METHODS DESCRIPTION

API BDL

VERSION 1

Table of Contents

1. BASIC INFORMATION ABOUT THE SERVICE

LOCAL DATA BANK (POL.: BDL) is Poland's largest database of the economy, society and the environment. BDL offers more than 40 thousands statistical characteristics grouped thematically. In the API BDL service, data is made available through a programming interface.

The API BDL service is available at the address: https://bdl.stat.gov.pl/api/v1/

2. COMMUNICATION WITH THE SERVICE

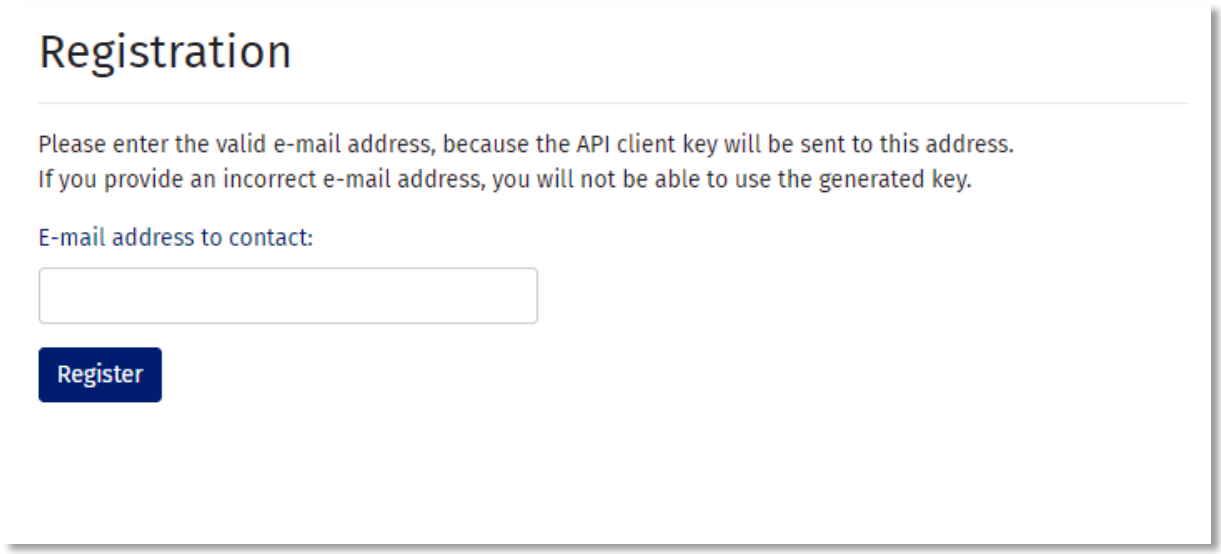
2.1 CLIENT AUTHENTICATION AND SESSION SUPPORT MECHANISM

The API BDL is publicly available to all interested parties, including anonymous users. For performance reasons, the number of user requests is limited.

In order to use the higher limits an API key is required. Users who need a key can register by providing us the email address to which the key will be sent.

2.2 REGISTRATION

To register, use the form:



The obtained API Key should be provided in the request as an HTTP header named X-ClientId, e.g.

```
GET https://bdl.stat.gov.pl/api/v1/data/by-variable/3643?format=xml&year=2000&year=2010 HTTP/1.1
Host: hostname
X-ClientId: 1A6B2AB1-0DF0-4DAC-74A3-07D7C07FC3BE
```

2.3 LIMITS

In the field of requests, the following user limits apply:

Category Period	Anonymous user	Registered user
1s	5	10
15m	100	500
12h	1 000	5 000
7d	10 000	50 000

2.4 RESOURCES

The API BDL service makes the resources available from the Local Data Bank via the API (type REST) using the parameterized "get" methods. Each defined resource, in addition to specific requests for data, supports a metadata request - a description of the content of the resource.

The following resources are defined and made available in the application:

Aggregates - data aggregation levels for which data are available, e.g. municipalities or urban-rural communes.

Attributes - descriptions of specific situations in the data that are associated with each numerical value, e.g. 'Value smaller than the adopted presentation format'.

Data - numerical data in the form of three [real number, attribute identifier, year identifier], in the form of a data set for a specific variable or set of data for one unit (more information below).

