



Pošta Slovenije d.o.o.
Slomškov trg 10
2500 Maribor
e-pošta: info@posta.si
www.posta.si

NAVODILA ZA UPORABO SPLETNEGA SERVISA REST

KAZALO

KAZALO	2
Opis dokumenta	4
REST Servis	5
Dostop do servisa	5
Grafični prikaz komunikacije s servisom.....	5
Zahteve za delovanje servisa	5
Objekti za komunikacijo s servisom	6
ProviderRequest	6
Authentication.....	6
Data.....	7
DataType	7
PaperSize.....	8
ProviderResponse	8
Code.....	8
Value	9
ProcessStatus.....	9
ResultCode	10
Error.....	10
ParcelsNumbers.....	11
eWebOddajaSpremniceERSresponse.....	11
eWebOddajaNapake.....	12
METODE post.....	13
R_GetGuid – Pridobivanje žetona za posredovanje podatkov.....	13
C_PostData – Oddaja podatkov o pošiljkah.....	13
R_GetStatus – Poizvedba o statusu obdelave oddanih pošiljk	14
R_GetDocument – Pridobivanje dokumenta	14
R_GetDocumentUnregistered – Pridobivanje neknjiženega dokumenta	15

R_GetERSDocument – Pridobivanje ERS-dokumenta in posredovanje e-podatkov za ERS-pošiljke	15
C_PostDataERS – Posredovanje e-podatkov za ERS-pošiljke.....	16
D_PostDeleteParcel – Storniranje oddanih e-podatkov	16
Primer strukture podatkov pri klicu post metode C_PostData	17
Implementacija C#	18
Pomožne metode.....	18
Decompress.....	18
Post	18
Implementacija v PHP	19
Pomožne metode.....	19
Decompress.....	19
Post	19
IMPLEMENTACIJA V GOOGLE APPS SCRIPT	20
UPORABA V POSTMANU	21
Psevdokoda za druge programske jezike.....	22
Dekodiranje odgovora iz strežnika.....	22
Kazalo tabel	23
Kazalo implementacij	23

OPIS DOKUMENTA

Dokument opisuje strukturo podatkovnega modela za elektronsko izmenjavo podatkov z informacijskim sistemom Pošte Slovenije (v nadaljevanju: PS). Omogoča izmenjavo podatkov o pošiljkah v domačem in mednarodnem prometu ter pridobivanje ustreznih obrazcev v PDF-formatu.

Datum	Verzija dokumenta	Opomba
14.1.2021	1.0	Priprava dokumentacije
17.2.2022	1.1	Dopolnitve za klic ERS-dokumenta
16.9.2022	1.2	Dopolnitev Apps Script, Postman
12.10.2022	1.3	Dodana metoda za storniranje e-podatkov
3.1.2024	1.4	Dopolnitev za velikost dokumentov

