily shipments of all users. If the value =1 then we finalize the shipments of the specific username (useful only if the users >1)

#### Error messages (Can be changed in the future without notice):

- **You cannot issue a pick list. Found X unprinted vouchers** if there are unprinted vouchers (i.e., we did not call the Print Voucher for them). In this case, the response will show the number of unprinted vouchers and their codes.

Example:

Request	Response
<pre>{   "ACSAlias": "ACS_Issue_Pickup_List",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": 'EN',     "Pickup_Date": '2019-01-11',     "MyData": null,   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "PickupList_No": null,         "Unprinted_Found": 2,         "Error_Message": "Αδύνατη η έκδοση λίστας παραλαβής.         Βρέθηκαν 2 ατύπωτες αποστολές."       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "Unprinted_Vouchers": "7227889841"         },         {           "Unprinted_Vouchers": "7227889874"         }       ]     }   } }</pre>

In this case, we should either print or delete these vouchers, and then run again the pickup list to get the **PickupList\_No**.

## PRINT PICKUP LIST

To print the pickup list, we call the method

```
{
  "ACSAlias": "ACS_Print_Pickup_List",
  "ACSInputParameters": {
    "Company_ID": 'demo',
    "Company_Password": 'demo',
    "User_ID": 'demo',
    "User_Password": 'demo',
    "Language": 'GR',
    "Mass_Number": XXXXXXXXXXXX,
    "Pickup_Date": '2020-10-01'
  }
}
```

### Notes:

- In the field "Mass\_Number" where XXXXXXXXXXXX insert the PickupList\_No given by the method ACS\_Issue\_Pickup\_List

## Pickup List Sample

<b>Μαζικό Αποδεικτικό</b>  7227889933		<b>ΣΥΝΟΠΤΙΚΗ ΚΑΤΑΣΤΑΣΗ ΠΑΡΑΛΑΒΗΣ</b> <b>ΑΠΟΣΤΟΛΩΝ ΠΡΟΣ ΜΕΤΑΦΟΡΑ</b>		 Τα πάντα, παντού.			
Αρ.Αποστολής	Αποστολέας	Όνομα Παραλήπτη	Διεύθυνση/Περιοχή	T.K	Τηλέφωνο	Κωδ.Πελ.	Προ/αγός
Κύριος Αρ.Απ.	Ημ.Παράλ.	Ταμ.	Βάρος	Παρόδ.Έως	Ποσό Αντ.	Τρ.Πλ.	Ποσό Ασφ.
Κατάσταση	Προϊόντα	Σχετικό 1	Παρατηρήσεις				
7227889900	ESHOP	11/01/2019	1	0,50	00:00	0,00 €	M
7227889911	ESHOP	11/01/2019	1	0,50	00:00	0,00 €	M
7227889922	ESHOP	11/01/2019	1	0,50	00:00	0,00 €	M
ΣΥΝΟΛΟ ΑΠΟΣΤΟΛΩΝ: 3				Ο ΠΑΡΑΔΟΥΣ		Ο ΠΑΡΑΛΑΒΩΝ	
				ESHOP			

At the top left corner, the bar code corresponds to the **PickupList\_No**.

## PICKUP\_LIST\_DISPLAY\_VOUCHER

This method displays the vouchers contained in a specific pickup list and the respective Reference\_Key1 - Reference\_Key2 (if they had been completed in the create voucher method).

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```
{
  "ACSAlias": "ACS_Pickup_List_Display_Voucher ",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response

Request	Response
<pre>{   "ACSAlias": "ACS_Pickup_List_Display_Voucher",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": null,     "PickupList_No": XXXXXXXX,     "Pickup_Date": '2019-01-15'   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "List_Vouchers_Count": 1,         "Error_Message": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "Voucher_no": "XXXXXXXXXX",           "Reference_Key1": null,           "Reference_Key2": null         }       ]     }   } }</pre>

