



Manual for REST API restPrsInfo

(technical upgrade of legacy SOAP web service wsPrsInfo)

Last update: January 2026

1. Description

New **restPrsInfo** REST API is intended for the direct integration of users' software solutions with the Business Register of Slovenia (for simplicity, Slovenian abbreviation PRS will be used hereafter, which refers to Slovenian naming *Poslovni Register Slovenije*). The service enables searching for and retrieving official data on business entities from the PRS in minimal, narrow, extended and protected data sets.

Users' software solutions can search for business entities using various criteria, as described below. For the business entities found, they obtain the registration number (i.e. *matična številka* in Slovene), which is the unique identifier of an individual business entity. Using this identifier, the corresponding data can then be retrieved from the PRS database.

For updating data that has already been retrieved, the API provides an additional method that enables obtaining a list of registration numbers (identifiers) of those business entities for which any change was made during the specified period—i.e. a change that resulted in any data of the business entity being modified (not necessarily the data actually retrieved by the user based on the selected data set). Based on the obtained list of registration numbers, the user's software solution can re-retrieve the individual, updated, and (potentially) changed data from the PRS.

2. Technical specifications

Developers can test **restPrsInfo** API via the **Swagger UI** in a web browser at the following URL:

Test environment: <https://www.ajpes.si/restPrsInfo/swagger/index.html>

Production environment: <https://www.ajpes.si/restPrsInfo/swagger/index.html>

The **restPrsInfo** service contains two GET methods—one for testing connectivity (Ping!) and one for retrieving PRS code lists—and five POST methods: one for checking the user's available funds (which also returns the user's contact details) and four methods that return business data from the PRS.

By clicking on an individual method and selecting "Try it out" (top right in the respective tab), the user is presented with a form in which they enter the parameters, the *Content-Type* header, and the input message (*Request Body*). The request body can be either in JSON or in XML format, depending on the selected *Content-Type* header as follows:

- `application/json` – the user must provide the input message in JSON format
- `application/xml` – the input message must be in XML format

The output data format can also be JSON (default) or XML. This is controlled by two boolean URL parameters: *xml* and *raw*. If *xml=true*, the data is returned in XML format, whereas *xml=false* or NULL returns the data in the default JSON format. Furthermore, if *raw = true*, the "business logic" data is returned as a string, while *raw=false* or NULL returns the data in the default structured format (i.e. data model) as defined for the individual API method. The **restPrsInfo** API contains four POST methods that return content-equivalent data from the PRS as its predecessor, the SOAP service **wsPrsInfo**, but they are named differently, as follows from Table 1:

restPrsInfo		wsPrsInfo
<code>~/ping</code>	(GET)	Ping
<code>~/sifranti</code>	(GET)	-
<code>~/checkSredstva</code>	(POST)	-
<code>~/get</code>	(POST)	PrsDataGet
<code>~/find</code>	(POST)	PrsDataFind
<code>~/modified</code>	(POST)	PrsDataMod
<code>~/dissolved</code>	(POST)	PrsDataDissolved

Table 1: The names (and types) of the controllers for invoking the methods of the new **restPrsInfo** service that are functionally equivalent to the methods of the legacy SOAP web service **wsPrsInfo**.

3. Validation of input parameters

The data that the user sends in the input message (request body) is first validated. If validation fails, the user receives an appropriate notification with a more or less detailed description of the error or the reason for the validation failure. Based on this notification, the user can correct the input message themselves. The output message of all four POST methods has the same structure and contains the following fields:

- **status** – a unique 4-digit error or service response code (data type: *integer*)
- **message** – a notification describing the error or another response (data type: *string*)
- **payload** – a field containing data from the PRS (data type: *string* or an appropriate data model, depending on the value of boolean parameter *raw*)

If validation of the input data fails or any other system error occurs, the value of the payload field is always NULL. The message field contains information about the type and details of the error, while the status field contains a unique 4-digit integer code that encodes each expected and unexpected error. Status codes for expected errors start with 4 (4xxx), while status codes for unexpected, i.e. system errors (exceptions) start with 5 (5xxx).

If there are no errors when calling a specific method, two scenarios are possible. If the API finds the requested data in the database, the status is 2000 and the message is "OK", and the payload field is populated with the requested data that the user expects. The second scenario is when no error occurs, but the requested data is not found in the database. In this case, the payload field is also NULL, status = 2001, and the message is either "Ni zadetkov!", which means »No hits!«, or »V Poslovnem registru Slovenije ni subjekta z matično številko ***matična številka***.«, which means "There is no entity with registration number ***registration number*** in the Business Register of Slovenia.". The third option for status=2001 is »Subjekt z matično številko ***matična številka*** je izbrisan iz Poslovnega registra Slovenije. Za izpis podatkov izbrisanega subjekta v vhodnem sporočilu izberite vrednost polja 'izbrisan': true.«, which means "The entity with registration number ***registration number*** has been deleted from the Business Register of Slovenia. To display data for a deleted entity, set the field value 'izbrisan' to true in the input message." The payload field, which contains the data that the user wants to display in their GUI, is therefore populated only when the status is 2000; in all other cases, the value of the payload field is NULL!

Example of an output message when validation of user identification fails (e.g. incorrect password):

```
{
  "payload": null,
  "status": 4002,
  "message": "Napaka pri avtentikaciji uporabnika. Podrobnosti: Napačno uporabniško ime ali geslo. "
}
```

where 'Napaka pri avtentikaciji uporabnika. Podrobnosti: Napačno uporabniško ime ali geslo.' means 'User authentication error. Details: Incorrect username or password.'

Corresponding screen-shot of Swagger UI in web browser:



Picture 1: An example of a restPrsInfo REST API response (GET method) – User authentication error. The input message contained an incorrect password '*****'. Since there is no business data, the payload field is NULL.

