

Navodila za uporabo servisa restRzpplInfo

za razvijalce in uporabnike programske opreme

AJPES, Služba za informacijsko tehnologijo
oktober 2026

REST API restRzppInfo je tehnična nadgradnja starega SOAP spletnega servisa wsRzppInfo. Poslovna logika se ne spreminja. Dodana je metoda checkSredstva, s katero uporabnik lahko preverja razpoložljiva sredstva za pridobivanje podatkov iz Registra zastavnih pravic premičnin (RZPP). Tako kot pri obstoječem servisu, se tudi tu enote odštevajo za vsak vpogled v register z NOVO VIN številko. Če je uporabnik z neko VIN številko že poizvedoval, mu vse nadaljnje poizvedbe s to VIN številko ne bremenijo računa. Za pomoč in dodatna pojasnila, ki jih ne najdete v teh navodilih se obrnite na naš Center za pomoč uporabnikom cpu.it@ajpes.si.

URL do Swagger vmesnika (testno okolje):

<https://www.ajpes.si/restRzppInfo/swagger/index.html>

URL do Swagger vmesnika (produksijsko okolje):

<https://www.ajpes.si/restRzppInfo/swagger/index.html>

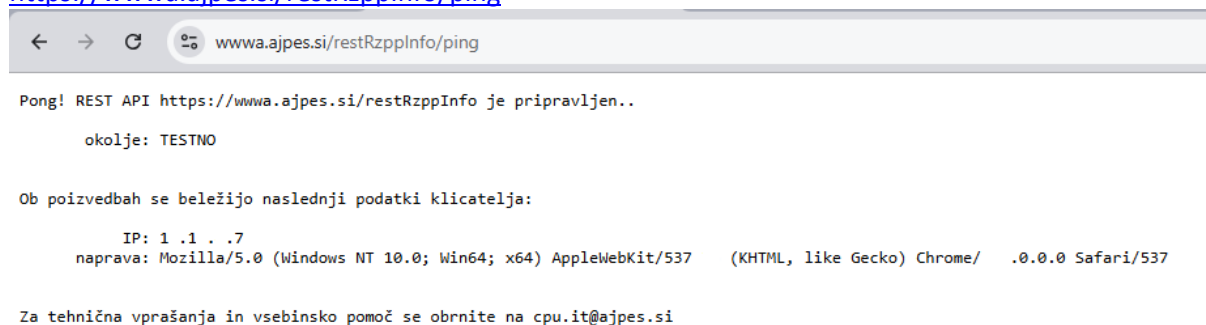
Spletni servis ima 3 metode, in sicer **ping** (GET), **checkSredstva** (POST) in **get** (POST):



Slika 1: Pogled na Swagger UI vmesnik (v testnem okolju).

Z metodo **ping** preverjamo **povezljivost** do servisa. Ker gre za GET metodo, lahko to storimo tudi v spletnem brskalniku z vpisom URL-ja

<https://www.ajpes.si/restRzppInfo/ping>



Slika 2: Klic metode ping s spletnim brskalnikom Chrome.

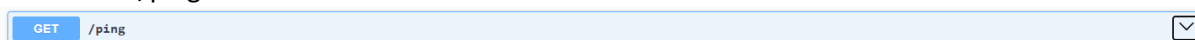
Izhodno sporočilo ping metode vsebuje sledeče podatke:

- Okolje: testno ali produkcijsko
- IP naslov klicatelja
- Naprava klicatelja: Postman, Chrome, ..
- Elektronski naslov AJPEŠ Službe za pomoč uporabnikom cpu.it@ajpes.si

Pozor, IP naslovi klicateljev se poleg ostalih parametrov beležijo v dnevnik klicev!