## GET PICKUP LISTS

If for any reason a reissue of a pickup list is required, the **Get\_Pickup\_Lists** method for a given date can be used to return all **PickupList\_No** numbers issued for that date.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```
{
  "ACSAlias": "ACS_Get_Pickup_Lists",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_Get_Pickup_Lists",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": 'EN',     "Pickup_Date": '2019-01-15'   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSExecutionResponse": {     "ACSValueOutput": [       {         "Error_Message": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "Pickup_date": "2019-01-15T00:00:00",           "Pickup_List_DateTime": "2019-01-15T11:05:03.943",           "User_ID": "demo",           "PickupList_No": "7227890935",           "List_Vouchers_Count": 2         },         {           "Pickup_date": "2019-01-15T00:00:00",           "Pickup_List_DateTime": "2019-01-15T11:05:34.11",           "User_ID": "demo",           "PickupList_No": "7227890950",           "List_Vouchers_Count": 1         }       ]     }   } }</pre>

In our example for the date 2019-01-15 there are 2 lists. The response shows the codes (PickupList\_No), the date (Pickup\_date) and the time (Pickup\_List\_DateTime) that were created, as well as the number of shipments that each contains (List\_Vouchers\_Count).

The reprint of a pickup list can be done up to six months behind.

## ACS\_COD\_BENEFICIARY\_INFO

This method displays cash on delivery performance information. The vouchers, the amounts of cash on delivery as well as the method of collection of cash on delivery are displayed (i.e., if the recipient paid in cash or by card).

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>



and in the body:

```
{
  "ACSAlias": "ACS_COD_Beneficiary_Info"
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_COD_Beneficiary_Info",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "User_locals": 'GR',     "COD_Payment_Date": 'EEEE-MM-HH'   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSExecutionResponse": {     "ACSValueOutput": [       {         "Error_msg": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "Customer_Code": "2A0XXXX",           "POD": "XXXXXXXXXX",           "Parcel_Sender": "Sender",           "Parcel_Receiver": "Recipient",           "Parcel_Pickup_Date": "2020-09-05T00:00:00",           "Parcel_Delivery_Date": "2020-09-07T00:00:00",           "Parcel_COD_Amount": 120.30,           "Customer_RefNo_1": "XXXX",           "Customer_RefNo_2": "",           "COD_Amount_Cach": 0.00,           "COD_Amount_CreditCard": 120.30         }       ]     }   } }</pre>

Notes:

- The date in the field **COD\_Payment\_Date** does not bring results in the sense from that date but brings results for that day.
- In the response the fields Customer\_RefNo\_1 & Customer\_RefNo\_2 show which keys were used in the create voucher method in the fields Reference\_Key1 & Reference\_Key2.

## TRACKING SUMMARY

Through this method we have the latest status update for a shipment.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>[help](#)

and in the body:

```
{
  "ACSAlias": "ACS_Trackingsummary",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

## demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_Trackingsummary",   "ACSInputParameters": {     "Company_ID": 'XXXXXXX',     "Company_Password": XXXXXX,     "User_ID": XXXXXX,     "User_Password": XXXXXXXX,     "Language": 'EN',     "Voucher_No": XXXXXXXXXXXX   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "Error_Message": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "voucher_no": XXXXXXXXXXXX,           "acs_station_origin": "AΘ",           "acs_station_origin_descr": "AΘHNA",           "acs_station_destination": "AΘ",           "acs_station_destination_descr": "AΘHNA",           "pickup_date": "2018-12-21T00:00:00",           "delivery_flag": 1,           "returned_flag": 0,           "delivery_date": "2018-12-21T00:00:00",           "consignee": "XXXXXXXXXX",           "non_delivery_reason_code": "",           "delivery_date_expected": "2018-12-24T00:00:00",           "delivery_info": "Shipment has been delivered to destination. Date: 21/12/2018 Name: XXXXXX",           "sender": "XXXXXXXXXX",           "recipient": "XXXXXXXXXX",           "recipient_address": "XXXXXXXXXX 25",           "shipment_status": 4,           "phone_acs_station_origin": "210-8190000",           "phone_acs_station_destination": "210-8190000"         }       ]     }   } }</pre>

In the example we see which the starting point and the destination of the shipment is, when the courier received it from the sender, who is the sender and the recipient/ consignee, what is the delivery address, when delivered and to whom.