4. Validation of output data

All data retrieved from the PRS database is, before being returned to the user by the service, first validated against the appropriate **XSD schema**, depending on the method user called. The names, links to the XSD schemas used, and their versions are listed in the Table 2.

XSD schemas:

Schema name	URL of XSD schema	Namespace
PrsInfo	http://www.ajpes.si/xml_sheme/prs/prs_info_v1_9.xsd	http://www.ajpes.si/xml_sheme/prs-info-20080729
(included)	http://www.ajpes.si/xml_sheme/prs/prs_elements_v1_6.xsd	http://www.ajpes.si/xml_sheme/prs-info-20080729
PrsDataFind	http://www.ajpes.si/xml_sheme/prs/PrsDataFind_v1_1.xsd	http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind
PrsDataMod	http://www.ajpes.si/xml_sheme/prs/PrsDataMod_v1_0.xsd	http://www.ajpes.si/xml_sheme/prs-info/PrsDataMod

Table 2: A list of XSD schemas that were used to validate the output data and to generate data models (classes) for returning output data in deserialized form.

5. Methods

Individual methods of the restPrsInfo REST API are called using the corresponding controller in the URL, which follows the **base URL**:

TEST: <https://www.ajpes.si/restPrsInfo>
 PRODUCTION: <https://www.ajpes.si/restPrsInfo>

a) ping

URL: ~/ping
 Type: GET

By calling the GET method ping, the user can check the connectivity to the service. Upon a successful call, it returns the user's IP address and device name, and if a digital certificate is used (required for returning personal data from the PRS), it also returns the serial number and issuer of the digital certificate. The ping method also informs the user which client data will be recorded in the AJPEs database for each PRS query.

The following URL can be entered in a web browser without user identification or an input message to check the connectivity:

TEST: <https://www.ajpes.si/restPrsInfo/ping>
PRODUCTION: <https://www.ajpes.si/restPrsInfo/ping>

The remaining four POST methods, which return data from the PRS, all use the 'ident' field in the input message for user authentication (username and password) and authorization (user PRS permission level). ALL fields in ident must always be filled:

- uporabnik – the username registered on the portal www.ajpes.si
- geslo – the password corresponding to the username
- shema – the code of the PRS schema that the user has registered with AJ PES

**Special case – protected data (not available on public servers):*

To view protected (personal) data, the user must also have a digital certificate registered with AJ PES. The service retrieves the digital certificate directly from the caller. The caller must have the private key of the digital certificate installed on the device from which they are calling restPrsInfo.

In the public test environment, the service can only be tested with the user **wsPrsInfoTest**. AJ PES provides the password for the test user to developers by email. Please, send request to cpu.it@ajpes.si. In the publicly accessible test environment, the restPrsInfo service returns data in a limited, predefined set of entities from the production database. Details on obtaining the list of allowed business entities for testing are described at method c) find.

b) sifranti

URL: ~/sifranti
Type: GET

By calling the GET method sifranti, the user retrieves all code lists of the Business Register of Slovenia. The following URL can be entered in a web browser without user identification or an input message:

TEST: <https://www.ajpes.si/restPrsInfo/sifranti>
PRODUCTION: <https://www.ajpes.si/restPrsInfo/sifranti>

c) checkSredstva

URL: ~/checkSredstva
Type: POST

The POST method checkSredstva is used to check the user's available funds (it also returns contact information).

Field name	Data type	Description	Obligatory?
uporabnik	string	the username registered on the portal www.ajpes.si	YES
geslo	string	the password corresponding to the username	YES

Table 3: Fields of the input message for calling the **checkSredstva** method.

An example of input JSON message:

```
{
  "uporabnik": "***username***",
  "geslo": "***password***"
}
```

The service returns a list of all registered PRS schemas, and for each of them, a list of all purchased packages, including those already used. The data for each package includes:

- **datumNakupa** – purchase date
- **veljaDo** – expiration date
- **kupljenihEnot** – number of units purchased
- **porabljenihEnot** – number of units used

If the user has no PRS schema registered in their profile (for example: a call with the test user in the production environment, or vice versa), the JSON response is as follows:

```
{
  "uporabnik": {
    "ime": null,
    "priimek": null,
    "podjetje": null,
    "email": null,
    "telefon": null,
    "nivoPrsInfo": null,
    "digitalnoPotrdilo": {
      "izdajatelj": null,
      "serijskaStevilka": null
    },
    "registriraneSheme": []
  },
  "status": 2000,
  "message": "OK"
}
```

If the user has one or more PRS schemas registered, the response is as follows, an example JSON structure:

```
{
  "uporabnik": {
    "ime": "Janez",
    "priimek": "Novak",
    "podjetje": "Podjetje d.o.o.",
    "email": "janez.novak@podjetje.si",
    "telefon": "",
    "nivoPrsInfo": "",
    "digitalnoPotrdilo": {
      "izdajatelj": "",
      "serijskaStevilka": ""
    },
    "registriraneSheme": [
      {
        "shema": "PRS_MN_E",
        "opis": "wsPrsInfo enota - minimalni nabor-do 500 enot",