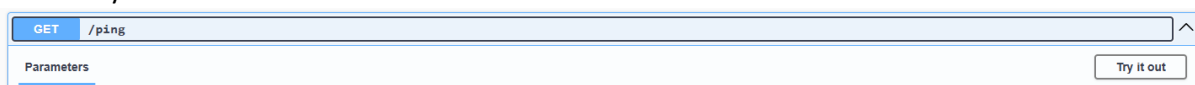
Metodo *ping* lahko kličemo tudi preko UI vmesnika Swagger. Postopek:

1. Klik na GET /ping



GET /ping

2. Klik na Try it out

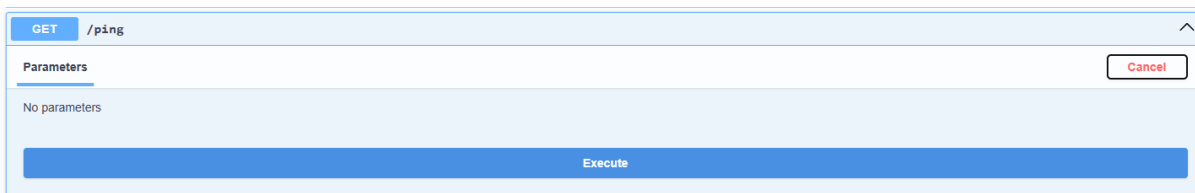


GET /ping

Parameters

Try it out

3. Klik na Execute



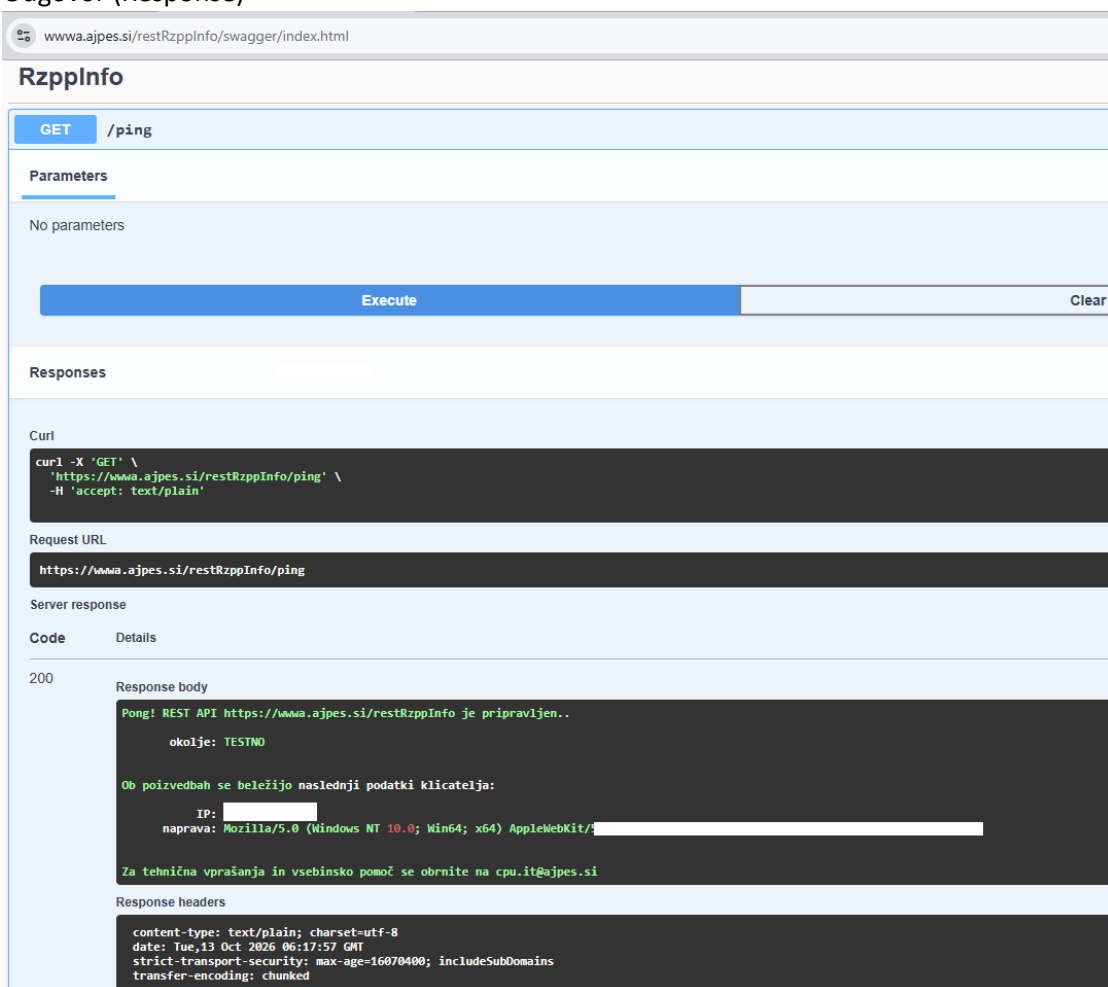
GET /ping

Parameters

No parameters

Execute

4. Odgovor (Response)



www.ajpes.si/restRzppInfo/swagger/index.html

RzppInfo

GET /ping

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://www.ajpes.si/restRzppInfo/ping' \
  -H 'accept: text/plain'
```

Request URL

