

ACS Courier

ACS REST API

Developer integration guide

Version 2.0 (draft)

October 2026

Version history

Version	Date	Changes
1.x	—	Previous version of the document.
2.0	October 2026	Complete restructuring: chapters by function, a uniform layout per method (request, parameters, response, errors), cURL and code examples, error-handling guide. Voucher printing with ACS_Print_Voucher_V2. Updated ACS Point types. Testing tools moved to appendices. First English edition.

Contents

1. Introduction and general specifications	3
1.1 Purpose of this document	3
1.2 The API at a glance.....	3
1.3 What you need before you start	3
1.4 Endpoints.....	4
1.5 HTTP headers.....	4
1.6 Request structure	4
1.7 Response structure.....	4
1.8 Error handling.....	5
1.9 Quick start: your first call.....	6
1.10 Usage limits and best practices	6
1.11 Method catalog	6
2. Creating shipments.....	7
2.1 Overview.....	7
2.2 Create a voucher — ACS_Create_Voucher	7
2.3 Delivery to an ACS store or ACS Smartpoint	12
2.4 Multi-part shipment — ACS_Get_Multipart_Vouchers.....	12
2.5 Common scenarios	13
3. Printing and managing vouchers	14
3.1 Print a voucher — ACS_Print_Voucher_V2	14
3.2 Delete a voucher — ACS_Delete_Voucher	16
3.3 Content of shipments to Cyprus — ACS_Get_Content_Types	18
4. Pickup lists	19
4.1 Overview.....	19
4.2 Issue pickup list — ACS_Issue_Pickup_List	19
4.3 Print pickup list — ACS_Print_Pickup_List.....	21
4.4 Pickup lists of a day — ACS_Get_Pickup_Lists.....	22
4.5 Shipments in a pickup list — ACS_Pickup_List_Display_Voucher.....	23
5. Shipment tracking.....	24
5.1 Overview.....	24
5.2 Tracking summary — ACS_Trackingsummary	25
5.3 Tracking details — ACS_TrackingDetails.....	27
5.4 Find voucher by reference key — ACS_POD_FROM_REFERENCE_NO	28
6. Cash on delivery	29
6.1 COD remittance details — ACS_COD_Beneficiary_Info.....	29
7. Pricing and addresses	31

7.1 Overview	31
7.2 Cost calculation — ACS_Price_Calculation	32
7.3 Address validation — ACS_Address_Validation	33
7.4 Area lookup by postal code — ACS_Area_Find_By_Zip_Code	35
8. ACS stores	37
8.1 Overview	37
8.2 Store serving a postal code — ACS_Find_Station_By_Zip_Code	37
8.3 List of ACS Points — ACS_Stations	38
Appendix A. Testing with Insomnia	41
A.1 Installation	41
A.2 Creating a request	41
A.3 URL and header	42
A.4 Execution	42
Appendix B. Generating code from Insomnia	42

1. Introduction and general specifications

1.1 Purpose of this document

This guide is intended for developers who will connect the customer's system (ERP, e-shop, WMS, etc.) to the ACS Web Services. It describes how to make a call, what the server returns and how to handle errors, and then covers each method in detail with complete request and response examples.

We recommend that you read this entire chapter first: the rules described here apply to **all** methods and are not repeated in the following chapters.

1.2 The API at a glance

Characteristic	Value
API type	REST, data exchanged in JSON
Protocol	HTTPS
HTTP method	POST for all calls
Base URL	https://webservices.acscourier.net/ACSRestServices/api
Authentication	API Key in the ACSApiKey header and customer access credentials in the body
Method selection	With the ACSAlias parameter in the body (e.g. ACS_Create_Voucher)
Rate limit	10 calls per second (default)

1.3 What you need before you start

ACS provides you with the following details when your connection is activated:

Item	Where it is used	Description
API Key	HTTP header ACSApiKey	Identifies your application. Determines which methods (ACSAlias) you have access to.
Company_ID Company_Password	Body (ACSInputParameters)	Credentials of the customer company.
User_ID User_Password	Body (ACSInputParameters)	Credentials of the user making the call.
Billing_Code	Body (shipment creation and pricing methods)	The customer's billing code.
IP whitelist (optional)	—	If the connection is IP-restricted, calls are accepted only from the IP addresses you have registered with ACS.

Credential security

The API Key and access credentials are confidential. Store them on your application server (e.g. in configuration settings or a secrets manager) and **never embed them** in code that runs in a browser or in a mobile app.

1.4 Endpoints

All methods are executed through the same endpoint. The method to be executed is determined by the value of **ACSAlias** in the body, not by the URL.

Endpoint	Use
POST https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest	Executes a method. This is the endpoint you use in production.
POST https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRestHelp	Help. Returns the description of a method or the JSON template with the parameters it accepts (see §1.9).

1.5 HTTP headers

Header	Value	Required
Content-Type	application/json	Yes
ACSApiKey	The API Key provided to you by ACS	Yes

1.6 Request structure

Every request has the same format: the method name (**ACSAlias**) and an object containing its parameters (**ACSInputParameters**).

General request format
<pre>{ "ACSAlias": "<Method_Name>", "ACSInputParameters": { "Parameter_1": "value", "Parameter_2": 123, "Parameter_n": null } }</pre>

Field	Type	Required	Description
ACSAlias	string	Yes	The name of the method you want to execute, e.g. ACS_Create_Voucher . The full list is in §1.11.
ACSInputParameters	object	Yes	The method parameters, by name (name : value). The available parameters are described in the chapter for each method.

Field names are case-sensitive

All JSON field names are **case-sensitive**: **Voucher_No**, **voucher_no** and **VOUCHER_NO** are three different names. A field whose name differs even in one letter is not recognized.

Always use the field names **exactly** as returned by the help endpoint **ACSAutoRestHelp** (§1.9), which gives you the parameter template of every method. Note that some methods use a different spelling, e.g. **Company_Id** in **ACS_POD_FROM_REFERENCE_NO**.

Value formatting rules

- The body must be **valid JSON**: field names and strings in **double** quotes ("**...**"), not single quotes.
- Dates: format **YYYY-MM-DD**, e.g. "**2026-10-12**".
- Decimal numbers (type **decimal**: weight, cash on delivery, insurance): use a **period** as the decimal separator and up to **2** decimal places, e.g. **0.5**, **50.50**.
- Optional parameters you do not use: value **null**.
- Character encoding: **UTF-8** (Greek characters are supported).

1.7 Response structure

All methods return the same envelope. The data of each method is inside **ACSOutputResponse**.

General response format

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOuputResponse": {
    "ACSValueOutput": [
      { ...method results... }
    ],
    "ACSTableOutput": { }
  }
}
```

Field	Type	Description
ACSExecution_HasError	boolean	false : the call was executed. true : the call failed (see §1.8).
ACSExecutionErrorMessage	string	Description of the error. Has a value only when ACSExecution_HasError = true .
ACSOuputResponse	object	The method results. Has content only when ACSExecution_HasError = false .
ACSValueOutput	array	List of result records (e.g. voucher number, method error message).
ACSTableOutput	object	Results in table form, when the method returns multiple rows (e.g. tracking, lists). The rows are usually in Table_Data . If there is a Return_Data field, it can be deserialized directly into a .NET DataSet .

Watch the spelling

The field is named **ACSOuputResponse** (with a **c**, not "Response"). Use exactly this name when you read or deserialize the response.

1.8 Error handling

An error can occur at three levels. Check them **in this order** in every response:

Level 1 — HTTP status code

HTTP status	Meaning	What to do
200 OK	The call reached the service. The server returns 200 in all other cases, even when execution failed .	Continue to Level 2.
403 Forbidden	The ACSApiKey header is missing, or the API Key is not valid.	Check the name and value of the header.
406 Not Acceptable	You exceeded the allowed limit of concurrent calls per second (default: 10/sec).	Wait briefly and retry the call (see §1.10).

Level 2 — Execution error

If **ACSExecution_HasError** is **true**, the method was not executed. The cause is in **ACSExecutionErrorMessage**. Common causes: a wrong or nonexistent **ACSAlias**, the API Key has no permission for that method, the IP is not on the whitelist, wrong parameter names or types.

Level 3 — Method error

Even when **ACSExecution_HasError** is **false**, the method itself may reject the request for a business reason (e.g. there are no vouchers for issuing a pickup list). In this case the message is returned in the **Error_Message** field inside **ACSValueOutput**:

Example: the call was executed, but the method returned an error

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOuputResponse": {
    "ACSValueOutput": [
      {
        "Error_Message": "You cannot issue a pick list ..."
      }
    ]
  }
}
```

The examples in this guide use **Language = EN**, so error messages are shown in English.

The correct check for every response

1. HTTP status = 200?
 2. `ACSExecution_HasError` = false?
 3. `Error_Message` empty or null?
- Only if all three are true has the call completed successfully and you can use the results.

1.9 Quick start: your first call

The fastest and safest way to get started with a new method is to ask the API for its parameter template and then fill it in. This gives you the field names exactly as the method expects them (with the correct upper and lower case).

Step 1 — Request the parameter template

Send only the `ACSAlias` to the `ACSAutoRestHelp` endpoint. The response is the JSON with all the parameters the method accepts.

cURL — parameter template

```
curl -X POST "https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRestHelp" \
-H "Content-Type: application/json" \
-H "ACSApiKey: <YOUR_API_KEY>" \
-d '{ "ACSAlias": "ACS_Create_Voucher" }'
```

If instead of an object you send just the method name as a JSON string (e.g. `"ACS_Create_Voucher"`), the same endpoint returns a text description of the method.

Step 2 — Execute the method

Fill in the values and send the request to the `ACSAutoRest` endpoint:

cURL — method execution

```
curl -X POST "https://webservices.acscourier.net/ACSRestServices/api/ACSAutoRest" \
-H "Content-Type: application/json" \
-H "ACSApiKey: <YOUR_API_KEY>" \
-d '{
  "ACSAlias": "ACS_Trackingsummary",
  "ACSInputParameters": {
    "Company_ID": "<Company_ID>",
    "Company_Password": "<Company_Password>",
    "User_ID": "<User_ID>",
    "User_Password": "<User_Password>",
    "Voucher_No": "7227889174",
    "Language": "EN"
  }
}'
```

Step 3 — Read the response

Check the response in the order given in §1.8 and read the results from `ACSOutputResponse`.

Testing without code

For testing you can use a REST client such as Insomnia or Postman. Instructions for Insomnia are in Appendix A.

1.10 Usage limits and best practices

- **Always start from the help endpoint:** for every method, get the parameter template from `ACSAutoRestHelp` (§1.9) and build your request from it, so that the field names are exactly right.
- **Rate limit:** up to 10 calls per second (default). If you receive **406**, retry the call after a short wait, gradually increasing the delay (e.g. 1, 2, 4 seconds).
- **Timeouts:** set a timeout on your HTTP client and log any calls that failed.
- **Retrying creation calls:** if a voucher creation call times out without a response and you retry it, a duplicate voucher may be created. Before issuing the pickup list, find and delete any duplicate vouchers (§3.2). The pickup list will return as "unprinted" any vouchers you did not print (§4.2).
- **Logging:** keep the request and response of every call (without the credentials). This makes support from ACS much easier.

1.11 Method catalog

The available methods, grouped by function. Access to each method depends on the permissions of your API Key.

Function	ACS Alias	What it does	§
Shipments	ACS_Create_Voucher	Create a voucher (including delivery to an ACS Point)	2.2, 2.3
	ACS_Get_Multipart_Vouchers	Vouchers of a multi-part shipment (multiple parcels)	2.4
	ACS_Print_Voucher_V2	Print a voucher (PDF)	3.1
	ACS_Delete_Voucher	Delete a voucher	3.2
	ACS_Get_Content_Types	Content types for shipments to Cyprus	3.3
Pickup lists	ACS_Issue_Pickup_List	Issue a pickup list (finalizes vouchers)	4.2
	ACS_Print_Pickup_List	Print a pickup list	4.3
	ACS_Get_Pickup_Lists	Pickup lists of a day	4.4
	ACS_Pickup_List_Display_Voucher	Shipments of a pickup list	4.5
Tracking	ACS_Trackingsummary	Tracking summary (current status)	5.2
	ACS_TrackingDetails	Tracking details (all events)	5.3
	ACS_POD_FROM_REFERENCE_NO	Find a voucher by reference key	5.4
Cash on delivery	ACS_COD_Beneficiary_Info	COD remittance details	6.1
Pricing & addresses	ACS_Price_Calculation	Calculate shipping cost	7.2
	ACS_Address_Validation	Validate a destination address	7.3
	ACS_Area_Find_By_Zip_Code	Find an area by postal code	7.4
ACS stores	ACS_Find_Station_By_Zip_Code	ACS store serving a postal code	8.2
	ACS_Stations	List of ACS Points (stores, Smartpoints) with codes, opening hours and coordinates	8.3

2. Creating shipments

2.1 Overview

The methods in this chapter create ACS shipment documents (vouchers). Shipments **within Greece** and **to Cyprus** (Xpress or Cyprus Economy) are supported. Shipments to other countries are not supported.