## Explanation of parameters

"delivery_flag"	If 1 = delivered shipment, if 0 = undelivered shipment
"returned_flag"	If 1 = returned as non-deliverable back to the sender
"non_delivery_reason_code"	the non-delivery reason code
"shipment_status"	Shipment's status code

## Notes:

If the shipment status is 4 it is a delivered shipment.

The Non-delivery codes in the "non\_delivery\_reason\_code" field in conjunction with the description in the "delivery\_flag" field and the code in the "shipment\_status" field:

- AD1 - non-delivery reason: ORDER FOR OFFICE PICKUP - STATUS 5.
- AD3 - non-delivery reason: FORCE MAJEURE - STATUS 5.
- AD8 - non-delivery reason: Pick-up from ACS Shop / Sender's Request - STATUS 5.
- AP1 - non-delivery reason: Delivery Refusal (due to charge)- STATUS 1.
- AP2 - non-delivery reason: UNABLE TO PAY - STATUS 1.

- AP3 - non-delivery reason: NOT ACCEPTABLE SHIPMENT - STATUS 1.
- AP4 - non-delivery reason: DECEASED- STATUS 1.
- AS1 - non-delivery reason: ABSENT - STATUS 3.
- DP1- non-delivery reason: DIFFICULT TO ACCESS AREA - STATUS 5.
- LS1 - non-delivery reason: UNKNOWN CONSIGNEE - STATUS 2.
- LS2 - Shipment is on transit to destination – STATUS 2.
- LS3 - non-delivery reason: ERROR OR INADEQUATE ADDRESS - STATUS 2.
- PA1 - non-delivery reason: NEW DELIVERY DATE ORDER OF SHIPPER - STATUS 5.
- PA2 - non-delivery reason: NEW DELIVERY DATE ORDER OF CONSIGNEE - STATUS 5.
- PA4 - non-delivery reason: Reschedule / Redirect - STATUS 5.

If a shipment is not delivered after 10 days, it returns to the sender. In this case there will initially be shipment\_status 6 – to be returned and finally shipment\_status 7 when delivered to the sender along with "returned\_flag": 1 and "delivery\_flag": 1.

## TRACKING DETAILS

Through this method we have the picture of the whole course of a shipment.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```
{
  "ACSAlias": "ACS_TrackingDetails",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response (reds are mandatory):

<pre>{   "ACSAlias": "ACS_TrackingDetails",   "ACSInputParameters": {     "Company_ID": 'XXXXXXX',     "Company_Password": 'XXXXXXX',     "User_ID": 'XXXXXXX',     "User_Password": 'XXXXXXX',     "Language":     "Voucher_No": XXXXXXXXXX   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSExecutionResponse": {     "ACSValueOutput": [       {         "Error_Message": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "checkpoint_date_time": "2019-01-26T13:05:57.263",           "checkpoint_action": "Departure to destination MAROUSI",           "checkpoint_location": "KAVALA",           "checkpoint_notes": " "         },         {           "checkpoint_date_time": "2019-01-28T00:25:54.723",           "checkpoint_action": "Arrival-departure from HUB",           "checkpoint_location": "HUB P. RALLI",           "checkpoint_notes": " "         },         {           "checkpoint_date_time": "2019-01-28T00:25:56.567",           "checkpoint_action": "Arrival-departure from HUB",           "checkpoint_location": "HUB P. RALLI",           "checkpoint_notes": " "         }       ]     }   } }</pre>
--	--