```
https://www.ajpes.si/restRzppInfo/ping
```

Server response

Code	Details
200	Response body <pre>Pong! REST API https://www.ajpes.si/restRzppInfo je pripravljen..</pre> okolje: TESTNO Ob poizvedbah se beležijo naslednji podatki klicatelja: IP: [redacted] naprava: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/ Za tehnična vprašanja in vsebinsko pomoč se obrnite na cpu.it@ajpes.si Response headers <pre>content-type: text/plain; charset=utf-8 date: Tue, 13 Oct 2026 06:17:57 GMT strict-transport-security: max-age=16070400; includeSubDomains transfer-encoding: chunked</pre>

Slika 3: Postopek klicanja metode *ping* z vmesnikom Swagger UI.

Z metodo **checkSredstva** preverjamo **razpoložljivost sredstev** uporabnika, tako da kličemo URL <https://www.ajpes.si/restRzppInfo/checkSredstva> v Body request-a pa navedemo uporabniško ime in geslo, ki ju imamo registrirana na portalu Ajpes (velja za produkcijsko okolje). Za testno okolje uporabimo naslednji JSON Body

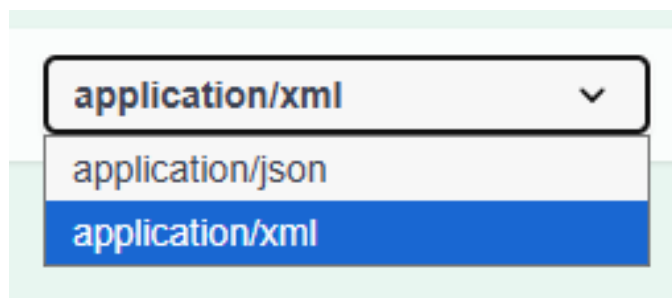
Request:

```
{
  "uporabnik": "wsRzppInfoTest",
  "geslo": "wsrzppinfo"
}
```

oziroma XML Body Request:

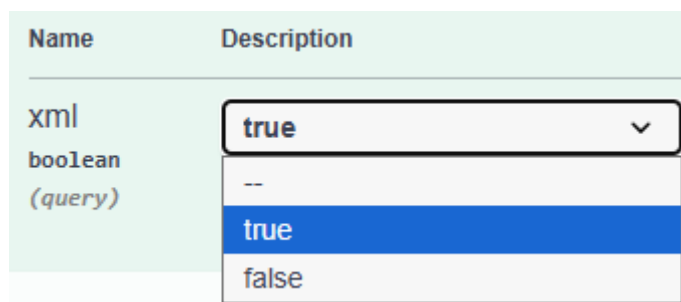
```
<Login>
  <uporabnik>wsRzppInfoTest</uporabnik>
  <geslo>wsrzppinfo</geslo>
</Login>
```

V Swagger UI header Content-Type Requesta med JSON in XML preklapljam z:



Slika 4: Izbira headerja Content-Type vhodnega sporočila v Swagger UI vmesniku.

Tudi Odgovor (response) je lahko bodisi v JSON bodisi v XML obliki, odvisno od vrednosti boolean parametra *xml* v URL naslovu, kjer *xml=False* ali NULL vrne JSON response, medtem ko *xml=True* vrne XML response. V Swagger UI vmesniku to storimo z



Slika 5: Izbira Content-Type Response-a v Swagger UI vmesniku. Tu gre za parameter v URL.

Content-Type odgovora sicer določimo v URL naslovu, kjer osnovni vrstici dodamo parameter *xml*. Če želimo odgovor v XML obliki, izberemo *?xml=true*. Takole potem izgleda celoten URL:

<https://www.ajpes.si/restRzppInfo/checkSredstva?xml=true>

Pripadajoč odgovor (Response):