Method (ACS Alias)	Use	Section
ACS_Create_Voucher	Creates a voucher. With the appropriate parameters it also covers delivery to an ACS store or ACS Smartpoint.	2.2, 2.3
ACS_Get_Multipart_Vouchers	Returns the numbers of the sub-vouchers of a multi-part shipment.	2.4

The lifecycle of a shipment

Creating the voucher is the first step. For the shipment to be picked up by the courier, follow this sequence:

1. **Create the voucher:** [ACS_Create_Voucher](#) (Chapter 2).
2. **Print the voucher** and attach it to the parcel: [ACS_Print_Voucher_V2](#) (Chapter 3).
3. **Issue the day's pickup list:** [ACS_Issue_Pickup_List](#) (Chapter 4).
4. **Print the pickup list** and hand the parcels over to the courier: [ACS_Print_Pickup_List](#) (Chapter 4).

The pickup list is mandatory

Vouchers are finalized as shipments **only** when the day's pickup list is issued. If no list is issued, the voucher barcodes are not recognized by the ACS network.

2.2 Create a voucher — [ACS_Create_Voucher](#)

Creates one voucher for one recipient. If [Item_Quantity](#) is greater than 1, a multi-part shipment is created (see §2.4). If [With_Return_Voucher](#) is 1, a return voucher is also created.

Example request

Shipment of 1 parcel of 0.5 kg within Greece, with cash on delivery of €50.50 in cash and shipping charged to the sender:

Request — ACS_Create_Voucher

```
{
  "ACSAlias": "ACS_Create_Voucher",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Pickup_Date": "2026-10-13",
    "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": null,
    "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_Amount": 50.50,
    "Cod_Payment_Way": 0,
    "Acs_Delivery_Products": "COD",
    "Insurance_Amount": null,
    "Delivery_Notes": null,
    "Appointment_Until_Time": null,
    "Recipient_Email": null,
    "Reference_Key1": "ORDER-10045",
    "Reference_Key2": null,
    "With_Return_Voucher": null,
    "Content_Type_ID": null,
    "Language": "EN"
  }
}
```

Example response

Response — voucher created successfully

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      {
        "Voucher_No": "7227889174",
        "Voucher_No_Return": null,
        "Error_Message": ""
      }
    ],
    "ACSTableOutput": {}
  }
}
```

Response field	Description
Voucher_No	The number of the voucher created. Store it: you need it for printing, deletion and tracking. We recommend trimming any whitespace before storing it.
Voucher_No_Return	The number of the return voucher, when With_Return_Voucher = 1. Otherwise null .
Error_Message	Empty or null on success. Otherwise, the error message (see “Error messages” below).

Request parameters

Parameters are presented by group. Send any you do not use with the value **null**. In the “Required” column, “Conditional” means the parameter is required only in the case described. In the “Type” column, a number in parentheses is the maximum length (in characters), e.g. **string (50)**.

Access credentials

Field	Type	Required	Description
Company_ID	string	Yes	Company ID, provided by ACS.
Company_Password	string	Yes	Company password, provided by ACS.
User_ID	string	Yes	User ID, provided by ACS.
User_Password	string	Yes	User password, provided by ACS.

Pickup and sender

Field	Type	Required	Description
Pickup_Date	string (date)	Yes	Date of pickup by the courier, format YYYY-MM-DD . Past dates, Sundays and public holidays are not accepted.
Sender	string	Yes	Sender name as it will appear on the voucher.

Recipient

Field	Type	Required	Description
Recipient_Name	string (50)	Yes	Recipient full name or company name. Greek or Latin characters only.
Recipient_Company_Name	string	No	Recipient company name, if any.
Recipient_Address	string (50)	Yes	Recipient street, without number, area or locality. Greek or Latin characters only.
Recipient_Address_Number	integer	Yes	Street number.
Recipient_Zipcode	integer (5)	Yes	Postal code, without spaces, hyphens or other characters: 5 digits for Greece (e.g. 17778 , not 177 78), 4 digits for Cyprus (e.g. 1010). Must match the country in Recipient_Country .
Recipient_Region	string (50)	Yes	Area or locality. Greek or Latin characters only.
Recipient_Floor	string	No	Floor.
Recipient_Country	string	Yes	Destination country: GR (Greece) or CY (Cyprus).
Recipient_Phone	integer or string (25)	Conditional	Landline phone of a Greek or Cypriot provider . Without a prefix, send it as a number (e.g. 2115005000). With a prefix, it must be sent as a string (Greece: " +302115005000 " or "00302115005000" , Cyprus: " +35722123456 " or "0035722123456"). At least one of the two phone numbers is required.
Recipient_Cell_Phone	integer or string (25)	Conditional	Mobile phone of a Greek or Cypriot provider . Without a prefix, send it as a number (e.g. 6999999999). With a prefix, it must be sent as a string (Greece: " +306999999999 " or "00306999999999" , Cyprus: " +35799123456 " or "0035799123456"). Always recommended , because it enables automatic SMS notification of the recipient (e.g. after a missed delivery notice). Required for delivery to an ACS Smartpoint.
Recipient_Email	string (100)	Conditional	Recipient email.

Recipient data format

Name, street, area (**Recipient_Name**, **Recipient_Address**, **Recipient_Region**): Greek or Latin characters only.

Postal code (**Recipient_Zipcode**): no spaces, hyphens or other characters; 5 digits for Greece, 4 digits for Cyprus.

Phone numbers (**Recipient_Phone**, **Recipient_Cell_Phone**): numbers of a Greek or Cypriot provider. The prefix is optional; if present, send the value in quotes (string), e.g. **" +306999999999 "** / **"00306999999999 "** for Greece or **" +35799123456 "** / **"0035799123456 "** for Cyprus.

ACS destination store

Field	Type	Required	Description
Acs_Station_Destination	string	No	Code (2 letters) of the ACS store that will make the delivery. Usually null : the system determines it from the postal code. Fill it in for delivery to an ACS Smartpoint (see §2.3).

Field	Type	Required	Description
Acs_Station_Branch_Destination	integer	No	Usually null . For delivery to an ACS Smartpoint, the Smartpoint code (e.g. 502 , see §2.3).

Billing

Field	Type	Required	Description
Billing_Code	string	Yes	The customer's (credit) billing code, provided by ACS.
Charge_Type	integer	Yes	Who pays for the shipping: 2 = sender, 4 = recipient.
Cost_Center_Code	string	No	Cost center of the billing code, if any.

Parcel

Field	Type	Required	Description
Item_Quantity	integer	Yes	Number of items (parcels), from 1 to 99. A value greater than 1 creates a multi-part shipment (see §2.4).
Weight	decimal	Yes	Weight in kg, from 0.5 to 999. Up to 2 decimal places.
Dimension_X_In_Cm Dimension_Y_in_Cm Dimension_Z_in_Cm	integer	No	Dimensions in centimeters. Volumetric weight is not calculated: the voucher always shows Weight .

Cash on delivery and insurance

Field	Type	Required	Description
Cod_Ammount	decimal	Conditional	Cash on delivery (COD) amount in €. If there is no COD: null . When filled in, Acs_Delivery_Products must contain COD . Up to 2 decimal places.
Cod_Payment_Way	integer	Conditional	COD payment method: 0 = cash, 1 = check. If there is no COD: null .
Insurance_Ammount	decimal	No	Insurance amount in €, up to 3000. A higher amount requires a business insurance commercial agreement. Otherwise null . Up to 2 decimal places.

Additional services

Field	Type	Required	Description
Acs_Delivery_Products	string	No	Additional service codes, comma-separated, e.g. "COD,SAT" . Order does not matter. See the table "Additional service codes".
Appointment_Until_Time	string	No	Delivery time commitment, format HH:MM . Up to 10:00, Morning delivery (MDV) is added automatically; after 10:00, the 2-hour delivery appointment (TDD). May incur an extra charge.
With_Return_Voucher	integer	No	1 = also create a return voucher, null = no. Applies only to the Document Return Service (RDO), where the courier delivers and at the same time collects something back. It does not apply to product returns after delivery.
Delivery_Notes	string	No	Notes for the courier (free text).

References and other

Field	Type	Required	Description
Reference_Key1	string	No	Your own reference key (e.g. order number). Recommended: it lets you find the voucher later (see Chapter 5, ACS_POD_FROM_REFERENCE_NO). For the search to work, the value must be unique per shipment.
Reference_Key2	string	No	Second reference key. Can also be used for searching with ACS_POD_FROM_REFERENCE_NO .
Content_Type_ID	integer	Conditional	Shipment content code. Required for shipments to Cyprus (CY) . The values are returned by the ACS_Get_Content_Types method (Chapter 3).
Language	string	No	Message language: null or GR = Greek, EN = English.

Field names and values

Use field names **exactly** as shown (they are case-sensitive), e.g. `Cod_Ammount` and `Insurance_Ammount` (with two `ms`), `Dimension_X_In_Cm` but `Dimension_Y_in_Cm`.

Numeric values (`Cod_Ammount`, `Weight`, `Insurance_Ammount`, `Cod_Payment_Way`, `Charge_Type`, `Item_Quantity`) are sent **without quotes** and with a **period** as the decimal separator.

Additional service codes (Acs_Delivery_Products)

Code	Service	Applies to	Extra charge
<code>COD</code>	Cash on delivery	Greece, Cyprus	Possibly
<code>INS</code>	Shipment insurance	Greece, Cyprus	Possibly
<code>SAT</code>	Saturday delivery	Greece, Cyprus	Possibly
<code>MDV</code>	Morning delivery	Greece, Cyprus	Possibly
<code>TDD</code>	2-hour delivery appointment	Greece, Cyprus	Possibly
<code>PRO</code>	Protocol	Greece, Cyprus	Possibly
<code>RDO</code>	Return documents (requires <code>With_Return_Voucher = 1</code>)	Greece, Cyprus	—
<code>REC</code>	Reception delivery (store pickup)	Greece, Cyprus	—
<code>CEC</code>	Cyprus Economy	Cyprus only	—
<code>P2P</code>	Point to point	Cyprus only	—
<code>D2P</code>	Door to point	Cyprus only	—
<code>P2D</code>	Point to door	Cyprus only	—

Combination rules:

- Multiple codes are given comma-separated, in any order: `"COD,SAT"` or `"SAT,COD"`.
- `REC` **cannot** be combined with `SAT`, `MDV` or `TDD`.
- For remote destinations (DP: remote area, with surcharge / DX: remote area, no surcharge), `SAT`, `MDV` and `TDD` are not allowed. `SAT` is also not supported at all destinations.
- If you fill in `Cod_Ammount` and `Cod_Payment_Way`, the `COD` code must also be present.

Error messages

Returned in the `Error_Message` field of the response. The texts may change in the future, so do not base your application logic on their exact wording.

Message	Cause / what to check
Invalid pick-up date	<code>Pickup_Date</code> is invalid or in the past.
Pickup date is not allowed on Sunday or national holiday	<code>Pickup_Date</code> is a Sunday or a public holiday.
The recipient's name cannot be empty	<code>Recipient_Name</code> is empty.
The recipient's address cannot be empty	<code>Recipient_Address</code> is empty.
Unacceptable zip code or country	<code>Recipient_Zipcode</code> is wrong or does not match <code>Recipient_Country</code> (e.g. <code>GR</code> with a Cypriot postal code).
Unacceptable ACS Destination Value	Wrong value in <code>Acs_Station_Destination</code> and/or <code>Acs_Station_Branch_Destination</code> .
Not more than 99 pieces per shipment are supported	<code>Item_Quantity</code> > 99.
Unacceptable weight value (0.5-999)	<code>Weight</code> out of range.
Unacceptable shipping charge value	<code>Charge_Type</code> is not 2 or 4.
Unacceptable payment method for COD	<code>Cod_Payment_Way</code> is not 0 or 1.
Cannot find the COD	A COD amount and payment method are given, but the <code>COD</code> code is missing from <code>Acs_Delivery_Products</code> .
No valid credit code	<code>Billing_Code</code> is wrong.
The Acs-SmartPoint destination must have 1 mobile phone	Delivery to a Smartpoint without <code>Recipient_Cell_Phone</code> .
For an Acs-SmartPoint destination with cash on delivery (COD) there must be a recipient email	Delivery to a Smartpoint parcel locker with COD but without <code>Recipient_Email</code> .
Acs-SmartPoint destination can not be combined with multiple shipment (items>1)	Delivery to a Smartpoint with <code>Item_Quantity</code> > 1.

Message	Cause / what to check
The product "RVO" is only combined with return voucher (with_return = 1)	The RDO code is present but With_Return_Voucher is not 1 .
You cannot create Cyprus Economy (EC) shipments with the billing code: 2XXXXXXX	The Billing_Code is not enabled for Cyprus Economy.
You cannot combine these products together	Invalid combination of codes, e.g. TDD, REC or MDV, REC .
The destination was found as a remote area(REM) and can not be combined with the other Acs products you entered.	Remote destination with SAT, MDV or TDD .
The product 5Σ is not supported for this destination.	SAT (Saturday delivery) is not available for the destination postal code.
For shipments from Greece to Cyprus the shipment Content_Type_ID field must have a valid value	Shipment to CY with a missing or wrong Content_Type_ID .

2.3 Delivery to an ACS store or ACS Smartpoint

The recipient can collect the shipment from an ACS Point instead of at their door. ACS Points are the ACS stores and the ACS Smartpoints. You use the same **ACS_Create_Voucher** method, with the differences described below. You get the codes of all ACS Points from the **ACS_Stations** method (Chapter 8).

A. Collection from an ACS store

Add the **REC** code to **Acs_Delivery_Products**. If there is also cash on delivery: **"REC,COD"**.

Request excerpt — collection from an ACS store

