MyGLS API
for system integration

GLS.

Table of contents

Change log

Change No.	Date version	Description	Since
1		Brand new API for MyGLS	2019-02-01
2		Address class HouseNumberInfo; APIRequestBase class ClientNumberList: do not use; code samples have been extended with service sample code.	2019-10-21
3		Code samples have been extended with pickup date.	2019-12-13
4		Error code 23.	2019-12-17
5		Services parameters.	2020-01-22
6		Housenumber valid value: number.	2020-01-31
7		GetParcelList; GetParcelStatuses; code samples.	2020-04-17
8		PrintDataInfo class -> DepotNumber: MODIFIED CONTENT; TourNumber -> Driver; Depot, Sort: NEW	2020-06-09
9		New error messages: 28, 29.	2020-07-02
10		New error messages: 30, 31.	2020-08-13
11		MyGLS_API_services_json.txt	2020-08-27
12		New error messages: 32, 33.	2020-09-02
13		GetParcelStatusesRequest → LanguageIsoCode	2020-09-14
14		PrintLabelsRequest, GetPrintedLabelsRequest → TypeOfPrinter; GetPrintDataRequest → ParcelIdList	2020-11-18
15		RS country has been added New error messages: 34, 43, 44, 48	2023-01-01
16		LRS service has been added	2023-11-13
17		WebshopEngine field has been added to PrintLabels and PrepareLabels methods New TypeOfPrinter: ThermoZPL	2025-02-12
18		Optional parcel properties field	2025-05-29
19		GetClientReturnAddress; code samples, update all the code samples	2025-07-08
20		Add new TypeOfPrinter „ShipItThermoPdf"	2025-09-02
21		Add new TypeOfPrinter „ThermoZPL_300DPI"	2025-09-03

Getting started

Before you can start using this **API** for MyGLS system, you got to have an agreement with GLS. If you don't have the required MyGLS login credentials please contact GLS company.

MyGLS API communication is designed for using via **HTTPS**.

MyGLS API supports two approaches: **SOAP** (format **XML**) and **REST** (format **JSON** or **XML**).

Used HTTP response codes: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/401

The "Type of printer" property is set in the "Settings".

API service data assembly

API development team has decided to release shared assembly *GLS.MyGLS.APIServiceData.dll* **and sample codes (C#, PHP, JAVA).**
You can download it e.g. from https://api.mygls.hu/.
Library contains declaration of all API request/response classes, enumerations, etc.
It can be used by own developed client implementation.

Authentication and authorization

Every calling of API method has to be authenticated via request parameter.
You need *user name (email)* and *password*.
Don't forget to fill it in all requests.

APIRequestBase

C# code

```csharp
public class APIRequestBase
{
    public List<int> ClientNumberList { get; set; }
    public byte[] Password { get; set; }
    public string Username { get; set; }
    public string WebshopEngine { get; set; }
}
```

Password in string representation must be encrypted with **SHA512** algorithm to byte array. (see
Appendix C: Copy / Paste snippet section)

Country domain API URLs

We are supporting MyGLS *API* for 6 countries now. Do not forget to use the **right country domain**.

{**service name**} = ParcelService

	Country		API URL
1	**Croatia**	SOAP	https://api.mygls.**hr**/{**service name**}.svc?singleWsdl
		REST	https://api.mygls.**hr**/{**service name**}.svc/{format}/{methodName}
		testing and development	https://api.**test**.mygls.**hr**/...
2	**Czechia**	SOAP	https://api.mygls.**cz**/{**service name**}.svc?singleWsdl
		REST	https://api.mygls.**cz**/{**service name**}.svc/{format}/{methodName}
		testing and development	https://api.**test**.mygls.**cz**/...
3	**Hungary**	SOAP	https://api.mygls.**hu**/{**service name**}.svc?singleWsdl
		REST	https://api.mygls.**hu**/{**service name**}.svc/{format}/{methodName}
		testing and development	https://api.**test**.mygls.**hu**/...
4	**Romania**	SOAP	https://api.mygls.**ro**/{**service name**}.svc?singleWsdl
		REST	https://api.mygls.**ro**/{**service name**}.svc/{format}/{methodName}
		testing and development	https://api.**test**.mygls.**ro**/...
5	**Slovenia**	SOAP	https://api.mygls.**si**/{**service name**}.svc?singleWsdl
		REST	https://api.mygls.**si**/{**service name**}.svc/{format}/{methodName}
		testing and development	https://api.**test**.mygls.**si**/...
6	**Slovakia**	SOAP	https://api.mygls.**sk**/{**service name**}.svc?singleWsdl
		REST	https://api.mygls.**sk**/{**service name**}.svc/{format}/{methodName}
		testing and development	https://api.test.mygls.sk/...
7	**Serbia**	SOAP	https://api.mygls.**rs**/{**service name**}.svc?singleWsdl
		REST	https://api.mygls.**rs**/{**service name**}.svc/{format}/{methodName}
		testing and development	https://api.**test**.mygls.**rs**/...

Used data structures in request/response objects

Parcel class

Object containing necessary data for printing labels.

