

GService

web service manual

About GService

GService is a web service designed for customers of the Geis Group. It enables customers to fully automate the process of submitting transport data within Geis systems. The web service is used to create shipments (in export or order mode), retrieve shipment statuses and related information, arrange pickups, and obtain information about available additional services.

The web service is primarily intended for full integration with customers' production systems, warehouse management systems, or e-commerce platforms. Requests and responses are structured in JSON format (REST) or XML format (SOAP), enabling further automated processing within the customer's systems.

GService is based on REST (<https://en.wikipedia.org/wiki/REST>) or SOAP architecture (<https://en.wikipedia.org/wiki/SOAP>). For new integrations, the **REST API** is recommended. Access is available exclusively via the **secure HTTPS** protocol. TLS versions **1.2** and **1.3** are supported. A comparison table mapping REST and SOAP methods is available in the "[REST and SOAP Method Comparison](#)" section.

To test the REST web service, you can use Postman (<https://www.postman.com/>) or similar tools.

The SOAP web service can be tested using open-source tools such as SoapUI (<http://www.soapui.org>).

Access to the web service requires authentication credentials – a customer code and password. These credentials are provided upon request by the assigned business manager of the Geis Group.

In order to ensure the availability of service for all customers, the number of calls by individual customers is monitored. Each web service method has a defined call limit for a certain time period. If this number is exceeded, access to that method will be temporarily restricted. In this case, the customer receives an error message when calling the method.

Before going live, integration testing in the test environment is mandatory. These tests are performed by the customer's technical department in cooperation with the Geis Group EDI department.

The web service is available for customers from the Czech Republic at the following addresses:

- <https://gservice.geis.cz> - access to the production environment
- <https://gservicetest.geis.cz> – access to the test environment

The web service is available for customers from Poland at the following addresses:

- <https://gservice.geis.pl> - access to the production environment
- <https://gservicetest.geis.pl> – access to the test environment

The web service is available for customers from Slovakia at the following addresses:

- <https://gservice.geis.sk> - access to the production environment
- <https://gservicetest.geis.sk> - access to the test environment

For the successful implementation of the web service in the customer environment and for a better understanding of the broader context, it is recommended to review the *System Integration Manual*, which is available on the technical support web site of the Geis group under link Technical support in the downloads section at:

- www.geis-group.cz or www.geis-group.sk (Czech, Slovak and English version)
- www.geis.pl (Polish and English version)

Content

About GService.....	2
ChangeLog	6
Implementation steps on the customer side	8
Recommended set of services for GService implementation:.....	8
Minimal test scenario.....	9
REST	10
REST basics	10
Swagger	10
Authentication	10
Language.....	10
Endpoint overview	11
GET /api/customer/service	11
GET /api/Status	11
GET /api/PackingType	11
GET /api/Routing/Delivery.....	11
GET /api/Shipment/Detail	11
GET /api/Routing/Table.....	11
POST /api/Pickup.....	11
POST /api/Shipment/Export.....	11
POST /api/Shipment/Order	11
POST /api/Customer/Range	12
GET /healthcheck	12
GET /api/Shipment/Label	12
GET /api/Shipment.....	12
GET /api/Shipment/Status	12
GET /api/PickupList	12
DELETE /api/Shipment	13
GET /api/Customer/Range	13
GET /api/Pickup	13
Overview of HTTP statuses	13
SOAP.....	15
Structure of the XML request	15
Web service base classes	15
Web service methods	17
ServiceList.....	17
AddServiceList.....	18

StatusList	19
WrapList	20
DeliveryRouting	21
ShipmentDetail	22
RouteTable	24
CreatePickUp	25
InsertExport	27
InsertOrder	30
AssignRange	32
IsHealthy	33
GetLabel	34
ListOfShipments	36
ShipmentStatus	37
GetPickupList	38
DeleteShipment	39
RangeDetail	39
PickupInfo	41
An examples of method calls and responses	42
ServiceList	42
AddServiceList	43
StatusList	45
WrapList	46
DeliveryRouting	47
ShipmentDetail	48
RouteTable	51
CreatePickUp	53
InsertExport	54
InsertOrder	56
AssignRange	58
IsHealthy	59
GetLabel	60
ListOfShipments	62
ShipmentStatus	63
GetPickupList	64
DeleteShipment	65

RangeDetail.....	66
PickupInfo	67
REST and SOAP Method Comparison.....	69
An examples of error codes	70

ChangeLog

Version	Date	Change
1.0	2015-05-29	Service creation. Basic methods: - ServiceList - AddServiceList - StatusList - WrapList - DeliveryRouting - ShipmentDetail - RouteTable - CreatePickUp - InsertExport - InsertOrder - AssignRange
1.1		Added list of possible currencies Currency
1.2	2015-08-03	Changes in input: - To ExportService added Parameter_5 - InsertOrder corrected parameter MergedOrder - Edited mandatory parameters in InsertExport - Language CS changed to CZ - Password for logging extended to 20 characters Changes in output: - method ShipmentDetail – PackageRow added customer references (Reference) - method InserExport added list of numbers merged shipments (MergedPackNumbers) - method InsertOrder added list of numbers merged orders (MergedPackNumbers)
1.3	2015-08-24	Changes in input: - InsertOrder: added parameter for entering merged orders (MergedOrders) - InsertOrder: removed parameter CountItems - InsertExport: added parameter for entering merged shipments (MergedPackages) - Corrected control of class Contact, now it is checking the format of Email and Telephone number.
1.4	2015-08-27	Changes in input: - DeliveryRouting parameter CountryIn changed to optional for Parcel
1.5	2015-09-09	Updating the obligations parameters. Correction.
1.6	2015-09-21	Supplementing dimensions parameters for order.
1.7	2015-11-13	Changed description of parameters for method DeliveryRouting
1.8	2015-12-02	Added web service description, implementation scenario
1.9	2016-01-06	Error code list update
2.0	2016-02-10	New method GetLabel . Update of method IsHealthy. Update of response of method ShipmentDetail.

2.1	2016-05-09	New methods ListOfShipments , ShipmentStatus , GetPickupList , DeleteShipment . Added new parameter BurstId in methods Insertorder and InsertExport.
2.2	2016-06-06	GDO service added to list of services (in part of method InsertExport)

2.3	2016-07-13	Modification of methods ListOfShipments, ShipmentDetail
2.4	2016-08-23	Correction of parameter Range for AssignRange method (parameter is mandatory)
2.5	2016-10-06	New web service addresses for K-EX customers
2.6	2016-12-05	Changed description of parameters for method AddServiceList
2.7	2017-01-12	Complementing services for method InsertExport
2.8	2017-01-14	Changed description of parameters for method InsertExport
2.9	2017-02-07	The total revision of the document
3.0	2017-05-02	Changed description for method DeleteShipment , editing references to documentation
3.1	2017-08-11	Changed description for method AssignRange
3.2	2017-09-19	Changed description for method InsertExport
3.3	2017-11-02	Updated table of services parameters for method InsertExport
3.4	2017-11-03	Added new limit information in ShipmentStatus , DeleteShipment , GetLabel and InsertExport methods
3.5	2018-01-25	Updated minimum test scenario and added a recommended set of service for Gservice implementation
3.6	2018-03-15	New method RangeDetail
3.7	2018-05-23	Updated table of Error codes
3.8	2018-07-09	Updated table of additional services
3.9	2018-08-24	Updated table of additional service (format of Dispensing point code)
4.0	2019-02-07	Service Insurance changed
4.1	2019-02-22	New method PickupInfo
4.2	2019-06-27	Structure of the XML request - new article
4.3	2019-11-11	Change of the manual formatting
4.4	2020-02-25	Notes for weight for the merged shipments removed (InsertExport , InsertOrder)
5.0	2021-03-30	Removal of the type of transport Parcel
5.1	2021-04-19	Deleted INS service
5.2	2021-04-27	Replacement of the B2C service with the HDS service
5.3	2021-06-22	Updated table of additional services (phone number prefix)
5.4	2022-07-18	Update of table "An examples of error codes", update of the whole documentation – tables, formats, unseeable marks. Section " An examples of method call and responses " was added a button "Go Back" which takes you back to the " Web service methods " by the topic where you work.
5.5	2023-02-22	Updated method ShipmentStatus
5.6	2023-10-03	Adding the ExWorks and PersonalPickUp parameters to the InsertExport and InsertOrder
5.7	2024-03-04	Deleted B2x services
5.8	2024-06-24	Change of obligation for parameters - Width, Length and Height in the InsertExport method, now these parameters are mandatory.
5.9	2026-01-19	Adding secondary handling units
6.0	2026-03-11	Adding REST section, reorganizing the document, adding comparison REST and SOAP comparison table
6.1	2026-05-25	Adding information about sub items and information about CEE orders.

Implementation steps on the customer side

The following steps are required for implementation on the customer side:

- Verify the ability of the customer's logistics system to send API requests and process responses via the HTTPS protocol.
- Review the *System Integration Manual* to understand the basic concepts (product types, shipment routing, additional services, label printing, and pickup lists), as well as this documentation
- Request access to the test API environment from the assigned Geis business manager
- Test the basic methods and classes, process responses, and implement proper handling of return codes
- Generate shipment labels and the pick up list and submit them to the Geis business manager for approval
- Agree on the go-live date and proceed with production deployment

Recommended set of services for GService implementation:

REST	SOAP	Popis
Communication check		
GET /healthcheck	IsHealthy	Verifies that the web service is responding.
Creation of shipments		
POST /api/Customer/Range	AssignRange	Assigns a customer-specific parcel number range (required only if the customer uses custom numbering).
–	ServiceList	Retrieves the allowed transport types (no longer used – no REST equivalent).
GET /api/Customer/Service	AddServiceList	Retrieves the list of services available for a specific customer.
POST /api/Pickup	CreatePickUp	Creates a pickup.
POST /api/Shipment/Export	InsertExport	Creates an export shipment.
POST /api/Shipment/Order	InsertOrder	Creates a shipment order.
Label printing, pickup list printing and shipment deletion		
GET /api/Shipment/Label	GetLabel	Prints a shipment label. If this method is not used, up-to-date routing tables must be maintained and the required parcel and cargo label formats must be followed (see the System Integration Manual).
GET /api/Shipment/PickupList	GetPickupList	Prints the pickup list (handover protocol).
DELETE /api/Shipment	DeleteShipment	Deletes a created shipment.
Checking of shipment status		
GET /api/Shipment/Status	ShipmentStatus	Returns the latest shipment status.
GET /api/Shipment/Detail	ShipmentDetail	Returns the complete list of shipment statuses (Track & Trace).