```
"Acs_Delivery_Products": "REC,COD"
```

B. Delivery to an ACS Smartpoint

Fill in the code of the Smartpoint chosen by the recipient in the two destination fields:

Request excerpt — delivery to an ACS Smartpoint

```
"Acs_Station_Destination": "AK",
"Acs_Station_Branch_Destination": 502,
"Recipient_Cell_Phone": 6999999999,
"Item_Quantity": 1
```

Requirements for a Smartpoint shipment:

- Recipient **mobile phone** (**Recipient_Cell_Phone**): required.
- Recipient **email** (**Recipient_Email**): required when the shipment has cash on delivery.
- **One item** only (**Item_Quantity = 1**): multi-part shipments are not allowed.

2.4 Multi-part shipment — ACS_Get_Multipart_Vouchers

When the contents of a shipment do not fit in one parcel, you create a **multi-part shipment** by passing an **Item_Quantity** greater than 1 to **ACS_Create_Voucher**. A multi-part shipment consists of one **main voucher** and the **sub-vouchers**. All items are delivered together and the sender is charged for one shipment.

ACS_Create_Voucher returns only the number of the main voucher. If you also need the numbers of the sub-vouchers, call the **ACS_Get_Multipart_Vouchers** method with the main voucher number.

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Main_Voucher_No	integer	Yes	The main voucher number, as returned by ACS_Create_Voucher .
Language	string	No	null or GR = Greek, EN = English.

Example

The main voucher **7227890681** was created with **Item_Quantity = 2**. The method returns the number of the sub-voucher:

Request — ACS_Get_Multipart_Vouchers

```
{
  "ACSAlias": "ACS_Get_Multipart_Vouchers",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Main_Voucher_No": 7227890681,
    "Language": "EN"
  }
}
```

Response — ACS_Get_Multipart_Vouchers

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_Message": null }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        { "MultiPart_Voucher_No": "8052453426" }
      ]
    }
  }
}
```

Response field	Description
<code>Table_Data[].MultiPart_Voucher_No</code>	The number of each sub-voucher. One row is returned for each additional item (that is, <code>Item_Quantity</code> – 1 rows).
<code>Error_Message</code>	Empty or <code>null</code> on success. Otherwise the error message.

2.5 Common scenarios

The excerpts below show only the fields that differ from the full example in §2.2.

Simple shipment, no cash on delivery

```
"Cod_Amount": null,
"Cod_Payment_Way": null,
"Acs_Delivery_Products": null
```

Shipping charged to the recipient

```
"Charge_Type": 4
```

Shipment of 3 parcels (multi-part shipment)

```
"Item_Quantity": 3,
"Weight": 12.5
```

Shipment to Cyprus

```
"Recipient_Country": "CY",
"Recipient_Zipcode": 1010,
"Content_Type_ID": 14
```

In this example, code `14` corresponds to `CLOTHING` (see §3.3).

Return of documents

```
"Acs_Delivery_Products": "RDO",
"With_Return_Voucher": 1
```

3. Printing and managing vouchers

3.1 Print a voucher — ACS_Print_Voucher_V2

Returns the vouchers in a **single PDF file**, ready to print and attach to the parcels. The PDF format adapts to the customer's printer: **laser** (A4 page with 3 labels) or **thermal** (label roll). When you request multiple vouchers, they are all returned together in the same PDF.

Print before the pickup list

Vouchers must be printed **before** the pickup list is issued (Chapter 4). Once the list has been issued, they can no longer be printed.

Example request

Printing two vouchers on a thermal printer:

Request — ACS_Print_Voucher_V2

```
{
  "ACSAlias": "ACS_Print_Voucher_V2",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Voucher_No": "7227889174,7227889175",
    "Language": "EN",
    "Print_Type": 1,
    "Start_Position": 1
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Voucher_No	string	Yes	The voucher number. For bulk printing, up to 10 comma-separated numbers, e.g. "7227889174,7227889175".
Print_Type	integer	Yes	Printer type: 1 = thermal, 2 = laser (A4 page).
Start_Position	integer	Conditional	The position of the first label on the A4 page: 1 , 2 or 3 . Applies only to laser (Print_Type = 2). Lets you use pages whose remaining labels are still blank.
Language	string	No	Message language: null or GR = Greek, EN = English.

What gets printed

In some cases, printing a voucher automatically returns the related vouchers as well. Provide **only the main voucher number**:

Case	What you pass in Voucher_No	What is returned
Simple shipment	The voucher number	The voucher
Multi-part shipment (Item_Quantity > 1)	The main voucher number	The main voucher and all sub-vouchers
With return voucher (With_Return_Voucher = 1)	The main voucher number	The main and the return voucher

Response

The PDF is returned in the **ACSObjectOutput** object, inside the **ACSValueOutput** list, encoded in **Base64**. The format is the same whether you request one or many vouchers.

Response — ACS_Print_Voucher_V2 (Base64 truncated)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOuputResponse": {
    "ACSValueOutput": [
      {
        "ACSObjectOutput": {
          "Voucher_No": "7227889174,7227889175",
          "PDFData": "JVBERi0xLjQNCiWio4+TDQo2IDAgb2JqDQo8PC9U...=="
        }
      }
    ],
    "ACSTableOutput": {}
  }
}
```

Response field	Type	Description
ACSObjectOutput.Voucher_No	string	The numbers of the vouchers contained in the PDF, comma-separated.
ACSObjectOutput.PDFData	string	The PDF file in Base64. The Base64 text of a PDF always starts with JVBERi (corresponding to “%PDF”).

Decode **PDFData** to bytes and save it as a **.pdf** file or send it directly to the printer. For example:

Saving the PDF — C#

```
// C# (.NET) — with System.Text.Json
using var doc = JsonDocument.Parse(responseJson);
var obj = doc.RootElement.GetProperty("ACSOuputResponse")
    .GetProperty("ACSValueOutput")[0]
    .GetProperty("ACSObjectOutput");
byte[] pdf = Convert.FromBase64String(obj.GetProperty("PDFData").GetString());
File.WriteAllBytes("vouchers.pdf", pdf);
```

Saving the PDF — Python

```
# Python
import base64
obj = data["ACSOuputResponse"]["ACSValueOutput"][0]["ACSObjectOutput"]
with open("vouchers.pdf", "wb") as f:
    f.write(base64.b64decode(obj["PDFData"]))
```

Saving the PDF — PHP

```
// PHP
$data = json_decode($responseJson, true);
$obj = $data['ACSOuputResponse']['ACSValueOutput'][0]['ACSObjectOutput'];
file_put_contents('vouchers.pdf', base64_decode($obj['PDFData']));
```

Print samples

 ΑΡΙΘ. ΑΠΟΣΤΟΛΗΣ 7400271992		DEMO ACS CONNECT ΕΑΡΑ: ΑΣΚΗΠΙΟΥ 25 ΑΦΜ: 094360202 ΤΚ: 14565 ΔΟΥ: ΦΑΕ ΑΘΗΝΩΝ	
ΑΠΟΣΤΟΛΕΑΣ ΟΝΟΜΑ: TEST ΔΙΕΥ/ΝΣΗ-ΥΠ/ΜΑ: ΑΣΚΗΠΙΟΥ 25 ΤΚ / ΠΟΛΗ: 14565 / ΑΘΗΝΑ ΤΗΛ. / ΚΙΝ.: 8190000		ΚΩΔ. ΠΕΛΑΤΗ: 2Α0999999 ΗΜΕΡΟΜΗΝΙΑ: 14/06/2021 ΒΑΡΟΣ (KG): 1,00 ΤΕΜΑΧΙΑ: 1 ΠΟΣΟ ΑΝΤ/ΛΗΣ: ΠΑΡΑΔΟΣΗ ΜΕΤΑΞΥ: ΚΥΡΙΟ ΑΝΘ ΟΔΗΓΙΕΣ:	
ΠΑΡΑΛΗΠΤΗΣ ΟΝΟΜΑ: TEST RECIPIENT ΕΤΑΙΡΙΑ / ΟΡΟΦΟΣ: ΟΔΟΣ/ΑΡ.: Π. ΡΑΛΛΗ 45 ΤΚ / ΠΟΛΗ: 17778 / ΤΑΥΡΟΣ ΤΗΛ. / ΚΙΝ.: 2108190000		ΠΑΡΑΤΗΡΗΣΕΙΣ:	
ACS Connect ACS H.A.E.E. — Ασκήπιο 25 Τ.Κ. 14568 Κεντρικά Αθηνών ΑΦΜ: 094360202 ΔΟΥ: ΦΑΕ Α.Ε. ΑΘΗΝΩΝ, Τηλ. 210-8190000, Fax: 210-8190261		ΕΠΤ.Α.Μ. : 99-122 Εθνική & Γενική Δύση Ταχυδρομικών Υπηρεσιών	
		  Τα πάντα, παντού.	ΠΡΟΟΡΙΣΜΟΣ 

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

Laser — A4 page with 3 labels (Print_Type = 2)

Αρ. Αποστολής/AWB No 7400271992	Προέλευση/Origin ΑΘ	Τεκ./Pieces 1	Βάρος/Weight 1,00	Κωδ. Πηλ./Cust Code 2ΑΘ999999	Ημερομηνία/Date 14/06/2021	Courier	Παράδοση/Time 12:07	Όργανο ΗΥ/PC ACSIS19
ΑΠΟΣΤΟΛΕΑΣ / SENDER		ΛΟΓΙΑ ΣΤΟΙΧΕΙΑ/ADD.INFO						
Όνομα TEST		Χρέωση/Debit: ΑΠΟΣΤΟΛΕΑ			Προϊόντα/Products:			
Διεύθυνση Address ΑΣΚΛΗΠΙΟΥ 25		Ταχ.Κωδ. Zip Code 14565			Αντ/An/C.o.d ☐ Αγορά/Buy ☐			
Πόλη City ΑΘΗΝΑ		Τηλ. Tel. 8190000			Προσδ/Total:			
Επάγγελμα Profession ΤΑΧΥΜΕΤΑΦΟΡΕΣ		Δ.Ο.Υ. A.F.M. ΦΑΕ ΑΘΗΝΩΝ			Παράδοση Έως/Delivery Until:			
ΕΓΓΡΑΦΩΝ & ΔΕΜΑ		094360202			ΑΝΘ Επιστροφής: Κόστος AWB:			
ΠΑΡΑΛΗΠΤΗΣ / RECEIVER		ΠΑΡΑΤΗΡΗΣΕΙΣ/NOTES						
Όνομα TEST RECIPIENT		Εταιρεία/Όργανο Company/Storey /			Προορισμός Destination ΑΘ			
Διεύθυνση Address Π. ΡΑΛΛΗ 45		Ταχ.Κωδ. Zip Code 17778			www.acscourier.net			
Πόλη City ΤΑΥΡΟΣ		Τηλ. Tel. 2108190000			ΕΤΕΤ Α.Μ.: 09-122			
ΑΦΜ/094360202 ΔΟΥ-Φ.Α.Ε. ΑΘΗΝΩΝ, Τηλ. 210-8190000 Fax 210-8190					Εδωκή & Γενική Πάση Ταχυδρομικών Υπηρεσιών			
*Οι χρονικές προημιές παράδοσης δεν δεσμεύουν την ACS εάν δεν συνδυάζονται με την αντίστοιχη επιλογή υπηρεσίας.								