Property	DataType	Description
ClientNumber	Integer	Unique client number provided by GLS company. REQUIRED
ClientReference	String	Client custom tag identifying parcel. STRONGLY RECOMMENDED (It is optionally REQUIRED, depends on the user right.)
Count	Integer	Count of parcels sent in one shipment. (maximum 99) DEFAULT 1
CODAmount	Decimal	Cash on delivery amount. NOT REQUIRED
CODReference	String	Cash on delivery client reference number used for payment pairing. STRONGLY RECOMMENDED if CODAmount is filled.
CODCurrency	String	ISO code of the currency NOT REQUIRED
Content	String	Parcel info printed on label. (It is optionally REQUIRED, depends on the user right. It is MANDATORY in Serbia.)
PickupDate	DateTime	Pick up date. DEFAULT actual date
PickupAddress	Address	The address of place where courier pick up the shipment. REQUIRED
DeliveryAddress	Address	The address of destination place. REQUIRED
ServiceList	List<Service>	Services and their special parameters.
SenderIdentityCardNumber	String	Only in Serbia! ID card number or PIB. REQUIRED
PickupType	Integer	Only used for LRS service! Parameter is always 2. Available only in HU!
ParcelPropertyList	List<ParcelProperty>	Optional list where parcel dimensions and packaging properties can be specified.

Address class

Object containing address of pick up place and parcel destination.

Property	DataType	Description
Name	String	Name of the person or organization. REQUIRED

Property	DataType	Description
Street	String	Name of the street. REQUIRED
HouseNumber	String	Number of the house. (ONLY NUMBER)
HouseNumberInfo	String	Additional information. (Building, stairway, etc.)
City	String	Name of the town or village. REQUIRED
ZipCode	String	Area Zip code. REQUIRED
CountryIsoCode	String	Two letter country code defined in ISO 3166-1. More… REQUIRED
ContactName	String	Name of person which can be asked or inform about shipment details by GLS.
ContactPhone	String	Phone number of person which can be asked or inform about shipment details by GLS.
ContactEmail	String	Email address of person which can be asked or inform about shipment details by GLS.

Service class

Object containing specific service settings.

Property	DataType	Description
Code	String	Service code (see Appendix B: List of services). REQUIRED
ADRParameter	ServiceParameterADR	Settings for ADR service REQUIRED FOR "ADR" SERVICE
AOSParameter	ServiceParameterString	Settings for AOS service REQUIRED FOR "AOS" SERVICE
CS1Parameter	ServiceParameterString	Settings for CS1 service REQUIRED FOR "CS1" SERVICE CODE
DDSParameter	ServiceParameterDateTime	Settings for DDS service REQUIRED FOR "DDS" SERVICE CODE
DPVParameter	ServiceParameterStringDecimal	Settings for DPV service REQUIRED FOR "DPV" SERVICE CODE
FDSParameter	ServiceParameterString	Settings for FDS service REQUIRED FOR "FDS" SERVICE CODE
FSSParameter	ServiceParameterString	Settings for FSS service REQUIRED FOR "FSS" SERVICE CODE
INSParameter	ServiceParameterDecimal	Settings for INS service REQUIRED FOR "INS" SERVICE CODE
MMPParameter	ServiceParameterDecimal	Settings for MMP service REQUIRED FOR "MMP" SERVICE CODE
PSDParameter	ServiceParameterStringInteger	Settings for PSD service REQUIRED FOR "PSD" SERVICE CODE
SDSParameter	ServiceParameterTimeRange	Settings for SDS service REQUIRED FOR "SDS" SERVICE CODE

Property	DataType	Description
SM1Parameter	ServiceParameterString	Settings for SM1 service REQUIRED FOR "SM1" SERVICE CODE
SM2Parameter	ServiceParameterString	Settings for SM2 service REQUIRED FOR "SM2" SERVICE CODE
SZLParameter	ServiceParameterString	Settings for SZL service REQUIRED FOR "SZL" SERVICE CODE
Value	String	Service value without previous special service settings

ParcelProperty class

Object containing parcel's dimensions and packaging type.

Property	DataType	Description
Content	String	Description of the parcel contents
PackageType	Integer	Type of packaging, which can be: Colli = 1 Box = 2 Roll = 3 Can = 4 Case = 5 Reel = 6 Sack = 7
Height	Integer	Height of parcel
Length	Integer	Length of parcel
Width	Integer	Width of parcel
Weight	Decimal	Weight of parcel

ErrorInfo class

In case of ParcelService.

Property	DataType	Description
ErrorCode	Integer	Appendix A
ErrorDescription	String	Appendix A
ClientReferenceList	List<String>	List of client parcel tags identifying parcels where specific error happened.
ParcelIdList	List<Integer>	List of database parcel ID identifying parcel records where specific error happened.

Parcel service operations

PrepareLabels

Validates parcel data for labels and adds valid parcel data to database.