	<pre> {   "checkpoint_date_time": "2019-01-28T06:53:16.93",   "checkpoint_action": "Arrival to ",   "checkpoint_location": "MAROUSI",   "checkpoint_notes": " " }, {   "checkpoint_date_time": "2019-01-28T07:56:00.587",   "checkpoint_action": "Arrival to ",   "checkpoint_location": "MAROUSI",   "checkpoint_notes": " " }, {   "checkpoint_date_time": "2019-01-28T07:56:00.587",   "checkpoint_action": "On delivery",   "checkpoint_location": "MAROUSI",   "checkpoint_notes": " " }, {   "checkpoint_date_time": "2019-01-28T09:21:00",   "checkpoint_action": "Delivery to consignee",   "checkpoint_location": "MAROUSI",   "checkpoint_notes": "XXXXXXX" } </pre>
--	--

## REFERENCE KEY FOR ACS VOUCHER CODE

Through this method it is possible to find out which ACS shipment code corresponds to the reference key entered in the corresponding field of the create voucher method. **The method returns data up to 2 months back.**

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```

{
  "ACSAlias": "ACS_POD_FROM_REFERENCE_NO ",
}

```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

**demo request – response** (reds are mandatory):

```

{
  "ACSAlias": "ACS_POD_FROM_REFERENCE_NO",
  "ACSInputParameters": {
    "Company_Id": 'demo',
    "Company_Password": 'demo',
    "User_ID": 'demo',
    "User_Password": 'demo',
    "User_locals": 'GR',
    "reference_no": 'XXXXXX'
  }
}

```

### Notes:

- The method returns data up to 2 months back.

- Provided that the search refers to final shipments, i.e. shipments that are on the PickupList pickup list.
- The search reference key must correspond to one final shipment / voucher.
- If it is a multiple shipment (items> 1) then in the response the main voucher appears but also the sub-piece.

## PRICE CALCULATION

Returns the shipping cost based on the sender's pricelist, considering several parameters such as: the type of destination of the shipment (e.g. city, island etc.), the type of additional service (e.g., COD, morning delivery, etc.), the weight (real or volumetric), etc.

**The cost of the shipments to Cyprus or abroad cannot be calculated.**

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```
{
  "ACSAlias": "ACS_Price_Calculation",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request:

Request	Response
<pre>{   "ACSAlias": "ACS_Price_Calculation",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Billing_Code": '2A0999999',     "Billing_Category": 2,     "Acs_Station_Origin": 'AO',     "Acs_Station_Destination": 'XN',     "Weight": '0,5',     "Pickup_Date": '2019-01-14',     "Acs_Delivery_Products": null,     "Charge_Type": 2,     "Delivery_Zone": null,     "Insurance_Ammount": null,     "Dimension_X_In_Cm": null,     "Dimension_Y_In_Cm": null,     "Dimension_Z_In_Cm": null,     "Language": null   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSExecutionResponse": {     "ACSExecutionOutput": [       {         "Basic_Ammount": 11.2200,         "Extra_Service_Ammount": 0.0000,         "Total_Ammount": 11.22,         "Total_Vat_Ammount": 2.69,         "Info_Message": "",         "Error_Message": ""       }     ],     "ACSExecutionTableOutput": {}   } }</pre>

In the example, we have a dispatch of 0.5 kilos (no dimensions therefore no volumetric calculation), from Athens to Chania (i.e., island destination), with delivery date 14/01/2019 and Charge\_Type = 2.

## Notes:

- **Acs\_Station\_Origin** and **Acs\_Station\_Destination** codes must be in Greek and uppercase.

- These codes can be found from the **Find by Zipcode** method discussed in this documentation.
- If we register weight and dimensions, then the cost will be based on actual or volumetric weight (whichever is greater).
- The volumetric weight is computed by the formula:  $L \times W \times H / 5000$
- The response is: **Basic Service Cost (Basic\_Ammount) | ACS delivery products Cost (Extra\_Service\_Ammount) | Total Excl. VAT (Total\_Ammount) | VAT (Total\_Vat\_Ammount)**.
- In the **Acs\_Delivery\_Products** field you can enter the Greek product codes as described in the create voucher method.
- **The Insurance\_Amount** field refers to the security amount, if additional security of the transported item is requested e.g., "550.50" euro. Maximum accepted price up to € 3000.

## ADDRESS VALIDATION

The method returns a list of possible valid addresses or the only valid address based on a combination of data such as street, number, Zipcode, and region.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```
{
  "ACSAlias": "ACS_Address_Validation ",
}
```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_Address_Validation",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Language": null,     "Address": 'ΔΩΔΕΚΑΝΗΣΟΥ 25 13231'     "AddressID": null   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "ACSObjectOutput": [           {             "GeoID": 631466,             "Resolved_Street": "ΔΩΔΕΚΑΝΗΣΟΥ",             "Resolved_Street_Num": "25",             "Resolved_Zip": "13232",             "Resolved_Area": "ΠΕΤΡΟΥΠΟΛΗ",             "Resolved_Long": 23.67857,             "Resolved_Lat": 38.04628,             "Resolved_GeoDataType": 1,             "Resolved_GeoDataID": 631466,             "Resolved_Station_ID": "BΘ",             "Resolved_Branch_ID": 1,             "Resolved_As_Inaccessible_Area_With_Cost": 0,             "Resolved_As_Inaccessible_Area_Without_Cost": 0,             "Resolved_Confidence": 0,             "Resolved_GeoRegionType": 1,             "Resolved_Providence": "N. ΑΤΤΙΚΗΣ",             "Resolved_Correction": "110",             "Resolved_Station_Descr": "ΑΓΙΟΙ ΑΝΑΓΥΡΟΙ (210-2693523 &amp; 210-2693553)"             "AddressID": ""           }         ]       }     ]   } }</pre>