```

```

"paketi": [
  {
    "datumNakupa": "2023-03-30",
    "veljaDo": "2024-03-29",
    "kupljenihEnot": 1000,
    "porabljenihEnot": 1000
  },
  {
    "datumNakupa": "2024-03-30",
    "veljaDo": "2025-03-30",
    "kupljenihEnot": 500,
    "porabljenihEnot": 202
  },
  {
    "datumNakupa": "2025-03-31",
    "veljaDo": "2026-03-31",
    "kupljenihEnot": 500,
    "porabljenihEnot": 59
  }
]
},
{
  "shema": "PRS_MN_P",
  "opis": "wsPrsInfo poizvedba - minimalni nabor",
  "paketi": [
    {
      "datumNakupa": "2019-07-02",
      "veljaDo": "",
      "kupljenihEnot": 500,
      "porabljenihEnot": 500
    }
  ]
},
{
  "shema": "PRS_SN_P",
  "opis": "wsPrsInfo poizvedba - širši nabor",
  "paketi": [
    {
      "datumNakupa": "2023-03-14",
      "veljaDo": "",
      "kupljenihEnot": 500,
      "porabljenihEnot": 500
    },
    {
      "datumNakupa": "2024-03-05",
      "veljaDo": "",
      "kupljenihEnot": 500,
      "porabljenihEnot": 221
    }
  ]
},
{
  "shema": "PRS_ON_E",
  "opis": "wsPrsInfo enota - ožji nabor",
  "paketi": [

```

```

{
  "datumNakupa": "2024-03-25",
  "veljaDo": "2025-03-25",
  "kupljenihEnot": 500,
  "porabljenihEnot": 25
},
{
  "datumNakupa": "2025-03-31",
  "veljaDo": "2026-03-31",
  "kupljenihEnot": 500,
  "porabljenihEnot": 2
}
]
},
{
  "shema": "PRS_SN_E",
  "opis": "wsPrsInfo enota - širši nabor",
  "paketi": [
    {
      "datumNakupa": "2017-12-05",
      "veljaDo": "2018-12-05",
      "kupljenihEnot": 500,
      "porabljenihEnot": 500
    },
    {
      "datumNakupa": "2024-12-17",
      "veljaDo": "2025-12-17",
      "kupljenihEnot": 500,
      "porabljenihEnot": 404
    }
  ]
}
]
},
"status": 2000,
"message": "OK"
}

```

d) find

URL: ~/find

Type: POST

The POST method find allows searching for data on business entities using the following search criteria.

Field name	Data type	Description	Obligatory?
uporabnik	string	the username registered on the portal www.ajpes.si	YES
geslo	string	the password corresponding to the username	YES
shema	string	shema – the code of the PRS schema that the user has registered with AJPES	YES
naziv	string	full or partial name of business entity	NO
maticna	string	registration number of the entity (10-digit number)	NO
davcna	string	tax number of the entity (8-digit number)	NO
naslov	string	address of the entity's registered office (street name, road, etc.)	NO

hisnaStevilka	string	house number	NO
naselje	string	settlement	NO
obcina	string	municipality	NO
posta	string	postal code	NO
dejavnost	string	main activity	NO
sektor	string	sector (SKIS)	NO
oblika	string	organization form code	NO
najdiPodruznice	boolean	true - return parent (head) entity information false - return subsidiary entities (branches)* data * When searching for subsidiaries, the registration number of the parent entity must be provided. If the parameter is not specified (NULL), the default value is false.	NO
maxRecords	ushort	Defines the number of returned records. Unlimited in the production environment and limited to 3 records in the test environment. If the parameter is not specified (NULL), the default value is 3.	NO

Table 4: Fields of the input message for calling the **find** method.

One or more parameters can be provided. If the registration number or tax number of an entity is provided, only that parameter is used for the search. If multiple parameters are provided, a logical AND operation is applied between them. The returned results match all specified parameters.

The search is case-insensitive and does not support special wildcard characters. In the test environment, the find method returns data for up to three business entities from one of the smaller Slovenian municipalities that match the search criteria. The list of allowed registration numbers for testing can be obtained by providing the method with a registration number not on the test list, for example:

```
"maticna": "1234567890"
```

The list of allowed tax numbers intended for testing can be obtained by providing the method with a tax number not on the test list, for example:

```
"davcna": "12345678"
```

As mentioned, the restPrsInfo REST API can be called with a request body in either JSON or XML format. Table 5 presents two input messages for calling the find method that are functionally identical, but in different formats.

header Content-Type: application/json	header Content-Type: application/xml
<pre>{ "ident": { "uporabnik": "wsprsinfotest", "geslo": "***password***", "shema": "PRS_SN_E" }, "maticna": "", "davcna": "", "naziv": "", "obcina": "", "hisnaStevilka": "" }</pre>	<pre><Find> <ident> <uporabnik>wsprsinfotest</uporabnik> <geslo>***password***</geslo> <shema>PRS_SN_E</shema> </ident> <maticna></maticna> <davcna></davcna> <naziv></naziv> <obcina></obcina></pre>

<pre>"naslov": "", "naselje": "", "posta": "", "dejavnost": "", "sektor": "", "oblika": "", "najdiPodruznice": false, "maxRecords": 10 }</pre>	<pre><hisnaStevilka></hisnaStevilka> <naslov></naslov> <naselje></naselje> <posta></posta> <dejavnost></dejavnost> <sektor></sektor> <oblika></oblika> < najdiPodruznice >false</ najdiPodruznice > <maxRecords>10</maxRecords> </Find></pre>
--	--

Table 5: An example of equal input message in JSON format (application/json) and in XML format (application/xml).

An example of **default response**, format JSON (parameter *xm*==NULL), model PrsData (parameter *raw*==NULL):

```
{
  "payload": {
    "raw": null,
    "formatted": {
      "prsData": [
        {
          "popolno_ime": "Mia Erbus, računalniško programiranje, s.p.",
          "kratko_ime": "Mia Erbus s.p.",
          "maticna": "9089357000",
          "ulica": "Placar 042 A",
          "posta": "2250 Ptuj",
          "st_pod_enot": null,
          "podenota": "0",
          "zbrisano": null,
          "datum_izbrisa": "0001-01-01T00:00:00",
          "datum_izbrisaSpecified": false
        },
        {
          "popolno_ime": "MM ELEKTROMONTAŽA IN AVTOMATIKA, montaža industrijskih strojev in naprav,
            d.o.o.",
          "kratko_ime": "MM ELEKTROMONTAŽA IN AVTOMATIKA d.o.o.",
          "maticna": "8896941000",
          "ulica": "Placar 031",
          "posta": "2253 Destrnik",
          "st_pod_enot": null,
          "podenota": "0",
          "zbrisano": null,
          "datum_izbrisa": "0001-01-01T00:00:00",
          "datum_izbrisaSpecified": false
        },
        {
          "popolno_ime": "Nataša Čeh s.p., posredništvo pri prodaji",
          "kratko_ime": "Nataša Čeh s.p.",
          "maticna": "6453147000",
          "ulica": "Jiršovci 015 B",
          "posta": "2253 Destrnik",
          "st_pod_enot": null,