C# code

```
PrepareLabelsResponse PrepareLabels(PrepareLabelsRequest prepareLabelsRequest);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/PrepareLabels **(LATEST VERSION, format JSON, for HU)**
POST	https://api.**test**.mygls.cz/ParcelService.svc/xml/PrepareLabels_20190201 **(SPECIFIC VERSION, format XML, for CZ)**

Request class PrepareLabelsRequest

Property	DataType	Description
ParcelList	List<Parcel>	List of labels/parcels data . REQUIRED NOT EMPTY
WebshopEngine	String	Webshop engine of the website REQUIRED

C# code

```
public class PrepareLabelsRequest : APIRequestBase
{
    public List<Parcel> ParcelList { get; set; }
}
```

Response class PrepareLabelsResponse

Property	DataType	Description
ParcelInfoList	List<ParcelInfo>	List of successfully prepared records (ID and ClientReference) for generating labels.
PrepareLabelsError	List<ErrorInfo>	List of potential errors (see Error lists in response classes)

C# code

```
public class PrepareLabelsResponse
{
    public List<ParcelInfo> ParcelInfoList { get; set; }
    public List<ErrorInfo> PrepareLabelsError { get; set; }
}
```

ParcelInfo class

Property	DataType	Description
ClientReference	String	Client custom tag identifying parcel.
ParcelId	Integer	Label/Parcel database record ID.

C# code

```csharp
public class ParcelInfo
{
    public string ClientReference { get; set; }
    public int ParcelId { get; set; }
}
```

PrepareLabelsV2

The PrepareLabelsV2 provides the same functionality as the original PrepareLabels, Validates parcel data for labels and adds valid parcel data to database. However, it introduces an improved and more efficient internal logic. The request and response structures are identical to those of PrepareLabels, ensuring full compatibility.

GetPrintedLabels

Generates parcel numbers and PDF document contains labels in byte array format.

C# code

```
GetPrintedLabelsResponse GetPrintedLabels(
    GetPrintedLabelsRequest getPrintedLabelsRequest
);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/GetPrintedLabels **(LATEST VERSION, format JSON, for HU)**
POST	https://api.**test**.mygls.cz/ParcelService.svc/xml/GetPrintedLabels_20190201 **(SPECIFIC VERSION, format XML, for CZ)**

Request class GetPrintedLabelsRequest

Property	DataType	Description		
ParcelIdList	List<Integer>	List of labels/parcels database record ID. REQUIRED NOT EMPTY MAX. 99 ITEMS PER REQUEST		
PrintPosition	Integer	Number of page quarter ACCEPTED ONLY FOR A4-FORMAT	1	2
			3	4
ShowPrintDialog	Bool	Flag for third party PDF reader (if supported application shows print dialog immediately after opening document)		
TypeOfPrinter	String	Possible values: A4_2x2, A4_4x1, Connect, Thermo, ThermoZPL, ShipItThermoPdf, ThermoZPL_300DPI		

C# code

```
public class GetPrintedLabelsRequest : APIRequestBase
{
    public List<int> ParcelIdList { get; set; }
    public int PrintPosition { get; set; }
    public bool ShowPrintDialog { get; set; }
    public string TypeOfPrinter { get; set; }
}
```

Response class GetPrintedLabelsResponse

Property	DataType	Description
Labels	Byte[]	PDF document in byte array.
GetPrintedLabelsErrorList	List<ErrorInfo>	List of potential errors (see Error lists in response classes)
Labels	byte[]	Labels in PDF format
PrintDataInfoList	List<PrintDataInfo>	Additional label information

C# code

```csharp
public class GetPrintedLabelsResponse
{
    public List<ErrorInfo> GetPrintedLabelsErrorList { get; set; }
    public byte[] Labels { get; set; }
    public List<PrintDataInfo> PrintDataInfoList { get; set; }
}
```

PrintDataInfo class

Additional label information in label creation process response

Property	DataType	Description
B2CChar	String	Appendix A
ClientReference	String	Appendix A
Depot	String	List of client parcel tags identifying parcels where specific error happened.
DepotNumber	String	List of database parcel ID identifying parcel records where specific error happened.
Driver	String	Driver identification
Parcel	Parcel	Parcel details
ParcelId	Integer	Parcel identification
ParcelNumber	Long	Parcel number without check digit
ParcelNumberWithCheckdigit	Long	Parcel number with check digit
Sort	String	Sorting information
Displaylanguage	String	Language ISO code

C# code

```csharp
public class PrintDataInfo
{
    public string B2CChar { get; set; }
    public string ClientReference { get; set; }
    public string Depot { get; set; }
    public string DepotNumber { get; set; }
    public string Driver { get; set; }
    public Parcel Parcel { get; set; }
    public int ParcelId { get; set; }
    public long ParcelNumber { get; set; }
    public long ParcelNumberWithCheckdigit { get; set; }
    public string Sort { get; set; }
    public string Displaylanguage { get; set; }
}
```

PrintLabels

Under the hood it calls both [PrepareLabels](#) and [GetPrintedLabels](#) in one step.

So, it validates parcel data for labels, adds valid parcel data to database, generates parcel numbers and PDF document containing labels in byte array format.