```
<RzppInfo_InterOfUporabnik xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <status>2000</status>
  <message>OK</message>
  <timestamp>2026-10-13 10:34:42</timestamp>
  <payload>
    <ime>RzppInfoTest</ime>
    <priimek>Rzpp info test</priimek>
    <podjetje>testiranje servisa wsRzppInfo</podjetje>
    <email></email>
    <telefon />
    <nivoPrsInfo />
    <digitalnoPotrdilo>
      <izdajatelj />
      <serijskaStevilka />
    </digitalnoPotrdilo>
    <registriraneSheme>
      <Shema>
        <shema>RZPP_MON_P</shema>
        <opis>wsRzppInfo poizvedba, motorna vozila</opis>
        <paketi>
          <Paket>
            <datumNakupa>2015-03-03</datumNakupa>
            <veljaDo>2099-01-01</veljaDo>
            <kupljenihEnot>50000</kupljenihEnot>
            <porabljenihEnot>429</porabljenihEnot>
          </Paket>
        </paketi>
      </Shema>
    </registriraneSheme>
  </payload>
</RzppInfo_InterOfUporabnik>
```

Z metodo [get](#) pridobimo željene podatke iz Registra zastavnih pravic premičnin - RZPP. Primer JSON request-a za vozilo z VIN številko UU15SDAG453175072.

```
{
  "prijava": {
    "uporabnik": "wsRzppInfoTest",
    "geslo": "wsrzppinfo"
  },
  "vin": "UU15SDAG453175072"
}