Minimal test scenario

1. Test the implemented methods
 - a. Verify that the API is responding
 - b. Verification of allowed types of transport and allowed services
 - c. Generate a number range using the appropriate method¹
 - d. Create a pickup or verify whether a pickup was automatically generated
 - e. Create a test set of shipments
 1. shipment without services
 2. shipment with HDS/HDP category – **REST**²
 3. shipment with HDS/HDP service – **SOAP**²
 4. shipment with COD service
 5. shipment with multiple packages/pieces
2. Generate shipment labels and pickup list for the created shipments and send them for review to the Geis IT department (edi@geis.cz or the specific contact person responsible for the implementation).
3. Retrieve the statuses of selected shipments.
4. Delete the created shipments.

¹ If not used, implementation and testing are not required.

² In SOAP, HDP and HDS are defined as services. In REST, they are specified as the ProductCategory field in the request body (see the Swagger schema for GET /api/Shipment/Export)

REST

REST basics

- REST API uses HTTP methods (GET, POST, DELETE)
- Query parameters are part of the URL (e.g. ?deliveryCountry=CZ)
- Request and response bodies are in JSON format
- The dot "." is used as the decimal separator
- UTF-8 encoding is used

Swagger

The current specification of methods and request/response schemas is available in the Swagger documentation.

Access to the test Swagger:

- <https://gservicetest.geis.cz/swagger/> - access to the CZ test environment
- <https://gservicetest.geis.pl/swagger/> - access to the PL test environment
- <https://gservicetest.geis.sk/swagger/> - access to the SK test environment

Swagger is the primary source of technical specifications for the REST API.

To access all Swagger functionality, authorization is required. Customer code and password are required for authentication. You can request these credentials from your assigned business manager. To log in to Swagger – use the Authorize button.

Authentication

The API uses Basic Authentication. The customer code (address code) and password are used for authentication.

The authentication request header is structured as follows:

Authorization: Basic {**Base64**(customercode:password)}

E.g.:

- **Authorization: Basic a8OzZHrDoWthem7DrWthOmhlc2xv**

Mandatory request header:

Key	Value
Accept	application/json
Authorization	Basic { Base64 (customercode:password)}
Content-Type	application/json

Language

The response language can be controlled using the HTTP header Accept-Language.

If the header is not provided, the API will return responses using the default language configured on the server instance. This default may vary depending on the environment or deployment configuration.

Clients are recommended to explicitly specify the desired language using the Accept-Language header. Supported values are cs-CZ, en-GB, pl-PL, sk-SK.

Endpoint overview

GET /api/customer/service

Returns the list of additional services that the customer has set up in the primary system. E.g. COD – Cash on delivery, POJ – Insurance etc...

GET /api/Status

Returns a list of the shipment statuses which are visible in Track & Trace.

GET /api/PackingType

Returns a list of manipulation units allowed for the customer in the primary system.

GET /api/Routing/Delivery

Returns the country of the recipient, depot and direction. This information is important for generation of the label - see the [System Integration Manual](#) for Customers, section 2.3 "Specification for Printing Labels", page 23.

GET /api/Shipment/Detail

Returns information about the sender, recipient, status, and services of the shipment.

GET /api/Routing/Table

Returns the routing table. It is possible to return the table for 7 days in advance. This information is important for creation of the label see [System integration manual](#) for customers section 2.3. Specification for printing labels on the page 23.

POST /api/Pickup

Creates a pickup request associated with expedition type of shipments. The request serves as a notification to arrange the collection of shipments according to the specified requirements. Pickups can be ordered for the same day the request is created, but no **later than 10:00 AM**. After this time, the pickup date must be set according to the following rule: **current date + 1 business day**.

POST /api/Shipment/Export

Creates a shipment of expedition type. Expedition shipments can be created for current day with some limitations – it is required to have a pickup otherwise the method will return an error. To correctly fill an expedition request body you may need to know the list of services (**GET /api/Customer/Service**) or the list of transport items (**GET /api/PackingType**). List of services and required parameters can be found [here](#)

For subitems, the following rules apply: they can only be used for domestic shipments. The maximum number of subitems per shipment item is 10. For returnable non-exchangeable subitems, the **Number** field must also always be provided. This number must be unique within the shipment.

POST /api/Shipment/Order

Creates a shipment of order type. An order can be created no earlier than the next business day. To complete the order, it may be necessary to retrieve certain data, such as the list of services (**GET /api/Customer/Service**) or the list of packing types (handling units, **GET /api/PackingType**). The list of services and required parameters can be found [here](#)

Orders from countries other than the customer's country cannot be shipped to third-party recipients. In such cases, the recipient must always be the customer's address. The countries from which orders can be placed will be provided by the assigned sales manager.

POST /api/Customer/Range

This method is used to assign number ranges from which individual shipment numbers are generated. It returns a range defined by a lower bound (**RangeLow**) and an upper bound (**RangeHigh**). Both values must be stored for future use, and both values can be used as shipment numbers. Upon the first call, each customer is assigned a range of 5,000 numbers. A new range can be assigned once fewer than 500 numbers remain available in the current range.

GET /healthcheck

Returns "Healthy" and the current API version if the service is running. Used for basic service availability testing.

GET /api/Shipment/Label

This method generates a shipment label in the requested format. The shipment must belong to the authenticated customer and must have been created via the web service; otherwise, the method returns an error. Labels are returned only for shipments created using the POST /api/Shipment/Export method. The label data is encoded in **Base64 format**.

This method allows retrieving labels for multiple shipments in a single request. The limit is set to 200 shipments per method call.

GET /api/Shipment

This method retrieves a list of shipments created for a specific pickup date. The method returns a list of shipments along with a flag indicating whether each record represents an order or an export shipment. The resulting list of shipments depends on the specified input parameters.

GET /api/Shipment/Status

This method returns the status of one or more shipments based on the submitted request. It allows retrieving the status of multiple shipments in a single request. The limit is set to 200 shipments per method call.

GET /api/PickupList

This method generates a pickup list (handover protocol) in PDF format. The PDF format is identical to the one used in the GClient and GPortal applications. Shipments included in the pickup list must not be real, i.e., they must not have any operational statuses assigned. As input parameters, the method requires the pickup date (mandatory) and optionally a shipment number or an array of shipment numbers. If no shipment number is provided, all shipments scheduled for pickup on the specified date will be included. The PDF data is returned in **Base64-encoded format**.

DELETE /api/Shipment

This method is used to cancel a shipment. A shipment can be deleted via this method only if it is still in a data-only state (i.e., no operational statuses have been assigned) and belongs to the authenticated customer. Additionally, a shipment of type *order* can be cancelled no later than the day before pickup (by 24:00). Cancelling an order shipment will also automatically cancel the associated pickup. As input parameters, the method accepts a shipment number or an array of shipment numbers. The response returns a list of shipment numbers together with a flag indicating whether each shipment was successfully deleted (true / false). Multiple shipments can be deleted in a single request. The limit is set to 200 shipments per method call.

GET /api/Customer/Range

This method retrieves details of the number ranges assigned to the customer.

GET /api/Pickup

Returns information about pick-ups for specified date.

Overview of HTTP statuses

HTTP Status	Meaning	Description
200	Success	The request was successfully processed. The response contains a response body.
201	Created	The request resulted in the creation of a new entity. The response contains details of the created record.
400	Bad Request	The request contains invalid or missing data. Error details are provided in the response body.
401	Unauthorized	Invalid or missing authentication credentials.
404	Not Found	The requested resource or entity was not found.
409	Conflict	The request conflicts with the current state of the system (e.g., a record already exists)
415	Unsupported Media Type	Incorrect Content-Type or invalid request format.
429	Rate Limit Exceeded	The allowed number of method calls within the specified time period has been exceeded.

Examples of error codes can be found at the [end of the document](#)

Service parameters - REST

List of services with specific parameters (mandatory in bold).

Code	Service	Parameter _1	Parameter _2	Parameter _3	Parameter _4	Availability	Cannot be used with service:
COD	Cash on delivery	Cash on delivery value	Cash on delivery currency	Variable symbol*	IBAN	CZ/SK/PL	-
POJ	Insurance	The value of insurance	Currency			CZ/SK/PL	-
PHO	Telephone notification	Phone number**				CZ/SK/PL	HDS, HDP
SMS	SMS notification	Phone number**				CZ/SK/PL	HDS, HDP
RSE	SENT registration	Reference	Key			CZ/SK/PL	-
HDS** *	Home Delivery Standard	Phone number**				CZ/SK	HDP, SMS, PHO, DTR, OSO, ADR, ADQ
HDP** *	Home Delivery Premium	Phone number**				CZ/SK	HDS, SMS, PHO, DTR, OSO, ADR, ADQ

*Variable symbol – this field is used in the COD service. If it is filled in, it replaces the shipment number in the report.

** For SMS, HDP, HDS and PHO services, the recipient's number must be added with the prefix in the section and parameter 1. The recipient's prefix must agree with the recipient's country (eg if the shipment is going to Germany, the phone number must have the prefix +49!). The only exceptions are Slovak telephone numbers in the Czech Republic and vice versa.

***The HDS and HDP services are not used in REST GService. They are replaced by the productCategory parameter, which is included in the request body when creating a shipment. The parameter can have the following values: B2B, HDS, HDP.

Services that are not listed in the table do not require parameters.

SOAP

Structure of the XML request

- Webservice GService uses SOAP protocol for sending and receiving **XML messages**.
- It is very crucial that all **XML elements** of the same level **are alphabetically sorted (A->Z)**.
- Data contained in the elements out of the alphabetical order **will not be processed**.
- The dot "." is used as a separator for decimal places.