Tabela 1 – Spremembe dokumenta

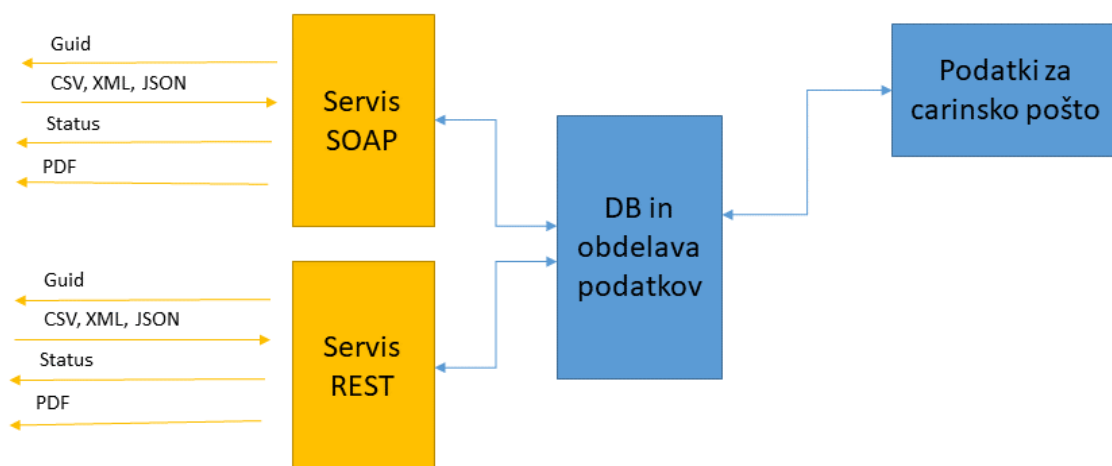
REST SERVIS

Dostop do servisa

Test: <https://tstePortal.posta.si/Services/eSpremnica.Provider/WebApi/api/>

Produkcija: <https://ePortal.posta.si/Services/eSpremnica.Provider/WebApi/api/>

Grafični prikaz komunikacije s servisom



Zahteve za delovanje servisa

Za pravilno implementacijo je treba zagotoviti naslednje:

- komunikacija deluje samo prek HTTPS;
- izmenjava podatkov je omogočena samo registriranim uporabnikom – komitentom;
- za vzpostavitev komunikacije s servisom je treba zagotoviti avtorizacijo med sistemoma (BasicAuth). Prijavne podatke posreduje skrbnik pogodbe na strani Pošte Slovenije;
- odgovor (Response) je komprimiran string. Uporabljena je GZip-oblika s štimi zlogi odmika, zato klic s programom Postman in podobnimi ni uspešen.
- Za izklop kompresije in odmika je treba uporabiti glavo (header) "Accept-Encoding" : "identity".

OBJEKTI ZA KOMUNIKACIJO S SERVISOM

ProviderRequest

Objekt je namenjen posredovanju informacij na servis REST.

Element	Tip	Opis	Obvezen
Authentication	Authentication	Avtentikacija uporabnika	O
Guid	string	Žeton za oddajo podatkov	O
Data	object	Tip odvisen od metode klica	O
Route	string		N
DataType	int	Tip oddaje podatkov	PO
PaperSize	string	Velikost papirja	PO

Tabela 2 - ProviderRequest

C# implementacija

```
public class ProviderRequest
{
    public Authentication Authentication {get;set; }
    public string Guid {get;set; }
    public object Data { get;set; }
    public string Route { get; set; }
    public int DataType {get;set; }
    public string PaperSize { get; set; }
}
```

Implementacija 1 - ProviderRequest

Authentication

Objekt je namenjen avtentikaciji uporabnika pri izmenjavi podatkov s sistemom PS.

Element	Tip	Opis	Obvezen
KomitentId	int	(*)	O
Pogodbald	int	(*)	O
Podružnicald	int	(*)	O
Postald	string	(*)	O
PogodbaPPNId	int	(*)	P

Tabela 3 – Authentication

(*) Podatke za prijavo uporabnika posreduje PS.

C# implementacija

```
public class Authentication
{
    public int KomitentId { get; set; }
    public int PogodbaId { get; set; }
    public int PodruznicaId { get; set; }
    public string PostaId { get; set; }
    public string PogodbaPPNID { get; set; }
}
```

Implementacija 2 – Authentication

Data

Objekt je namenjen posredovanju podatkov v sistem PS. Njegov tip je odvisen od metode klica spletnega servisa.

Metoda	Tip	Obvezen
R_GetGuid		N
C_PostData	List<eOddajaData>	O
R_GetStatus		N
R_GetDocument	string	PO
R_GetDocumentUnregistered	List<eOddajaData>	O
R_GetERSDocument	List<eOddajaData>	O
C_PostDataERS	List<eOddajaData>	O
D_PostDeleteParcel	string	O

Tabela 4 – Data

DataType

Tip oddanih podatkov pri klicu metode C_PostData.

Vrednost	Tip	Opis
2	int	CSV-oblika
3	int	XML-oblika
4	int	JSON-oblika
5	int	XML 2-oblika

Tabela 5 – DataType

PaperSize

Velikost spremnice pri klicu metode R_GetDocument.