C# code

```
PrintLabelsResponse PrintLabels(PrintLabelsRequest printLabelsRequest);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/PrintLabels **(LATEST VERSION, format JSON, for HU)**
POST	https://api.**test**.mygls.cz/ParcelService.svc/xml/PrintLabels_20190201 **(SPECIFIC VERSION, format XML, for CZ)**

Request class PrintLabelsRequest

Property	DataType	Description		
ParcelList	List<[Parcel](#)>	List of labels/parcels data . REQUIRED NOT EMPTY		
WebshopEngine	String	Webshop engine of the website REQUIRED		
PrintPosition	Integer	Number of page quarter ACCEPTED ONLY FOR A4-FORMAT	1 / 3	2 / 4
ShowPrintDialog	Bool	Flag for third party PDF reader (if supported application shows print dialog immediately after opening document)		
TypeOfPrinter	String	Possible values: A4_2x2, A4_4x1, Connect, Thermo, ThermoZPL, ShipItThermoPdf, ThermoZPL_300DPI		

C# code

```csharp
public class PrintLabelsRequest : APIRequestBase
{
    public List<Parcel> ParcelList { get; set; }
    public int PrintPosition { get; set; }
    public bool ShowPrintDialog { get; set; }
    public string TypeOfPrinter { get; set; }
}
```

Response class PrintLabelsResponse

Property	DataType	Description
Labels	Byte[]	PDF document in byte array.
PrintLabelsErrorList	List<[ErrorInfo](#)>	List of potential errors (see [Error lists in response classes](#))

Property	DataType	Description
PrintLabelsInfoList	List<PrintLabelsInfo>	List of successfully prepared records (ID and ClientReference) for generating labels.

C# code

```csharp
public class PrintLabelsResponse
{
    public byte[] Labels { get; set; }
    public List<ErrorInfo> PrintLabelsErrorList { get; set; }
    public List<PrintLabelsInfo> PrintLabelsInfoList { get; set; }
}
```

PrintLabelsInfo class (inherits **ParcelInfo**)

Property	DataType	Description
ClientReference	String	Client custom tag identifying parcel.
ParcelId	Integer	Label/Parcel database record ID.
ParcelNumber	Long	Parcel number

C# code

```csharp
public class PrintLabelsInfo : ParcelInfo
{
    public long ParcelNumber { get; set; }
}
```

DeleteLabels

Set DELETED state for labels/parcels with specific database record ID.

C# code

```
DeleteLabelsResponse DeleteLabels(DeleteLabelsRequest deleteLabelsRequest);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/DeleteLabels **(LATEST VERSION, format JSON, for HU)**
POST	https://api.**test**.mygls.cz/ParcelService.svc/xml/DeleteLabels_20190201 **(SPECIFIC VERSION, format XML, for CZ)**

Request class DeleteLabelsRequest

Property	DataType	Description
ParcelIdList	List<Integer>	List of labels/parcels database ID. REQUIRED NOT EMPTY MAX. 50 ITEMS PER REQUEST

C# code

```
public class DeleteLabelsRequest : APIRequestBase
{
    public List<int> ParcelIdList { get; set; }
}
```

Response class DeleteLabelsResponse

Property	DataType	Description
Labels	Byte[]	PDF document in byte array.
DeleteLabelsErrorList	List<ErrorInfo>	List of potential errors (see Error lists in response classes)
SuccessfullyDeletedList	List<SuccessfullyDeleted>	List of successfully deleted labels/parcels database ID. When a deleted parcel wasn't the only one in shipment, there is a filled array of sub-parcels ID.

C# code

```
public class DeleteLabelsResponse
{
    public List<ErrorInfo> DeleteLabelsErrorList { get; set; }
    public List<SuccessfullyDeleted> SuccessfullyDeletedList { get; set; }
}
```

SuccessfullyDeleted class

Property	DataType	Description
ParcelId	Integer	Parcel identifier
SubParcelIdList	List<Integer>	List of successfully deleted subparcel identifiers

C# code

```csharp
public class SuccessfullyDeleted
{
    public int ParcelId { get; set; }
    public List<int> SubParcelIdList { get; set; }
}
```

ModifyCOD

Changes COD amount for specific parcel.

C# code

```
ModifyCODResponse ModifyCOD(ModifyCODRequest modifyCODRequest);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/ModifyCOD **(LATEST VERSION, format JSON, for HU)**
POST	https://api.**test**.mygls.cz/ParcelService.svc/xml/ModifyCOD_20190201 **(SPECIFIC VERSION, format XML, for CZ)**

Request class ModifyCODRequest

Property	DataType	Description
ParcelId	Integer	Label/parcel database ID. REQUIRED IF ParcelNumber IS NULL
ParcelNumber	Long	Parcel number. REQUIRED IF ParcelId IS NULL
CODAmount	Decimal	Cash on delivery amount. ZERO OR POSITIVE

C# code

```
public class ModifyCODRequest : APIRequestBase
{
    public decimal CODAmount { get; set; }
    public int? ParcelId { get; set; }
    public long? ParcelNumber { get; set; }
}
```

Response class ModifyCODResponse

Property	DataType	Description
Successful	Bool	True = modifying COD without error
ModifyCODError	List<ErrorInfo>	List of potential errors (see Error lists in response classes)

C# code