```

```

        "podenota": "0",
        "zbrisanost": null,
        "datum_izbrisa": "0001-01-01T00:00:00",
        "datum_izbrisaSpecified": false
    }
]
}
},
"status": 2000,
"message": "OK"
}

```

The **default response** means that the URL (<https://www.ajpes.si/restPrsInf/find>) does not include the *xml* and *raw* parameters, so their values are both false. This means the API returns the response in JSON format (appsettings/json), and the business logic data is populated in the payload/formatted field, following the data model defined by the XSD schema PrsInfo (see XSD schemas in Table 2, Chapter 4). If the *xml=true* parameter is added to the URL, the API returns the exact same data in XML format (appsettings/xml), **even if the input message is in JSON format**.

URL: <https://www.ajpes.si/restPrsInf/find?xml=true>

```

<PrsInfoResponseOfPrsFind xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <status>2000</status>
  <message>OK</message>
  <payload>
    <formatted>
      <PrsData xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind">
        <Popolno_ime>Mia Erbus, računalniško programiranje, s.p.</Popolno_ime>
        <Kratko_ime>Mia Erbus s.p.</Kratko_ime>
        <Maticna>9089357000</Maticna>
        <Ulica>Placar 042 A</Ulica>
        <Posta>2250 Ptuj</Posta>
        <Podenota>0</Podenota>
      </PrsData>
      <PrsData xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind">
        <Popolno_ime>MM ELEKTROMONTAŽA IN AVTOMATIKA, montaža industrijskih strojev in naprav,
          d.o.o.</Popolno_ime>
        <Kratko_ime>MM ELEKTROMONTAŽA IN AVTOMATIKA d.o.o.</Kratko_ime>
        <Maticna>8896941000</Maticna>
        <Ulica>Placar 031</Ulica>
        <Posta>2253 Destrnik</Posta>
        <Podenota>0</Podenota>
      </PrsData>
      <PrsData xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind">
        <Popolno_ime>Nataša Čeh s.p., posredništvo pri prodaji</Popolno_ime>
        <Kratko_ime>Nataša Čeh s.p.</Kratko_ime>
        <Maticna>6453147000</Maticna>
        <Ulica>Jiršovci 015 B</Ulica>
        <Posta>2253 Destrnik</Posta>
        <Podenota>0</Podenota>
      </PrsData>
    </formatted>
  </payload>
</PrsInfoResponseOfPrsFind>

```

The business logic data is populated in the <formatted> sub-element of the <payload> element, using the same data model as in the previous example, which was generated from the PrsInfo XSD schema (see Table 2 – XSD schemas in Chapter 4).

IMPORTANT NOTE FOR DEVELOPERS when replacing existing wsPrsInfo with restPrsInfo:

The legacy SOAP web service wsPrsInfo returns business logic data in raw XML string format. To make the transition from wsPrsInfo to restPrsInfo for developers as smooth as possible, new restPrsInfo service does the same, if `xml=true&raw=true` parameters are used in URL.

URL: <https://www.ajpes.si/restPrsInf/find?xml=true&raw=true>

```
<PrsInfoResponseOfPrsFind xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <status>2000</status>
  <message>OK</message>
  <payload>
<raw>&lt;PrsFind xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;&lt;PrsData&gt;&lt;Popolno_ime&gt;Mia Erbus,
računalniško programiranje, s.p.&lt;/Popolno_ime&gt;&lt;Kratko_ime&gt;Mia Erbus
s.p.&lt;/Kratko_ime&gt;&lt;Maticna&gt;9089357000&lt;/Maticna&gt;&lt;Ulica&gt;Placar 042
A&lt;/Ulica&gt;&lt;Posta&gt;2250
Ptuj&lt;/Posta&gt;&lt;Podenota&gt;0&lt;/Podenota&gt;&lt;PrsData&gt;&lt;PrsData&gt;&lt;Popolno_ime&gt;M
M ELEKTROMONTAŽA IN AVTOMATIKA, montaža industrijskih strojev in naprav,
d.o.o.&lt;/Popolno_ime&gt;&lt;Kratko_ime&gt;MM ELEKTROMONTAŽA IN AVTOMATIKA
d.o.o.&lt;/Kratko_ime&gt;&lt;Maticna&gt;8896941000&lt;/Maticna&gt;&lt;Ulica&gt;Placar
031&lt;/Ulica&gt;&lt;Posta&gt;2253
Destrnik&lt;/Posta&gt;&lt;Podenota&gt;0&lt;/Podenota&gt;&lt;PrsData&gt;&lt;PrsData&gt;&lt;Popolno_ime&gt;
;Nataša Čeh s.p., posredništvo pri prodaji&lt;/Popolno_ime&gt;&lt;Kratko_ime&gt;Nataša Čeh
s.p.&lt;/Kratko_ime&gt;&lt;Maticna&gt;6453147000&lt;/Maticna&gt;&lt;Ulica&gt;Jiršovci 015
B&lt;/Ulica&gt;&lt;Posta&gt;2253
Destrnik&lt;/Posta&gt;&lt;Podenota&gt;0&lt;/Podenota&gt;&lt;PrsData&gt;&lt;PrsFind&gt;</raw>
  </payload>
</PrsInfoResponseOfPrsFind>
```

Raw (serialized) **XML string** s poslovnimi podatki se napolni v podelement <raw> elementa <payload>. Podobno lahko dobimo tudi raw (serialized) **JSON string** s poslovnimi podatki, če v URL uporabimo samo `raw=true`, brez xml parametra oziroma vrednostjo `xml=false`:

The raw (serialized) XML string containing the business data is populated in the <raw> sub-element of the <payload> element. Similarly, a raw (serialized) JSON string containing the business data can be obtained by using `raw=true` in the URL without the xml parameter, or with `xml=false`.