Vrednost	Tip	Opis
Null	string	Dejanska velikost dokumenta
A4	string	Dokument pozicioniran na papirju formata A4. Možnost tiskanja 4 spremnic formata A6

Tabela 6 – PaperSize

ProviderResponse

Objekt se vrača po klicu servisa REST s potrebno vsebino. V primeru uspešne poizvedbe (Code=200) se podatki nahajajo v atributu Value. V nasprotnem primeru element vsebuje vsebino o napaki.

Element	Tip	Opis	Obvezen
Code	int	Status o poizvedbi	O
Value	object	Vsebina odgovora	O
Route	string	Klicana pot	P

Tabela 6 – ProviderResponse

C# implementacija

```
public class ProviderResponse
{
    public int Code { get; set; }
    public object Value { get; set; }
    public string Route { get; set; }
}
```

Implementacija 3 – ProviderResponse

Code

Vrednost	Opis
200	OK
400	BadRequest
401	Unauthorized
403	Forbidden
408	TimeOut
412	InvalidArguments
500	InternalServerError

501	NotImplemented
550	ValidationError
560	NoData
600	TryAgain

Tabela 7 – Code

Value

Tip objekta je odvisen od klica metode spletnega servisa. V primeru uspešne poizvedbe (200) vsebuje objekt, drugače opis napake, ki je tipa string.

Metoda	Tip objekta	Opis
R_GetGuid	string	Vsebuje vrednost GUID
C_PostData	string	Code – opis
R_GetStatus	ProcessStatus	Vsebuje objekt ProcessStatus
R_GetDocument	Byte[]	Vsebuje generirano spremnico v datoteki .PDF, ki je zapisana kot besedilo Base64.
R_GetERSDocument	Byte[]	Vsebuje generirano ERS-spremnico v datoteki .PDF, ki je zapisana kot besedilo Base64.
R_GetDocumentUnregistered	Byte[]	Vsebuje generirano spremnico v datoteki .PDF, ki je zapisana kot besedilo Base64.
C_PostDataERS	Byte[0]	
D_PostDeleteParcel	string	

Tabela 8 – Value

ProcessStatus

Objekt je namenjen posredovanju informacij o statusu obdelave posredovanih pošilk.

Element	Tip	Opis
ResultCode	int	Tip
Errors	List<Error>	Seznam morebitnih napak
ProcessNumber	int	Številka oddajnega popisa
ParcelsNumbers	List<ParcelsNumbers>	Generirane sprejemne številke za oddane podatke

Tabela 9 – ProcessStatus

resultCode

Seznam statusov obdelave podatkov

Vrednost		Tip	Opis
0	InTransmission	int	Podatki se še posredujejo
1	Transmitted	int	Podatki so posredovani
2	InProgress	int	Oddani podatki so v stanju obdelave
3	ProcessingWithoutErrors	int	Podatki so obdelani brez napak
4	ProcessingWithErrors	int	Podatki so obdelani z napako
5	InvalidInputData	int	Posredovani podatki so napačni
6	NoData	int	Za zahtevani žeton ni podatkov
7	ProcessingWithCN23Errors	int	Podatki so obdelani z napakami v strukturi CN23
8	None	int	Inicializirana vrednost

Tabela 10 – ResultCode

Error

Objekt je namenjen posredovanju informacij o napakah, če so se pojavile pri obdelavi posredovanih pošilk, samo kadar je ResultCode = 4 ali 7.

Element	Tip	Opis
RowNumber	int	Številka napake
Description	string	Opis morebitnih napak v slovenskem jeziku
DescriptionEN	string	Opis morebitnih napak v angleškem jeziku

Tabela 11 – Error

C# implementacija

```
public class Error
{
    public int RowNumber{ get; set; }
    public string Description{ get; set; }
    public string DescriptionEN{ get; set; }
}
```

Implementacija 4 – Error

ParcelsNumbers

Objekt je namenjen posredovanju informacij o sprejetih pošiljkah iz oddanih podatkov.

Element	Tip	Opis
RecordNumber	int	Zaporedna številka pošiljke
ParcelNumber	string	Sprejemna številka
Comment	string	

Tabela 12 – ParcelsNumbers

C# implementacija

```
public class ParcelsNumbers
{
    public int RecordNumber{ get; set; }
    public string ParcelNumber{ get; set; }
    public string Comment{ get; set; }
}
```

Implementacija 5 – ParcelsNumbers

eWebOddajaSpremniceERSresponse

Objekt je namenjen posredovanju podatkov o ERS-pošiljki.

