

Web API

Geniki's web API is a SOAP/WSDL web service located at:

<https://voucher.taxydromiki.gr/JobServices.asmx>

For test purposes you can use the web service located at:

<https://testvoucher.taxydromiki.gr/JobServices.asmx>

using a test user that we will provide for you.

The API provides the following methods. The objects that are used are explained later.

Important Notice:

In order to get credentials for the live environment you must first confirm to our sales department that you have implemented, in your solution for the end user, the function below:

- Authorization (Usage of the Authenticate method)
- Voucher creation (Usage of the CreateJob method)
- Print voucher (Usage of GetVouchersPdf method, if using our template. If you are using your own template validation will be required)
- Ability to cancel a voucher (Usage of the CancelJob method)
- Finalize created vouchers (Usage of the ClosePendingJobs, ClosePendingJobsByDate methods)

Methods

AuthenticateResult Authenticate(string sUserName, string sUserPwd, string applicationKey)

User name, password and application key are passed and an object with the result code and an authentication key is returned. This authentication key is used with all other methods.

CreateJobResult CreateJob(string sAuthKey, Record oVoucher, JobType eType)

The voucher or order info is passed in Record object. The record should have been filled with the minimum info for creating a voucher or order (Name, Address, City). Job id, voucher number and subvouchers (if any) are returned (for order job type, voucher number and subvouchers will be empty since no voucher is created when placing the order).

GetVoucherJobResult GetVoucherJob(string sAuthKey, long nJobId)

The job with the specified id is returned.

int ClosePendingJobs(string sAuthKey)

Closes and sends to the Geniki shop, the pending (that have not closed yet) created vouchers and orders that have been created **TODAY**. An error code is returned (explained later).

int ClosePendingJobsByDate(string sAuthKey, DateTime dFr, DateTime dTo)

As the previous function, but instead of today's jobs it closes jobs that have been created in the range dateFrom - dateTo.

int CancelJob(string sAuthKey, long nJobId, bool bCancel)

Cancels or un-cancels the the job with the specified id. If cancel parameter is true the job is canceled. If cancel is false, the job is made active again. An error code is returned (explained later).

GetJobsResult GetJobsFromOrderId(string sAuthKey, string sOrderId)

Gets all the jobs referenced by this order id. Can be more than one if the client does not use unique order ids for the jobs he creates.

TrackAndTraceResult TrackAndTrace(string authKey, string voucherNo, string language)

Gets a history of checkpoints for the specified voucher number and its current status (delivered/not delivered). Language can be "el" (for Greek) or "en" (for English).

void GetVouchersPdf(string authKey, string[] voucherNumbers, MediaFormat format, ExtraInfoFormat extraInfoFormat)

Get a pdf document to print on Geniki's supported paper types, for the specified vouchers. This method differs from the others in that it does not return a SOAP message as result. In case of success it returns an "application/pdf" document and in case of failure a "text/plain" document with the error result code (see later). "voucherNumbers" argument is an array with the voucher numbers you need to generate a PDF document for. "format" argument can have either of the values "Flyer" or "Sticker", depending on the paper type that will be used for printing on. Lastly extraInfoFormat should always have the value "None". You can use an HTTP GET request if this is easier, like this:

<https://voucher.taxydromiki.gr/JobServices.asmx/GetVouchersPdf?authKey=key&voucherNumbers=voucherNo1&voucherNumbers=voucherNo2...&voucherNumbers=voucherNoN&Format=Flyer&extraInfoFormat=None>

GetShopsListResult GetShopsList(string authKey)

Gets a list of Geniki Taxydromiki shops.