Web service base classes

The base classes are mentioned below. They are used in almost all methods of service.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	Generic	Input object (specific for call method)	Y

RequestHeader

Name	Type	Description	Mandatory
CustomerCode	String	Customer code	Y*
Password	String	Password for login	Y*
Language	Language	Language for communication	Y

**it does not have to be mandatory for all calls methods.*

Contact

Name	Type	Length	Description	Mandatory
Email	String	50	E-mail	Y*
FullName	String	50	Full name	
Phone	String	50	Phone number in international format (e.g +420 111 222 333)	Y*

Address

Name	Type	Length	Description	Mandatory
Name	String	50	Name	Y
Name2	String	50	Name supplement	
Street	String	50	Street	Y
City	String	50	City	Y
ZipCode	String	10	ZipCode	Y
Country	String	2	Country in ISO 3166-1 alpha-2 format	Y

Language

Value	Name/Description
CZ	Czech
SK	Slovak
EN	English
DE	German
PL	Polish

Currency

Value	Name/Description
UND	Undefined value currency
CZK	Czech crown
EUR	Euro
CHF	Swiss franc
USD	American dollar
PLN	Polish Zloty
HUF	Hungarian forint

Response

Name	Type	Description
Request	Request	Input for the called method (it is set only in the case when it ends with an error).
ResponseObject	Generic	Output object (specified for call method)
ErrorMessage	String	Error message.
ErrorCode	String	Error code (List of error codes is here)
Status	ResponseStatus	Status of method processing

ResponseStatus

Value	Name/Description
Inserted	Insert was successful.
Deleted	Delete was successful.
Updated	Update was successful.
ErrorOccurred	Error occurred during the processing.
AccesDenied	Access denied (invalid combination of customer code and password)
Processed	Listing/Process in passed right.
NoDataFound	No data found.

ZipCodes

Name	Type	Length	Description
ZipCodeFrom	String	10	Zip code from
ZipCodeTo	String	10	Zip code to

Web service methods

If the inbound request for a web services exceeds the limit specified for each method, the server rejects the request and show response with an error state in the http header: **HTTP Error 413 Request entity too large**

ServiceList

It returns list of transportation services that are set in the primary system. It can return these values:

Value	Name/Description
20	Cargo expedition
21	Cargo orders

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	ServiceRequest	Empty class	Y

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in the case when it ends with an error).
ResponseObject	List< Service >	List of transport services
ErrorMessage	String	Error message.
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing

Service

Name	Type	Length	Description
Code	String	3	Transport service code
Name	String	30	Transport service name

[An example of ServiceList method](#)

AddServiceList

It returns the list of additional services that the customer has set up in the primary system. In these service may be included for example **COD** –Cash on delivery, **POJ** – Insurance etc.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	AddServiceRequest	Instance of the class AddServiceRequest	Y

AddServiceRequest

Name	Type	Length	Description	Mandatory
Service	String	2	Type of transport service (according to the output ServiceList)	Y
DeliveryCountry	String	2	Country of recipient in ISO 3166-1 alpha-2 form	Y

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in the case when it ends with an error)
ResponseObject	List< AddService >	List of services which customer can use.
ErrorMessage	String	Error message.
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing.

AddService

Name	Type	Length	Description
Code	Int	2	Service Id
Abbreviation	String	3	Service abbreviation
Description	String	30	Service description.
ZipCodeFrom	String	10	Zip code from (if set)
ZipCodeTo	String	10	Zip code to (if set)

[An example of AddServiceList call method](#)

StatusList

It returns a list of the shipments statuses which are visible in Track & Trace.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	StatusRequest	StatusRequest instance class	Y

StatusRequest

Name	Type	Description	Mandatory
DistributionChannel	Int	Type of transport service (2 – Cargo)	Y

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case when it ends with an error)
ResponseObject	List< Status >	List of statuses according to the transport service
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status process method.

Status

Name	Type	Length	Description
Code	String	3	Status code
Description	String	20	Service description
DistributionChannel	Int	1	Type of transport service (2 – Cargo)

[An example of StatusList call method](#)

WrapList

This returns a list of manipulation units allowed for the customer in the primary system.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	WrapRequest	Instance of the WrapRequest class (empty class)	Y

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	List< Wrap >	List of manipulation units which are allowed for the customer.
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status process method.

Wrap

Detail of the manipulation unit.

Name	Type	Length	Description
Code	String	3	Manipulation unit code
Description	String	50	Manipulation unit description
Returnable	Bool		Returnable package

[An example of WrapList call method](#)

DeliveryRouting

It returns country of the recipient, depot and direction. This information is important for compilation of the label - see [System integration manual for customers](#) section 2.3. Specification for printing labels on the page 23.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	DeliveryRoutingRequest	Instance of class DeliveryRoutingRequest	Y

DeliveryRoutingRequest

Name	Type	Length	Description	Mandatory
DistributionChannel	Int		Type of transport service (2 – Cargo)	Y
CountryIn	String	2	Sender's country	Y
ZipCodeIn	String	10	Sender's Zip code	Y
CountryOut	String	2	Recipient's country	Y
ZipCodeOut	String	10	Recipient's Zip code	Y
TemperatureMode	String		Temperature mode (not used)	

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	DeliveryRouting	Information about routing.
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing

DeliveryRouting

Name	Type	Length	Description
DeliveryRoute	String	4	Delivery routes code
DepCode	String	3	Depot code

[An example of DeliveryRouting call method](#)

ShipmentDetail

The method returns information about the sender, recipient, status, and services of the shipment.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	ShipmentRequest	Instance of the class ShipmentRequest	Y

ShipmentRequest

Name	Type	Length	Description	Mandatory
DistributionChannel	Int		Type of transport service (2 – Cargo)	Y
ShipmentNumber	String	13	Number of the shipment cargo	Y

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	Shipment	Details about shipment
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing.

Shipment

Name	Type	Length	Description
ShipmentNumber	String	13	Number of the shipment cargo
ShipmentNumberCust	String	50	Shipment references
DelivPerson	String	50	Name of the shipment recipient
Weight	Decimal	(5,2)	Weight of the shipment
CodVarCode	String	10	Payment identification code for cash on delivery
CodValue	Decimal	(12,2)	Cash on delivery value
DelivDate	DateTime		Date of delivery
BorderDate	DateTime		Sending date (date of loading to main run)
Volume	Decimal	(9,3)	Volume
RecAddress	Address		Address of the recipient
SendAddress	Address		Address of the sender
History	List< PackageHistory >		List of shipment statuses
ColliHistory	List< ColliHistory >		List of statuses for individual pieces of shipment
Services	List< PackageService >		List of shipment services
Rows	List< PackageRow >		List of shipment rows

PackageHistory

Name	Type	Length	Description
StatusDate	DateTime		Date of status
StatusName	String	50	Name of status
StatusCode	String	10	Code of status
DepName	String	50	Depot of status
Description	String	50	Note

ColliHistory

Name	Type	Length	Description
StatusDate	DateTime		Date of status
StatusName	String	50	Name of status
StatusCode	String	10	Code of status
DepName	String	50	Depot of status
Description	String	50	Note
ColliNumber	String	25	Number of the piece

PackageService

Name	Type	Length	Description
Name	String	30	Name of service
Price	Decimal	(12,2)	Service price
Currency	Currency		Currency

PackageRow

Name	Type	Length	Description
CountItems	Int		Number of pieces
PackingType	String	30	Package
Description	String	30	Description
Weight	Decimal	(9,2)	Weight
Reference	String	50	References of the item
Volume	Decimal	(9,3)	Volume
Dimension	String		Dimension (return empty value)

Error codes

Code	Shipment status	Description
0	OK	Non-delivery, it has status
1	OK	Delivery, it has status
2	OK	Does not have status yet
3	STORNO	Shipment was cancelled.
4	Není	Shipment does not exist.

[An example of ShipmentDetail call method](#)

RouteTable

It returns route table. It is possible to return the table for 7 days in advance. This information is important for creation of label see [System integration manual for customers](#) section 2.3. Specification for printing labels on the page 23.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	RouteTableRequest	Instance of the class RouteTableRequest	Y

RouteTableRequest

Name	Type	Description	Mandatory
DistributionChannel	Int	Type of transport service (2 – Cargo)	Y
ValidDate	DateTime	Validity date	Y

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	List< RouteTable >	List of items from routing table
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing

RouteTable

Name	Type	Length	Description
DistributionChannel	Int		Service
Country	String	2	Country
Depot	String	2	epot code
Route	String	4	Route
ZipCodes	ZipCodes		Zip code from/to
ValidDate	Datetime		Validity

[An example of RouteTable call method](#)

CreatePickUp

It creates pick up that belongs to expedition. It is an advice that shipments have been picked up according to the requirement. The pickup may be ordered on the same day when you create requirement, but maximum until 10:00 o'clock. After that, the date of pick up must be set up according to the definition: **current day + 1, but it must be a working day.**

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	PickUpRequest	Instance of class PickUpRequest	Y

PickUpRequest

Name	Type	Length	Description	Mandatory
DistributionChannel	Int		Type of transport service(2 – Cargo)	Y
DateFrom	DateTime		Date and time of pick up from	Y
DateTo	DateTime?		Date and time of pick up to (day and month must be same as DateFrom)	
CountItems	Int		Number of pieces	Y
TotalWeight	Decimal	(9,2)	Total weight (kg)	Y
Contact	Contact		Contact information	Y
Note	String	50	Note	
PickUpItems	List< PickUpItem >		List of pick up items	
TemperatureMode	Int		Code of temperature mode *	

* Assumes values 1, 2 (only for PL)

Contact

Name	Type	Length	Description	Mandatory
Email	String	50	E-mail	Y
FullName	String	50	Full name	
Phone	String	50	Phone number in international format (e.g +420 111 222 333)	Y

PickUpItem

Name	Type	Length	Description	Mandatory
CountItems	Int		Number of items	Y
Weight	Decimal	(9,2)	Weight (kg)	Y
Type	String	3	Type of manipulating item	Y
Reference	String	50	Shipment references	

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	PickUp	Empty class
ErrorMessage	String	Error message
ErrorCode	String	Error code (List of error codes is here)
Status	ResponseStatus	Status of method processing

[An example of CreatePickUp call method](#)

InsertExport

It creates type of shipment expedition for the product (Cargo). Expedition can be created for current day, but there must be pick up for this day, otherwise the method ends with an error. To complete the expedition you may need to know several data such as list of services – method [AddServiceList](#), list of transport items (manipulating items, [WrapList](#) method). Object InsertExport consists of several objects which can be entered as field of objects (e.g. List of services) see next table ExportRequest.