URL: <https://www.ajpes.si/restPrsInfo/find?raw=true>

```
{
  "payload": {
    "raw": "{\\"PrsData\\":{\\"Popolno_ime\\":\\"Mia Erbus, ra\u010dunalni\u0161ko programiranje,
s.p.\\",\\"Kratko_ime\\":\\"Mia Erbus s.p.\\",\\"Maticna\\":\\"9089357000\\",\\"Ulica\\":\\"Placar 042 A\\",\\"Posta\\":\\"2250
```

```

Ptuj\", \"Podenota\": \"0\", \"Datum_izbrisa\": \"0001-01-
01T00:00:00\", \"Datum_izbrisaSpecified\": false}, { \"Popolno_ime\": \"MM ELEKTROMONTAŽA IN AVTOMATIKA,
montaža industrijskih strojev in naprav, d.o.o.\", \"Kratko_ime\": \"MM ELEKTROMONTAŽA IN AVTOMATIKA
d.o.o.\", \"Maticna\": \"8896941000\", \"Ulica\": \"Placar 031\", \"Posta\": \"2253
Destrikn\", \"Podenota\": \"0\", \"Datum_izbrisa\": \"0001-01-
01T00:00:00\", \"Datum_izbrisaSpecified\": false}, { \"Popolno_ime\": \"Nataša Čeh s.p., posredništvo pri
prodaji\", \"Kratko_ime\": \"Nataša Čeh s.p.\", \"Maticna\": \"6453147000\", \"Ulica\": \"Jiršovci 015
B\", \"Posta\": \"2253 Destrikn\", \"Podenota\": \"0\", \"Datum_izbrisa\": \"0001-01-
01T00:00:00\", \"Datum_izbrisaSpecified\": false}}\",
  \"formatted\": null
},
  \"status\": 2000,
  \"message\": \"OK\"
}

```

All of the above-mentioned rules also apply to the remaining three POST methods that return data from the Business Register of Slovenia. A special case is the search for registration numbers of subsidiary companies or branches, which requires that, in addition to the registration number of the parent business entity (head office), the *najdiPodruznice = true* parameter is also provided in the input message (Body) of the find method.

An example of such request:

header Content-Type: application/json	header Content-Type: application/xml
<pre> { "ident": { "uporabnik": "wsprsinfortest", "geslo": "***password***", "shema": "PRS_SN_E" }, "maticna": "5048664000", "davcna": "", "naziv": "", "obcina": "", "hisnaStevilka": "", "naslov": "", "naselje": "", "posta": "", "dejavnost": "", "sektor": "", "oblika": "", "najdiPodruznice": true, "maxRecords": 10 } </pre>	<pre> <Find> <ident> <uporabnik>wsprsinfortest</uporabnik> <geslo>***password***</geslo> <shema>PRS_SN_E</shema> </ident> <maticna>5048664000</maticna> <davcna></davcna> <naziv></naziv> <obcina></obcina> <hisnaStevilka></hisnaStevilka> <naslov></naslov> <naselje></naselje> <posta></posta> <dejavnost></dejavnost> <sektor></sektor> <oblika></oblika> <najdiPodruznice>true</ najdiPodruznice > <maxRecords>10</maxRecords> </Find> </pre>

Table 6: An example of an input message for the find method for searching branches of an entity, in JSON and XML format.

A response example (JSON, parameter *xml=false* or NULL):

```
{
  "payload": {
    "raw": null,
    "formatted": {
      "prsData": [
        {
          "popolno_ime": "FRUCTAL Živilska industrija d.o.o.",
          "kratko_ime": "FRUCTAL d.o.o.",
          "maticna": "5048664000",
          "ulica": "Tovarniška cesta 007",
          "posta": "5270 Ajdovščina",
          "st_pod_enot": null,
          "podenota": "000",
          "zbrisano": null,
          "datum_izbrisa": "0001-01-01T00:00:00",
          "datum_izbrisaSpecified": false
        },
        {
          "popolno_ime": "FRUCTAL Živilska industrija d.o.o., Proizvodni obrat Duplica pri Kamniku",
          "kratko_ime": "FRUCTAL d.o.o., Proizvodni obrat Duplica pri Kamniku",
          "maticna": "5048664004",
          "ulica": "Korenova cesta 007",
          "posta": "1241 Kamnik",
          "st_pod_enot": null,
          "podenota": "004",
          "zbrisano": null,
          "datum_izbrisa": "0001-01-01T00:00:00",
          "datum_izbrisaSpecified": false
        },
        {
          "popolno_ime": "FRUCTAL Živilska industrija d.o.o., MALOPRODAJA",
          "kratko_ime": "FRUCTAL d.o.o., MALOPRODAJA",
          "maticna": "5048664026",
          "ulica": "Tovarniška cesta 007",
          "posta": "5270 Ajdovščina",
          "st_pod_enot": null,
          "podenota": "026",
          "zbrisano": null,
          "datum_izbrisa": "0001-01-01T00:00:00",
          "datum_izbrisaSpecified": false
        },
        {
          "popolno_ime": "FRUCTAL Živilska industrija d.o.o., Maloprodaja Kamnik",
          "kratko_ime": "FRUCTAL d.o.o., Maloprodaja Kamnik",
          "maticna": "5048664028",
          "ulica": "Korenova cesta 007",
          "posta": "1241 Kamnik",
          "st_pod_enot": null,
          "podenota": "028",
          "zbrisano": null,
          "datum_izbrisa": "0001-01-01T00:00:00",
          "datum_izbrisaSpecified": false
        }
      ]
    }
  }
}
```

```

    ]
  }
},
"status": 2000,
"message": "OK"
}