Thermal printer — label roll (Print_Type = 1)

3.2 Delete a voucher — ACS_Delete_Voucher

Deletes one or more vouchers, for example when a shipment is canceled before it is handed over to the courier.

Only before the pickup list

Deletion is possible **only** for vouchers that have not been included in a pickup list (Chapter 4). Once the list has been issued, the voucher is considered final: it can only be deleted by ACS, at the sender's request to the ACS store that serves them or to ACS headquarters.

Example request

Request — ACS_Delete_Voucher

```
{
  "ACSAlias": "ACS_Delete_Voucher",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Voucher_No": "7227889480",
    "Language": "EN"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Voucher_No	string	Yes	The voucher number. For bulk deletion, up to 20 comma-separated numbers, e.g. "7227889480,7227889481".
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

ACSTableOutput.Table_Data contains **one row for each voucher** in the call. If a row's **error_message** is **null**, the voucher was deleted; otherwise it contains the reason for the failure.

A. All vouchers deleted

Response — successful deletion

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_Message": null }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        { "voucher_no": "9815571404", "error_message": null },
        { "voucher_no": "9815571662", "error_message": null }
      ]
    }
  }
}
```

B. Some vouchers not deleted

Error_Message contains one of the errors, while **Table_Data** shows the reason for each voucher:

Response — bulk deletion with failures

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_Message": "No.9815541256 was not found for deletion." }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "voucher_no": "9815537756",
          "error_message": "Cannot delete ACS voucher when the courier pickup list is issued"
        },
        {
          "voucher_no": "9815541256",
          "error_message": "No.9815541256 was not found for deletion."
        }
      ]
    }
  }
}
```

Response field	Description
Error_Message	Empty or null : all vouchers were deleted. Otherwise, one of the errors of the call.
Table_Data[].voucher_no	The voucher number.
Table_Data[].error_message	null : the voucher was deleted. Otherwise, the reason it was not deleted.

How to check the deletion result

Check the **error_message** of **each row** in **Table_Data**: it is the reliable indicator of whether that specific voucher was deleted. In a bulk deletion, **Error_Message** shows only one error.

Note that the **Table_Data** fields are written in **lowercase** (**voucher_no**, **error_message**).

Notes

- **Multi-part shipment:** deleting the main voucher automatically deletes all its sub-vouchers. You do not need to provide them separately.
- **Bulk deletion:** up to 20 vouchers per call. For more, make successive calls.

Error messages

Returned in **Error_Message** and, for bulk deletion, in **Table_Data**. The texts may change in the future.

Message	Cause / what to do
Cannot delete ACS voucher when the courier pickup list is issued	The voucher has been included in a pickup list and is now final. Contact the ACS store that serves you or ACS headquarters.
No.... was not found for deletion.	The voucher number does not exist or has already been deleted. Check the number.

3.3 Content of shipments to Cyprus — ACS_Get_Content_Types

For shipments to Cyprus (`Recipient_Country = "CY"`), declaring the type of content is **mandatory**, by decision of the Cyprus customs authority. This method returns the available content codes. Send the matching code in the `Content_Type_ID` field of `ACS_Create_Voucher` (§2.2).

Consequences of an incomplete declaration

A shipment to Cyprus without a correct `Content_Type_ID` is not created. In addition, an incorrect content description may cause delays and fines from Larnaca customs.

Example request

Request — ACS_Get_Content_Types

```
{
  "ACSAlias": "ACS_Get_Content_Types",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Language	string	No	Message language: <code>null</code> or <code>GR</code> = Greek, <code>EN</code> = English.

Response

The codes are returned as a list in `ACSTableOutput.Table_Data`:

Response — ACS_Get_Content_Types (abridged)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_Message": null }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        { "content_type_id": 1, "content_type_descr": "SPARE PARTS" },
        { "content_type_id": 2, "content_type_descr": "BOOKS" },
        ...
        { "content_type_id": 18, "content_type_descr": "OTHER" }
      ]
    }
  }
}
```

Response field	Type	Description
<code>Table_Data[].content_type_id</code>	integer	The content code. This is what you send in <code>Content_Type_ID</code> of <code>ACS_Create_Voucher</code> .
<code>Table_Data[].content_type_descr</code>	string	The content description.

Content codes

The codes valid at the time of writing, with the descriptions returned when `Language = EN`:

Code	Content	Code	Content
1	SPARE PARTS	10	DRY FOOD
2	BOOKS	11	OPTICAL
3	VITAMINS, PARAPHARMACEUTICALS	12	TOYS-GAMES

Code	Content	Code	Content
4	DECORATION ITEMS	13	PERSONAL ITEMS
5	BEAUTIFICATION ITEMS	14	CLOTHING
6	ELECTRONICS	15	SHOES
7	COSMETICS	16	PHARMACEUTICALS
8	CELL PHONE	17	DOCUMENTS - SECURITIES
9	FEAU BIJOUX JEWELRY	18	OTHER

Recommended implementation

The list may change. Call the method periodically (e.g. once a day) and store the codes in your system, instead of hard-coding them. This also lets you map your e-shop's product categories to the ACS codes.

4. Pickup lists

4.1 Overview

The **pickup list** (or manifest) is the summary document of the day's shipments, handed to the courier together with the parcels. Issuing it **finalizes** the vouchers: without a pickup list, the vouchers are not recognized by ACS systems.

Method (ACSAlias)	Use	Section
ACS_Issue_Pickup_List	Issues the pickup list. Returns the list's mass number.	4.2
ACS_Print_Pickup_List	Prints the pickup list as a PDF.	4.3
ACS_Get_Pickup_Lists	The pickup lists issued on a given date (e.g. for reprinting).	4.4
ACS_Pickup_List_Display_Voucher	The vouchers contained in a pickup list.	4.5

The daily flow

1. Create and **print** all of the day's vouchers (Chapters 2 and 3).
2. At the end of the day, **issue** the pickup list: [ACS_Issue_Pickup_List](#).
3. **Print** the list using the mass number you received: [ACS_Print_Pickup_List](#).
4. Hand the parcels to the courier together with the printed list.

After the list is issued

Vouchers included in a pickup list are **final**: they can no longer be printed or deleted through the API (see §3.1 and §3.2). Issue the list only when you have completed the day's shipments.

4.2 Issue pickup list — [ACS_Issue_Pickup_List](#)

Closes the shipments of the pickup date into one list and returns its **mass number** ([PickupList_No](#)), which you need to print it.

Example request

Request — [ACS_Issue_Pickup_List](#)

```
{
  "ACSAlias": "ACS_Issue_Pickup_List",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN",
    "Pickup_Date": "2026-10-13",
    "MyData": null
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID , Company_Password , User_ID , User_Password	string	Yes	Access credentials (see §1.3).

Field	Type	Required	Description
Pickup_Date	string (date)	Yes	Date of the list, format YYYY-MM-DD. Must be the same as the Pickup_Date of the vouchers you want to include.
MyData	integer	No	Whose shipments go into the list: 0 or null = all users of the company, 1 = only the calling user (User_ID). Useful only when the company has more than one user.
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

A. Successful issue

Response — successful issue

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      {
        "PickupList_No": "7227889830",
        "Unprinted_Found": 0,
        "Error_Message": ""
      }
    ],
    "ACSTableOutput": {
      "Table_Data": []
    }
  }
}
```

B. Unprinted vouchers exist

If there are vouchers for the day that have not been printed, the list is **not issued**. The response returns how many there are and their numbers:

Response — unprinted vouchers

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      {
        "PickupList_No": null,
        "Unprinted_Found": 2,
        "Error_Message": "You cannot issue a pick list. Found 2 unprinted vouchers"
      }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        { "Unprinted_Vouchers": "7227889841" },
        { "Unprinted_Vouchers": "7227889874" }
      ]
    }
  }
}
```

Response field	Description
PickupList_No	The list's mass number. Store it: you need it for printing (§4.3). null if the list was not issued.
Unprinted_Found	Number of unprinted vouchers. 0 on success.
Error_Message	Empty on success. Otherwise, the reason for the failure.
Table_Data[].Unprinted_Vouchers	The number of each unprinted voucher.

What to do with unprinted vouchers

For each voucher in the **Unprinted_Vouchers** list, either **print** it (§3.1) or **delete** it (§3.2) if the shipment will not be sent. Then call **ACS_Issue_Pickup_List** again.

Error messages

Returned in the **Error_Message** field. The texts may change in the future.

Message	Cause / what to do
You cannot issue a pick list. Found X unprinted vouchers	There are vouchers for the day that have not been printed. Print or delete them and repeat the call.

4.3 Print pickup list — ACS_Print_Pickup_List

Returns the pickup list as a **PDF**, for printing and handing to the courier.

Example request

Request — ACS_Print_Pickup_List

```
{
  "ACSAlias": "ACS_Print_Pickup_List",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN",
    "Mass_Number": "7227889830",
    "Pickup_Date": "2026-10-13"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Mass_Number	string	Yes	The list's mass number: the PickupList_No returned by ACS_Issue_Pickup_List (§4.2) or ACS_Get_Pickup_Lists (§4.4).
Pickup_Date	string (date)	Yes	The date of the list, format YYYY-MM-DD.
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

The list's PDF is returned in the **ACSObjectOutput** object, encoded in **Base64**:

Response — ACS_Print_Pickup_List (Base64 truncated)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      {
        "ACSObjectOutput": {
          "Mass_Voucher_No": "7227889830",
          "PDFData": "JVBERi0xLjQNCiWio4+TDQo2IDAgb2JqDQo8PC9U...==",
          "ACSExecutionErrorMessage": "",
          "ACSExecution_HasError": false
        }
      }
    ],
    "ACSTableOutput": {}
  }
}
```

Field (in ACSObjectOutput)	Type	Description
Mass_Voucher_No	string	The mass number of the printed list.
PDFData	string	The list's PDF in Base64.
ACSExecution_HasError	boolean	true if printing the list failed.
ACSExecutionErrorMessage	string	The reason the printing failed.

Double error check

In addition to the **ACSExecution_HasError** of the overall response, also check the **ACSExecution_HasError** inside **ACSObjectOutput**: it refers specifically to generating the list's PDF.

Sample code for saving the PDF:

Save PDF — C#

```
// C# (.NET) — with System.Text.Json
using var doc = JsonDocument.Parse(responseJson);
var obj = doc.RootElement.GetProperty("ACSOutputResponse")
    .GetProperty("ACSValueOutput")[0]
    .GetProperty("ACSObjectOutput");
if (!obj.GetProperty("ACSExecution_HasError").GetBoolean())
{
    string no = obj.GetProperty("Mass_Voucher_No").GetString();
    File.WriteAllBytes($"pickup_{no}.pdf",
        Convert.FromBase64String(obj.GetProperty("PDFData").GetString()));
}
```

Save PDF — Python

```
# Python
import base64
obj = data["ACSOutputResponse"][0]["ACSObjectOutput"]
if not obj["ACSExecution_HasError"]:
    with open(f'pickup_{obj["Mass_Voucher_No"]}.pdf', "wb") as f:
        f.write(base64.b64decode(obj["PDFData"]))
```

Save PDF — PHP

```
// PHP
$data = json_decode($responseJson, true);
$obj = $data['ACSOutputResponse'][0]['ACSObjectOutput'];
if (!$obj['ACSExecution_HasError']) {
    file_put_contents('pickup_' . $obj['Mass_Voucher_No'] . '.pdf',
        base64_decode($obj['PDFData']));
}
```

Sample pickup list

The top left shows the barcode of the mass number (**PickupList_No**), which the courier scans at pickup.