```
public class ModifyCODResponse
{
    public List<ErrorInfo> ModifyCODError { get; set; }
    public bool Successful { get; set; }
}
```

GetParcelList

Get parcel(s) information by date ranges.

C# code

```
GetParcelListResponse GetParcelList(GetParcelListRequest getParcelListRequest);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/GetParcelList **(format JSON, for HU)**
POST	https://api.**test**.mygls.cz/ParcelService.svc/xml/GetParcelList_20190201 **(SPECIFIC VERSION, format XML, for CZ)**

Request class GetParcelListRequest

Property	DataType	Description
PickupDateFrom	DateTime	Pickup date.
PickupDateTo	DateTime	Pickup date.
PrintDateFrom	DateTime	Print Date.
PrintDateTo	DateTime	Print date.

C# code

```
public class GetParcelListRequest : APIRequestBase
{
    public DateTime? PickupDateFrom { get; set; }
    public DateTime? PickupDateTo { get; set; }
    public DateTime? PrintDateFrom { get; set; }
    public DateTime? PrintDateTo { get; set; }
}
```

Response class GetParcelListResponse

Property	DataType	Description
GetParcelListErrors	List<ErrorInfo>	List of potential errors (see Error lists in response classes)
PrintDataInfoList	List<PrintDataInfo>	Parcel(s) information by date ranges.

C# code

```
public class GetParcelListResponse
{
    public List<ErrorInfo> GetParcelListErrors { get; set; }
    public List<PrintDataInfo> PrintDataInfoList { get; set; }
}
```

GetParcelStatuses

Get parcel statuses with/or without POD.

C# code

```
GetParcelStatusResponse GetParcelStatuses(
    GetParcelStatusesRequest getParcelStatusesRequest
);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/GetParcelStatuses **(format JSON, for HU)**
POST	https://api.**test**.mygls.cz/ParcelService.svc/xml/GetParcelStatuses_20190201 **(SPECIFIC VERSION, format XML, for CZ)**

Request class GetParcelStatusesRequest

Property	DataType	Description
ParcelNumber	Long	Parcel number. REQUIRED
ReturnPOD	Bool	True = get POD file in PDF format.
LanguageIsoCode	String	Default: EN. Possible values: HR, CS, HU, RO, SK, SL. (ISO 639-1)

C# code

```
public class GetParcelStatusesRequest : APIRequestBase
{
    public string LanguageIsoCode { get; set; }
    public long ParcelNumber { get; set; }
    public bool ReturnPOD { get; set; }
}
```

Response class GetParcelStatusResponse

Property	DataType	Description
ClientReference	String	Client custom tag identifying parcel.
DeliveryCountryCode	String	Two letter country code defined in ISO 3166-1.
DeliveryZipCode	String	Delivery area Zip code.
GetParcelStatusErrors	List<ErrorInfo>	List of potential errors (see Error lists in response classes)
ParcelNumber	Long	Parcel number.
ParcelStatusList	List<ParcelStatus>	Parcel status.
POD	Byte[]	PDF document in byte array.
Weight	Double	Weight. (NULLABLE)

C# code

```
public class GetParcelStatusResponse
{
    public string ClientReference { get; set; }
    public string DeliveryCountryCode { get; set; }
    public string DeliveryZipCode { get; set; }
    public List<ErrorInfo> GetParcelStatusErrors { get; set; }
    public long ParcelNumber { get; set; }
    public List<ParcelStatus> ParcelStatusList { get; set; }
    public byte[] POD { get; set; }
    public double? Weight { get; set; }
}
```

ParcelStatus class

Property	DataType	Description
DepotCity	String	Depot city.
DepotNumber	String	Depot number.
StatusCode	String	Status code.
StatusDate	DateTime	Status date.
StatusDescription	String	Status description.
StatusInfo	String	Status info.

C# code

```
public class ParcelStatus
{
    public string DepotCity { get; set; }
    public string DepotNumber { get; set; }
    public string StatusCode { get; set; }
    public DateTime StatusDate { get; set; }
    public string StatusDescription { get; set; }
    public string StatusInfo { get; set; }
}
```

GetClientReturnAddress

Get return address of unique clientnumber.

C# code

```
GetClientReturnAddressResponse GetClientReturnAddress(
    GetClientReturnAddressRequest getClientReturnAddressRequest
);
```

Request

Method	URL examples
POST	https://api.**test**.mygls.hu/ParcelService.svc/json/GetClientReturnAddress **(format JSON, for HU)**

Request class GetClientReturnAddressRequest

Property	DataType	Description
ClientNumber	Long	Unique client number. REQUIRED
LanguageIsoCode	String	Possible values: HR, CS, HU, RO, SK, SL. (ISO 639-1) REQUIRED
LinkTypeId	Integer	Possible link type values: 1 = ClientPayment, 2 = ConsigneePayment
Name	String	Link name REQUIRED

C# code