Levels - levels of data availability, corresponding to the level of territorial units, e.g. 2 - the voivedships level.

Measures - units of measurement appearing in the data, related to specific variables, e.g. 'thousand liters'.

Subjects - hierarchically related groups of variables according to the substantive scope, eg. 'Population', 'Population by age and gender groups' (more information below).

Units - hierarchically related list of territorial units (from Poland up to municipalities) and statistical localities.

Variables - these are multidimensional characteristics representing a specific phenomenon, with a defined validity in years and at a specific level of units, e.g. the number of employees for the ages of '20 -26 'and gender' men '.

Years - years of validity for which data may exist, e.g. '2003', '2004'.

Version - application version.

2.5 RESOURCES AND THEIR METHODS

All methods, except the parameters mentioned, have 4 additional:

Accept-Language enum (header)	Expected response content language - optional (if “lang” parameter is specified, the “Accept-Language” header will be ignored) Available values : pl, en
Accept enum (header)	Expected response content type - optional (if the “format” parameter is specified, the “Accept” header will be ignored) Available values : application/json, application/vnd.api+json, application/xml
If-None-Match string (header)	The header of conditional request If-None-Match (entity tag)/Conditional Requests header If-None-Match (entity tag)
If-Modified-Since string (header)	The header of conditional request If-Modified-Since/Conditional Requests header If-Modified-Since

2.5.1 AGGREGATES

Aggregates

Methods:

Type	Syntax	
GET	**Aggregates/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	Expected response content language - optional (if parameter specified, request header “Accept-Language” is ignored) *Available values* : pl, en
	format **enum** *(query)*	Expected response content type - optional (if parameter specified, request header “Accept” is ignored) *Available values* : json, xml
	Example	aggregates/metadata?lang=en&format=xml
GET	**Aggregates/{id}** Aggregation level with a given Id	
	Parameters	Description
	id * **integer($int32)** *(path)*	Aggregation level Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	aggregates/3?format=xml *Aggregate with an identifier Id=3*

GET	**Aggregates** List of aggregation levels	
	Parameters	Description
	sort **string** *(query)*	Expected order *Available values* : Id, -Id, Id,Name, Id,-Name, -Id,Name, -Id,-Name, Name, -Name, Name,Id, Name,-Id, -Name,Id, -Name,-Id **
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	aggregates?format=xml *List of all aggregates*

* required parameter ** list of all possible sorting combinations for this resource

2.5.2 ATTRIBUTES

Attributes

Methods:

Type	Syntax	
GET	**Attributes/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above
	Example	attributes/metadata?format=json
GET	**Attributes /{id}** Attribute with a given Id	
	Parameters	Description
	id * **integer($int32)** *(path)*	Attribute with a given Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	attributes/3?format=json *Aggregate with an identifier Id=K3*

GET	**Attributes** List of attributes	
	Parameters	Description
	sort **string** *(query)*	Expected order *Available values* : Display, -Display, Display,Id, Display,-Id, -Display,Id, -Display,-Id, Display,Id,Symbol, Display,Id,-Symbol, Display,-Id,Symbol, Display,-Id,-Symbol, -Display,Id,Symbol, -Display,Id,-Symbol, -Display,-Id,Symbol, -Display,-Id,-Symbol, Display,Symbol, Display,-Symbol, -Display,Symbol, -Display,-Symbol, Display,Symbol,Id, Display,Symbol,-Id, Display,-Symbol,Id, Display,-Symbol,-Id, -Display,Symbol,Id, -Display,Symbol,-Id, -Display,-Symbol,Id, -Display,-Symbol,-Id, Id, -Id, Id,Display, Id,-Display, -Id,Display, -Id,-Display, Id,Display,Symbol, Id,Display,-Symbol, Id,-Display,Symbol, Id,-Display,-Symbol, -Id,Display,Symbol, -Id,Display,-Symbol, -Id,-Display,Symbol, -Id,-Display,-Symbol, Id,Symbol, Id,-Symbol, -Id,Symbol, -Id,-Symbol, Id,Symbol,Display, Id,Symbol,-Display, Id,-Symbol,Display, Id,-Symbol,-Display, -Id,Symbol,Display, -Id,Symbol,-Display, -Id,-Symbol,Display, -Id,-Symbol,-Display, Symbol, -Symbol, Symbol,Display, Symbol,-Display, -Symbol,Display, -Symbol,-Display, Symbol,Display,Id, Symbol,Display,-Id, Symbol,-Display,Id, Symbol,-Display,-Id, -Symbol,Display,Id, -Symbol,Display,-Id, -Symbol,-Display,Id, -Symbol,-Display,-Id, Symbol,Id, Symbol,-Id, -Symbol,Id, -Symbol,-Id, Symbol,Id,Display, Symbol,Id,-Display, Symbol,-Id,Display, Symbol,-Id,-Display, -Symbol,Id,Display, -Symbol,Id,-Display, -Symbol,-Id,Display, -Symbol,-Id,-Display
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	attributes?format=xml *List of attributes*