	<pre>         }       ]     }   ],   "ACSTableOutput": {} } } </pre>
--	--

## Notes

- The address must be in Greek and not in Latin characters.

## FIND AREA BY ZIPCODE

The method searches for Greek and Cyprian regions based on the zipcode and returns the ACS store and branch that serves it, as well as whether it is a hard-to-reach region or not. It can display either all regions or only the hard-to-reach regions of the search query.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```

{
  "ACSAlias": "ACS_Area_Find_By_Zip_Code",
}

```

Otherwise (if we already know the parameters) the url is

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

demo request – response:

Request	Response
<pre> {   "ACSAlias": "ACS_Area_Find_By_Zip_Code",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "Zip_Code": null,     "Show_Only_Inaccessible_Areas": 0,     "Language": 'EN',     "Country": 'GR'   } } </pre>	<pre> {   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "Error_Message": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "Description": "ΑΓΙΑ ΤΡΙΑΔΑ ΠΑΡΝΗΘΑΣ",           "Area": "ΑΓΙΑ ΤΡΙΑΔΑ ΠΑΡΝΗΘΑΣ",           "Description_Eng": "AGIA TRIADA PARNITHAS",           "Zip_Code": "13679",           "Municipality": "",           "Prefecture": "N. ΑΤΤΙΚΗΣ",           "Station_ID": "BA",           "Branch_ID": 1,           "Inaccessible_Area_Kind": "ΔΠ"         },         {           "Description": "ΑΜΥΓΔΑΛΕΖΑ",           "Area": "ΑΜΥΓΔΑΛΕΖΑ",           "Description_Eng": "AMIGDALEZA",           "Zip_Code": "13679",           "Municipality": "",           "Prefecture": "N. ΑΤΤΙΚΗΣ",           "Station_ID": "BA", </pre>

	<pre> "Branch_ID": 1, "Inaccessible_Area_Kind": "" }, {   "Description": "ΞΕΝΙΑ ΠΑΡΝΗΘΑΣ",   "Area": "ΞΕΝΙΑ ΠΑΡΝΗΘΑΣ",   "Description_Eng": "KSENIA PARNITHAS",   "Zip_Code": "13679",   "Municipality": "",   "Prefecture": "N. ΑΤΤΙΚΗΣ",   "Station_ID": "BA",   "Branch_ID": 1,   "Inaccessible_Area_Kind": "ΔΠ" }, {   "Description": "ΠΑΡΝΗΘΑ",   "Area": "ΠΑΡΝΗΘΑ",   "Description_Eng": "PARNITHA",   "Zip_Code": "13679",   "Municipality": "",   "Prefecture": "N. ΑΤΤΙΚΗΣ",   "Station_ID": "BA",   "Branch_ID": 1,   "Inaccessible_Area_Kind": "ΔΠ" } } } } </pre>
--	---

#### Notes:

- In the Country field the acceptable values are GR for Greece, CY for Cyprus, BG for Bulgaria, and AL for Albania.
- For Bulgaria and Albania, we give only 1 result in the response.
- In the Show\_Only\_Inaccessible\_Areas field if the value is 0 all areas are displayed while if it is 1 then only the remote areas are displayed. **This filter is only valid for Greece Country "Country": 'GR'.**
- Those areas that are remote to the response have the indication ΔΠ in the field Inaccessible\_Area\_Kind.
- In the Zip\_Code field if the value is 0 or null, then in the response there will be all the areas and all the zip codes of the selected country.

#### Error messages (can be changed in the future without notice):

- **Data not found with these criteria.** For example, when the zip code is non-existent or filter 1 is set in the Show\_Only\_Inaccessible\_Areas field for a zip code in which there is no remote areas. (inaccessible) area.

### ACS STATION CODE BASED ON ZIPCODE

The method refers to the search for ACS station codes based on zipcode for Greece and Cyprus.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url

<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoResthelp>

and in the body:

```

{
  "ACSAlias": "ACS_Find_Station_By_Zip_Code",
}

```



Otherwise (if we already know the parameters) the url is  
<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

#### demo request

```
{
  "ACSAlias": "ACS_Find_Station_By_Zip_Code",
  "ACSInputParameters": {
    "Company_ID": 'demo',
    "Company_Password": 'demo',
    "User_ID": 'demo',
    "User_Password": 'demo',
    "Zip_Code": 'XXXXX'
    "language": 'GR',
    "Country": 'GR',
  }
}
```

#### Notes:

- In the Zip\_Code field insert the 5-digit Greek or 4-digit Cypriot ZIP code.
- In the Country field, the accepted values are GR for Greece and CY for Cyprus.
- In the language field, if the value is GR, results are returned in Greek and if EN in English.

## ACS STATIONS

The method is to search for ACS store details in Greece.

The method checks the accuracy of the recipient's Zipcode. It can show either all regions or only the inaccessible areas (REM) of the search pointer.

The request is created as described in the **START INSOMNIA** chapter.

To see the parameters of the method we can use the url  
<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>[help](#)

and in the body:

```
{
  "ACSAlias": "Acs_Stations ",
}
```

Otherwise (if we already know the parameters) the url is  
<https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest>

#### demo request – response:

Request	Response
<pre>{   "ACSAlias": "ACS_Stations",   "ACSInputParameters": {     "Company_ID": 'demo',     "Company_Password": 'demo',     "User_ID": 'demo',     "User_Password": 'demo',     "language": 'EN',     "ACS_SHOP_COUNTRY_ID": 'GR',     "ACS_SHOP_KIND": 1   } }</pre>	<pre>{   "ACSExecution_HasError": false,   "ACSExecutionErrorMessage": "",   "ACSOutputResponse": {     "ACSValueOutput": [       {         "error_message": null       }     ],     "ACSTableOutput": {       "Table_Data": [         {           "ACS_SHOP_COUNTRY_ID": 1,</pre>

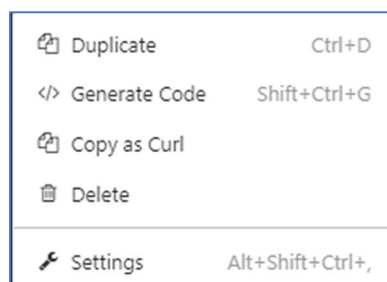