```

A response example (XML, parameter *xml=true*):

```

<PrsInfoResponseOfPrsFind xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <status>2000</status>
  <message>OK</message>
  <payload>
    <formatted>
      <PrsData xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind">
        <Popolno_ime>FRUCTAL Živilska industrija d.o.o.</Popolno_ime>
        <Kratko_ime>FRUCTAL d.o.o.</Kratko_ime>
        <Maticna>5048664000</Maticna>
        <Ulica>Tovarniška cesta 007</Ulica>
        <Posta>5270 Ajdovščina</Posta>
        <Podenota>000</Podenota>
      </PrsData>
      <PrsData xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind">
        <Popolno_ime>FRUCTAL Živilska industrija d.o.o., Proizvodni obrat Duplica pri Kamniku</Popolno_ime>
        <Kratko_ime>FRUCTAL d.o.o., Proizvodni obrat Duplica pri Kamniku</Kratko_ime>
        <Maticna>5048664004</Maticna>
        <Ulica>Korenova cesta 007</Ulica>
        <Posta>1241 Kamnik</Posta>
        <Podenota>004</Podenota>
      </PrsData>
      <PrsData xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind">
        <Popolno_ime>FRUCTAL Živilska industrija d.o.o., MALOPRODAJA</Popolno_ime>
        <Kratko_ime>FRUCTAL d.o.o., MALOPRODAJA</Kratko_ime>
        <Maticna>5048664026</Maticna>
        <Ulica>Tovarniška cesta 007</Ulica>
        <Posta>5270 Ajdovščina</Posta>
        <Podenota>026</Podenota>
      </PrsData>
      <PrsData xmlns="http://www.ajpes.si/xml_sheme/prs-info/PrsDataFind">
        <Popolno_ime>FRUCTAL Živilska industrija d.o.o., Maloprodaja Kamnik</Popolno_ime>
        <Kratko_ime>FRUCTAL d.o.o., Maloprodaja Kamnik</Kratko_ime>
        <Maticna>5048664028</Maticna>
        <Ulica>Korenova cesta 007</Ulica>
        <Posta>1241 Kamnik</Posta>
        <Podenota>028</Podenota>
      </PrsData>
    </formatted>
  </payload>
</PrsInfoResponseOfPrsFind>