Note: It is necessary to preserve the order of individual elements in accordance with tables below. Otherwise, the submitted information may be skipped.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	ExportRequest	Instance of class ExportRequest	Y

ExportRequest

Name	Type	Length	Description	Mandatory
BurstId	String	50	ID batch (according to this parameter to choose the shipment by method ListOfShipments)	
CoverAddress	Address		Cover address	
DeliveryAddress	Address		Delivery address	Y
DeliveryContact	Contact		Contact for the recipient	
DistributionChannel	Int		Type of transport service (2 – Cargo)	Y
ExWorks	Bool	1	ExWorks service (0 - no; 1 - yes)	
ExportItems	List< ExportItem >		Transport items	Y
ExportServices	List< ExportService >		List of shipment services	
MergedPackages	List< MergedItem >		It was used for Parcel	
MergedShipment	Bool		It was used for Parcel. Always "false"	
Note	String	50	Note for recipient	
NoteDriver	String	50	Note for driver	
PartnerNumber	String	40	External number of shipment	
PersonalPickUp	Bool	1	Personal collection service (0 - no; 1 - yes)	
PickUpDate	DateTime		Date of pick up (pick up must be created for this date)	Y
Reference	String	30	Shipment references	
SenderContact	Contact		Sender's contact	
ShipmentNumber	String	13	Number of shipment (when is empty, shipment number is assigned automatically. Is possible fill in according to output method AssignRange)	
Weight	Decimal	(9,2)	It was used for Parcel	

ExportService

Name	Type	Description	Mandatory
Code	String	Service code	Y
Parameter_1	String	Parameter 1	
Parameter_2	String	Parameter 2	
Parameter_3	String	Parameter 3	
Parameter_4	String	Parameter 4	
Parameter_5	String	Parameter 5	
Parameter_6	String	Parameter 6	
Parameter_7	String	Parameter 7	
Parameter_8	String	Parameter 8	
RODParameters	List< RODParameter>	It was used for Parcel (PL)	

Services parameter - SOAP

List of services with specific parameters (mandatory in bold).

Code	Service	Parameter_1	Parameter_2	Parameter_3	Parameter_4	Availability	Cannot be used with service:
COD	Cash on delivery	Cash on delivery value	Cash on delivery currency	Variable symbol*	IBAN	CZ/SK/PL	-
POJ	Insurance	The value of insurance	Currency			CZ/SK/PL	-
PHO	Telephone advice	Phone number**				CZ/SK/PL	HDS, HDP
SMS	SMS advice	Phone number**				CZ/SK/PL	HDS, HDP
RSE	SENT registration	Reference	Key			CZ/SK/PL	-
HDS	Home Delivery Standard	Phone number**				CZ/SK	HDS, SMS, PHO, DTR, OSO, ADP, ADQ
HDP	Home Delivery Premium	Phone number**				CZ/SK	HDS, SMS, PHO, DTR, OSO, ADP, ADQ

*Variable symbol - this field is used in the COD service. If it is filled in, it replaces the shipment number in the report.

** For SMS, HDP, HDS and PHO services, the recipient's number must be added with the prefix in the <DeliveryContact> section and parameter 1. The recipient's prefix must agree with the recipient's country (eg if the shipment is going to Germany, the phone number must have the prefix +49 !). The only exceptions are Slovak telephone numbers in the Czech Republic and vice versa.

Services that are not listed in the table do not require assignment of parameters.

ExportItem

Name	Type	Length	Description	Mandatory
CountItems	Int		Number of items	Y
Description	String	50	Description	
Type	String	3	Manipulation unit code (WrapList)	Y
Reference	String	50	References of the item	
SubItems	List< SubItems >		Secondary handling units	
Weight	Decimal	(9,2)	Weight (kg)	Y
Volume	Decimal	(9,3)	Volume (v m^3)	Y*
Height	Decimal	(9,2)	Height (in meters)	Y
Width	Decimal	(9,2)	Width (in meters)	Y
Length	Decimal	(9,2)	Length (in meters)	Y

*Optional in case of entering dimensions (it is calculated from dimensions)

SubItems

For subitems, the following rules apply: they can only be used for domestic shipments. The maximum number of subitems per shipment item is 10. For returnable non-exchangeable subitems, the **Number** field must also always be provided. This number must be unique within the shipment.

Název	Type	Délka	Popis	Povinný
Type	String	3	Code of secondary handling unit(WrapList)	Y
Number	String	50	Number of secondary handling unit	Y*

*The type of Handling unit must be Returnable. The type of Handling unit can be identified using the [WrapList](#) method. If a secondary handling unit of type Returnable specific is entered, the Number field is mandatory. If a secondary handling unit of any other type than Returnable specific is entered, the Number field must not be filled in. At this moment, no Handling unit is of the Returnable specific type.

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	Export	Empty class
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing

[An full example of InsertExport call method for merged shipment](#)

InsertOrder

It creates type of shipment order. The order can be created first for the next working day. To complete the order some data are required such as list of services (method [AddServiceList](#)) or list of transport pieces (manipulating items, [WrapList](#) method).

Orders from countries other than the customer's country cannot be shipped to third-party recipients. In such cases, the recipient must always be the customer's address. The countries from which orders can be placed will be provided by the assigned sales manager.

Object InsertOrder is created from several objects that may be entered as field of objects (e.g. list of services) see following table OrderRequest.

Note: It is necessary to preserve the order of individual elements in accordance with tables below. Otherwise, the submitted information may be skipped.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	OrderRequest	Instance of class OrderRequest	Y

OrderRequest

Name	Type	Length	Description	Mandatory
DistributionChannel	Int		Type of transport service (2 – Cargo)	Y
ExWorks	Bool	1	ExWorks service (0 - no; 1 - yes)	
PartnerNumber	String	40	External partner number	
PickUpDate	DateTime		Date of pick up	Y
Reference	String	30	Shipment references	
Note	String	50	Note to recipient	
NoteDriver	String	50	Note for driver	
Weight	Decimal	(9,2)	Weight (kg)	
Volume	Decimal	(9,3)	Volume (m^3)	
MergedOrder	Bool		It was used for Parcel. Always "false"	
ExportItems	List< ExportItem >		List of transport pieces	Y
SenderAddress	Address		Sender's address	Y
SenderContact	Contact		Sender's contact	
DeliveryAddress	Address		Recipient's address	Y
DeliveryContact	Contact		Recipient's contact	
ExportServices	List< ExportService >		List of services	
MergedOrders	List< MergedItem >		It was used for Parcel.	
BurstId	String	50	ID batch (according to this parameter to choose the shipment by method ListOfShipments)	

Response



Geis CZ s.r.o.
Zemská 211/I
337 01 Ejovice
Česká republika

Telephone: +420 951 277 777

E-mail: info@geis.cz
Internet: www.geis-group.cz
IČO: 44567359
DIČ: CZ44567359

Bankovní spojení:
UniCredit Bank Czech Republic and Slovakia a.s.

Č.ú.: 1545471501/2700 (CZK)
CZK IBAN:
CZ97 2700 0000 0015 4547 1501
Č.ú.: 1545471560/2700 (EUR)
EUR IBAN
CZ56 2700 0000 0015 4547 1560
SWIFT: BACXCZPP

Společnost je zapsaná v obchodním rejstříku, vedeném Krajským soudem v Plzni, oddíl C, číslo vložky 14475.

Pracujeme na základě našich všeobecných obchodních podmínek - aktuální stav.

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	Order	Information about entered order
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error code is here)
Status	ResponseStatus	Status of method processing

[An example of InsertOrder call method](#)

AssignRange

Method for allocating customer number series, which are used as shipment numbers. Method returns range bounded by minimum value (RangeLow) and maximum value (RangeHigh), both values should be saved for later use and both values could be used as shipment numbers. Each customer will be assigned a range of 5000 numbers on the first call. A new set can be assigned after less than 500 numbers remain from the old range.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	RangeRequest	Instance of class RangeRequest	Y

RangeRequest

Name	Type	Length	Description	Mandatory
DistributionChannel	Int		Type of transport service (2 – Cargo)	Y
TransportType	String	1	It was used for Parcel	
Range	Int		It was used for Parcel	

* the parameter is mandatory and is listed only for parcel

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it ends with an error)
ResponseObject	Range	Information about range
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error code is here)
Status	ResponseStatus	Status of method processing

Range

Name	Type	Description
RangeLow	String	Lower limit of range
RangeHigh	String	Upper limit of range

[An example of AssignRange method call](#)

IsHealthy

In case of successful call method, this service answers with the text "Healthy". This method is used for simple text in case that service is running.

[An example of IsHealthy method call](#)

GetLabel

This method allows to get labels for shipments in specified format. The shipment has to be created via GService and must belong to the logged-on customer, otherwise it ends with error message. The method returns the label only for expeditions (shipments based on the InsertExport method).

Using this method, it is possible to obtain labels for multiple shipments at once. The limit is set at 200 cargo shipments in one method call.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	LabelRequest	Instance of class LabelRequest	Y

LabelRequest

Name	Type	Description	Mandatory
ShipmentNumbers	List <LabelItem>	List of objects LabelItem. Objects LabelItem contains attribute ShipmentNumber , this attribute is filled with number of shipment printed on the label.	Y
Format	Int	Determines format of generated file. Allowed values: 1 – PDF 2 – EPL 3 – ZPL 4 – BMP (BMP file is transformed from TIFF file.) 5 – PDF (Thermal printer format 10x15cm)	Y
Resolution	Int	Resolution. Fill this field only in case EPL/ZPL label. Allowed values : 200 – 200 DPI etiketa 300 – 300 DPI etiketa	Y*
DistributionChannel	Int	Type of transport service (2 – Cargo)	Y
Position	Int	Determines the position of label in PDF file (one-page A4 allows print 4 labels). Allowed values: 1, 2, 3, 4. For Format no. 5 "PDF for thermal printers" always use value 1.	Y**

*Required for Format 2 and 3

**Required for Format 1

As response the GService returns the object LabelResponse

LabelResponse



Geis CZ s.r.o.
Zemská 211/I
337 01 Ejovice
Česká republika

Telefon: +420 951 277 777

E-mail: info@geis.cz
Internet: www.geis-group.cz
IČO: 44567359
DIČ: CZ44567359

Bankovní spojení:
UniCredit Bank Czech Republic and Slovakia a.s.

Č.ú.: 1545471501/2700 (CZK)
CZK IBAN:
CZ97 2700 0000 0015 4547 1501
Č.ú.: 1545471560/2700 (EUR)
EUR IBAN
CZ56 2700 0000 0015 4547 1560
SWIFT: BACXCZPP

Společnost je zapsaná v obchodním rejstříku, vedeném Krajským soudem v Plzni, oddíl C, číslo vložky 14475.

Pracujeme na základě našich všeobecných obchodních podmínek - aktuální stav.