2.5.3 DATA

Data

Methods:

Type	Syntax	
GET	**Data/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, xml
	Example	data/metadata?format=json
GET	**/Data/ByVariable/{var-Id}** Data for one variable	
	Parameters	Description
	varId * **integer($int32)** *(path)*	Variable Id
	year **array[integer]** *(query)*	Years list
	unitParentId **string** *(query)*	The identifier of the superior territorial unit

	unitLevel **integer($int32)** *(query)*	Territorial units level
	aggregateId **integer($int32)** *(query)*	Aggregation level Id
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	data/ByVariable/3643?format=json&year=2000&year=2010 *Data for the variable with Id=3643 for the years 2000,2010*
GET	**Data/ByUnit/{unit-Id}** Data for one territorial unit	
	Parameters	Description
	unitId * **string** *(path)*	Territorial unit id

	varId **array[integer]** *(query)*	Variable Id list
	year **array[integer]** *(query)*	Years list
	aggregateId **integer($int32)** *(query)*	Aggregation level Id
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	data/ByVariable/65809?format=json&year=2004&year=2005&year=2006 *Data for the variable with Id=65809 for the years 2004-2006*
GET	**Data/Data/Localities/ByVariable/{var-Id}** Data for statistical localities for one variable	

	Parameters	Description
	varId * **integer($int32)** *(path)*	Variable Id
	year **array[integer]** *(query)*	Years list
	unitParentId **string** *(query*	The identifier of the superior territorial unit
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	data/Localities/ByVariable/420?format=xml&unitParentId=011200000000 *Data for the variable with identifier Id=420, for the locality from the Małopolska voivodeship.*

GET	**Data/Localities/ByUnit/{unit-Id}** Data for one statistical locality	
	Parameters	Description
	unitId * **string** *(path)*	Statistical locality Id
	varId **array[integer]** *(query)*	Variable Id list
	year **array[integer]** *(query)*	Years list
	aggregateId **integer($int32)** *(query)*	Aggregation level Id
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	lang **enum** *(query)*	as above
	format	as above

	enum *(query)*	*Available values* : json, jsonapi, xml
	Example	data/ByUnit/012400000000?format=xml&varId=3643&year=2004&year=2005&year=2006 *Data for a territorial unit with Id=012400000000, for the variable with Id=3643, for the years 2004-2006*

2.5.4 LEVELS

Levels

Methods:

Type	Syntax	
GET	**Levels/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, xml
	Example	levels/metadata?format=json
GET	**Levels/{id}** The level of data availability with the given Id	
	Parameters	Description
	id * **integer($int32)** *(path)*	Level Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	/levels/3?format=xml *The level of data availability with identifier Id=3*

GET	**Levels** List of accessibility levels	
	Parameters	Description
	sort **string** *(query)*	Expected order *Available values* : Id, -Id, Id,Name, Id,-Name, -Id,Name, -Id,-Name, Name, -Name, Name,Id, Name,-Id, -Name,Id, -Name,-Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	Levels?sort=id *List of all accessibility levels - sorting by column Id*

2.5.5 MEASURES

Measures

Methods:

Type	Syntax	
GET	**Measures/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, xml
	Example	measures/metadata?format=json
GET	**Measures/{id}** Unit of measure with given Id	
	Parameters	Description
	id * **integer($int32)** *(path)*	Unit of measure Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	/measures/3?format=xml *Unit of measure with identifier Id=3*