Enostavno vračilo pošiljk (ERS) je funkcionalnost, ki omogoča kreiranje in natis paketne spremnice ERS, ki jo posredujete naslovnikom, ki želijo vrniti pošiljke na vaš naslov. Vračilo pošiljk je možno samo za vrste pošiljk MojPaket Mini, Midi in Maxi ter za komitente, ki imajo s Pošto Slovenije sklenjeno pogodbo PPN A ali PPN B. Če v vaši pogodbi za prenos pošiljk še nimate opredeljene uporabe pošiljk MojPaket Mini, Midi ali Maxi ter pogodbe PPN A ali PPN B, se za dodatne informacije obrnite na vašega skrbnika pogodbe.

Element	Tip	Opis
PdfReport	Byte[]	Generirana ERS-spremnica v PDF-formatu
ParcelNumber	string	Sprejemna številka ERS-pošiljke
Comment	string	Polje Komentar pri oddaji podatkov
Napaka	string	Opis napake v slovenskem jeziku
Error	string	Opis napake v angleškem jeziku
TipNapake	string	Tip napake

Tabela 133 – eWebOddajaSpremniceERSresponse

C# implementacija

```
public class eWebOddajaSpremniceERSresponse
{
    public byte[] PdfReport{ get; set; }
    public string ParcelNumber{ get; set; }
    public string Comment{ get; set; }
    public string Napaka{ get; set; }
    public string Error{ get; set; }
    public string TipNapake{ get; set; }
}
```

Implementacija 6 – eWebOddajaSpremniceERSresponse

eWebOddajaNapake

Objekt je namenjen posredovanju napak za neknjižene pošiljke.

Element	Tip	Opis
Guid	string	Guid – žeton za posredovane e-podatke
ZapSt	int	Zaporedna številka napake
Napaka	string	Opis napake v slovenskem jeziku
Error	string	Opis napake v angleškem jeziku
TipNapake	string	Tip napake

Tabela 144 – eWebOddajaNapake

C# implementacija

```
public class eWebOddajaNapake
{
    public string Guid { get; set; }
    public int ZapSt{ get; set; }
    public string Napaka{ get; set; }
    public string Error{ get; set; }
    public string TipNapake{ get; set; }
}
```

Implementacija 7 – eWebOddajaNapake

METODE POST

Za uspešno oddajo podatkov je potrebno naprej pridobiti unikatno GUID-številko. Pridobljeni GUID se nato uporablja v vseh klicanih metodah za posamezno oddajo.

R_GetGuid – Pridobivanje žetona za posredovanje podatkov

Metoda je namenjena za pridobivanje unikatnega žetona – (GUID-a).

URL: **/eOddaja/R_GetGuid**

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe `ProviderResponse.Code=200` se v `ProviderResponse.Value` nahaja unikatni žeton.

C_PostData – Oddaja podatkov o pošiljkah

Metoda je namenjena za oddajo podatkov o pošiljkah.

URL: **/eOddaja/C_PostData**

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe `ProviderResponse.Code=200`.

R_GetStatus – Poizvedba o statusu obdelave oddanih pošilk

Metoda je namenjena za pridobivanje statusa obdelave oddanih podatkov o pošilkah. Metodo pokličite z zakasnitvijo ~3 sekund šele po uspešnem klicu metode C_PostData. Proces lahko ponavljate v ne prepogostih intervalih, dokler je ResultCode enak 1.

URL: /eOddaja/R_GetStatus

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe ProviderResponse.Code=200 se v ProviderResponse.Value nahaja objekt ProcessStatus.

R_GetDocument – Pridobivanje dokumenta

Metoda je namenjena za pridobivanje dokumenta o oddani pošilki. Metodo pokličite z zakasnitvijo ~3 sekund po uspešnem klicu metode C_PostData.

URL: /eOddaja/R_GetDocument

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe ProviderResponse.Code=200 se v objektu Value nahaja PDF-dokument tipa byte array. V primeru, da servis ne najde pravih podatkov ali da pride do napake, je objekt Value prazen.