Name	Type	Description	Mandatory
ShipmentNumbers	List <LabelItem>	List of objects LabelItem, which should be generated.	Y
LabelData	String	In response are the binary data Base64 encoded. Allowed formats: Format – 1 (PDF) - .pdf Format – 2 (EPL) - .txt Format – 3 (ZPL) - .txt Format – 4 (BMP) - .bmp Format – 5 (PDF) - .pdf	Y

[An example of GetLabel method call](#)

ListOfShipments

This method returns a list of shipments created for specific pickup date. Method has three input parameters, first is date of pickup (DateExpOrOrder), the second parameter is identifier of the shipments groups (BurstId - obligatory) and the third is the creation date (Created). The method returns a list of shipments and an indication of whether it is an order or an expedition. The resulting list of shipments is based on the entered input parameters.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	ShipmentRequest	Instance of class ShipmentRequest	Y

ShipmentRequest

Name	Type	Description	Mandatory
DateExpOrOrder	DateTime	Date of order or expedition	Y*
BurstId	String (50)	Identifier of shipments group	Y*
Created	DateTime	Date of creation of order or expedition	Y*

No parameter needs to be entered - In this case, the current date is automatically added to the "created" parameter.

ShipmentResponse

Name	Type	Description	Mandatory
ShipmentNumber	String	Number of shipment	Y
IsOrder	Bool	Identifier is i tis order or expedition	

[An example of ListOfShipment method call:](#)

ShipmentStatus

Method returns status of shipment / shipments based on request. Statuses are equal with statuses of GClient in list of expeditions / orders. As input parameter method demands number / array numbers of shipment.

Within this method, it is possible to identify multiple shipments at once. The limit is set to 200 cargo shipments in one method call.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	ShipmentStatusRequest	Instance of class ShipmentStatusRequest	Y

List of states:

Code	Name
ZGC	Created
PCK	Picked up
NCT	On the road
NDC	On delivery
CDR	Partial delivery
DLV	Delivered
BCK	Back to sender
SGC	Canceled

ShipmentStatusRequest

Name	Type	Description	Mandatory
ShipmentNumber		List of object of type ShipmentStatusItem, which includes array of ShipmentNumber	Y

ShipmentStatusResponse

Name	Type	Description
ShipmentNumber	String	Number of shipment
StatusCode	String	Code of status
StatusName	String	Name of status

[An example of ShipmentStatus method call](#)

GetPickupList

Method creates pickup list (handover protocol) in PDF format. The PDF format is the same as in application GClient or GPortal. Shipments on handover protocol must not be real, that means the shipment must not have any operating status. As input parameter method accepts date of pickup of shipments (mandatory), type of transport (cargo x parcel - mandatory) and number / array of numbers of shipment (not mandatory). If there is any shipment number, the method returns all shipments with specific date of pickup.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	PickupListRequest	Instance of class PickupListRequest	Y

PickupListRequest

Name	Type	Description	Mandatory
ExpeditionCreateDate	DateTime	Date of pickup	Y*
DistributionChannel	Int	Distribution channel (cargo = 2)	Y
ShipmentNumbers	List	List of object of type PickupListShipmItem, which includes the parameter ShipmentNumber	Y*

* Mandatory is to provide pickup-date or list of shipments

PickupListReponse

Name	Type	Description
PickupListData	Byte []	Array of bytes. Includes byte representation of the PDF pickuplist

[An example of GetPickupList method call](#)

DeleteShipment

Method for deleting shipments. This method is able to delete shipment, but only in case, that the shipments is not real (does not have any operating status) and belongs to the logged in customer. In addition, the shipment type order can be deleted at the latest on the day before the pickup date (until 24:00) and the pickup for the deleted order will be canceled at the same time.

As input parameters method accepts number / array of numbers of shipment. Method returns list of numbers of shipment and information if the shipment was deleted or not (true x false).

Within this method, multiple shipments can be deleted at once. The limit is set to 200 cargo shipments in one call method.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	DeleteShipmentRequest	Instance of class DeleteShipmentRequest	Y

DeleteShipmentRequest

Name	Type	Description	Mandatory
ShipmentsNumbers	List	List of objects of type DeleteShipmentItem	Y

DelteShipmentItem

Name	Type	Description	Mandatory
ShipmentNumber	String	Number of shipment	Y
DistributionChannel	Int	Distribution channel	Y

DeleteShipmentResponse

Name	Type	Description
ShipmentNumber	String	Number of shipment
IsStorno	Bit	Information, if the shipment was deleted or not
DistributionChannel	Int	Distribution channel

[An example of DeleteShipment method call](#)

RangeDetail

Method to find out the details about the number ranges that customer has assigned. The input parameters for the method are customer logins credentials, type of transport service and type of transport.

Request



Geis CZ s.r.o.
Zemská 211/I
337 01 Ejovice
Česká republika

Telefon: +420 951 277 777

E-mail: info@geis.cz
Internet: www.geis-group.cz
IČO: 44567359
DIČ: CZ44567359

Bankovní spojení:
UniCredit Bank Czech Republic and Slovakia a.s.

Č.ú.: 1545471501/2700 (CZK)
CZK IBAN:
CZ97 2700 0000 0015 4547 1501
Č.ú.: 1545471560/2700 (EUR)
EUR IBAN
CZ56 2700 0000 0015 4547 1560
SWIFT: BACXCZPP

Společnost je zapsaná v obchodním rejstříku, vedeném Krajským soudem v Plzni, oddíl C, číslo vložky 14475.

Pracujeme na základě našich všeobecných obchodních podmínek - aktuální stav.

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	RangeDetailRequest	Instance of class RangeDetailRequest	Y

RangeDetailRequest

Name	Type	Length	Description	Mandatory
DistributionChannel	Int	1	Type of transport service (2 – Cargo)	Y
TransportType	Int	1	It was used for Parcel	

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it end with an error)
ResponseObject	RangeDetail	Information about number ranges
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing

RangeDetail

Name	Type	Description
AssignAllowed	Boolean	Specified whether you can assign a new number range
AssignRangeLimit	String	Limit for AssingRange method
RangeDetailItem	List< RangeDetailItem >	List of number ranges

RangeDetailItem

Name	Type	Description
LastUsedNumber	String	Last used number from range
RangeHigh	String	Upper limit of range
RangeLow	String	Lower limit of range
RemainingNumber	String	Number of remaining numbers from range

[An example of RangeDetail method call](#)

PickupInfo

Method to find out the details about the pickups on selected day. The input parameters for the method are customer logins credentials, pickup date, type of transport service and element isOrder.

Request

Name	Type	Description	Mandatory
Header	RequestHeader	Header of request	Y
RequestObject	PickupInfoRequest	Instance of class PickupInfoRequest	Y

RangeDetailRequest

Name	Type	Délka	Description	Mandatory
DistributionChannel	Int	1	Type of transport service (2 – Cargo)	Y
Date	DateTime	1	Pickup date	Y
IsOrder	Bool	1	Is this pickup for order? (1 – Yes, 0 – No)	Y
TemperatureMode	Int		Code of temperature mode**	

* Assumes values 1, 2 (only for PL)

Response

Name	Type	Description
Request	Request	Input data for the called method (it is set only in case it end with an error)
ResponseObject	PickupInfo	Information about number ranges
ErrorMessage	String	Error message
ErrorCode	String	Error code (list of error codes is here)
Status	ResponseStatus	Status of method processing

PickupInfo

Name	Type	Description
PickupInfoItem	List< PickupInfoItem >	List of the pickups

PickupInfoItem

Name	Type	Description
City	String	City
CountryCode	String	Country
Date	DateTime	Pickup date
DistributionChannel	Int	Type of transport service (2 - Cargo)
IsOrder	Bool	Is this pickup for order? (1 – Yes, 0 – No)
Name	String	Name of the customer
State	String	Status of the pickup
Street	String	Street
ZipCode	Int	Zip code

[An example of PickupDetail method call:](#)

An examples of method calls and responses

ServiceList

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ServiceList>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject/>
      </tem:Request>
    </tem:ServiceList>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ServiceListResponse xmlns="http://tempuri.org/">
      <ServiceListResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:Service>
            <a:Code>20</a:Code>
            <a:Name>cargo expedice</a:Name>
          </a:Service>
          <a:Service>
            <a:Code>21</a:Code>
            <a:Name>cargo objednávky</a:Name>
          </a:Service>
        </a:ResponseObject>
        <a:Status>Processed</a:Status>
      </ServiceListResult>
    </ServiceListResponse>
  </s:Body>
</s:Envelope>
```

AddServiceList

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:AddServiceList>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:DeliveryCountry>CZ</gser:DeliveryCountry>
          <gser:Service>20</gser:Service>
        </gser:RequestObject>
      </tem:Request>
    </tem:AddServiceList>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <AddServiceListResponse xmlns="http://tempuri.org/">
      <AddServiceListResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:AddService>
            <a:Abbreviation>HDS</a:Abbreviation>
            <a:Code>1117</a:Code>
            <a:Description>Home Delivery Standard</a:Description>
            <a:Service>20</a:Service>
            <a:ZipCodesList i:nil="true"/>
          </a:AddService>
          <a:AddService>
            <a:Abbreviation>COD</a:Abbreviation>
            <a:Code>1002</a:Code>
            <a:Description>COD (Dobírka)</a:Description>
            <a:Service>20</a:Service>
            <a:ZipCodesList i:nil="true"/>
          </a:AddService>
          <a:AddService>
            <a:Abbreviation>SMS</a:Abbreviation>
            <a:Code>1074</a:Code>
            <a:Description>SMS avízo</a:Description>
            <a:Service>20</a:Service>
            <a:ZipCodesList i:nil="true"/>
          </a:AddService>
        </a:ResponseObject>
        <a>Status>Processed</a>Status>
      </AddServiceListResult>
    </AddServiceListResponse>
  </s:Body>
</s:Envelope>
```

```
</AddServiceListResult>  
</AddServiceListResponse>  
</s:Body>  
</s:Envelope>
```