```
public class GetClientReturnAddressRequest : APIRequestBase
{
    public int ClientNumber { get; set; }
    public string LanguageIsoCode { get; set; }
    public int? LinkTypeId { get; set; }
    public string Name { get; set; }
}
```

Response class GetClientReturnAddressResponse

Property	DataType	Description
ReturnAddress	Address	Client return address.
LinkStatus	Integer	Possible values: 0 = NotExists, 1 = Active, 2 = Inactive, 3 = Deleted.
GetClientReturnAddressErrors	List<ErrorInfo>	List of potential errors (see Error lists in response classes)

C# code

```csharp
public class GetClientReturnAddressResponse
{
    public Address ReturnAddress { get; set; }
    public LinkStatus LinkStatus { get; set; }
    public List<ErrorInfo> GetClientReturnAddressErrors { get; set; }
}
```

Appendix A: API error codes

Error number	Meaning	Since change number
1	Request parameter is null	1
2	Parcel ID list is null	1
3	Parcel ID list is empty	1
4	Parcel ID not exists	1
5	Access denied for this parcel ID	1
6	Parcel with this ID has different status than PRINTED	1
7	Missing parcel data in request	1
8	COD amount has to be >= 0	1
9	Parcel number not exists	1
10	Parcel number was not assigned yet	1
11	Parcel list is null	1
12	Parcel list is empty	1
13	Parcel validation issue	1
14	User not exists	1
15	User is not authorized to access parcel	1
16	Label is empty	1
17	There are no parcel numbers	1
18	Parcel label is already generated	1
19	Parcel number generator failed	1
20	Parcel numbers were not generated	1
21	There are no printable labels	1
22	Count of parcels for deleting is out of limit	1
23	The house number cannot be 0	4
24	Wrong print date interval	7
25	Wrong pickup date interval	7
26	Parcel not found with current settings	7
27	User is not authorized to access to Client	7
28	The Count must be 1 because of the INS service	9
29	Parcel(s) count must be between 1 and 99	9
30	FSS service is not available without FDS	10
31	Same request sent 5 times within last 5 minutes (ATTENTION: the numbers in the message are variables. It is important because of the any retry logic.)	10
32	The ClientReference field is required	12
33	The Content field is required	12
34	Value of TypeOfPrinter is invalid	14
43	Missing senderId card number (only in RS)	14
44	Missing content (only in RS)	14
48	Invalid COD currency ISO code	14
1000	Unexpected exception happened	1
1001	Internal Problem	1

Appendix B: List of services

Not all services are available in each country or area. JSON examples: Appendix E.

Service code	Service name	Parameter
24H	Service guaranteed delivery shipment in 24 Hours	
ADR	Agreement about Dangerous goods by Road	ADRParameter(AdrItemType,AmountUnit,InnerCount, PackSize,UnNumber) REQUIRED (C# .NET example: "new Service() { Code = "ADR", ADRParameter = new ServiceParameterADR() { AdrItemType = AdrItemType.EQ, AmountUnit = AmountUnit.ml, InnerCount = 1, PackSize = 1, UnNumber = 1002 } }" → "EQ1002 1x1ml" (on the label))
AOS	Addressee Only Service	AOSParameter(Value) REQUIRED (Name of the recipient / contact person can be used)
COD	Cash On Delivery service	
CS1	Contact Service	CS1Parameter(Value) REQUIRED (Recipient phone number / contact phone number can be used)
DDS	Day Definite Service	DDSParameter(Value) REQUIRED (Date of delivery in format yyyy-MM-dd)
DPV	Declared Parcel Value service	DPVParameter(StringValue ,DecimalValue) REQUIRED (Used in case of HR, 20xxx zip codes, to declare value of the parcel) (C# .NET example: "new Service() { Code = "DPV", DPVParameter = new ServiceParameterStringDecimal() { StringValue = "APITEST", DecimalValue = 12500 } }")
FDS	Flexible Delivery Service	FDSParameter(Value) REQUIRED (Email address)
FSS	Flexible delivery Sms Service	FSSParameter(Value) REQUIRED (SMS phone number in international format; not available without FDS)
INS	Insurance Service	INSParameter(Value) REQUIRED (Value of the parcel)
PRS	Pick & Return Service	
PSD	Parcel Shop Delivery service	PSDParameter(IntegerValue=DropOffPoint ID /StringValue) REQUIRED (C# .NET example: "new Service() { Code = "PSD", PSDParameter = new ServiceParameterStringInteger() { StringValue = "2351-CSOMAGPONT" } }") Please note, the PSD service requires **additonal attributes** to be provided, as **mandatory** (located in Address class): - ContactName - ContactPhone - ContactEmail
PSS	Pick & Ship Service	
SAT	SATurday service	
SDS	Scheduled Delivery Service	SDSParameter(TimeFrom,TimeTo) REQUIRED (The format the same as the PickupDate, but focus on the time (UTC).)

Service code	Service name	Parameter
SM1	SMs service	SM1Parameter(Value) REQUIRED (SMS Phone number and SMS text in format "phone nr in international format|sms text". Variables that can be used in the text of the SMS: #ParcelNr#,#COD#,#PickupDate#,#From_Name#,#ClientRef#.)
SM2	SMs pre-advice	SM2Parameter(Value) REQUIRED (SMS Phone number in international format)
SRS	ShopReturnService	Available only in HU and SI
SZL	document return service (SZáLlítólevél visszaforgatás)	SZLParameter(Value) REQUIRED (Document number – string, max. 15 char)
T09	Express service	
T10	Express service	
T12	Express service	
TGS	Think Green Service	
XS	Exchange Service	
LRS	LockerReturn Service	Available only in HU. PickupType must be used.

Appendix C: Copy / Paste snippet section

Password SHA512 implementations

https://api.**test**.mygls.hu/index_en.html → Sample files → DOWNLOAD

Appendix D: Code Samples

https://api.mygls.hu/

Appendix E: Service parameter examples in JSON format

JSON