GET	**Measures** List of units of measure	
	Parameters	Description
	sort **string** *(query)*	Expected order *Available values* : Id, -Id, Id,Name, Id,-Name, -Id,Name, -Id,-Name, Name, -Name, Name,Id, Name,-Id, -Name,Id, -Name,-Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	Measures?sort=id *List of all units* of measure - *sorting by column Id*

2.5.6 SUBJECTS

Subjects

Methods:

Type	Syntax	
GET	**Subjects/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, xml
	Example	Subjects/metadata
GET	**Subjects** List of subjects	
	Parameters	Description
	parentId **string** *(query)*	Parent subject Id
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	sort	Expected order *Available values* : Id, -Id, Id,Name, Id,-Name, -Id,Name, -Id,-Name, Name, -Name,

	string *(query)*	Name,Id, Name,-Id, -Name,Id, -Name,-Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	subjects?format=xml *List of all top-level subjects*
GET	**Subjects/search** Search for subjects by name	
	Parameters	Description
	name **string** *(query)*	Fragment of the subject name
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	sort **string** *(query)*	Expected order *Available values* : Id, -Id, Id,Name, Id,-Name, -Id,Name, -Id,-Name, Name, -Name, Name,Id, Name,-Id, -Name,Id, -Name,-Id
	lang **enum**	as above

	(query)	
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	subjects/search?name=samochody *Searching subjects with the word 'samochody' in the name*
GET	**Subjects/{id}** Subject with a given Id	
	Parameters	Description
	name **string** *(query)*	Subject Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	*subjects/K3* *Subject with identifier Id=K3*

2.5.7 UNITS

Units

Methods:

Type	Syntax	
GET	**Units/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, xml
	Example	**units** /metadata
GET	**Units/{id}** A territorial unit (up to the municipality level) with a given Id	
	Parameters	Description
	id * **string** *(path)*	Territorial unit Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	**units/010000000000?format=xml** *Territorial unit with identifier Id=010000000000*

GET	**Units** List of territorial units	
	Parameters	Description
	parentId **string** *(query)*	Parent territorial unit Id
	level **array[integer]** *(query)*	Level 0 – Poland, 1 – Macroregion, 2 – Voivodeship, 3 – Region, 4 – Subregion, 5 – County, 6 - Municipality
	page **integer($int32)** *(query)*	as above
	pageSize **integer($int32)** *(query)*	as above
	sort **string** *(query)*	Expected order Available values : Id, -Id, Id,Name, Id,-Name, -Id,Name, -Id,-Name, Name, -Name, Name,Id, Name,-Id, -Name,Id, -Name,-Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	**units**
GET	**Units/search** Search for units by name	

	Parameters	Description
	name **string** *(query)*	A fragment of the name of the unit
	level **array[integer]** *(query)*	as above
	year **array[integer]** *(query)*	Years list
	kind **string** *(query)*	Type of unit

	sort **string** *(query)*	Expected order as above
	page **integer($int32)** *(query)*	as above
	pageSize **integer($int32)** *(query)*	as above
	lang **enum** *(query)*	as above
	format **enum**	as above

	(query)	*Available values* : json, jsonapi, xml
	Example	units/search?level=1&year=2010&page=0&pageSize=10
GET	**Units/localities** List of statistical localities	
	Parameters	Description
	parentId * **string** *(query)*	Parent territorial unit Id
	page **integer($int32)** *(query)*	as above
	pageSize **integer($int32)** *(query)*	as above
	sort **string** *(query)*	as above
	lang **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	format **enum** *(query)*	as above
	Example	units/localities?parentId=011212001011&page=0&pageSize=10

GET	Units/localities/{id} Statistical locality with a given Id	
	Parameters	Description
	id * **string** *(path)*	Statistical locality Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above
	Example	units/localities/011212001011-0981682

GET	Units/localities/search Searching for statistical localities by name	
	Parameters	Description
	name **string** *(query)*	A fragment of the name of the unit
	year **array[integer]** *(query)*	Years list
	sort **string** *(query)*	as above
	page **integer**($int32) *(query)*	as above
	pageSize	as above

	integer($int32) *(query)*	
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	units/localities/search?year=2007&sort=name&page=0&pageSize=10

2.5.8 VARIABLES

Variables

Methods:

Type	Syntax	
GET	**Variables/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	**Variables** /metadata
GET	**Variables/{id}** Variable with given Id	
	Parameters	Description
	id * **integer($int32)** *(path)*	Unit of measurement Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	**variables/1688** Variable with identifier Id=1688

GET	**Variables** List of variables	
	Parameters	Description
	subjectId **string** *(query)*	Subject Id
	level **integer($int32)** *(query)*	Level
	year **array[integer]** *(query)*	Years list
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	sort **string** *(query)*	Expected order *Available values* : Id, -Id, Id,SubjectId, Id,-SubjectId, -Id,SubjectId, -Id,-SubjectId, SubjectId, -SubjectId, SubjectId,Id, SubjectId,-Id, -SubjectId,Id, -SubjectId,-Id
	lang **enum** *(query)*	as above
	format **enum**	as above *Available values* : json, jsonapi, xml

	(query)	
	Example	**/variables?subjectId=P2137&year=2005&year=2006&format=xml** List of variables for the subject with Id=P2137, occurring in the years 2005,2006
GET	**Variables/search** Searching for variables by conditions	
	Parameters	Description
	subjectId **string** *(query)*	Parent subject Id
	name **string** *(query)*	Text in N1…N5 fields
	level **integer($int32)** *(query)*	Level
	year **array[integer]** *(query)*	Years list
	page **integer($int32)** *(query)*	Page index
	pageSize **integer($int32)** *(query)*	Page size
	sort **string**	Expected order *Available values* : Id, -Id, Id,SubjectId, Id,-SubjectId, -Id,SubjectId, -Id,-SubjectId,

	(query)	SubjectId, -SubjectId, SubjectId,Id, SubjectId,-Id, -SubjectId,Id, -SubjectId,-Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	**/variables/search?name=samochody** Searching for variables with the word 'samochody' in the names of dimensions

2.5.9 YEARS

Years

Methods:

Type	Syntax	
GET	**Years/metadata** Metadata	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, xml
	Example	**Years** /metadata
GET	**Years /{id}** Variable with given Id	
	Parameters	Description
	id * **string** *(path)*	Year Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	**Years /2003** Year with identifier Id=2003
GET	**Years**	

	A list of years of data validity	
	Parameters	Description
	subjectId **string** *(query)*	Subject Id
	sort **string** *(query)*	Expected order *Available values* : Id, -Id
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above *Available values* : json, jsonapi, xml
	Example	**/ Years?subjectId=P2137&year=2005&year=2006&format=xml** List of variables for the subject with Id=P2137, occurring in the years 2005,2006

2.5.10 VERSION

Version

Methods:

Type	Syntax	
GET	**Version** Application version	
	Parameters	Description
	lang **enum** *(query)*	as above
	format **enum** *(query)*	as above
	Example	**Version**

3 REGULATIONS FOR USING THE SERVICE

3.1. DEFINITIONS

1. **The Service Provider** - Statistics Poland, based at al. Niepodległości 208, 00-925 Warsaw.

2. **Recipient** - each User (anonymous or not) using the resources of the Local Data Bank (BDL) via the API programming interface to the Local Data Bank.

3. **Service - Data sharing service from the Local Data Bank resources (API BDL Service)** - software, web service that allows you to download data from the Local Data Bank resources directly from the level of external IT systems or via the tools provided on the Statistics Poland websites.

4. **User's key** - a 36 string of letters and numbers that allows the Recipient to use the Service with increased queries limits.

3.2. RULES OF USING THE SERVICE

1. Using the API BDL service to share data from the resources of the Local Data Bank, hereinafter referred to as the Service, is related to the acceptance of these regulations (along with subsequent changes).

2. Access to the Service is free of charge.

3. The methods provided by the Service may be implemented in any ICT systems used or produced by the Recipient.

4. Actions that could lead to overloading the system or its malfunction are prohibited.

5. Please send technical notifications to the address bdl.kontakt@stat.gov.pl (giving in the subject: API-Pomoc).

3.3. RESPONSIBILITY

1. The Service Provider reserves the right to introduce modifications to the system, carry out maintenance works and other activities that may cause temporary unavailability or difficulties in using the Service.

2. The Service Provider is not liable for the lost benefits incurred by the Recipient or third parties using the Service.

3. The Service Provider is not liable for the lack of access to services resulting from the use of the API BDL Service not in accordance with the technical specification.