StatusList

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:StatusList>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:DistributionChannel>2</gser:DistributionChannel>
        </gser:RequestObject>
      </tem:Request>
    </tem:StatusList>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <StatusListResponse xmlns="http://tempuri.org/">
      <StatusListResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:StatusInfo>
            <a:Code>10</a:Code>
            <a:Description>Převzetí zásilky</a:Description>
            <a:DistributionChannel>2</a:DistributionChannel>
          </a:StatusInfo>
          ... (zde jsou další statusy)
          <a:StatusInfo>
            <a:Code>50</a:Code>
            <a:Description>Naložení na rozvoz</a:Description>
            <a:DistributionChannel>2</a:DistributionChannel>
          </a:StatusInfo>
          <a:StatusInfo>
            <a:Code>70</a:Code>
            <a:Description>Doručeno</a:Description>
            <a:DistributionChannel>2</a:DistributionChannel>
          </a:StatusInfo>
        </a:ResponseObject>
        <a:Status>Processed</a:Status>
      </StatusListResult>
    </StatusListResponse>
  </s:Body>
</s:Envelope>
```

WrapList

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:WrapList>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject/>
      </tem:Request>
    </tem:WrapList>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <WrapListResponse xmlns="http://tempuri.org/">
      <WrapListResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0</a:ErrorCode>
        <a:ErrorMessage i:nil="true"/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:WrapInfo>
            <a:Code>EP</a:Code>
            <a:Description>JEDNOCESTNA PALETA</a:Description>
            <a:Returnable>false</a:Returnable>
          </a:WrapInfo>
          <a:WrapInfo>
            <a:Code>FP</a:Code>
            <a:Description>EUROPALETA</a:Description>
            <a:Returnable>true</a:Returnable>
          </a:WrapInfo>
          <a:WrapInfo>
            <a:Code>CC</a:Code>
            <a:Description>COLLI</a:Description>
            <a:Returnable>false</a:Returnable>
          </a:WrapInfo>
          <a:WrapInfo>
            <a:Code>HP</a:Code>
            <a:Description>PŮLPALETA</a:Description>
            <a:Returnable>false</a:Returnable>
          </a:WrapInfo>
        </a:ResponseObject>
        <a:Status>Processed</a:Status>
      </WrapListResult>
    </WrapListResponse>
  </s:Body>
</s:Envelope>
```

DeliveryRouting

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:DeliveryRouting>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:CountryIn>CZ</gser:CountryIn>
          <gser:CountryOut>CZ</gser:CountryOut>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser:ZipCodeIn>31200</gser:ZipCodeIn>
          <gser:ZipCodeOut>18000</gser:ZipCodeOut>
        </gser:RequestObject>
      </tem:Request>
    </tem:DeliveryRouting>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <DeliveryRoutingResponse xmlns="http://tempuri.org/">
      <DeliveryRoutingResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0</a:ErrorCode>
        <a:ErrorMessage i:nil="true"/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:DeliveryRoute>8808</a:DeliveryRoute>
          <a:DepCode>88</a:DepCode>
        </a:ResponseObject>
        <a>Status>Processed</a>Status>
      </DeliveryRoutingResult>
    </DeliveryRoutingResponse>
  </s:Body>
</s:Envelope>
```

ShipmentDetail

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ShipmentDetail>
      <!--Optional:-->
      <tem:Request>
        <gser:Header>
          <!--Optional:-->
          <gser:CustomerCode>22108305</gser:CustomerCode>
          <gser:Language>EN</gser:Language>
          <!--Optional:-->
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <!--Optional:-->
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <!--Optional:-->
          <gser:ShipmentNumber>3110000259115</gser:ShipmentNumber>
        </gser:RequestObject>
      </tem:Request>
    </tem:ShipmentDetail>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ShipmentDetailResponse xmlns="http://tempuri.org/">
      <ShipmentDetailResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage>Shipment is undelivered.</a:ErrorMessage>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:BorderDate>0001-01-01 T00:00:00</a:BorderDate>
          <a:CodValue>50000.000</a:CodValue>
          <a:CodVarCode>0000259115</a:CodVarCode>
          <a:ColliHistory>
            <a:ColliHistory>
              <a:ColliNumber>311000025911501</a:ColliNumber>
              <a:DepName>Ejpovice</a:DepName>
              <a:Description/>
              <a:StatusCode>20</a:StatusCode>
              <a:StatusDate>2022-04-11 T10:22:55.82</a:StatusDate>
              <a:StatusName>Acceptance at depot</a:StatusName>
            </a:ColliHistory>
            <a:ColliHistory>
              <a:ColliNumber>311000025911501</a:ColliNumber>
              <a:DepName>Ejpovice</a:DepName>
              <a:Description/>
              <a:StatusCode>50</a:StatusCode>
              <a:StatusDate>2022-04-11 T10:38:00</a:StatusDate>
            </a:ColliHistory>
          </a:ResponseObject>
        </a:ShipmentDetailResult>
      </ShipmentDetailResponse>
    </s:Body>
  </s:Envelope>
```

```

    <a:StatusName>Loading for distribution to consignee</a:StatusName>
  </a:ColliHistory>
  <a:ColliHistory>
    <a:ColliNumber>311000025911502</a:ColliNumber>
    <a:DepName>Ejpovice</a:DepName>
    <a:Description/>
    <a:StatusCode>20</a:StatusCode>
    <a:StatusDate>2022-04-11T10:23:03.173</a:StatusDate>
    <a:StatusName>Acceptance at depot</a:StatusName>
  </a:ColliHistory>
</a:ColliHistory>
<a:DelivDate>0001-01-01T00:00:00</a:DelivDate>
<a:DelivPerson i:nil="true"/>
<a:History>
  <a:PackageHistory>
    <a:DepName>Ejpovice</a:DepName>
    <a:Description/>
    <a:StatusCode>20</a:StatusCode>
    <a:StatusDate>2022-04-11T10:23:03.173</a:StatusDate>
    <a:StatusName>Acceptance at depot</a:StatusName>
  </a:PackageHistory>
</a:History>
<a:RecAddress>
  <a:City>Chotikov</a:City>
  <a:Country>Czech Republic</a:Country>
  <a:Name>Test</a:Name>
  <a:Name2 i:nil="true"/>
  <a:Street>Platina 88</a:Street>
  <a:ZipCode>32300</a:ZipCode>
</a:RecAddress>
<a:Rows>
  <a:PackageRow>
    <a:CountItems>1</a:CountItems>
    <a:Description/>
    <a:Dimensions/>
    <a:PackingType>EURO PALLET</a:PackingType>
    <a:Reference i:nil="true"/>
    <a:Volume>1.000</a:Volume>
    <a:Weight>100.00</a:Weight>
  </a:PackageRow>
  <a:PackageRow>
    <a:CountItems>1</a:CountItems>
    <a:Description/>
    <a:Dimensions/>
    <a:PackingType>EURO PALLET</a:PackingType>
    <a:Reference i:nil="true"/>
    <a:Volume>1.000</a:Volume>
    <a:Weight>100.00</a:Weight>
  </a:PackageRow>
</a:Rows>
<a:SendAddress>
  <a:City>Plzeň</a:City>
  <a:Country>Czech Republic</a:Country>
  <a:Name>test</a:Name>
  <a:Name2 i:nil="true"/>
  <a:Street>test420</a:Street>
  <a:ZipCode>30100</a:ZipCode>
</a:SendAddress>
<a:Services>

```

```

<a:PackageService>
  <a:Currency>CZK</a:Currency>
  <a:Name>Freight cost</a:Name>
  <a:Price>1518.000</a:Price>
</a:PackageService>
<a:PackageService>
  <a:Currency>CZK</a:Currency>
  <a:Name>Toll CZ</a:Name>
  <a:Price>29.080</a:Price>
</a:PackageService>
<a:PackageService>
  <a:Currency>CZK</a:Currency>
  <a:Name>Fuel surcharge</a:Name>
  <a:Price>450.000</a:Price>
</a:PackageService>
<a:PackageService>
  <a:Currency>CZK</a:Currency>
  <a:Name>Cash on delivery</a:Name>
  <a:Price>70.000</a:Price>
</a:PackageService>
<a:PackageService>
  <a:Currency>CZK</a:Currency>
  <a:Name>Home Delivery Standard</a:Name>
  <a:Price>50.000</a:Price>
</a:PackageService>
</a:Services>
<a:ShipmentNumber>3110000259115</a:ShipmentNumber>
<a:ShipmentNumberCust/>
<a:Volume>2.000</a:Volume>
<a:Weight>200.00</a:Weight>
</a:ResponseObject>
<a:Status>Processed</a:Status>
</ShipmentDetailResult>
</ShipmentDetailResponse>
</s:Body>
</s:Envelope>

```

RouteTable

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:RouteTable>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser:ValidDate>2021-01-18</gser:ValidDate>
        </gser:RequestObject>
      </tem:Request>
    </tem:RouteTable>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <RouteTableResponse xmlns="http://tempuri.org/">
      <RouteTableResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0</a:ErrorCode>
        <a:ErrorMessage i:nil="true"/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:RouteTableInfo>
            <a:Country>CZ</a:Country>
            <a:Depot>37</a:Depot>
            <a:DistributionChannel>2</a:DistributionChannel>
            <a:Route>3714</a:Route>
            <a:ValidDate>2021-01-18T00:00:00</a:ValidDate>
            <a:ZipCodes>
              <a:ZipCodeFrom>37303</a:ZipCodeFrom>
              <a:ZipCodeTo>37310</a:ZipCodeTo>
            </a:ZipCodes>
          </a:RouteTableInfo>
          ... (zde jsou další trasy)
          <a:RouteTableInfo>
            <a:Country>CZ</a:Country>
            <a:Depot>68</a:Depot>
            <a:DistributionChannel>2</a:DistributionChannel>
            <a:Route>6901</a:Route>
            <a:ValidDate>2021-01-18T00:00:00</a:ValidDate>
            <a:ZipCodes>
              <a:ZipCodeFrom>66458</a:ZipCodeFrom>
              <a:ZipCodeTo>66458</a:ZipCodeTo>
            </a:ZipCodes>
          </a:RouteTableInfo>
        </a:ResponseObject>
      </RouteTableResult>
    </RouteTableResponse>
  </s:Body>
</s:Envelope>
```