```
"ServiceList":[{
     "Code":"24H"
}]

"ServiceList":[{
     "Code": "ADR",
     "ADRParameter": {
          "AdrItemType": 2
          , "AmountUnit": 4
          , "InnerCount": 1
          , "PackSize": 1
          , "UnNumber": 1002
     }
}]

"ServiceList":[{
     "Code": "AOS",
     "AOSParameter": {
          "Value": "Test Addressee"
     }
}]

"ServiceList":[{
     "Code": "CS1",
     "CS1Parameter": {
          "Value": "+36701234567"
     }
}]

"ServiceList":[{
     "Code": "DDS",
     "DDSParameter": {
          "Value": "\/Date(1598911199000)\/"
     }
}]

"ServiceList":[{
```

```
        "Code": "DPV",
        "DPVParameter": {
                "StringValue": "APITEST",
                "DecimalValue": 12500
        }
}]

"ServiceList":[{
        "Code": "FDS",
        "FDSParameter": {
                "Value": "something@anything.hu"
        }
}]

"ServiceList":[{
                "Code": "FSS",
                "FSSParameter": {
                        "Value": "+36701234567"
                }
        }
        ,{
                "Code": "FDS",
                "FDSParameter": {
                        "Value": "something@anything.hu"
                }
}]

/*
In case e.g. HU: below 50000 HUF, there is a default insurance.
*/
"ServiceList":[{
        "Code": "INS",
        "INSParameter": {
                "Value": "50000"
        }
}]

"ServiceList":[{
        "Code": "PRS"
}]

"ServiceList":[{
        "Code":"PSD",
        "PSDParameter":{
                "StringValue":"2351-CSOMAGPONT"
        }
}]

"ServiceList":[{
        "Code": "PSS"
}]

"ServiceList":[{
```

```
        "Code": "SAT"
}]


/*
(SENT)1970-01-01 12:21:00 -> (SAVED)1970-01-01 12:30:00
(SENT)1970-01-01 15:51:00 -> (SAVED)1970-01-01 16:00:00
In case of PHP: date_default_timezone_set('UTC');
*/

"ServiceList":[{
        "Code": "SDS",
        "SDSParameter": {
                "TimeFrom": "\/Date(44460000)\/",
                "TimeTo": "\/Date(57060000)\/"
        }
}]

"ServiceList":[{
        "Code": "SM1",
        "SM1Parameter": {
                "Value": "+36701234567|#ParcelNr# - Test message."
        }
}]

"ServiceList":[{
        "Code": "SM2",
        "SM2Parameter": {
                "Value": "+36701234567"
        }
}]

"ServiceList":[{
        "Code": "SRS"
}]

"ServiceList":[{
        "Code": "SZL",
        "SZLParameter": {
                "Value": "DOCUMENT ID"
        }
}]

"PickupDate": "'.$pickupDate.'",
        "ServiceList":[{
                "Code": "T09"
}]

"PickupDate": "'.$pickupDate.'",
        "ServiceList":[{
                "Code": "T10"
}]

"PickupDate": "'.$pickupDate.'",
```

```
      "ServiceList":[{
            "Code": "T12"
}]


"ServiceList":[{
      "Code": "TGS"
}]


"ServiceList":[{
      "Code": "XS"
}]


"PickupType": 2,                    // PickupType always 2
"ServiceList": [{
      "Code": "LRS"
 }]
```

Appendix E: Parcel property list examples in JSON format

JSON