Μαζικό Αποδεικτικό



7227889933

ΣΥΝΟΠΤΙΚΗ ΚΑΤΑΣΤΑΣΗ ΠΑΡΑΛΑΒΗΣ

ΑΠΟΣΤΟΛΩΝ ΠΡΟΣ ΜΕΤΑΦΟΡΑ



Τα πάντα, παντού.

Αρ.Αποστολής	Αποστολέας			Όνομα Παραλήπτη			Διεύθυνση/Περιοχή			T.K	Τηλέφωνο	Κωδ.Πελ.	Προ/σμός
Κύριος.Αρ.Απ.	Ημ.Παράλ.	Τεκ.	Βάρος	Παράδ.Έως	Ποσό Αντ.	Τρ.Πλ.	Ποσό Ασφ	Προϊόντα	Σχετικό 1			Παρατηρήσεις	
7227889900	ESHOP			T*** R*****			P. RALLI 45 TAVROS	17778	2*****00	2ΑΘ999999	BT		
	11/01/2019	1	0,50	00:00	0,00 €	M	0,0 €						
7227889911	ESHOP			T*** R*****			P. RALLI 45 TAVROS	17778	2*****00	2ΑΘ999999	BT		
	11/01/2019	1	0,50	00:00	0,00 €	M	0,0 €						
7227889922	ESHOP			T*** R*****			P. RALLI 45 TAVROS	17778	2*****00	2ΑΘ999999	BT		
	11/01/2019	1	0,50	00:00	0,00 €	M	0,0 €						
ΣΥΝΟΛΟ ΑΠΟΣΤΟΛΩΝ: 3				Ο ΠΑΡΑΔΟΥΣ				Ο ΠΑΡΑΛΑΒΩΝ					
ESHOP													

Sample printed pickup list

4.4 Pickup lists of a day — ACS_Get_Pickup_Lists

Returns all pickup lists issued on a given date, with their mass numbers. Used mainly to **reprint** a list when you have not kept its number.

Example request

Request — ACS_Get_Pickup_Lists

```
{
  "ACSAlias": "ACS_Get_Pickup_Lists",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN",
    "Pickup_Date": "2026-10-13"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID , Company_Password , User_ID , User_Password	string	Yes	Access credentials (see §1.3).
Pickup_Date	string (date)	Yes	The date of the lists, format YYYY-MM-DD . Up to 6 months back.
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

In the example, two lists were issued for the date:

Response — ACS_Get_Pickup_Lists

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_Message": null }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "Pickup_date": "2026-10-13T00:00:00",
          "Pickup_List_DateTime": "2026-10-13T11:05:03.943",
          "User_ID": "demo",
          "PickupList_No": "7227890935",
          "List_Vouchers_Count": 2
        },
        {
          "Pickup_date": "2026-10-13T00:00:00",
          "Pickup_List_DateTime": "2026-10-13T11:05:34.11",
          "User_ID": "demo",
          "PickupList_No": "7227890950",
          "List_Vouchers_Count": 1
        }
      ]
    }
  }
}
```

Field (per list)	Description
PickupList_No	The list's mass number. Pass it in Mass_Number of ACS_Print_Pickup_List to reprint.
Pickup_date	The pickup date of the list.
Pickup_List_DateTime	Date and time the list was issued.
User_ID	The user who issued the list.
List_Vouchers_Count	Number of shipments in the list.

Reprinting lists

ACS_Print_Pickup_List accepts one mass number per call. To reprint several lists, call it once for each **PickupList_No**. Reprinting is possible for lists up to 6 months old.

4.5 Shipments in a pickup list — ACS_Pickup_List_Display_Voucher

Returns the vouchers contained in a pickup list, together with the reference keys (**Reference_Key1**, **Reference_Key2**) you provided when creating them. Useful for reconciling with the orders in your system.

Example request

Request — ACS_Pickup_List_Display_Voucher

```
{
  "ACSAlias": "ACS_Pickup_List_Display_Voucher",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN",
    "PickupList_No": "7227890950",
    "Pickup_Date": "2026-10-13"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
PickupList_No	string	Yes	The list's mass number (§4.2 or §4.4).
Pickup_Date	string (date)	Yes	The date of the list, format YYYY-MM-DD.
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

Response — ACS_Pickup_List_Display_Voucher

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOuputResponse": {
    "ACSValueOutput": [
      {
        "List_Vouchers_Count": 1,
        "Error_Message": null
      }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "Voucher_no": "7227889174",
          "Reference_Key1": "ORDER-10045",
          "Reference_Key2": null
        }
      ]
    }
  }
}
```

Response field	Description
List_Vouchers_Count	Number of shipments in the list.
Table_Data[].Voucher_no	The voucher number. Note: it is spelled Voucher_no (with a lowercase n).
Table_Data[].Reference_Key1 Table_Data[].Reference_Key2	The reference keys provided in ACS_Create_Voucher . null if none were provided.

5. Shipment tracking

5.1 Overview

Method (ACSAlias)	Use	Section
ACS_Trackingsummary	The current status of a shipment: whether it was delivered, when and to whom, or why it was not delivered.	5.2
ACS_TrackingDetails	The full history of a shipment's movements, from label printing to delivery.	5.3

Method (ACSAlias)	Use	Section
ACS_POD_FROM_REFERENCE_NO	Finds the voucher integer from your own reference key (e.g. order integer).	5.4

Which method to use

To update the status of your orders automatically, use **ACS_Trackingsummary**: it returns everything you need in a single record. Use **ACS_TrackingDetails** when you want to show the detailed history to your customer.

5.2 Tracking summary — ACS_Trackingsummary

Returns the latest status update of a shipment: origin and destination, pickup and delivery dates, the recipient and, if the shipment has not been delivered, the reason.

Example request

Request — ACS_Trackingsummary

```
{
  "ACSAlias": "ACS_Trackingsummary",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN",
    "Voucher_No": "7227889174"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Voucher_No	string	Yes	The voucher number.
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

Example of a delivered shipment:

Response — ACS_Trackingsummary (demo data)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_Message": "" }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "voucher_no": "7227889174",
          "acs_station_origin": "A0",
          "acs_station_origin_descr": "A0HNA",
          "acs_station_destination": "00",
          "acs_station_destination_descr": "00EPMH",
          "pickup_date": "2026-09-29T00:00:00",
          "delivery_flag": 1,
          "returned_flag": 0,
          "delivery_date": "2026-09-30T00:00:00",
          "consignee": "TEST RECIPIENT",
          "non_delivery_reason_code": "",
          "delivery_date_expected": "2026-10-01T00:00:00",
          "delivery_info": "Η αποστολή παρεδόθη στον προορισμό της. Ημερομηνία: 30/09/2026 Ονομα: TEST RECIPIENT",
          "sender": "ESHOP",
          "recipient": "TEST RECIPIENT",
          "recipient_address": "P. RALLI 45 - TAVROS",
          "shipment_status": 4,
          "phone_acs_station_origin": "2108190000",
          "phone_acs_station_destination": "2310000000",
          "return_doc_voucher_no": null
        }
      ]
    }
  }
}
```

```

    }
  }
}

```

Response fields

Field	Description
<code>voucher_no</code>	The voucher number.
<code>acs_station_origin</code> <code>acs_station_origin_descr</code>	Code and name of the origin ACS store.
<code>acs_station_destination</code> <code>acs_station_destination_descr</code>	Code and name of the destination ACS store.
<code>pickup_date</code>	Date of pickup from the sender.
<code>delivery_date_expected</code>	Estimated delivery date.
<code>delivery_flag</code>	1 = the shipment was delivered, 0 = it has not been delivered.
<code>delivery_date</code>	Delivery date (when <code>delivery_flag</code> = 1).
<code>consignee</code>	The name of the person who received the shipment.
<code>returned_flag</code>	1 = the shipment was returned to the sender as undelivered.
<code>shipment_status</code>	Shipment status code (see the table below).
<code>non_delivery_reason_code</code>	Non-delivery reason code (see the table below). Empty if there is none.
<code>delivery_info</code>	Text description of the status, suitable for display to your customer.
<code>sender</code> , <code>recipient</code> , <code>recipient_address</code>	Sender, recipient and delivery address.
<code>phone_acs_station_origin</code> <code>phone_acs_station_destination</code>	Phone numbers of the origin and destination ACS stores.
<code>return_doc_voucher_no</code>	The return voucher number, for shipments with return of documents (RDO). Otherwise null .

Field names

The `Table_Data` fields are written in **lowercase** (e.g. `voucher_no`, `shipment_status`), unlike the request fields.

Status codes (`shipment_status`)

Code	Meaning
4	Successful delivery of the shipment.
1, 2, 3, 5	The shipment has not been delivered . The reason is given by the <code>non_delivery_reason_code</code> , together with the description in <code>delivery_info</code> (see the non-delivery code table below).
6	To be returned to the sender.
7	Returned to the sender (with <code>returned_flag</code> = 1 and <code>delivery_flag</code> = 1).

Return of undelivered shipments

If a shipment is not delivered within **10 days**, it is returned to the sender: first it gets `shipment_status` = **6** (to be returned) and, once delivered to the sender, `shipment_status` = **7** with `returned_flag` = **1** and `delivery_flag` = **1**.

Note: `delivery_flag` = **1** also appears on a return. To identify delivery to the recipient, check for `shipment_status` = **4**.

Non-delivery reason codes (`non_delivery_reason_code`)

The table shows each non-delivery code, the reason and the `shipment_status` code it appears with. The corresponding description is also returned in the `delivery_info` field.

Code	Reason	<code>shipment_status</code>
AD1	Order for office pickup	5
AD3	Force majeure	5
AD8	Pick-up from ACS shop / sender's request	5
AP1	Delivery refusal (due to charge)	1
AP2	Unable to pay	1
AP3	Not acceptable shipment	1

Code	Reason	shipment_status
AP4	Deceased	1
AS1	Absent	3
DP1	Difficult to access area	5
ED1	Lack of requested shipment	5
LS1	Unknown consignee	2
LS2	Address change	2
LS3	Error or inadequate address	2
PA1	New delivery date, order of shipper	5
PA2	New delivery date, order of consignee	5
PA4	Reschedule / redirect	5

5.3 Tracking details — ACS_TrackingDetails

Returns all movements (checkpoints) of a shipment, in chronological order.

Example request

Request — ACS_TrackingDetails

```
{
  "ACSAlias": "ACS_TrackingDetails",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN",
    "Voucher_No": "7227889174"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Voucher_No	string	Yes	The voucher number.
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

Example of a shipment delivered to an ACS Smartpoint parcel locker (abridged):

Response — ACS_TrackingDetails (demo data)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSExecutionResponse": {
    "ACSValueOutput": [
      { "Error_Message": "" }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "checkpoint_date_time": "2026-09-14T21:02:17.073",
          "checkpoint_action": "ΕΚΤΥΠΩΣΗ ΕΤΙΚΕΤΑΣ ACSCONNECT/WEBTOOLS",
          "checkpoint_location": "ΘΕΣΣΑΛΟΝΙΚΗ",
          "checkpoint_notes": " "
        },
        {
          "checkpoint_date_time": "2026-09-21T12:12:18.047",
          "checkpoint_action": "ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΚΕΝΤΡΟ ΔΙΑΛΟΓΗΣ",
          "checkpoint_location": "HUB ΘΕΣΣΑΛΟΝΙΚΗΣ",
          "checkpoint_notes": " "
        },
        {
          "checkpoint_date_time": "2026-09-22T09:35:57.72",
          "checkpoint_action": "ΑΦΙΞΗ ΣΕ ΚΑΤΑΣΤΗΜΑ",
          "checkpoint_location": "ΦΛΩΡΙΝΑ",
          "checkpoint_notes": " "
        }
      ]
    }
  }
}
```

```

    "checkpoint_notes": " "
  },
  {
    "checkpoint_date_time": "2026-09-22T13:22:05",
    "checkpoint_action": "Η ΑΠΟΣΤΟΛΗ ΒΡΙΣΚΕΤΑΙ ΣΕ SMART POINT",
    "checkpoint_location": "SMARTPOINT LOCKER ΦΛΩΡΙΝΑ",
    "checkpoint_notes": " "
  },
  {
    "checkpoint_date_time": "2026-09-22T20:49:15",
    "checkpoint_action": "ΠΑΡΑΔΟΣΗ",
    "checkpoint_location": "SMARTPOINT LOCKER ΦΛΩΡΙΝΑ",
    "checkpoint_notes": "TEST RECIPIENT"
  }
]
}
}
}

```

Field (per movement)	Description
<code>checkpoint_date_time</code>	Date and time of the movement.
<code>checkpoint_action</code>	The movement as text (e.g. «ΑΦΙΞΗ ΣΕ ΚΑΤΑΣΤΗΜΑ» (arrival at store), «ΠΑΡΑΔΟΣΗ» (delivery)).
<code>checkpoint_location</code>	The store, sorting center or Smartpoint where the movement took place.
<code>checkpoint_notes</code>	Additional information. On delivery it contains the recipient's name. It often contains just a space (" ").