{
  "prijava": {
    "uporabnik": "wsRzppInfoTest",
    "geslo": "wsrzppinfo"
  },
  "vin": "UU15SDAG453175072"
}
```

Poleg uporabniškega imena in gesla servis kliče tudi z **VIN številko vozila**, za katero želimo pridobiti podatke. V testnem okolju so za testiranje na voljo naslednje VIN številke (Tabela 1):

ZFA16900001874885	WAUZZZ4F48N169935	VF1BT110644143504
ZAR93200001009108	W0LOAHM757G164728	VF1BT0J0640486946
WVWZZZ1JZ1W067803	VZ2260026NTT02419	VF1BR1J0H40238837
WVWZZZ1GZNW190163	VX1128A0000943704	VF1BA0E0514567701
WVWZZZ1GZMW400865	VX1100D0000910723	VANL820882Y042142
WVWZZZ19ZHW512781	VSSZZZ6KZZR728191	VADA704A090X25465
WP1ZZZ9PZ9LA41878	VF7CHRHYB39719840	UU15SDAG453175072
WF0BXXGCABVP18422	VF36ERHRH21173014	U5YFF23229L034884
WDB9525621L099337	VF1RFB00658859943	TMBEFF613V0387800
WDB4271011W166065	VF1KW41B156288236	TMBAM6NH7D4040519
WBSHD91060BK01313	VF1KTDN0642499573	JMZDK6WE601512805
WBAVW11060A280216	VF1KM1R0H40486924	JHMGD57704S209401
WBAPD11020WF83689	VF1KG0G0624771509	

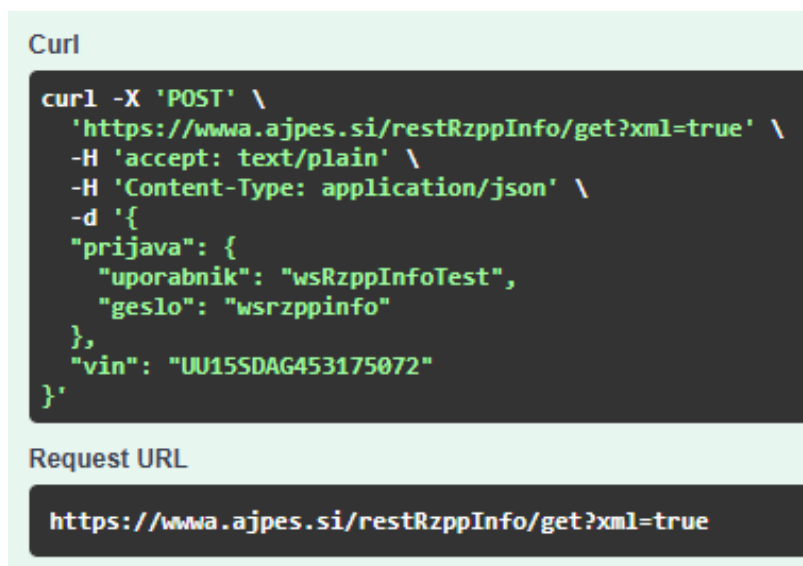
Tabela 1: Seznam VIN številke za testiranje.

Podobno kot pri metodi `checkSredstva` tudi tu Content-Type izhodnega sporočila izbiramo z parametrom `xml`, ki ga damo v URL vrstico. Če želimo odgovor v XML obliki, izberemo `?xml=true`. Takole potem izgleda celoten URL:

<https://www.ajpes.si/restRzppInfo/get?xml=true>

Če želimo izpis v (deserializirani) JSON obliki, parametra `xml` ne uporabimo (`null`) ali pa ga nastavimo na vrednost `False` (`xml=false`).

Klic REST API lahko izvedemo tudi z ukazom `curl` v CMD terminalu. Primer klica metode **get** je prikazan na Sliki 6.



Slika 6: Curl klic metode **get**, kjer je vhodno sporočilo (Request-Body) v obliki JSON (`-H Content-Type: application/json`), izhodno sporočilo (Response-Body) pa v obliki XML (`?xml=true`). Ekranska kopija s Swagger UI vmesnika.

Pripadajoč odgovor (Response):

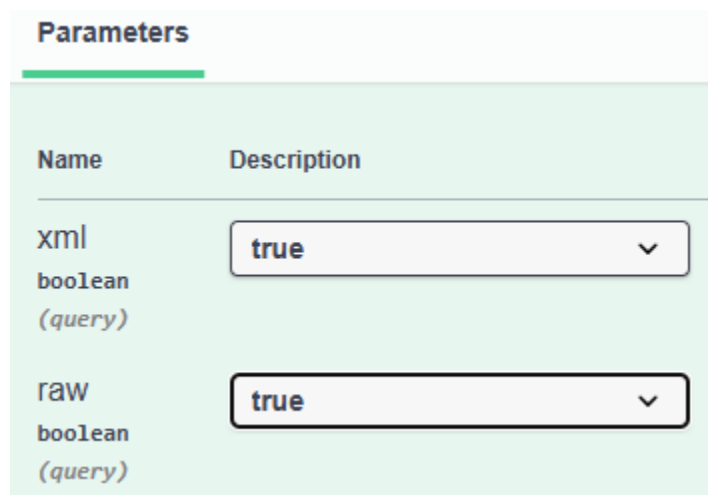
```
<RzppInfoResponseOfRzppInfoVrni xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <status>2000</status>
  <message>OK</message>
  <timestamp>2026-10-13 10:47:32</timestamp>
  <payload>
    <formatted>
      <Ident datum="2026-10-13T10:47:32" xmlns="http://www.ajpes.si/xml_sheme/rzpp/rzpp_info-20150131" />
      <Premicnina identifikacijskaStevilka="UU15SDAG453175072" xmlns="http://www.ajpes.si/xml_sheme/rzpp/rzpp_info-20150131">
        <Status>Zastavljena</Status>
        <Znamka>DACIA</Znamka>
        <Vrsta>tovorno vozilo</Vrsta>
        <KomerzialnaOznaka>SANDERO / 1.2 / 16V</KomerzialnaOznaka>
        <Zadeve tip="Prepoved" stevilka="200057" />
        <Zadeve tip="Rubež" stevilka="200066" />
        <Zadeve tip="Prepoved" stevilka="200072" />
      </Premicnina>
    </formatted>
  </payload>
</RzppInfoResponseOfRzppInfoVrni>