```

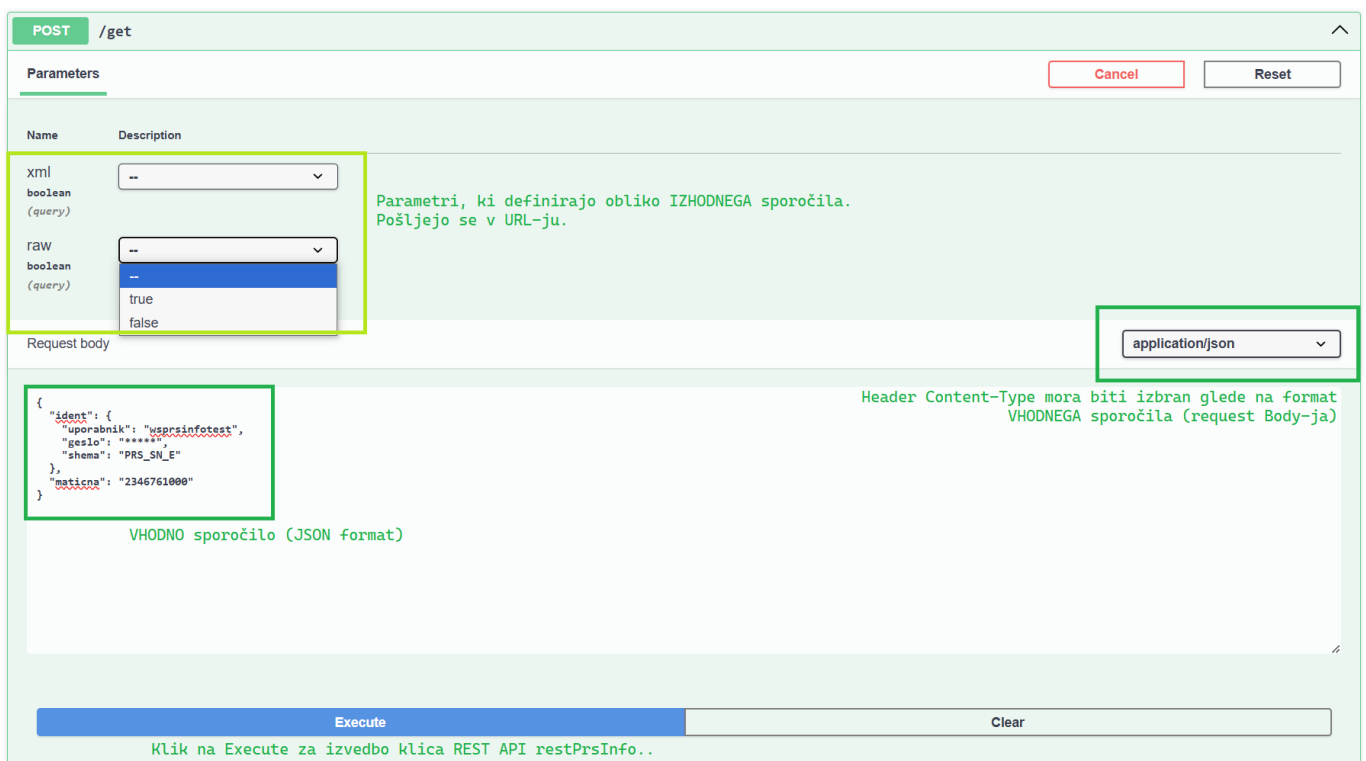
e) get

URL: ~/get
Type: POST

Field name	Data type	Description	Obligatory?
*uporabnik	string	the username registered on the portal www.ajpes.si	YES
geslo	string	the password corresponding to the username	YES
shema	string	shema – the code of the PRS schema that the user has registered with AJPES	YES
maticna	string	registration number of the entity (10-digit number)	YES
izbrisan	boolean	true – returns data of all, including erased entities false – returns data of active entities only If NULL, default value is false.	NO

Table 7: Fields of the input message for calling the **get** method.

Picture 2 explains the use of Swagger UI within web browser to call get method of restPrsInfo REST API.



Picture 2: An example of calling the **get** method from the **Swagger UI** in a web browser.

In the input message, within the 'ident' field, the user enters the test username **wsprsinfotest** in the 'uporabnik' field. The password for the test user, which you receive by email (send request to cpu.it@ajpes.si), is entered in the 'geslo' field. The 'shema' field contains the PRS data code, for example PRS_SN_E. The 'maticna' field is used to enter the registration number of the business entity being searched for. An example of a search by registration number of the Chamber of Commerce and Industry of Slovenia (Gospodarska zbornica Slovenije):

```

{
  "ident": {
    "uporabnik": "wsprsinfortest",
    "geslo": "***password***",
    "shema": "PRS_SN_E"
  },
  "maticna": "5021979000"
}

```

The method returns data about the business entity with the specified registration number if the entity is present in the collection of active business entities. If the business entity has been deleted from the collection of active business entities, the service informs the user that the entity with this registration number has been deleted. If you nevertheless want to retrieve data about a deleted entity from the PRS, the input message must include the field **izbrisan = true**. Example:

```

{
  "ident": {
    "uporabnik": "wsprsinfortest",
    "geslo": "***password***",
    "shema": "PRS_SN_E"
  },
  "maticna": "***registration number of deleted entity***",
  "izbrisan": true
}

```

The list of allowed registration numbers for testing the get method can be obtained by providing the method with a registration number that is not on the test list, for example:

```
"maticna": "1234567890"
```

The list of test registration numbers for the get method is the same as the list for the find method.

f) modified

URL: ~/modified

Type: POST

Field name	Data type	Description	Obligatory?
*uporabnik	string	the username registered on the portal www.ajpes.si	YES
geslo	string	the password corresponding to the username	YES
shema	string	shema – the code of the PRS schema that the user has registered with AJPES	YES
datumOd	string	The date (and time, optional*) from which the user wants to retrieve changes, i.e. <i>dateFrom</i> . Format must be: yyyy.MM.dd HH:mm:ss <i>*If the time is not specified, the default value 00:00:00 is used.</i>	YES
datumDo	string	The date (and time, optional*) up to which the user wants to retrieve changes, i.e. <i>dateTo</i> . Format must be: yyyy.MM.dd HH:mm:ss This parameter is optional. If it is not provided, the service returns changes up to the current date and	NO

		time. <i>*If the time is not specified, the default value 23:59:59 is used.</i>	
vseSpremembe	boolean	true – returns list of ALL modified entities, false – returns only the entities that have changed from the user’s package. When the parameter value is true and the user does not want to specify the end date (<i>datumDo</i>), datumDo must be NULL or an empty string. If the parameter value is true, the method returns data for a <u>maximum period of one month</u> . If the parameter is not specified (NULL), the default value is false.	NO

Table 8: Fields of the input message for calling the **modified** method.

Method **modified** returns a list of registration numbers of all business entities for which any data changes occurred in the PRS during the specified period. The set of registration numbers is generated programmatically from the list of registration numbers that the caller’s software has retrieved up to the time of the call from the PRS database via the get method within the purchased package (this does, however, not apply when the **vseSpremembe = true** parameter is used). An entity is included in the set even if a data element changes within the complete data set of the business entity that the user does not receive as part of their data package (e.g. personal data). When **vseSpremembe = true**, the method returns all business entities for which any change has occurred, regardless of whether the user has ever queried that entity using the get method or not. The only limitation in this case is that the service returns the list of all changed entities for a maximum period of one month.

This method is intended for all users who use the service with data refresh enabled (PRS_MN_E, PRS_ON_E, PRS_SN_E).

An example of request:

```
{
  "ident": {
    "uporabnik": "wsprsinfortest",
    "geslo": "***password***",
    "shema": "PRS_SN_E"
  },
  "datumOd": "2024.12.04 20:24:20",
  "datumDo": "",
  "vseSpremembe": false
}
```

Corresponding response (JSON, parameter *xml=false* or NULL):

```
{
  "payload": {
    "raw": null,
    "formatted": {
      "ps": [
        {
          "zb": "N",
          "dv": "2024-12-04T20:24:20",
          "mat": 5042437000
        }
      ]
    }
  }
}
```

```

    ]
  }
},
"status": 2000,
"message": "OK"
}

```

Corresponding response (XML, parameter *xml=true*):

```

<PrsInfoResponseOfPrsDataMod xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <status>2000</status>
  <message>OK</message>
  <payload>
    <formatted>
      <PS zb="N" dv="2024-12-04T20:24:20" mat="5042437000" xmlns="http://www.ajpes.si/xml_sheme/prs-
        info/PrsDataMod" />
    </formatted>
  </payload>
</PrsInfoResponseOfPrsDataMod>

```

Attribute code list in <PS> element:

mat = registration number of the entity

dv = date when the last change was entered into the PRS

ds = date of change corresponds to date when the entity was deleted. This attribute exists for deleted entities only.

zb = value 'N' means the entity is not deleted, whereas 'D' means the entity is deleted.