	<pre> "ACS_SHOP_COUNTRY_DESCR": "GREECE", "ACS_SHOP_AREA_ID": 19, "ACS_SHOP_AREA_DESCR": "THESSALONIKIS", "ACS_SHOP_STATION_ID": "ΘΣ", "ACS_SHOP_STATION_ID_EN": "SKG", "ACS_SHOP_STATION_DESCR": "THESSALONIKI", "ACS_SHOP_BRANCH_ID": 1, "ACS_SHOP_ADDRESS": "5 XLM.E.O. THES/KIS-ATHINON", "ACS_SHOP_ZIPCODE": "57009", "ACS_SHOP_PHONES": "2311 99 5500 &amp; 2310-573500", "ACS_SHOP_FAX": "2310-573524", "ACS_SHOP_WORKING_HOURS": "08:00-20:00", "ACS_SHOP_WORKING_HOURS_SATURDAY": "08:00-15:00", "ACS_SHOP_TRUCK_PICKUP_HOURS": "19:30", "ACS_SHOP_TRUCK_PICKUP_HOURS_SATURDAY": "15:00", "ACS_SHOP_LAT": "40.655865", "ACS_SHOP_LONG": "22.864653", "ACS_SHOP_DELIVERY_START_HOUR": "08:30", "ACS_SHOP_COORDINATES_VERIFIED": 1, "ACS_SHOP_KIND": 1, "ACS_SHOP_SERVICES": "All Services", "ACS_SHOP_EMAIL": "shops@acscourier.gr", "ACS_SHOP_PAYMENT_TYPES": "Acceptable use of bank cards. Exclusively for paying off shipment costs by retail customers, as well as invoices by corporate/credit customers and also COD amount for shipments", "ACS_SHOP_ID_CODE": "19003100" }, </pre>
--	---

#### Notes:

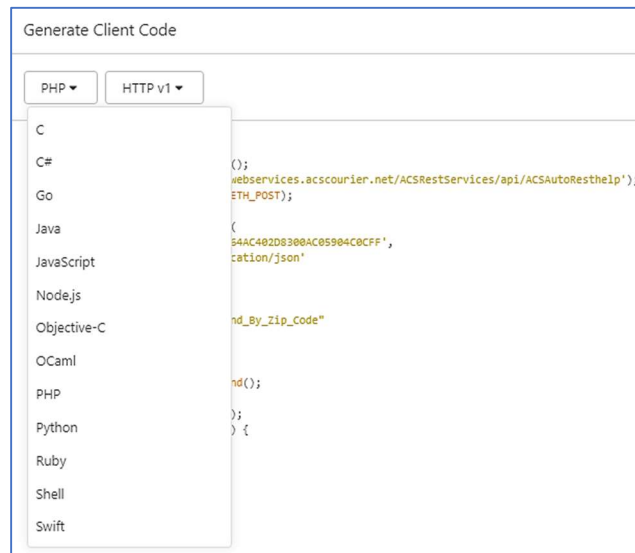
- In the field **ACS\_SHOP\_COUNTRY\_ID** if the value is **GR**, the result is returned by Greek stores and if field is **CY** for Cypriot.
- In the field **ACS\_SHOP\_KIND** and for country **GR**, if the value is:
  - **1** the central stores are returned.
  - **2** or **3** the subbranches are returned.
  - **4** is for the ACS Xpress points that perform only cash store pickups.
  - **5** the ACS Kiosks are returned to receive only standard folders.
  - **7** shows the ACS Smartpoints for receiving standard envelopes / packages up to 6kg.
  - **8** shows the ACS Smartpoints with Parcel locker for receiving standard envelopes / packages up to 6kg.
- In the field **ACS\_SHOP\_KIND** and for country **CY**, if the value is:
  - **1** the central stores are returned.
  - **2** or **3** no results.
  - **4** the shop in a shop.
  - **5** no results.
  - **7** no results.

## GENERATE CLIENT CODE

Through INSOMNIA with right click on the request we can see the options



By selecting the **Generate Code** option we have the list



From which we can choose the language we want and see how each request is formed.