Objects

Record		
Field	Type	Description
OrderId	string	User defined id
Name	string	Recipient name
Address	string	Recipient address
City	string	Recipient city
Telephone	string	Recipient telephone
Zip	string	Recipient zip code
Weight	decimal	Voucher weight
Pieces	int	Voucher pieces. If greater than one, subvouchers are created with different voucher numbers
Comments	string	Comments can be any length, but up to 40 (255 for orders) characters are inserted in the system
Services	string	Any services the voucher might have, delimited with comma (see the services later)
CodAmount	string	The amount of the cod. If greater than zero it must always have the service "αμ" or "αv"
InsAmount	decimal	The amount to be insured. If greater than zero it must have the service "ασ"
VoucherNo	string	Geniki's unique voucher number
SubCode	string	If you are using cost centers, this is the code of the cost center
BelongsTo	string	If it is a subvoucher, this field contains the number of the parent voucher
DeliverTo	string	Used with pickup jobs. Denotes the recipient of a pickup job. If it's empty recipient is the client who created the job.
ReceivedDate	DateTime	The date that the package is supposed to be received by Geniki's shop. Cannot be smaller than today.
ContentsDescription	string	A description of the packages' contents. Requested for abroad deliveries.

VoucherJob		
Field	Type	Description
Id	int	Distinct job id
Type	JobType	The type of the job (see enumeration below)
Voucher	Record	The main voucher that this job created
ReturningVoucher	Record	If the job is of type "SendAndReturn", this field holds the returning voucher created on server
SubVouchers	Record[]	The subvouchers created by this job, if the main voucher had more than one pieces
Date	DateTime	Date the job was created
OrderId	string	User defined id
IsClosed	bool	Is job closed?
IsCanceled	bool	Has the job been canceled?
Status	string	Job status (empty for now)
StatusDate	DateTime	The date of the status
User	string	The user who created the job

Type of the job can be any of the following enumerations

```
enum JobType {
    Voucher, //A normal shipment
    Pickup, //An order to go to a client and pickup a package to be returned to you
    SendAndReturn //Sent some documents to a client and get back from him some other documents
}
```

CreateJobResult		
Field	Type	Description
Result	int	The result code (explained later)
JobId	int	Distinct job id
Voucher	string	The main voucher number that this job created
SubVouchers	Record[]	The subvoucher records created by this job, if the main voucher had more than one pieces.

GetVoucherJobResult		
Field	Type	Description
Result	int	The result code (explained later)
JobId	int	Distinct job id
Job	VoucherJob	The voucher job

GetJobsResult		
Field	Type	Description
Result	int	The result code (explained later)
Jobs	VoucherJob[]	Array of jobs

AuthenticateResult		
Field	Type	Description
Result	int	The result code (explained later)
Key	string	The authentication key to be used with all functions (except Authenticate)

TrackAndTraceResult		
Field	Type	Description
Result	int	The result code (explained later)
Checkpoints	Checkpoint[]	An array holding the checkpoints history of a voucher
Status	string	The current status of the tracked package (delivered/not delivered)
DeliveryDate	DateTime	The date that the package was delivered
Consignee	string	The person who signed upon delivery

Checkpoint		
Field	Type	Description
Status	string	Checkpoint's status description
StatusDate	DateTime	The date and time the status occurred
Shop	string	The shop associated with the checkpoint

GetShopsListResult		
Field	Type	Description
Result	int	The result code (explained later)
Shops	Shop[]	An array holding the shops of Geniki Taxydromiki

Shop		
Field	Type	Description
Code	string	Shop code
Code2	string	Shop secondary code
Name	string	Shop description
State	string	Shop state
City	string	Shop city
Address	string	Shop address
Telephone	string	Shop telephone
Zip	string	Shop zip
Email	string	Shop email
Longitude	decimal	Shop longitude
Latitude	decimal	Shop latitude
SubShop	boolean	If true the shop is a subshop
Active	boolean	If true the shop is active