For more precise definition of the query time interval, an exact time can be specified in addition to the date. It is possible to specify only the hour (HH, 24-hour format); in this case, minutes and seconds (mm:ss) are set to 00:00. If hours and minutes (HH:mm) are specified, the seconds are set to 00. The 'datumOd' and 'datumDo' fields may use different DateTime formats within the same input message. See the example where 'datumOd' uses the format yyyy.MM.dd HH:mm:ss, while 'datumDo' uses the format yyyy.MM.dd HH. Time in addition to the date is not mandatory; the date however **is** mandatory. If the time is not provided in the 'datumOd' field, it is set to 00:00:00; if the time is not provided in the 'datumDo' field, it is set to 23:59:59. In this way, the 'datumOd' and 'datumDo' fields may have the same date, and the method **modified** will return the changes for that day, from 00:00:00 to 23:59:59.

A request example, where time is included beside date:

```

{
  "ident": {
    "uporabnik": "wsprsinfotest",
    "geslo": "***password***",
    "shema": "PRS_SN_E"
  },
  "datumOd": "2024.11.12 22:15:01",
  "datumDo": "2024.11.19 03",
  "vseSpremembe": false
}

```

g) dissolved

URL: ~/dissolved

Type: POST

The **dissolved** method returns a list of business entities from the PRS that were dissolved during a specified period. The period is defined by the dates in the 'datumOd' (*dateFrom*) and 'datumDo' (*dateTo*) fields. The period must not exceed seven days. In the test environment, the dissolved method returns data for a maximum of three dissolved business entities.

Field name	Data type	Description	Obligatory?
*uporabnik	string	the username registered on the portal www.ajpes.si	YES
geslo	string	the password corresponding to the username	YES
shema	string	shema – the code of the PRS schema that the user has registered with AJPES	YES
datumOd	string	The date (and time, optional*) from which the user wants to retrieve changes, i.e. <i>dateFrom</i> . Format must be: yyyy.MM.dd HH:mm:ss <i>*If the time is not specified, the default value 00:00:00 is used.</i>	YES
datumDo	string	The date (and time, optional*) up to which the user wants to retrieve changes, i.e. <i>dateTo</i> . Format must be: yyyy.MM.dd HH:mm:ss This parameter is optional. If it is not provided, the service returns changes up to the current date and time. <i>*If the time is not specified, the default value 23:59:59 is used.</i> The time frame for the deletion date of business entities can be a maximum of 7 days .	NO

Table 9: Fields of the input message for calling the **dissolved** method. The time frame for the deletion date of business entities can be a **maximum of 7 days**.

Request example (JSON):

```
{
  "ident": {
    "uporabnik": "wsprsinfortest",
    "geslo": "***password***",
    "shema": "PRS_SN_E"
  },
  "datumOd": "2024.11.12 06:15",
  "datumDo": "2024.11.19 21:00:17"
}
```

Response example (JSON):

```
{
  "payload": {
    "raw": null,
    "formatted": {
      "prsData": [
        {
```

```

    "popolno_ime": "Sabina Bizjak - sobodajalka",
    "kratko_ime": "/",
    "maticna": "2824108000",
    "ulica": "Jesenice, Jesenice 011 C",
    "posta": "8261 Jesenice na Dolenjskem",
    "st_pod_enot": null,
    "podenota": null,
    "zbrisano": "D",
    "datum_izbrisa": "2024-11-12T00:00:00",
    "datum_izbrisaSpecified": true
  },
  {
    "popolno_ime": "Sabina Bizjak - sobodajalka, APARTMENT ZALA",
    "kratko_ime": "/",
    "maticna": "2824108001",
    "ulica": "Savsko naselje 036 B",
    "posta": "4280 Kranjska Gora",
    "st_pod_enot": null,
    "podenota": null,
    "zbrisano": "D",
    "datum_izbrisa": "2024-11-12T00:00:00",
    "datum_izbrisaSpecified": true
  },
  {
    "popolno_ime": "WEB TRADE trgovina in storitve d.o.o. - v stečaju",
    "kratko_ime": "WEB TRADE d.o.o. - v stečaju",
    "maticna": "1934244000",
    "ulica": "Koper, Verdijeva ulica 001",
    "posta": "6000 Koper - Capodistria",
    "st_pod_enot": null,
    "podenota": null,
    "zbrisano": "D",
    "datum_izbrisa": "2024-11-12T00:00:00",
    "datum_izbrisaSpecified": true
  }
]
}
},
"status": 2000,
"message": "OK"
}

```

Since the **dissolved** method returns data on deleted business entities, the value of the 'zbrisano' attribute in the response is always "D" – YES. The validation of dates and times in the 'datumOd' and 'datumDo' fields is the same as in the modified method, where a detailed description can be found (see Chapter 4.f).

6. Codes of PRS schemas

Code	Data level	Request type	Notes
PRS_MN_E	Minimal	Unit	500, 1000, 2000 or 5000 units
PRS_MN_P	Minimal	Request	Up to 500 requests
PRS_ON_E	Narrow	Unit	500, 1000, 2000 or 5000 units
PRS_ON_P	Narrow	Request	Up to 500 requests
PRS_SN_E	Extended	Unit	500, 1000, 2000 or 5000 units
PRS_SN_P	Extended	Request	Up to 500 requests

E – unit (related to registration number): under this scheme, the user’s account is deducted by one unit for each NEW registration number that has not yet been queried. For the same business entity, the user can query without limits; only one unit is deducted at the first query.

P – query: under this scheme, the user’s account is deducted by one unit for EACH query, even if it is for the same registration number.