```json
"ParcelPropertyList": [
    {
            "Content": "Test",
            "PackageType": 1,
            "Height": 32,
            "Length": 28,
            "Width": 49,
            "Weight": 3.2
    }
]
```

GLS.

Appendix F: Jargon

API (Application Program Interface) is a set of routines, rules and tools for building software. In this case, it is a set of clearly defined methods for parcel processing. It helps to develop fast and clear communication between MyGLS online system and customer systems. More...

Domain (in this case means "country code top level domain") is used and reserved for country, sovereign state or territory identified with a country code. More...

HTTPS (Hypertext Transfer Protocol Secure) is used for encrypted communication over a computer network. More...

JSON (JavaScript Object Notation) is an open standard file format that uses human readable object consisting of attribute-value pair and array data. More...

REST (Representational State Transfer) is a software architectural style that defines a set of constraints to be used for creating web services. More...

SHA512 (Secure Hash Algorithm 512 bits) is a cryptographic hash function from set of SHA-2 family. Method computes 64 bytes (512 bits / 8 bits per byte) from any content. Every byte can store values 0-255 (0x00-0xFF), so hexadecimal string takes 128 chars (64 bytes * 2 characters per byte). More...

SOAP (Simple Object Access Protocol) is specification for exchanging structured information in web service implementation. It uses XML for describing message format. More...

URL (Uniform Resource Locator) is reference to specific web resource – network location and a mechanism for retrieving it. More...

XML (Extensible Markup Language) defines a set of rules for encoded documents in a format that is both human and machine readable. More...

GLS.

GLS.

Appendix G: GLS Status Codes

GLS system	Meaning text in English (EN)
1	The parcel was handed over to GLS.
2	The parcel has left the parcel center.
3	The parcel has reached the parcel center.
4	The parcel is expected to be delivered during the day.
5	The parcel has been delivered.
6	The parcel is stored in the parcel center.
7	The parcel is stored in the parcel center.
8	The parcel is stored in the GLS parcel center. The consignee has agreed to collect the goods himself.
9	The parcel is stored in the parcel center to be delivered at a new delivery date.
10	Check scan normal.
11	The parcel could not be delivered as the consignee is on holidays.
12	The parcel could not be delivered as the consignee was absent.
13	Sorting error at the depot.
14	The parcel could not be delivered as the reception was closed.
15	Not delivered lack of time.
16	The parcel could not be delivered as the consignee had no cash available/suitable.
17	The parcel could not be delivered as the recipient refused acceptance.
18	The parcel could not be delivered as further address information is needed.
19	The parcel could not be delivered due to the weather condition.
20	The parcel could not be delivered due to wrong or incomplete address.
21	Forwarded sorting error.
22	Parcel is sent from the depot to sorting center.
23	The parcel has been returned to sender.
24	The changed delivery option has been saved in the GLS system and will be implemented as requested.
25	Forwarded misrouted.
26	The parcel has reached the parcel center.
27	The parcel has reached the parcel center.
28	Disposed.
29	Parcel is under investigation.
30	Inbound damaged.
31	Parcel was completely damaged.
32	The parcel will be delivered in the evening.
33	The parcel could not be delivered due to exceeded time frame.

GLS system	Meaning text in English (EN)
34	The parcel could not be delivered as acceptance has been refused due to delayed delivery.
35	Parcel was refused because the goods was not ordered.
36	Consignee was not in, contact card couldn´t be left.
37	Change delivery for shipper´s request.
38	The parcel could not be delivered due to missing delivery note.
39	Delivery note not signed.
40	The parcel has been returned to sender.
41	Forwarded normal.
42	The parcel was disposed upon shipper´s request.
43	Parcel is not to locate.
44	Parcel is excluded from General Terms and Conditions.
46	Change completed for Delivery address.
47	The parcel has left the parcel center.
51	The parcel data was entered into the GLS IT system; the parcel was not yet handed over to GLS.
52	The COD data was entered into the GLS IT system.
53	Depot transit.
54	The parcel has been delivered to the parcel box.
55	The parcel has been delivered at the ParcelShop (see ParcelShop information).
56	Parcel is stored in GLS ParcelShop.
57	The parcel has reached the maximum storage time in the ParcelShop.
58	The parcel has been delivered at the neighbour's (see signature).
59	Parcelshop pickup.
60	Customs clearance is delayed due to a missing invoice.
61	The customs documents are being prepared.
62	Customs clearance is delayed as the consignee´s phone number is not available.
64	The parcel was released by customs.
65	The parcel was released by customs. Customs clearance is carried out by the consignee.
66	Customs clearance is delayed until the consignee´s approval is available.
67	The customs documents are being prepared.
68	The parcel could not be delivered as the consignee refused to pay charges.
69	The parcel is stored in the parcel center. It cannot be delivered as the consignment is not complete.
70	Customs clearance is delayed due to incomplete documents.
71	Customs clearance is delayed due to missing or inaccurate customs documents.
72	Customs data must be recorded.

GLS system	Meaning text in English (EN)
73	Customs parcel locked in origin country.
74	Customs clearance is delayed due to a customs inspection.
75	Parcel was confiscated by the Customs authorities.
76	Customs data recorded, parcel can be sent do final location.
80	The parcel has been forwarded to the desired address to be delivered there.
83	The parcel data for Pickup-Service was entered into the GLS system.
84	The parcel label for the pickup has been produced.
85	The driver has received the order to pick up the parcel during the day.
86	The parcel has reached the parcel center.
87	The pickup request has been cancelled as there were no goods to be picked up.
88	The parcel could not be picked up as the goods to be picked up were not packed.
89	The parcel could not be picked up as the customer was not informed about the pickup.
90	The pickup request has been cancelled as the goods were sent by other means.
91	Pick and Ship/Return cancelled.
92	The parcel has been delivered.
93	Signature confirmed.
97	Parcel is placed to parcellocker.
99	Consignee contacted Email delivery notification.
401	Parcellocker capacity problem
402	Parcellocker oversized
403	Parcel damaged
404	Parcellocker technical issue
420	Defect box