Error Codes		
Number	Description	Functions that can return it
0	Ok	All functions
1	Authentication failed	Authenticate
2	Not implemented	None right now
3	No data	CreateJob if null record is passed TrackAndTrace if null or empty voucher number is passed
4	Invalid operation	CancelJob if the job is closed
5	Max voucher No. reached	CreateJob if the server has reached the max voucher number. Should not happen (still many millions to go), but if it does you should really contact us
6	Max subvoucher No. reached	As above for subvoucher numbers
700	Validation failed	CreateJob if any of the required fields (Name, Address, City) is empty
701	Validation failed	CreateJob, if cod service set with no cod amount
702	Validation failed	CreateJob, if cod amount set with no cod service
703	Validation failed	CreateJob, if cod amount limit is exceeded
704	Validation failed	CreateJob, if insurance service set with no insurance amount
705	Validation failed	CreateJob, if insurance amount set with no insurance service
706	Validation failed	CreateJob, if received date is smaller than today
8	SQL error	All functions, internal error
9	Doesn't exist	GetVoucherJob or CancelJob, when specified job does not exist TrackAndTrace, if specified voucher number is not found
10	Not authorized	GetVoucherJob, CancelJob, when the requesting user has not right to access the specified job TrackAndTrace, when the requesting user has no right to access the specified voucher
11	Invalid key	All functions except Authenticate, if the specified authorization key is not valid. This is normal, the authorization key can change from time to time.
12	Run-time error	All functions, internal error
13	Job canceled	GetVoucherJob, CancelJob, if the job is canceled. For GetVoucherJob it should be treated as an error but rather as a status. The info of the job is still returned
14	Server busy	CreateJob, ClosePendingJobs, ClosePendingJobsByDate, when temporarily the operation can't be carried out.
15	Request limit reached	All functions, if a limit for requests per some time has been specified for the calling user and this limit has been exceeded

Geniki Services

Voucher.Services field can have the following values, separated with comma if more than one (case insensitive).

1Σ = SPECIAL ΠΡΩΙΝΗ ΠΑΡΑΔΟΣΗ
3Σ = ΑΥΘΗΜΕΡΟΝ ΠΟΛΗΣ
5Σ = ΠΑΡΑΔΟΣΗ ΣΑΒΒΑΤΟ
ΑΜ = ΑΝΤΙΚΑΤΑΒΟΛΗ ΜΕΤΡΗΤΟΙΣ
ΑΝ = ΑΝΤΙΚΑΤΑΒΟΛΗ ΑΞΙΟΓΡΑΦΑ
ΑΣ = ΑΣΦΑΛΙΣΗ
ΔΔ = ΔΙΚΑΙΟΛΟΓΗΤΙΚΑ ΔΙΑΓΩΝΙΣΜΩΝ
ΕΔ = ΕΙΔΙΚΗ ΧΡΕΩΣΗ
ΕΜ = ΠΑΡΑΛΑΒΗ ΔΙΚΑΙΟΛΟΓΗΤΙΚΩΝ

ΕΨ = ΕΙΔΗ ΨΥΓΕΙΟΥ
ΠΡ = ΠΑΡΑΛΑΒΗ ΠΡΩΤΟΚΟΛΟΥ
ΠΚ = ΕΠΙΣΤΡΟΦΗ ΠΑΚΕΤΟΥ
ΤΝ = ΑΕΡΟΜΕΤΑΦΟΡΑ
ΧΠ = ΧΡΕΩΣΗ ΠΑΡΑΛΗΠΤΗ
Τ1 = ΤΑΧΥΔΡ. ΚΙΒΩΤΙΟ Ν1 - 5 (ΕΩΣ 2 KG)
Τ6 = ΤΑΧΥΔΡ. ΚΙΒΩΤΙΟ Ν6 - 7 (ΕΩΣ 4KG)
ΤΕ = ΤΑΧΥΔΡ. ΚΙΒΩΤΙΟ ΕΓΓΡΑΦΩΝ (ΕΩΣ 2 KG)
ΦΡ = ΕΜΠΟΡΕΥΜΑΤΙΚΗ ΜΕΤΑΦΟΡΑ