Using the movements

Movements are free text, suitable for display. For automated logic (e.g. "was it delivered?") use the `shipment_status` of `ACS_Trackingsummary` (§5.2), not the text of `checkpoint_action`.

5.4 Find voucher by reference key — ACS_POD_FROM_REFERENCE_NO

Finds the voucher number that matches the reference key you provided when creating the shipment (e.g. order number). The search covers both `Reference_Key1` and `Reference_Key2` of `ACS_Create_Voucher`. It is useful when you have not stored the voucher number, or to check whether a shipment already exists for an order.

Example request

Request — ACS_POD_FROM_REFERENCE_NO

```

{
  "ACSAlias": "ACS_POD_FROM_REFERENCE_NO",
  "ACSInputParameters": {
    "Company_Id": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "User_locals": "EN",
    "reference_no": "ORDER-10045"
  }
}

```

Request parameters

Field	Type	Required	Description
<code>Company_Id</code> , <code>Company_Password</code> , <code>User_ID</code> , <code>User_Password</code>	string	Yes	Access credentials (see §1.3).
<code>reference_no</code>	string	Yes	The reference key provided in <code>Reference_Key1</code> or <code>Reference_Key2</code> of <code>ACS_Create_Voucher</code> . It must be unique : if the same key has been used in more than one shipment, the search fails.
<code>User_locals</code>	string	No	Message language: GR = Greek, EN = English.

Different parameter names

In this method some parameters are spelled differently from the other methods: `Company_Id` (with a lowercase **d**), `User_locals` instead of `Language`, and `reference_no` in lowercase. Use them exactly as shown.

Response

A. Shipment found

Response — shipment found

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOuputResponse": {
    "ACSValueOutput": [
      {
        "Voucher_No": "7227889174",
        "multipart_Voucher_No": null,
        "error_msg": null
      }
    ],
    "ACSTableOutput": {}
  }
}
```

B. Shipment not found

Response — shipment not found

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOuputResponse": {
    "ACSValueOutput": [
      {
        "Voucher_No": null,
        "multipart_Voucher_No": null,
        "error_msg": "There were not found any voucher numbers linked to the reference number you provided."
      }
    ],
    "ACSTableOutput": {}
  }
}
```

Response field	Description
Voucher_No	The voucher number (for a multi-part shipment, the main voucher). null if not found.
multipart_Voucher_No	For a multi-part shipment, the sub-voucher number. Otherwise null .
error_msg	null on success. Otherwise, the error message. Note: in this method the field is named error_msg , not Error_Message .

Notes

- The search covers shipments up to **2 months** back.
- It finds only **final** shipments, i.e. shipments that have been included in a pickup list (Chapter 4).
- The reference key must match **only one** shipment. For this reason, use unique values (e.g. order number), not values shared by many shipments (e.g. the e-shop name).

Error messages

Message	Cause / what to do
There were not found any voucher numbers linked to the reference number you provided.	There is no final shipment from the last 2 months with this reference key. Check the value and that the shipment has been included in a pickup list.
The reference number ... is not unique. You must have a unique reference number in order to find a voucher number.	The reference key has been given to more than one shipment. The search only works with unique keys.

6. Cash on delivery

6.1 COD remittance details — ACS_COD_Beneficiary_Info

Returns the cash on delivery (COD) amounts that ACS remitted to your bank account on a specific date: for each shipment, the voucher, the COD amount and how the recipient paid (cash or card). It is useful for **automatically marking orders as paid** in your system.

Example request

Request — ACS_COD_Beneficiary_Info

```
{
  "ACSAlias": "ACS_COD_Beneficiary_Info",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "User_locals": "EN",
    "COD_Payment_Date": "2026-10-13"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
COD_Payment_Date	string (date)	Yes	The remittance date, i.e. the date ACS deposited the amounts into your bank account, format YYYY-MM-DD . Only the remittances of that day are returned, not "from that date onward".
User_locals	string	No	Message language: GR = Greek, EN = English.

Different parameter names

In this method the language is given in **User_locals** (not **Language**) and the error message of the response is returned in **Error_msg** (not **Error_Message**).

Response

A. There are remittances on that date

Response — remittances of the day (demo data)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_msg": null }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "Customer_Code": "2A0999999",
          "POD": "7227889174",
          "Parcel_Sender": "ESHOP",
          "Parcel_Receiver": "TEST RECIPIENT",
          "Parcel_Pickup_Date": "2026-10-05T00:00:00",
          "Parcel_Delivery_Date": "2026-10-07T00:00:00",
          "Parcel_COD_Amount": 120.30,
          "Customer_RefNo_1": "ORDER-10045",
          "Customer_RefNo_2": "",
          "COD_Amount_Cach": 0.00,
          "COD_Amount_CreditCard": 120.30
        }
      ]
    }
  }
}
```

B. There are no remittances on that date

The call succeeds, but the **Table_Data** list is empty:

Response — no remittances

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSExecutionResponse": {
    "ACSValueOutput": [
      { "Error_msg": null }
    ],
    "ACSTableOutput": {
      "Table_Data": []
    }
  }
}
```

Response fields (per shipment)

Field	Type	Description
Customer_Code	string	The billing code (Billing_Code) of the shipment.
POD	string	The voucher number.
Parcel_Sender	string	The sender.
Parcel_Receiver	string	The recipient.
Parcel_Pickup_Date	date	Pickup date of the shipment.
Parcel_Delivery_Date	date	Delivery date of the shipment.
Parcel_COD_Amount	decimal	The COD amount in €.
COD_Amount_Cash	decimal	The part of the amount paid in cash .
COD_Amount_CreditCard	decimal	The part of the amount paid by card .
Customer_RefNo_1 Customer_RefNo_2	string	The reference keys you provided in Reference_Key1 and Reference_Key2 of ACS_Create_Voucher . Empty if not provided.

Watch the spelling

The cash field is named **COD_Amount_Cach** (not "Cash"). Use exactly this name.

Recommended implementation

- Call the method **once a day**, with the current date (or the previous business day), since each call returns only the remittances of a single date.
- Match each row to your order using **Customer_RefNo_1** (or **POD**) and mark the order as paid.
- For this reason, we recommend always passing the order number in **Reference_Key1** when you create the voucher (§2.2).
- An empty **Table_Data** list means there were no remittances on that day, not an error.

7. Pricing and addresses

7.1 Overview

Method (ACSAlias)	Use	Section
ACS_Price_Calculation	Calculates the cost of a shipment, based on the customer's price list.	7.2
ACS_Address_Validation	Checks and resolves an address; returns the ACS store that serves it.	7.3
ACS_Area_Find_By_Zip_Code	The areas of a postal code, with the ACS store that serves them and the remote area indicator.	7.4

Typical use in an e-shop checkout

With **ACS_Address_Validation** (or **ACS_Area_Find_By_Zip_Code**) you find the destination ACS store and whether the area is inaccessible. Then, with **ACS_Price_Calculation** you calculate the shipping cost to show to your customer.

7.2 Cost calculation — ACS_Price_Calculation

Returns the cost of a shipment based on the price list of the sender customer, considering the type of destination (e.g. within the city, island), additional services (e.g. cash on delivery) and the weight (actual or volumetric).

Domestic shipments within Greece only

The method does **not** calculate costs for shipments to Cyprus or abroad.

Example request

A 0.5 kg shipment from Athens to Chania (island destination), without dimensions and charged to the sender:

Request — ACS_Price_Calculation

```
{
  "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": "A0",
    "Acs_Station_Destination": "XN",
    "Weight": 0.5,
    "Pickup_Date": "2026-10-14",
    "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": "EN"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Billing_Code	string	Yes	The customer's billing code. The cost is calculated with the price list of this code.
Billing_Category	integer	Yes	Always 2 .
Acs_Station_Origin	string	Yes	Origin store code: the 2 Greek capital letters of the Billing_Code (e.g. A0 for 2A0999999).
Acs_Station_Destination	string	Yes	Destination store code. You find it with ACS_Address_Validation (Resolved_Station_ID , §7.3) or ACS_Area_Find_By_Zip_Code (Station_ID , §7.4).
Weight	decimal	Yes	Weight in kg, with a period as the decimal separator. Up to 2 decimal places.
Dimension_X_In_Cm Dimension_Y_In_Cm Dimension_Z_In_Cm	integer	No	Dimensions in centimeters. If provided, the cost is calculated with the greater of the actual and the volumetric weight.
Pickup_Date	string (date)	Yes	Pickup date, format YYYY-MM-DD .
Charge_Type	integer	Yes	2 = charged to sender, 4 = charged to recipient.
Acs_Delivery_Products	string	No	Additional service codes, separated by commas (see the table below). null if there are none.
Insurance_Ammount	decimal	No	Insurance amount in €, up to 3000, without quotes and with a period as the decimal separator (e.g. 550.50). Up to 2 decimal places.
Delivery_Zone	—	No	Not used. Send null .
Language	string	No	Message language: null or GR = Greek, EN = English.

Volumetric weight

Volumetric weight (kg) = Length × Width × Height (cm) / 5000. Example: a 40 × 30 × 20 cm parcel → 24,000 / 5000 = **4.8 kg**.

Additional service codes (Acs_Delivery_Products)

Code	Service
INS	Shipment insurance
SAT	Saturday delivery
COD	Cash on delivery
RDO	Return documents
REM	Remote area (identified with ACS_Area_Find_By_Zip_Code , \$7.4)
PRO	Protocol
REC	Reception delivery (from an ACS store)

Response

Response — ACS_Price_Calculation (indicative amounts)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      {
        "Basic_Amount": 2.6000,
        "Extra_Service_Amount": 0.1768,
        "Total_Amount": 2.78,
        "Total_Vat_Amount": 0.67,
        "Info_Message": "",
        "Error_Message": ""
      }
    ],
    "ACSTableOutput": {}
  }
}
```

In the example: basic service €2.60 + additional services and fuel surcharge €0.1768 = **€2.78** excluding VAT, plus VAT €0.67 = **€3.45** final cost.

Response field	Description
Basic_Amount	Cost of basic service (€, excluding VAT).
Extra_Service_Amount	Cost of additional services and fuel surcharge (€, excluding VAT). That is why it has value even when you did not request any additional service.
Total_Amount	Total excluding VAT (€), rounded to 2 decimal places.
Total_Vat_Amount	The VAT amount (€). The final cost including VAT is Total_Amount + Total_Vat_Amount .
Info_Message	Informational message, if any.
Error_Message	Empty on success. Otherwise, the error message.

Error messages

Message	Cause / what to do
Invalid ACS pickup station	Wrong code in Acs_Station_Origin .
Invalid ACS station destination	Wrong code in Acs_Station_Destination .
For insurance ammounts above 3000€ please contact ACS	Insurance_Amount > 3000.
For weights more than 100 kg please contact us	The Weight or the volumetric weight exceeds 100 kg.

7.3 Address validation — ACS_Address_Validation

Resolves an address and returns its valid form or, if there are several possible matches, a list of them. For each address it returns, among other things, the geographic coordinates, the ACS store that serves it and whether it is an remote area.

Example request

Request — ACS_Address_Validation

```
{
  "ACSAlias": "ACS_Address_Validation",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Language": "EN",
    "Address": "ΣΟΥΛΙΟΥ 10 ΠΕΤΡΟΥΠΟΛΗ 13232",
    "AddressID": null
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Address	string	Yes	The address in a single field, in this order: street and number, area, postal code (e.g. "ΣΟΥΛΙΟΥ 10 ΠΕΤΡΟΥΠΟΛΗ 13232").
AddressID	string	No	Your own identifier for the address (any value). It is returned in the response and helps with matching when you validate many addresses.
Language	string	No	Message language: null or GR = Greek, EN = English.