```
</a:ResponseObject>  
<a:Status>Processed</a:Status>  
</RouteTableResult>  
</RouteTableResponse>  
</s:Body>  
</s:Envelope>
```

CreatePickUp

[GoBack](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:CreatePickUp>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:Contact>
            <gser:Email>email@geis.cz</gser:Email>
            <gser:FullName>Kontaktní jméno</gser:FullName>
            <gser:Phone>+420000000000</gser:Phone>
          </gser:Contact>
          <gser:CountItems>2</gser:CountItems>
          <gser:DateFrom>2021-03-15T14:00:00</gser:DateFrom>
          <gser:DateTo>2021-03-15T15:00:00</gser:DateTo>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser>Note>Poznámka</gser>Note>
          <gser:PickUpItems>
            <gser:PickUpItem>
              <gser:CountItems>1</gser:CountItems>
              <gser:Reference>111</gser:Reference>
              <gser:Type>FP</gser:Type>
              <gser:Weight>4</gser:Weight>
            </gser:PickUpItem>
          </gser:PickUpItems>
          <gser:TotalWeight>4</gser:TotalWeight>
        </gser:RequestObject>
      </tem:Request>
    </tem:CreatePickUp>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <CreatePickUpResponse xmlns="http://tempuri.org/">
      <CreatePickUpResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0</a:ErrorCode>
        <a:ErrorMessage>Svoz přijat a potvrzen.</a:ErrorMessage>
        <a:Request i:nil="true"/>
        <a:ResponseObject i:nil="true"/>
        <a:Status>Inserted</a:Status>
      </CreatePickUpResult>
    </CreatePickUpResponse>
  </s:Body>
</s:Envelope>
```



InsertExport

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:InsertExport>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:BurstId>test</gser:BurstId>
          <gser:CoverAddress>
            <gser:City>City cover</gser:City>
            <gser:Country>CZ</gser:Country>
            <gser:Name>Name cover</gser:Name>
            <gser:Name2>Name2 cover</gser:Name2>
            <gser:Street>Street 1 cover</gser:Street>
            <gser:ZipCode>31200</gser:ZipCode>
          </gser:CoverAddress>
          <gser:DeliveryAddress>
            <gser:City>City</gser:City>
            <gser:Country>CZ</gser:Country>
            <gser:Name>Name</gser:Name>
            <gser:Name2>Name2</gser:Name2>
            <gser:Street>Street 311</gser:Street>
            <gser:ZipCode>18000</gser:ZipCode>
          </gser:DeliveryAddress>
          <gser:DeliveryContact>
            <gser:Email>email@geis.cz</gser:Email>
            <gser:FullName>Contact</gser:FullName>
            <gser:Phone>+420000000000</gser:Phone>
          </gser:DeliveryContact>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser:ExportItems>
            <gser:ExportItem>
              <gser:CountItems>1</gser:CountItems>
              <gser:Description>Popis</gser:Description>
              <gser:Height>0.8</gser:Height>
              <gser:Length>1.2</gser:Length>
              <gser:Reference>Reference kusu</gser:Reference>
              <gser:SubItems>
                <gser:SubItem>
                  <gser:Type>FP</gser:Type>
                </gser:SubItem>
                <gser:SubItem>
                  <gser:Type>FP</gser:Type>
                </gser:SubItem>
              </gser:SubItems>
              <gser:Type>FP</gser:Type>
              <gser:Weight>24</gser:Weight>
              <gser:Width>1.3</gser:Width>
            </gser:ExportItem>
          </gser:ExportItems>
          <gser:ExportServices>
```



Geis CZ s.r.o.
Zemská 211/I
337 01 Ejovice
Česká republika

Telefon: +420 951 277 777

E-mail: info@geis.cz
Internet: www.geis-group.cz
IČO: 44567359
DIČ: CZ44567359

Bankovní spojení:
UniCredit Bank Czech Republic and Slovakia a.s.

Č.ú.: 1545471501/2700 (CZK)
CZK IBAN:
 CZ97 2700 0000 0015 4547 1501
Č.ú.: 1545471560/2700 (EUR)
EUR IBAN
 CZ56 2700 0000 0015 4547 1560
SWIFT: BACXCZPP

Společnost je zapsaná v obchodním rejstříku, vedeném Krajským soudem v Plzni, oddíl C, číslo vložky 14475.

Pracujeme na základě našich všeobecných obchodních podmínek - aktuální stav.

```
<gser:ExportService>
  <gser:Code>HDS</gser:Code>
  <gser:Parameter_1>+420000000000</gser:Parameter_1>
</gser:ExportService>
</gser:ExportServices>
<gser:Note>Note</gser:Note>
<gser:NoteDriver>Note driver</gser:NoteDriver>
<gser:PartnerNumber>0000001</gser:PartnerNumber>
<gser:PickUpDate>2021-03-15T00:00:00</gser:PickUpDate>
<gser:Reference>Reference</gser:Reference>
<gser:SenderContact>
  <gser:Email>email@geis.cz</gser:Email>
  <gser:FullName>Sender contact</gser:FullName>
  <gser:Phone>+420000000000</gser:Phone>
</gser:SenderContact>
</gser:RequestObject>
</tem:Request>
</tem:InsertExport>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <InsertExportResponse xmlns="http://tempuri.org/">
      <InsertExportResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:BurstId>test</a:BurstId>
          <a:MergedPackNumbers i:nil="true" xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <a:PackNumber>3115000102902</a:PackNumber>
        </a:ResponseObject>
        <a>Status>Inserted</a>Status>
      </InsertExportResult>
    </InsertExportResponse>
  </s:Body>
</s:Envelope>
```

InsertOrder

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:InsertOrder>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:BurstId>12349</gser:BurstId>
          <gser:DeliveryAddress>
            <gser:City>Test delivery city</gser:City>
            <gser:Country>CZ</gser:Country>
            <gser:Name>Test delivery name</gser:Name>
            <gser:Name2>Test delivery name 2</gser:Name2>
            <gser:Street>Test delivery street</gser:Street>
            <gser:ZipCode>18000</gser:ZipCode>
          </gser:DeliveryAddress>
          <gser:DeliveryContact>
            <gser:Email>email@geis.cz</gser:Email>
            <gser:FullName>Test delivery kontakt</gser:FullName>
            <gser:Phone>+420000000000</gser:Phone>
          </gser:DeliveryContact>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser:ExportItems>
            <gser:ExportItem>
              <gser:CountItems>1</gser:CountItems>
              <gser>Description>Description</gser>Description>
              <gser:Height>1.1</gser:Height>
              <gser:Length>1.2</gser:Length>
              <gser:Reference>987</gser:Reference>
              <gser:SubItems>
                <gser:SubItem>
                  <gser:Type>FP</gser:Type>
                </gser:SubItem>
                <gser:SubItem>
                  <gser:Type>FP</gser:Type>
                </gser:SubItem>
                <gser:SubItem>
                  <gser:Type>EP</gser:Type>
                </gser:SubItem>
              </gser:SubItems>
              <gser:Volume>1.87</gser:Volume>
              <gser:Weight>17</gser:Weight>
              <gser:Width>1.4</gser:Width>
            </gser:ExportItem>
          </gser:ExportItems>
          <gser:ExportServices>
            <gser:ExportService>
              <gser:Code>HDS</gser:Code>
              <gser:Parameter_1>+420000000000</gser:Parameter_1>
            </gser:ExportService>
          </gser:ExportServices>
          <gser>Note>Note</gser>Note>
        </gser:RequestObject>
      </tem:Request>
    </tem:InsertOrder>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<gser:NoteDriver>Note driver</gser:NoteDriver>
<gser:PartnerNumber>123321</gser:PartnerNumber>
<gser:PickUpDate>2021-03-16</gser:PickUpDate>
<gser:Reference>125</gser:Reference>
<gser:SenderAddress>
  <gser:City>Test sender city</gser:City>
  <gser:Country>CZ</gser:Country>
  <gser:Name>Test sender name</gser:Name>
  <gser:Name2>Test sender name 2</gser:Name2>
  <gser:Street>Test sender street</gser:Street>
  <gser:ZipCode>32600</gser:ZipCode>
</gser:SenderAddress>
<gser:SenderContact>
  <gser:Email>email@geis.cz</gser:Email>
  <gser:FullName>Test sender kontakt</gser:FullName>
  <gser:Phone>+420000000000</gser:Phone>
</gser:SenderContact>
</gser:RequestObject>
</tem:Request>
</tem:InsertOrder>
</soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <InsertOrderResponse xmlns="http://tempuri.org/">
      <InsertOrderResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:BurstId>12349</a:BurstId>
          <a:MergedPackNumbers i:nil="true" xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <a:PackNumber>3111001033838</a:PackNumber>
        </a:ResponseObject>
        <a>Status>Inserted</a>Status>
      </InsertOrderResult>
    </InsertOrderResponse>
  </s:Body>
</s:Envelope>
```

AssignRange

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:AssignRange>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:DistributionChannel>2</gser:DistributionChannel>
        </gser:RequestObject>
      </tem:Request>
    </tem:AssignRange>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <AssignRangeResponse xmlns="http://tempuri.org/">
      <AssignRangeResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage>Číselný rozsah přidělen.</a:ErrorMessage>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:RangeHigh>3110001155000</a:RangeHigh>
          <a:RangeLow>3110001150001</a:RangeLow>
        </a:ResponseObject>
        <a:Status>Processed</a:Status>
      </AssignRangeResult>
    </AssignRangeResponse>
  </s:Body>
</s:Envelope>
```

IsHealthy

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:IsHealthy>
    </tem:IsHealthy>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <IsHealthyResponse xmlns="http://tempuri.org/">
      <IsHealthyResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Created>2021-03-24T15:17:25.4505618+01:00</a:Created>
        <a:Error i:nil="true"/>
        <a:Message i:nil="true"/>
        <a:Status>HEALTHY</a:Status>
        <a:Version>1.0.0.0</a:Version>
      </IsHealthyResult>
    </IsHealthyResponse>
  </s:Body>
</s:Envelope>
```

GetLabel

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetLabel>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108009</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser:Format>1</gser:Format>
          <gser:Position>1</gser:Position>
          <gser:ShipmentNumbers>
            <gser:LabelItem>
              <gser:ShipmentNumber>3115000103320</gser:ShipmentNumber>
            </gser:LabelItem>
          </gser:ShipmentNumbers>
        </gser:RequestObject>
      </tem:Request>
    </tem:GetLabel>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetLabelResponse xmlns="http://tempuri.org/">
      <GetLabelResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request>
          <a:Header>
            <a:CustomerCode>22108084</a:CustomerCode>
            <a:Language>CZ</a:Language>
            <a>Password>*****</a>Password>
          </a:Header>
          <a:RequestObject>
            <a:DistributionChannel>2</a:DistributionChannel>
            <a:Format>1</a:Format>
            <a:Position>1</a:Position>
            <a:Resolution>0</a:Resolution>
            <a:ShipmentNumbers>
              <a:LabelItem>
                <a:ShipmentNumber>3115000103320</a:ShipmentNumber>
              </a:LabelItem>
            </a:ShipmentNumbers>
          </a:RequestObject>
        </a:Request>
      </GetLabelResult>
    </GetLabelResponse>
  </s:Body>
</s:Envelope>
```