```

Podatki s poslovno logiko se polnijo v polju <payload><formatted>.

POZOR, POMEMBNO za razvijalce:

Za nadgradnjo informacijskih sistemov uporabnikov, kjer se stari SOAP spletni servis wsRzppInfo zamenjuje z novim REST API restRzppInfo, je najlažje, če pri klicu metode get v URL naslovu poleg parametra xml=true uporabite še parameter raw=true. Takole potem izgleda celoten URL:

<https://www.ajpes.si/restRzppInfo/get?xml=true&raw=true>



The image shows a 'Parameters' section in a Swagger UI. It contains two query parameters: 'xml' and 'raw'. Both are of type 'boolean' and are set to 'true' via dropdown menus. The 'xml' parameter is labeled '(query)' and the 'raw' parameter is also labeled '(query)'.

Name	Description
xml boolean (query)	true
raw boolean (query)	true

Slika 7: Ekranska kopija Swagger UI vmesnika, kjer sta izbrana parametra *xml=true* in *raw=true*.

Novi REST API na ta način vrača podatke v obliki, ki je najbližja tisti, ki jo vrača obstoječi SOAP spletni servis wsRzppInfo, t.j. **raw (serialized) string XML-a**. Pri tem je pomembno, da se podatki v tem načinu vrnejo v polju 'payload/raw'. V kolikor je parameter *raw* enak NULL ali False, so podatki formatirani (deserializirani) in se nahajajo v polju 'payload/formatted'.

Response metode **get** v načinu `xml=true&raw=true`

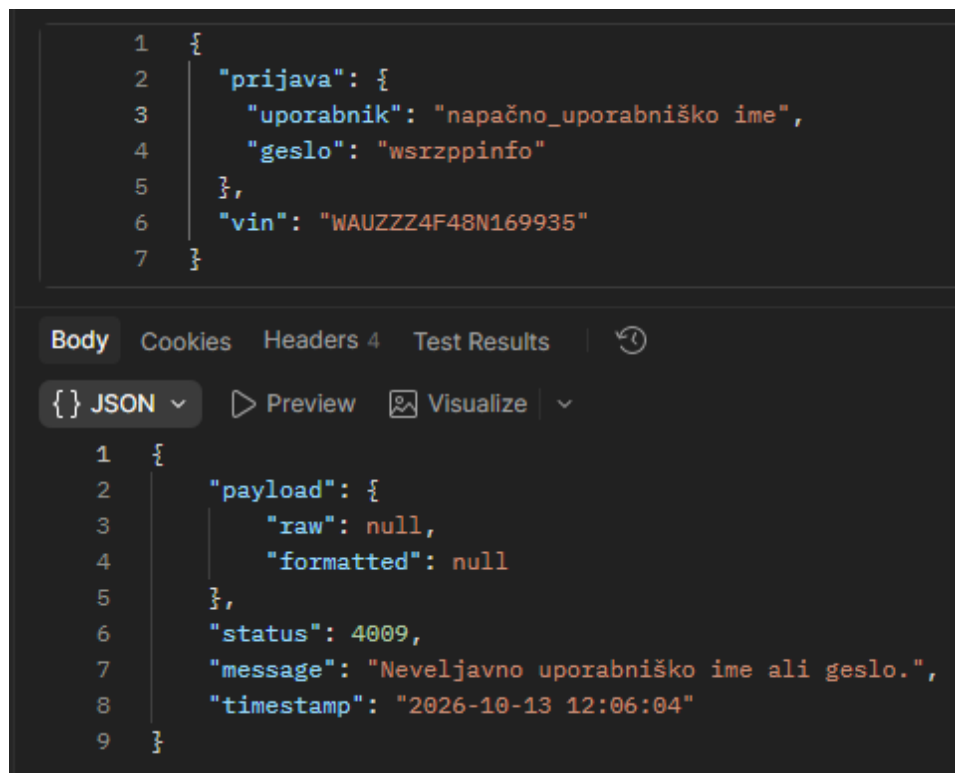
```
<RzppInfoResponseOfRzppInfoVrni xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <status>2000</status>
  <message>OK</message>
  <timestamp>2026-10-13 11:11:51</timestamp>
  <payload>
    <raw>
      &lt;RzppInfoVrni xmlns="http://www.ajpes.si/xml_sheme/rzpp/rzpp_info-20150131"&gt;&lt;Ident datum="2026-
      10-13T11:11:51" /&gt;&lt;Premicnina
      identifikacijskaStevilka="UU15SDAG453175072"&gt;&lt;Status&gt;Zastavljena&lt;/Status&gt;&lt;Znamka&gt;DACIA
      &lt;/Znamka&gt;&lt;Vrsta&gt;tovorno vozilo&lt;/Vrsta&gt;&lt;KomericalnaOznaka&gt;SANDERO / 1.2 /
      16V&lt;/KomericalnaOznaka&gt;&lt;Zadeve tip="Prepoved" stevilka="200057" /&gt;&lt;Zadeve tip="Rubež"
      stevilka="200066" /&gt;&lt;Zadeve tip="Prepoved" stevilka="200072"
      /&gt;&lt;/Premicnina&gt;&lt;/RzppInfoVrni&gt;
    </raw>
  </payload>
</RzppInfoResponseOfRzppInfoVrni>
```

Podatki s poslovno logiko (in obvestila o napakah) se polnijo v polju <payload><raw>.

VALIDACIJA VHODNIH PODATKOV

Vhodni podatki POST metod se najprej ustrezno validirajo. V primeru napake pri validaciji vhodnih podatkov (uporabniško ime, geslo in VIN številka) uporabnik prejme bolj ali manj jasno sporočilo, za kakšno napako gre. Nekaj primerov pričakovanih napak sledi v nadaljevanju. Za prikaz je uporabljeno orodje Postman:

- a) Neveljavno uporabniško ime ali geslo



b) Manjka polje 'prijava' za identifikacijo uporabnika