How to write the address

Prefer **Greek characters**. The method also recognizes addresses in Greeklish, but results are more reliable in Greek.

Uppercase or lowercase, accents and minor spelling mistakes do not matter.

Response

The address (or the possible addresses) is returned as a list in **ACSObjectOutput**:

Response — ACS_Address_Validation

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      {
        "ACSObjectOutput": [
          {
            "GeoID": 732310,
            "Resolved_Street": "ΣΟΥΛΙΟΥ",
            "Resolved_Street_Num": "10",
            "Resolved_Zip": "13232",
            "Resolved_Area": "ΠΕΤΡΟΥΠΟΛΗ",
            "Resolved_Long": 23.676939,
            "Resolved_Lat": 38.0465622,
            "Resolved_GeoDataType": 1,
            "Resolved_GeoDataID": 732310,
            "Resolved_Station_ID": "B0",
            "Resolved_Branch_ID": 1,
            "Resolved_As_Inaccessible_Area_With_Cost": 0,
            "Resolved_As_Inaccessible_Area_WithOut_Cost": 0,
            "Resolved_Confidence": 97,
            "Resolved_GeoRegionType": 1,
            "Resolved_Providence": "N. ΑΤΤΙΚΗΣ",
            "Resolved_Correction": "010",
            "Resolved_Station_Descr": "ΑΓΙΟΙ ΑΝΑΓΥΡΟΙ (2102693523)",
            "AddressID": ""
          }
        ]
      }
    ]
  },
  "ACSTableOutput": {}
}
```

Response fields (per address)

Field	Description
<code>Resolved_Street, Resolved_Street_Num</code>	Street and number, as resolved.
<code>Resolved_Zip, Resolved_Area</code>	Postal code and area, as resolved.
<code>Resolved_Providence</code>	Prefecture.
<code>Resolved_Lat, Resolved_Long</code>	Latitude and longitude.
<code>Resolved_Station_ID</code>	Code of the ACS store that serves the address. Used as <code>ACS_Station_Destination</code> in <code>ACS_Price_Calculation</code> and <code>ACS_Create_Voucher</code> .
<code>Resolved_Branch_ID</code>	The branch of the ACS store.
<code>Resolved_Station_Descr</code>	Name and phone number of the ACS store.
<code>Resolved_As_Inaccessible_Area_With_Cost</code>	1 = remote destination DP (remote area, with surcharge), 0 = no. See the table below.
<code>Resolved_As_Inaccessible_Area_WithOut_Cost</code>	1 = remote destination DX (remote area, no surcharge), 0 = no. See the table below.
<code>Resolved_Confidence</code>	Confidence level of the resolution, from 0 to 100 . The higher the value, the more accurate the address.
<code>AddressID</code>	The identifier you provided in the request.
<code>GeoID, Resolved_GeoDataID, Resolved_GeoDataType, Resolved_GeoRegionType, Resolved_Correction</code>	Technical ACS geocoding fields.

Remote destinations

Destination	<code>Inaccessible_Area_With_Cost</code>	<code>Inaccessible_Area_WithOut_Cost</code>
Regular (not inaccessible)	0	0
Inaccessible DP (with surcharge)	1	0
Inaccessible DX (no surcharge)	0	1

Using this in your e-shop

With `Resolved_As_Inaccessible_Area_With_Cost = 1` (DP) the shipment incurs an extra cost: you can inform your customer or include it in the shipping cost calculation. The **SAT**, **MDV**, **TDD** services are not available for remote destinations (\$2.2).

If `Resolved_Confidence` is low, ask the customer to confirm or correct the address before completing the order.

Field spelling

Field names are used exactly as shown, e.g. `Resolved_As_Inaccessible_Area_With_Cost` (with one **s**) and `Resolved_Providence` (for the prefecture).

7.4 Area lookup by postal code — ACS_Area_Find_By_Zip_Code

Returns the areas of a postal code, with the ACS store and branch that serve them and an indicator of whether it is an **remote area (DP)**. It can return all areas or only the inaccessible ones.

Example request

Request — `ACS_Area_Find_By_Zip_Code`

```
{
  "ACSAlias": "ACS_Area_Find_By_Zip_Code",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Zip_Code": "73500",
    "Show_Only_Inaccessible_Areas": 0,
    "Language": "EN",
    "Country": "GR"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
Zip_Code	string	No	The postal code to search for. With 0 or null , all postal codes and areas of the country are returned.
Show_Only_Inaccessible_Areas	integer	Yes	0 = all areas, 1 = only remote areas (DP). There are no remote areas in Cyprus, so with Country = "CY" and value 1 the response is always "Data not found with these criteria."
Country	string	Yes	GR = Greece, CY = Cyprus, BG = Bulgaria, AL = Albania. For Bulgaria and Albania only one postal code is returned.
Language	string	No	Message language: null or GR = Greek, EN = English.

Response

Example (abbreviated):

Response — ACS_Area_Find_By_Zip_Code
<pre>{ "ACSExecution_HasError": false, "ACSExecutionErrorMessage": "", "ACSOuputResponse": { "ACSValueOutput": [{ "Error_Message": null }], "ACSTableOutput": { "Table_Data": [{ "Description": "115 ΠΤΕΡΥΓΑ ΜΑΧΗΣ", "Area": "115 ΠΤΕΡΥΓΑ ΜΑΧΗΣ(ΣΟΥΔΑ)", "Description_Eng": "115 PTERIGA MAXIS(SOUDA)", "Zip_Code": "73500", "Municipality": "", "Prefecture": "N. ΧΑΝΙΩΝ", "Station_ID": "XN", "Branch_ID": 4, "Inaccessible_Area_Kind": "DP" }] } } }</pre>

Field (per area)	Description
Description, Area	The name of the area.
Description_Eng	The name of the area in Latin characters.
Zip_Code	The postal code of the area.
Municipality, Prefecture	Municipality and prefecture.
Station_ID	Code of the ACS store that serves the area. Used as Acs_Station_Destination in ACS_Price_Calculation .
Branch_ID	The branch of the ACS store.
Inaccessible_Area_Kind	DP = remote area (with surcharge).

Error messages

Message	Cause / what to do
Data not found with these criteria	The postal code does not exist, or you requested only remote areas (Show_Only_Inaccessible_Areas = 1) for a postal code that has none, e.g. any postal code in Cyprus.

8. ACS stores

8.1 Overview

Method (ACSAlias)	Use	Section
<code>ACS_Find_Station_By_Zip_Code</code>	The ACS store that serves a postal code (Greece or Cyprus).	8.2
<code>ACS_Stations</code>	The list of ACS Points (stores, Smartpoints, etc.) with addresses, opening hours, coordinates and codes.	8.3

Typical use in an e-shop

With `ACS_Stations` you load the list of ACS Points (e.g. once a day) and show it at checkout, as a list or on a map. When the customer selects a point, you use its codes in `ACS_Create_Voucher` (see §2.3 and §8.3).

8.2 Store serving a postal code — `ACS_Find_Station_By_Zip_Code`

Returns the ACS store that serves a postal code in Greece or Cyprus, together with the area description and the prefecture.

Example request

Request — `ACS_Find_Station_By_Zip_Code`

```
{
  "ACSAlias": "ACS_Find_Station_By_Zip_Code",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "Zip_Code": "12241",
    "Country": "GR",
    "Language": "EN"
  }
}
```

Request parameters

Field	Type	Required	Description
<code>Company_ID</code> , <code>Company_Password</code> , <code>User_ID</code> , <code>User_Password</code>	string	Yes	Access credentials (see §1.3).
<code>Zip_Code</code>	string	Yes	The postal code: 5 digits for Greece, 4 digits for Cyprus, without spaces or dashes.
<code>Country</code>	string	Yes	GR = Greece, CY = Cyprus.
<code>Language</code>	string	No	Results language: GR = Greek, EN = English.

Response

Different structure for Greece and Cyprus

The fields of `Table_Data` differ depending on the country (`Country`). Your code must handle the two schemas separately.

A. Greece ("Country": "GR") — excerpt for postal code 12241:

Response excerpt — Greece, postal code 12241

```
"ACSTableOutput": {
  "Table_Data": [
    {
      "PostalCode": 12241,
      "ACS_Station": "AB",
      "PostalCode_Description": "ΑΙΓΑΛΕΩ",
      "Prefecture": "ΑΤΤΙΚΗ",
      "ACS_Branch_ID": "01"
    }
  ]
}
```

Field	Type	Description
<code>PostalCode</code>	integer	The postal code searched for.

Field	Type	Description
ACS Station	string	The code (2 letters) of the ACS store that serves the postal code.
PostalCode Description	string	The area of the postal code.
Prefecture	string	The prefecture.
ACS_Branch_ID	string	The branch code, as text with a leading zero (e.g. "01").

Field names containing a space

The fields `ACS Station` and `PostalCode Description` contain a **space** in their name. In your code, use key-based access, e.g. `row["ACS Station"]` (JavaScript/Python) or `[JsonProperty("ACS Station")]` (C#).

B. Cyprus ("Country": "CY") — full response for postal code 1010:

Response — Cyprus, postal code 1010

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "Error_Message": null }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "Description": "Λευκωσία (NI)",
          "Area": "Λευκωσία (NI)",
          "Description_Eng": "LEFKOSIA (NI)",
          "Zip_Code": "1010",
          "Municipality": "Λευκωσία (NI)",
          "Prefecture": "NICOSIA HUB",
          "Station_ID": "NI",
          "Branch_ID": 1,
          "Inaccessible_Area_Kind": ""
        }
      ]
    }
  }
}
```

Field	Type	Description
Zip_Code	string	The postal code searched for (as text).
Description Description_Eng	string	The area description, in Greek and in Latin characters.
Area	string	The area.
Municipality	string	The municipality.
Prefecture	string	The ACS hub in Cyprus that serves the area (e.g. NICOSIA HUB).
Station_ID	string	The code (2 letters) of the ACS store that serves the postal code.
Branch_ID	integer	The branch code, as a number (in Greece it is returned as the text "01").
Inaccessible_Area_Kind	string	Remote area indicator. There are no remote areas in Cyprus, so it is always empty.

8.3 List of ACS Points — ACS_Stations

Returns the ACS service points in Greece or Cyprus, by type of point: central stores, sub-branches, ACS Smartpoints, etc. For each point it gives its codes, address, opening hours and coordinates.

Example request

Request — ACS_Stations

```
{
  "ACSAlias": "ACS_Stations",
  "ACSInputParameters": {
    "Company_ID": "demo",
    "Company_Password": "demo",
    "User_ID": "demo",
    "User_Password": "demo",
    "ACS_SHOP_COUNTRY_ID": "GR",
    "ACS_SHOP_KIND": 1,
    "Language": "EN"
  }
}
```

Request parameters

Field	Type	Required	Description
Company_ID, Company_Password, User_ID, User_Password	string	Yes	Access credentials (see §1.3).
ACS_SHOP_COUNTRY_ID	string	Yes	GR = points in Greece, CY = points in Cyprus.
ACS_SHOP_KIND	integer	Yes	The type of points to return (see the table below).
Language	string	No	Results language: GR = Greek, EN = English.

ACS_SHOP_KIND values

Value	Greece (GR)	Cyprus (CY)
1	Central stores.	Central stores.
2 or 3	Sub-branches.	Nothing is returned.
8	Locker at mini market, parking etc.	Locker at mini market, parking etc.
12	locker in front of an ACS branch.	—

All delivery points

Each call returns one type of point. To show your customer all the points where they can collect a shipment (stores and Smartpoints), call the method for each relevant **ACS_SHOP_KIND** value and merge the results.

Response

The response contains one row in **Table_Data** for each point. Below is one point (indicative values):

Response — ACS_Stations (one point, indicative data)