Example C# code

```
private void ExampleCode() {
    //JobServices is the referenced web service class
    JobServices services = new JobServices();
    AuthenticateResult authResult = services.Authenticate("UserName", "Password", "AppKey");
    if(authResult.Result != 0) {
        //Could not get key
        return;
    }

    Record voucher = new Record {
        OrderId = "00001",
        Name = "Test name",
        Address = "Test address",
        City = "Test city",
        Telephone = "2109999999",
        Zip = "12345",
        Comments = "Test comment",
        SubCode = "",
        Weight = 12.34m,
        Pieces = 3,
        Services = "αv",
        CodAmount = 1234.56m
    };

    CreateJobResult result = services.CreateJob(authResult.Key, voucher, JobType.Voucher);
    if(result.Result != 0) {
        //Error creating voucher
        return;
    }

    //Print them, store them, whatever...
    string voucherNumber = result.Voucher;

    foreach(Record subVoucher in result.SubVouchers) {
        string subVoucherNumber = subVoucher.VoucherNo;
        string name = subVoucher.Name;
        //...
        //...
    }

    //Not necessary to call with each print.
    //Can (and should, to avoid unnecessary burden) be called when all printing is done.
    services.ClosePendingJobs(authResult.Key);
    services.Dispose();
}
```

Example PHP code

Note: if coding under WAMP, don't forget to add the extension `php_soap`. Also the file has to be saved as UTF-8 for the greek services.

```
<html>
<body>
  <?php
    try {
      $soap = new SoapClient("http://testvoucher.taxydromiki.gr/JobServices.asmx?WSDL");

      echo "----- Authenticating -----<br>";
      $oAuthResult = $soap->Authenticate(
        array(
          'sUserName' => 'UserName',
          'sUserPwd' => 'Password',
          'applicationKey' => 'AppKey'
        )
      );
      print_r($oAuthResult);
      echo "<BR>";

      if ($oAuthResult->AuthenticateResult->Result != 0) {
        echo "Error authenticating!!<br>";
        return;
      }
      echo "Key = " . $oAuthResult->AuthenticateResult->Key . "<br>";
      echo "Authentication OK<br>";

      echo "----- Creating a voucher -----<br>";
      $oVoucher = array(
        'OrderId' => '00001',
        'Name' => 'Test name',
        'Address' => 'Test address',
        'City' => 'Test city',
        'Telephone' => '2109999999',
        'Zip' => '12345', 'Destination' => "",
        'Courier' => "",
        'Pieces' => 3,
        'Weight' => 12,
        'Comments' => 'Test comment',
      );
    }
  }
</body>
</html>
```

```

        'Services' => "αv",
        'CodAmount' => 1234.56,
        'InsAmount' => 0,
        'VoucherNumber' => "",
        'SubCode' => "",
        'BelongsTo' => "",
        'DeliverTo' => "",
        'ReceivedDate' => "2012-01-01"
    );

    $xml = array(
        'sAuthKey' => $oAuthResult->AuthenticateResult->Key,
        'oVoucher' => $oVoucher,
        'eType' => "Voucher"
    );
    echo "----- Result of the voucher creation -----<br>";
    print_r($xml);
    echo "<BR>";

    $oResult = $soap->CreateJob($xml);
    print_r($oResult);
    echo "<BR>";

    if($oResult->CreateJobResult->Result != 0) {
        echo "Error Creating a voucher!!<br>";
        return;
    }

    echo "----- Track and Trace a voucher -----<br>";
    $xml = array (
        'authKey' => $oAuthResult->AuthenticateResult->Key,
        'voucherNo' => $oResult->CreateJobResult->Voucher,
        'language' => 'el'
    );

    $TT = $soap->TrackAndTrace($xml);
    print_r($TT);
    echo "<BR>";

    echo "----- Closing Pending Jobs -----<br>";
    $soap->ClosePendingJobs(
        array('sAuthKey' => $oAuthResult->AuthenticateResult->Key)
    );
} catch(SoapFault $fault) {
    echo $fault;
}
?>
</body>
</html>

```