```
1 {
2   "vin": "WAUZZZ4F48N169935"
3 }
4
5
6
7
```

Body Cookies Headers 4 Test Results

{ JSON Preview Visualize

```
1 {
2   "payload": {
3     "raw": null,
4     "formatted": null
5   },
6   "status": 4000,
7   "message": "Napaka pri validaciji vhodnih podatkov! Napaka! Polje 'prijava' (uporabnik, geslo) je obvezno.",
8   "timestamp": "2026-10-13 12:08:31"
9 }
```

c) Neveljavno število znakov v polju 'vin'

```
1 {
2   "prijava": {
3     "uporabnik": "wsRzppInfoTest",
4     "geslo": "wsrzppinfo"
5   },
6   "vin": "WAUZZZ4F48N169"
7 }
8
```

Body Cookies Headers 4 Test Results

{ JSON Preview Visualize

```
1 {
2   "payload": {
3     "raw": null,
4     "formatted": null
5   },
6   "status": 4007,
7   "message": "Napaka pri validaciji vhodnih podatkov! Polje 'vin' mora vsebovati natančno 17 znakov.",
8   "timestamp": "2026-10-13 12:11:02"
9 }
```

d) VIN številke ni v registru

(v testnem okolju, status:2001, v polju message pa se izpiše seznam dovoljenih VIN števil za testiranje)

```
{
  "payload": {
    "raw": null,
    "formatted": {
      "ident": {
        "opisNapake": "Premičnine z vpisanim identifikatorjem VF1KG0G0624771588 med iskano vrsto premičnin ni v registru.",
        "datum": "2026-10-13T15:18:06"
      }
    },
    "premicnina": null
  },
  "status": 2001,
  "message": "V testnem okolju so na voljo naslednje VIN številke: ZFA16900001874885, ZAR93200001009108, WVVZZZ1JZ1W067803, WVVZZZ1GZNW190163, WVVZZZ1GZMW400865, WVVZZZ19ZHW512781, WP1ZZZ9PZ9LA41878, WF0BXGXCABVP18422, WDB9525621L099337, WDB4271811W166065, WBSHD91060BK01313, WBAVW11060A280216, WBAPD11020WF83689, WAUZZZ4F48N169935, W0L0AHM757G164728, VZ2260026NTT02419, VX1128A0000943704, VX110000000910723, VSSZZZ6KZZR728191, VF7CHRYH39719840, VF36ERHRH21173014, VF1RFB00658859943, VF1KW41B156288236, VF1KTON0642499573, VF1KM1R0H40486924, VF1KG0G0624771509, VF1BT110644143504, VF1BT0J0640486946, VF1BR1J0H40238837, VF1BA0E0514567701, VANL820882Y042142, VADA704A090X25465, UU15SDAG453175072, U5YFF23229L034884, TM8EFF613V0387800, TMBAM6NH7D4040519, JMZDK6WE601512805, JHMG057704S209401",
  "timestamp": "2026-10-13 15:18:06"
}
```

(v produkcijskem okolju, status:2001, message:OK)

```
{
  "payload": {
    "raw": null,
    "formatted": {
      "ident": {
        "opisNapake": "Premi nine z vpisanim identifikatorjem VF1KG0G0624771508 med iskano vrsto premi  nin ni v registru.",
        "datum": "2026-10-13T15:04:00"
      },
      "premi  nina": null
    }
  },
  "status": 2001,
  "message": "OK",
  "timestamp": "2026-10-13 15:04:00"
}
```