```
</a:Request>
<a:ResponseObject>
  <a:LabelData>
    <a:LabelItemData>
      <a:Data> Zde budou data ve formátu Base64 </a:Data>
    </a:LabelItemData>
  </a:LabelData>
  <a:ShipmentNumbers i:nil="true"/>
</a:ResponseObject>
<a:Status>Processed</a:Status>
</GetLabelResult>
</GetLabelResponse>
</s:Body>
</s:Envelope>
```

ListOfShipments

[Go Back](#)

Request :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ListOfShipments>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108084</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:Created>2021-03-22</gser:Created>
          <gser:DateExpOrOrder>2021-03-22</gser:DateExpOrOrder>
        </gser:RequestObject>
      </tem:Request>
    </tem:ListOfShipments>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ListOfShipmentsResponse xmlns="http://tempuri.org/">
      <ListOfShipmentsResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:ShipmentResponse>
            <a:IsOrder>false</a:IsOrder>
            <a:ShipmentNumber>3115000103273</a:ShipmentNumber>
          </a:ShipmentResponse>
          <a:ShipmentResponse>
            <a:IsOrder>false</a:IsOrder>
            <a:ShipmentNumber>3115000103274</a:ShipmentNumber>
          </a:ShipmentResponse>
          <a:ShipmentResponse>
            <a:IsOrder>false</a:IsOrder>
            <a:ShipmentNumber>3115000103275</a:ShipmentNumber>
          </a:ShipmentResponse>
          <a:ShipmentResponse>
            <a:IsOrder>false</a:IsOrder>
            <a:ShipmentNumber>3115000103276</a:ShipmentNumber>
          </a:ShipmentResponse>
        </a:ResponseObject>
        <a>Status>Processed</a>Status>
      </ListOfShipmentsResult>
    </ListOfShipmentsResponse>
  </s:Body>
</s:Envelope>
```

ShipmentStatus

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:ShipmentStatus>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108260</gser:CustomerCode>
          <gser:Language>EN</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:ShipmentsNumbers>
            <gser:ShipmentStatusItem>
              <gser:ShipmentNumber>3115000104310</gser:ShipmentNumber>
            </gser:ShipmentStatusItem>
          </gser:ShipmentsNumbers>
        </gser:RequestObject>
      </tem:Request>
    </tem:ShipmentStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ShipmentStatusResponse xmlns="http://tempuri.org/">
      <ShipmentStatusResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:ShipmentStatusResponse>
            <a:ShipmentNumber>3115000104310</a:ShipmentNumber>
            <a:StatusCode>CDR</a:StatusCode>
            <a:StatusName>Partial delivery</a:StatusName>
          </a:ShipmentStatusResponse>
        </a:ResponseObject>
        <a:Status>Processed</a:Status>
      </ShipmentStatusResult>
    </ShipmentStatusResponse>
  </s:Body>
</s:Envelope>
```

GetPickupList

[Go Back](#)

Request :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetPickupList>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108084</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser:ExpeditionCreateDate>2021-03-18</gser:ExpeditionCreateDate>
          <gser:ShipmentsNumbers>
            <gser:PickupListShipmItem>
              <gser:ShipmentNumber>3115000103222</gser:ShipmentNumber>
            </gser:PickupListShipmItem>
          </gser:ShipmentsNumbers>
        </gser:RequestObject>
      </tem:Request>
    </tem:GetPickupList>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetPickupListResponse xmlns="http://tempuri.org/">
      <GetPickupListResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:PickupListData> Zde budou data ve formátu Base64 </a:PickupListData>
        </a:ResponseObject>
        <a>Status>Processed</a>Status>
      </GetPickupListResult>
    </GetPickupListResponse>
  </s:Body>
</s:Envelope>
```

DeleteShipment

[Go Back](#)

Request :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem>DeleteShipment>
      <tem>Request</tem>
      <gser>Header</gser>
      <gser>CustomerCode<22108084</gser>CustomerCode>
      <gser>Language<CZ</gser>Language>
      <gser>Password<*****</gser>Password>
    </gser>Header>
    <gser>RequestObject</gser>
    <gser>ShipmentsNumbers</gser>
    <gser>DeleteShipmentItem</gser>
    <gser>DistributionChannel<2</gser>DistributionChannel>
    <gser>ShipmentNumber<3115000103222</gser>ShipmentNumber>
    </gser>DeleteShipmentItem</gser>
    </gser>ShipmentsNumbers</gser>
    </gser>RequestObject</gser>
  </tem>Request</tem>
</tem>DeleteShipment</tem>
</soapenv>Body>
</soapenv>Envelope>
```

Response:

```
<s>Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s>Body>
    <DeleteShipmentResponse xmlns="http://tempuri.org/">
      <DeleteShipmentResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a>ErrorCode<0000</a>ErrorCode>
        <a>ErrorMessage</a>
        <a>Request i:nil="true"/>
        <a>ResponseObject</a>
        <a>ShipmentsNumbers</a>
        <a>DeleteShipmentItemInfo</a>
        <a>ErrorMessage i:nil="true"/>
        <a>IsStorno<true</a>IsStorno>
        <a>ShipmentNumber<3115000103222</a>ShipmentNumber>
      </a>DeleteShipmentItemInfo</a>
    </a>ShipmentsNumbers</a>
    </a>ResponseObject</a>
    <a>Status<Processed</a>Status>
  </DeleteShipmentResult>
</DeleteShipmentResponse>
</s>Body>
</s>Envelope>
```

RangeDetail

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:RangeDetail>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108084</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:DistributionChannel>2</gser:DistributionChannel>
        </gser:RequestObject>
      </tem:Request>
    </tem:RangeDetail>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <RangeDetailResponse xmlns="http://tempuri.org/">
      <RangeDetailResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:AssignAllowed>false</a:AssignAllowed>
          <a:AssignRangeLimit>5000</a:AssignRangeLimit>
          <a:RangeDetailItem>
            <a:RangeDetailResponseItem>
              <a:LastUsedNumber>3110001165003</a:LastUsedNumber>
              <a:RangeHigh>3110001170000</a:RangeHigh>
              <a:RangeLow>3110001165001</a:RangeLow>
              <a:RemainingNumbers>4997</a:RemainingNumbers>
            </a:RangeDetailResponseItem>
          </a:RangeDetailItem>
        </a:ResponseObject>
        <a>Status>Processed</a>Status>
      </RangeDetailResult>
    </RangeDetailResponse>
  </s:Body>
</s:Envelope>
```

PickupInfo

[Go Back](#)

Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/"
xmlns:gser="http://schemas.datacontract.org/2004/07/GService.Manager">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:PickupInfo>
      <tem:Request>
        <gser:Header>
          <gser:CustomerCode>22108084</gser:CustomerCode>
          <gser:Language>CZ</gser:Language>
          <gser:Password>*****</gser:Password>
        </gser:Header>
        <gser:RequestObject>
          <gser:Date>2021-03-25</gser:Date>
          <gser:DistributionChannel>2</gser:DistributionChannel>
          <gser:IsOrder>0</gser:IsOrder>
        </gser:RequestObject>
      </tem:Request>
    </tem:PickupInfo>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <PickupInfoResponse xmlns="http://tempuri.org/">
      <PickupInfoResult xmlns:a="http://schemas.datacontract.org/2004/07/GService.Manager"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ErrorCode>0000</a:ErrorCode>
        <a:ErrorMessage/>
        <a:Request i:nil="true"/>
        <a:ResponseObject>
          <a:Message i:nil="true"/>
          <a:Pickups>
            <a:PickUpItemInfo>
              <a:City>Test city</a:City>
              <a:CountryCode>CZ</a:CountryCode>
              <a:Date>2021-03-25T00:00:00Z</a:Date>
              <a:DistributionChannel>2</a:DistributionChannel>
              <a:IsOrder>false</a:IsOrder>
              <a:Name>Test customer</a:Name>
              <a:State>Potvrzená</a:State>
              <a:StatId>2</a:StatId>
              <a:Street>Test street 31</a:Street>
              <a:ZipCode>31200</a:ZipCode>
            </a:PickUpItemInfo>
          </a:Pickups>
        </a:ResponseObject>
        <a>Status>Processed</a>Status>
      </PickupInfoResult>
    </PickupInfoResponse>
  </s:Body>
</s:Envelope>
```


REST and SOAP Method Comparison

REST method	SOAP method
GET /healthcheck	IsHealthy
GET /api/customer/service	AddServiceList
—	ServiceList (no longer used – no REST equivalent)
GET /api/Status	StatusList
GET /api/PackingType	WrapList
GET /api/Routing/Delivery	DeliveryRouting
GET /api/Shipment/Detail	ShipmentDetail
GET /api/Routing/Table	RouteTable
POST /api/Pickup	CreatePickUp
GET /api/Pickup	PickupInfo
GET /api/PickupList	GetPickupList
POST /api/Shipment/Export	InsertExport
POST /api/Shipment/Order	InsertOrder
DELETE /api/Shipment	DeleteShipment
GET /api/Shipment	ListOfShipments
GET /api/Shipment/Status	ShipmentStatus
GET /api/Shipment/Label	GetLabel
POST /api/CustomRange	AssignRange
GET /api/CustomRange	RangeDetail

An examples of error codes

Code	Description
0055	Cash on delivery price value is not valid for the specified currency.
0094	Colli weight exceeded allowed value
0098	VS cash on delivery can only contain numbers.
1000	Access denied. Please check login.
2000	[#Name of parameter#] is required parameter!
2002	Value [#Value from list of values#] was not found!
2003	No data was found.
2006	Range of dates is wrong! Please fill right date.
2015	Pickup for actual day could be created only until [#TEXT#] hours.
2016	You filled unspecified service [#TEXT#].
2017	[#TEXT#] must be in international format (eg.: +420111222333). Allowed prefixes are [#TEXT#].
2018	[#TEXT#] must be in correct format (eg.: correct.mail@email.cz)!
2029	For address [#TEXT#] service COD to individual account is not allowed.
2064	Number of requests for method is exceeded. – Limit will restore in one hour.
2083	There is no pick-up of this type on the specified day.
2091	Service HDP cannot be set for selected sender's / recipient's country combination
2096	The shipment with same number already exist
3000	Error service GClientWS! [#Error message from the service GClientWS #]
3044	The row must contain a weight greater than 0.
3146	Recipient's zip code is not in a valid format.
3228	The line must contain at least one piece.
9999	Unexpected Error!