```
{
  "ACSExecution_HasError": false,
  "ACSExecutionErrorMessage": "",
  "ACSOutputResponse": {
    "ACSValueOutput": [
      { "error_message": null }
    ],
    "ACSTableOutput": {
      "Table_Data": [
        {
          "ACS_SHOP_COUNTRY_ID": 1,
          "ACS_SHOP_COUNTRY_DESCR": "ΕΛΛΑΔΑ",
          "ACS_SHOP_AREA_ID": 19,
          "ACS_SHOP_AREA_DESCR": "ΘΕΣΣΑΛΟΝΙΚΗΣ",
          "ACS_SHOP_STATION_ID": "0Δ",
          "ACS_SHOP_STATION_ID_EN": "SDK",
          "ACS_SHOP_STATION_DESCR": "ΑΓ. ΔΗΜΗΤΡΙΟΥ",
          "ACS_SHOP_BRANCH_ID": 1,
          "ACS_SHOP_ADDRESS": "ΑΓ. ΔΗΜΗΤΡΙΟΥ 28-30, 54630",
          "ACS_SHOP_ZIPCODE": "54630",
          "ACS_SHOP_PHONES": "2310-000000",
          "ACS_SHOP_FAX": "",
          "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": "40.6419851",
          "ACS_SHOP_LONG": "22.9403949",
          "ACS_SHOP_DELIVERY_START_HOUR": "08:30",

```

```

    "ACS_SHOP_COORDINATES_VERIFIED": 1,
    "ACS_SHOP_KIND": 1,
    "ACS_SHOP_SERVICES": "Όλες οι Υπηρεσίες",
    "ACS_SHOP_EMAIL": "info@example.gr",
    "ACS_SHOP_PAYMENT_TYPES": "Αποδεκτή η χρήση των τραπεζικών καρτών ...",
    "ACS_SHOP_ID_CODE": "19003110",
    "ACS_SHOP_CITY": "ΘΕΣΣΑΛΟΝΙΚΗ",
    "ACS_SHOP_SMARTPOINT_KIND": 1,
    "ACS_SHOP_WORKING_HOURS_JSON": "[{"day":1, ...}]",
    "ACS_SHOP_MESSAGE": "*Παραλάβες που έχουν άφιξη στο κατάστημα ...",
    "ACS_SHOP_SMARTPOINT_CAPACITY": null,
    "ACS_SMARTPOINT_COD_SUPPORTED": null
  }
}
}
}
}

```

For an ACS Smartpoint with a parcel locker (**ACS_SHOP_KIND = 8 or 12**) the locker fields are also filled in:

Response excerpt — ACS Smartpoint with locker

```

"ACS_SHOP_KIND": 8,
"ACS_SHOP_SMARTPOINT_KIND": "APP (Automatic Box)",
"ACS_SHOP_SMARTPOINT_CAPACITY": "48",
"ACS_SMARTPOINT_COD_SUPPORTED": 1

```

Response fields (per point)

Field	Type	Description
ACS_SHOP_COUNTRY_ID	integer	The country, as a number (e.g. 1 = Greece), even though it is given as GR/CY in the request.
ACS_SHOP_COUNTRY_DESCR	string	The country description (e.g. ΕΛΛΑΔΑ , Greece).
ACS_SHOP_AREA_ID ACS_SHOP_AREA_DESCR	integer string	The code and description of the prefecture/area.
ACS_SHOP_STATION_ID	string	The code (2 letters) of the ACS store. For delivery to this point, pass it in Acs_Station_Destination of ACS_Create_Voucher .
ACS_SHOP_STATION_ID_EN	string	The store code in Latin characters.
ACS_SHOP_STATION_DESCR	string	The name of the point.
ACS_SHOP_BRANCH_ID	integer	The code of the point within the store. For delivery to an ACS Smartpoint, pass it in Acs_Station_Branch_Destination of ACS_Create_Voucher .
ACS_SHOP_ADDRESS ACS_SHOP_ZIPCODE	string	Address and postal code of the point.
ACS_SHOP_CITY	string	The name of the ACS store that serves the point (e.g. ΑΓΡΙΝΙΟ).
ACS_SHOP_PHONES ACS_SHOP_FAX ACS_SHOP_EMAIL	string	Contact details.
ACS_SHOP_WORKING_HOURS ACS_SHOP_WORKING_HOURS_SATURDAY	string	Opening hours on weekdays and Saturday.
ACS_SHOP_WORKING_HOURS_JSON	string (JSON)	The detailed opening hours per day (day 1–7), as JSON text : it must be parsed separately.
ACS_SHOP_TRUCK_PICKUP_HOURS ACS_SHOP_TRUCK_PICKUP_HOURS_SATURDAY	string	The time the ACS vehicle picks up shipments from the point (weekdays / Saturday).
ACS_SHOP_DELIVERY_START_HOUR	string	The time from which shipments are available for collection at the point.
ACS_SHOP_LAT ACS_SHOP_LONG	string	Latitude and longitude, as text . Convert them to numbers for use on a map.
ACS_SHOP_COORDINATES_VERIFIED	integer	1 = the coordinates have been verified.
ACS_SHOP_KIND	integer	The type of point (see the values table).
ACS_SHOP_SERVICES ACS_SHOP_PAYMENT_TYPES	string	The services of the point and the accepted payment methods, as free text in Greek (e.g. Όλες οι Υπηρεσίες , all services).
ACS_SHOP_ID_CODE	string	Internal code of the point.
ACS_SHOP_SMARTPOINT_KIND	string or integer	The type of point. For a Smartpoint with a parcel locker (ACS_SHOP_KIND 8 or 12): "APP (Automatic Box - SwipBox)" or "APP (Automatic Box)" . For stores (ACS_SHOP_KIND 1 or 2): the same value as ACS_SHOP_KIND .

Field	Type	Description
ACS_SHOP_SMARTPOINT_CAPACITY	string	The number of compartments in the locker (e.g. "48").
ACS_SMARTPOINT_COD_SUPPORTED	integer or boolean	Whether the Smartpoint accepts cash on delivery shipments: 1/true = yes, 0/false = no. Check it before offering the point for an order with cash on delivery.
ACS_SHOP_MESSAGE	string	A fixed informational message in Greek, the same for all points: shipments arriving at the store at least 20 minutes before departure leave for their destination the same day.

Linking to voucher creation

When the recipient chooses to collect from an ACS Smartpoint, the point's codes are passed to **ACS_Create_Voucher** as follows (see §2.3):

ACS_Stations field	ACS_Create_Voucher field
ACS_SHOP_STATION_ID	Acs_Station_Destination
ACS_SHOP_BRANCH_ID	Acs_Station_Branch_Destination

- For delivery from an ACS **store**, the **REC** code in **Acs_Delivery_Products** is sufficient (§2.3 A).
- For a **Smartpoint**, the requirements of §2.3 B apply: mobile phone number, email when there is cash on delivery, a single piece.

Appendix A. Testing with Insomnia

Before you write code, you can test the API methods with a REST client. This appendix describes **Insomnia**, but the same steps apply to any similar tool (e.g. Postman).

For testing only

Insomnia is recommended **for testing only**. Your production application calls the API directly from its own code.

Screenshots are indicative

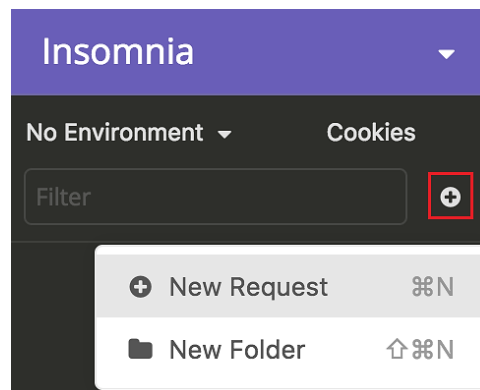
The Insomnia interface changes from version to version. The screenshots show where each setting is located; the option names (New Request, POST, JSON, Header) stay the same.

A.1 Installation

Download Insomnia from <https://insomnia.rest> and install it. The "Docs" section of the website has installation instructions for each operating system.

A.2 Creating a request

Click the + icon in the sidebar and select **New Request**.



Creating a new request

Give the request a name, e.g. the method name (**ACS_Create_Voucher**). Select the **POST** method (all ACS API calls are POST) and body type **JSON**. Click **Create**.

New Request

Name (defaults to your request URL if left empty)

ACS_Create_Voucher

POST JSON

* Tip: paste Curl command into URL afterwards to import it

Create

Name, POST method and JSON body

Import from cURL

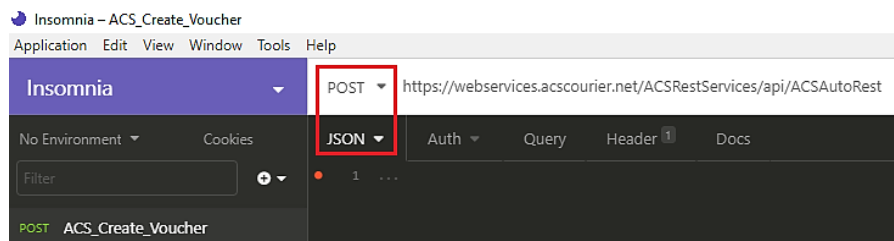
Instead of filling in the fields one by one, you can paste a cURL command, such as those in §1.9, into the URL field. Insomnia fills in the URL, headers and body automatically.

A.3 URL and header

In the URL field, enter the execution address:

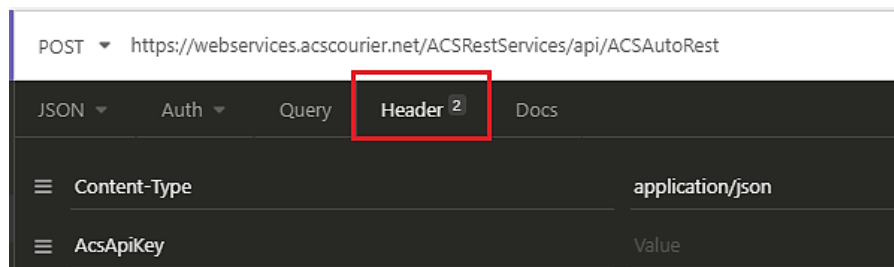
URL

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



The request with POST, JSON and the URL

On the **Header** tab, **Content-Type: application/json** is already filled in. Add a new header named **ACSApiKey** with the key provided to you by ACS as its value (see §1.3). The other tabs (Auth, Query, Docs) are not needed.



The Content-Type and ACSApiKey headers

A.4 Execution

1. In the body, enter the request JSON, e.g. one of the examples in this guide with your own access credentials.
2. Click **Send**.
3. Check the response as described in §1.8: HTTP status, **ACSExecution_HasError** and **Error_Message**.

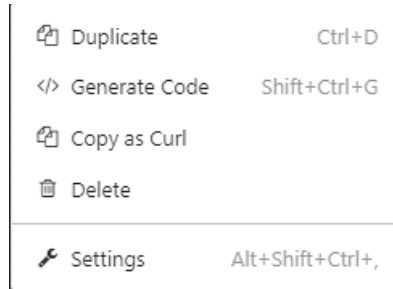
Parameter template

To see which parameters a method accepts, change the URL to **.../api/ACSAutoRestHelp** and send only the **ACSAlias** (see §1.9).

Appendix B. Generating code from Insomnia

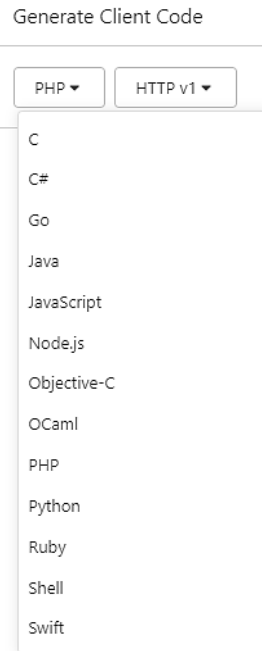
Once a request works correctly in Insomnia, you can get ready-made code that executes it, in the programming language you use.

1. Right-click the request in the sidebar and select **Generate Code**.



The Generate Code option

2. Select a language (e.g. C#, Java, Node.js, PHP, Python) and, where available, the HTTP library.



The available languages

3. Copy the code into your application.

Protect your API key

The generated code contains the **ACSApiKey** and the access credentials **in plain text**. Before you integrate it, move them to configuration settings or environment variables, and do not send them to third parties or commit them to public code repositories.