Primer vrednosti objekta Data za klic **PDF-spremnice**:

"Data": "CH123456711" za klic posamezne paketne spremnice.

"Data": "CH123456711SI;CH123456712SI;CH123456713SI;CH123456714SI" za klic štirih paketnih spremnic.

Če sprejemna številka ni dodana klicu, servis vrne celotni **oddajni popis** za vneseni 'Guid'.

Npr.: "Data": "",

Za pridobitev oddajnega popisa za **točno določen dan** se objekt Data napolni z datumom za željeni oddajni popis.

Npr.: "Data": "2022-01-25",

R_GetDocumentUnregistered – Pridobivanje neknjiženega dokumenta

Metoda je namenjena za pridobivanje dokumenta za neknjižene pošiljke (npr. standardno pismo, navadno pismo itd.) v PDF-formatu za poslane podatke.

URL: **/eOddaja/R_GetDocumentUnregistered**

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe ProviderResponse.Code=200 se v objektu Value nahaja PDF-dokument tipa byte array. V primeru napake se v objektu nahaja seznam napak tipa List<eWebOddajaNapake>

R_GetERSDocument – Pridobivanje ERS-dokumenta in posredovanje e-podatkov za ERS-pošiljke

Namenjeno za pridobivanje dokumenta o ERS-pošiljkah in posredovanje e-podatkov za ERS-pošiljke. Paketno spremnico za vračilo lahko pridobite šele potem, **ko so bili za prvotno pošiljko uspešno oddani e-podatki ali je bila pošiljka sprejeta v poštni sistem.**

URL: **/eOddaja/R_GetERSDocument**

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe ProviderResponse.Code=200 se v objektu Value nahaja objekt eWebOddajaSpremniceERSresponse, ki vsebuje PDF-dokument tipa byte array, črtno kodo, komentar in opis napake v primeru napake.

C_PostDataERS – Posredovanje e-podatkov za ERS-pošiljke

Namenjeno samo za posredovanje e-podatkov o ERS-pošiljkah. Navedeno metodo lahko uporabljajo stranke, ki same kreirajo paketne spremnice ERS in želijo le posredovati e-podatke o ERS-pošiljkah. Podatke o ERS-pošiljkah lahko oddamo šele po tem, **ko so bili za prvotno pošiljko uspešno oddani e-podatki ali je bila pošiljka sprejeta v poštni sistem.**

URL: **/eOddaja/C_PostDataERS**

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe `ProviderResponse.Code=200` se v objektu `Value` nahaja objekt `eWebOddajaSpremniceERSresponse`, ki vsebuje prazen byte array, črtno kodo, komentar in opis napake v primeru napake.

D_PostDeleteParcel – Storniranje oddanih e-podatkov

Namenjeno za storniranje posameznih oddanih e-podatkov o pošiljkah. E-podatke o pošiljkah lahko storniramo šele po tem, **ko so bili za prvotno pošiljko uspešno oddani e-podatki.**

URL: **/eOddaja/D_PostDeleteParcel**

Metoda: **POST**

Parametri: **ProviderRequest**

Odgovor: **ProviderResponse**

V primeru uspešne poizvedbe `ProviderResponse.Code=200` se v objektu `Value` nahaja string "Ok".

Primer strukture podatkov pri klicu post metode C_PostData

```
{
  "Authentication": {
    "Postald": "7773",
    "Podružnicald": 0,
    "Pogodbald": 1234,
    "Komitentld": 11111
  },
  "Guid": "17cf6ca8-e10e-40e0-83ff-621823338b49",
  "Data": "[{
    "parcelType": 101,
    "parcels": [
      {
        "additionalServices": [
          {
            "code": "DOB",
            "amoun": 0.0
          },
          {
            "code": "PR",
            "amoun": 0.0
          },
          {
            "code": "V",
            "amoun": 456.45
          },
          {
            "code": "ODKBN",
            "amoun": 123.45
          }
        ],
        "parcelNumber": "CF789456124SI",
        "weigh": 28200,
        "volume": 0.0,
        "cn23": null,
        "international": null
      }
    ],
    "commonServices": null,
    "addressee": {
      "addresseeName1": "Poštar Pavli",
      "addresseeName2": "Pošta Slovenije d.o.o.",
      "addresseeAddress": "Slomškov trg 10",
      "addresseePostld": "2500",
      "addresseePos": "Maribor",
      "addresseeCountryld": "705",
      "addresseePhone": "040999111",
      "addresseeEmail": postar.pavli@posta.si
    },
    "upn": {
      "UPNtype": 6,
      "UPNpurpose1": null,
      "UPNpurpose2": null,
      "UPNmodel": null,
      "UPNreference": "1235-7889-456",
      "UPNiban": null
    },
    "comment1": "opomba številka 1",
    "comment2": "opomba številka 2",
    "recordNumber": 22
  }]",
  "DataType": 4,
  "Route": null
}
```

IMPLEMENTACIJA C#

Pomožne metode

Decompress

```
public string Decompress(string aValue)
{
    byte[] gzBuffer = Convert.FromBase64String(aValue);
    using (MemoryStream ms = new MemoryStream())
    {
        int msgLength = BitConverter.ToInt32(gzBuffer, 0);
        ms.Write(gzBuffer, 4, gzBuffer.Length - 4);
        byte[] buffer = new byte[msgLength];
        ms.Position = 0;
        using (System.IO.Compression.GZipStream zip = new System.IO.Compression.GZipStream(ms,
            System.IO.Compression.CompressionMode.Decompress))
        {
            zip.Read(buffer, 0, buffer.Length);
        }
        return System.Text.Encoding.Unicode.GetString(buffer, 0, buffer.Length);
    }
}
```

Metoda 1 – C# - Pomožna metoda – Decompress

Post

```
private string Post(string aUsername, string aPassword, string aUrl, string aData)
{
    string result = string.Empty;
    using (var client = new WebClient())
    {
        client.Headers[HttpRequestHeader.ContentType] = "application/json";
        client.Encoding = Encoding.UTF8;
        if (!String.IsNullOrEmpty(aUsername) &&
            !String.IsNullOrEmpty(aPassword))
        {
            string credentials = Convert.ToBase64String(Encoding.ASCII.GetBytes(aUsername + ":" +
aPassword));
            client.Headers[HttpRequestHeader.Authorization] = string.Format("Basic {0}",
credentials);
        }
        result = client.UploadString(aUrl, "POST", aData);
        if (client.ResponseHeaders["Content-Encoding"] == "gzip")
            result = Decompress(result);
    }
    return result;
}
```

Metoda 2 – C# – Pomožna metoda – Post

IMPLEMENTACIJA V PHP

Pomožne metode

Decompress

```
<?php
function Decompress(string $arg1)
{   return gzdecode(substr(base64_decode($arg1), 4)); }
?>
```

Metoda 3 – PHP – Decompress

Post

```
<?php

$authentication = array("KomitentId"=>0, "Pogodbald"=>0, "Podruznicald"=>0, "Postald"=>"0");
$request = json_encode(array("Authentication" => $authentication ));

function Post($aUsername, $aPassword, $aUrl, $aData)
{
    $curl = curl_init($aUrl);
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
    curl_setopt($curl, CURLOPT_USERPWD, "$aUsername:$aPassword");
    curl_setopt($curl, CURLOPT_HTTPHEADER,
                array("Content-type: application/json"));
    curl_setopt($curl, CURLOPT_POST, true);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $aData);
    $json_response = curl_exec($curl);
    $status = curl_getinfo($curl, CURLINFO_HTTP_CODE);
    if ( $status != 200 ) {die("Error: call to URL $url failed with status $status, response
$json_response, curl_error " . curl_error($curl) . ", curl_errno " . curl_errno($curl));}
    curl_close($curl);
    return Decompress($json_response);
}?>
```

Metoda 4 – PHP – Post

IMPLEMENTACIJA V GOOGLE APPS SCRIPT

```
function posta() {  
    var username = "xxx";  
    var password = "xxx";  
    var header = {  
        "Content-Type": "application/json",  
        "Accept-Encoding" : "identity",  
        "Authorization": "Basic " + Utilities.base64Encode(username + ":" + password)  
    }  
  
    };  
    var payload = JSON.stringify({  
        "Authentication": {  
            "PostaId": "7773",  
            "PodruznicaId": 0,  
            "PogodbaId": x,  
            "KomitentId": x  
        },  
        "Data": "",  
        "DataType": 4,  
        "Route": null  
        //"Guid": "",  
    })  
    var options = {  
        "method" : "POST",  
        "headers" : header,  
        "payload": payload  
    }  
    Logger.log(UrlFetchApp.fetch("https://tstePortal.posta.si/Services/eSpremnic.  
Provider/WebApi/api/eOddaja/R_GetGuid" , options));  
    return ;  
}
```

Metoda 5 – Google Apps Script

UPORABA V POSTMANU

POST https://tsteportal.posta.si/Services/eSpremnica.Provider/WebApi/apl/eOddaja/R_GetGuid Send

Params Authorization Headers (13) Body Pre-request Script Tests Settings Cookies

Type Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username

Password

☐ Show Password

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

Params Authorization Headers (12) Body Pre-request Script Tests Settings

Headers 10 hidden

	KEY	VALUE
☒	Content-Type	application/json
☒	Accept-Encoding	identity
	Key	Value

Params Authorization Headers (12) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   ... "Authentication": {
3     ... "PostaId": "7773",
4     ... "PodruznicaId": x,
5     ... "PogodbaId": x,
6     ... "KomitentId": x
7   },
8   ... "Data": "",
9   ... "DataType": 4,
10  ... "Route": null,
11  ... "Guid": ""
12 }
13
```

PSEVDOKODA ZA DRUGE PROGRAMSKE JEZIKE

Dekodiranje odgovora iz strežnika

1. Niz iz strežnika v obliki Base64

"RAAAAB+LCAAAAAABACrZlBicGbIZ0hhSAWyrBhMGAwYDBl0gOwwhkSGHIZSqLgSEIJEG4BqSx1KkERrGQAjtv6kRAAAAA=="

2. Niz pretvorimo iz oblike Base64 v polje zlogov `byte[]`

V desetiškem zapisu po zlogih:

0	1	2	3	4	5	6	7	8	9
68	0	0	0	31	139	8	0	0	0
10	11	12	13	14	15	16	17	18	19
0	0	4	0	171	102	80	98	112	102
20	21	22	23	24	25	26	27	28	29
200	103	72	97	72	5	178	172	24	76
30	31	32	33	34	35	36	37	38	39
24	12	24	12	25	116	128	236	48	134
40	41	42	43	44	45	46	47	48	49
68	134	28	134	82	168	184	18	16	130
50	51	52	53	54	55	56	57	58	59
68	131	128	106	75	25	74	144	68	107
60	61	62	63	64	65	66	67	68	69
25	0	35	182	254	164	68	0	0	0

3. Odstranimo prve štiri zloge
4. Nato niz dekompresiramo po metodi gzip

Zapis v kodni tabeli unicode: {"Code":401,"Value":"","Route":""}

KAZALO TABEL

Tabela 1 – Spremembe dokumenta	4
Tabela 2 - ProviderRequest.....	6
Tabela 3 – Authentication	6
Tabela 4 – Data.....	7
Tabela 5 – DataType.....	7
Tabela 6 – ProviderResponse	8
Tabela 7 – Code.....	9
Tabela 8 – Value.....	9
Tabela 9 – ProcessStatus.....	9
Tabela 10 – ResultCode.....	10
Tabela 11 – Error.....	10
Tabela 12 – ParcelsNumbers	11
Tabela 13 – eWebOddajaSpremniceERSresponse	11
Tabela 14 – eWebOddajaNapake.....	12

KAZALO IMPLEMENTACIJ

Implementacija 1 - ProviderRequest.....	6
Implementacija 2 – Authentication.....	7
Implementacija 3 – ProviderResponse	8
Implementacija 4 – Error.....	10
Implementacija 5 – ParcelsNumbers.....	11
Implementacija 6 – eWebOddajaSpremniceERSresponse	12
Implementacija 7 – eWebOddajaNapake.....	12