



# RPSD – API pro taxislužby – Technická dokumentace

COMINT s.r.o.

Verze 1.0, 9.11.2020: Revize, informace o denním servisním okně

# Obsah

Historie dokumentu .....	1
1. Služba Prověření vozidla .....	2
1.1. Požadavek .....	3
1.2. Odpověď .....	5
2. Služba Prověření průkazu řidiče taxi .....	8
2.1. Požadavek .....	9
2.2. Odpověď .....	11
3. Společná datové typy .....	14
3.1. Třídy .....	14
3.2. Typy polí .....	14
3.3. Výčty .....	16
4. Zabezpečení a výkon .....	18
4.1. Předávání IP adresy .....	18
Odkazy .....	20
Použité zkratky .....	21

# Historie dokumentu

Datum	Verze	Autor	Popis
14.05.2020	0.1	Miroslav Exner	Založení dokumentu
19.05.2020	0.2	Miroslav Exner	Rozdělení služeb a dílčí úpravy
19.05.2020	0.3	Roman Knotek	Revize
25.05.2020	0.4	Miroslav Exner	Doplnění autentizace GUI
25.05.2020	0.5	Roman Knotek	Revize
15.07.2020	0.6	Roman Knotek	Úprava definice přípustného formátu registrační značky
9.11.2020	1.0	Roman Knotek	Revize, informace o denním servisním okně

# 1. Služba Prověření vozidla

Služba slouží k prověření vozidla, je-li evidováno v RPSD jako vozidlo taxislužby.

Služba využívá architekturu REST, komunikačním protokolem je HTTP(S), k výměně dat je využit formát JSON a je dostupná na těchto URL:

Pro produkční prostředí:

<https://rpsd.mdcz.cz/services/taxiApi/v1/checkVehicleRegistration>

Pro testovací prostředí:

<https://test-rpsd.mdcz.cz/services/taxiApi/v1/checkVehicleRegistration>

Pro volání služby je nutné použít HTTP metodu POST.



Provozní data API rozhraní jsou aktualizována dávkově jednou denně. Při plánování dotazů z vašich provozních systémů se prosím pokud možno vyhněte časovému oknu 21:30 až 23:00.

## 1.1. Požadavek

Požadavek je nutné zaslat metodou POST na URL endpointu s tělem požadavku ve formátu JSON (viz příklady).

*Příklad těla požadavku v JSON formátu — dotaz na jedno vozidlo*

```
{
  "registrationPlates" : ["ABC1234"]
}
```

*Příklad těla požadavku v JSON formátu — dotaz na více vozidel*

```
{
  "registrationPlates" : ["ABC1234", "1A23456", "DEF5678", "asda1421adawd"]
}
```

*Příklad těla požadavku v JSON formátu — s vyplněným IČO*

```
{
  "registrationPlates" : ["ABC1234", "1A23456", "DEF5678", "asda1421adawd"],
  "ico" : "01234567"
}
```

Strukturu požadavku je možné zapsat takto:



Obrázek 1. Datová struktura požadavku

V tabulce jsou uvedena jednotlivá pole požadavku, jejich datové typy i povinnost vyplnění.

### 1.1.1. Třída **CheckVehicleRegistrationRequest**

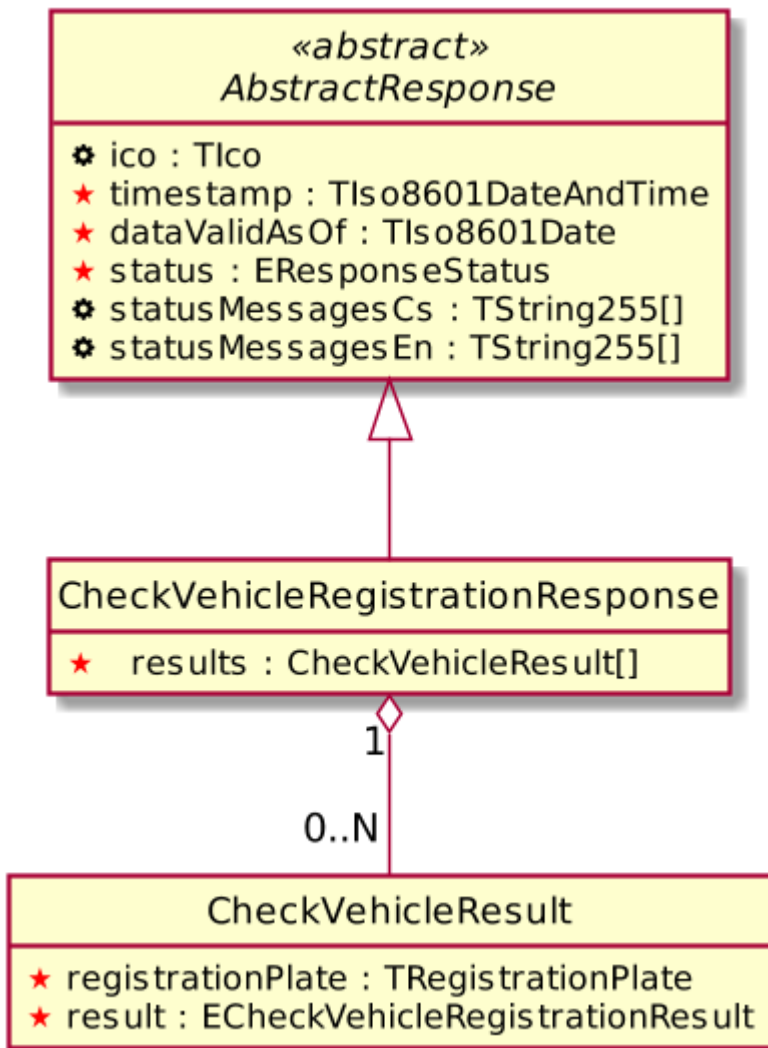
Pole	Datový typ	Povinné	Popis
<code>ico</code>	<code>TIco</code>	Ne	IČO subjektu, který se dotazuje. Pole nemusí být přítomné. Pokud je přítomné, jeho hodnota musí splňovat formát IČO včetně kontrolního součtu.
<code>registrationPlates</code>	<code>TRegistrationPlate[]</code>	Ano	Pole s registračními značkami, podle kterých se mají prověřovat daná vozidla. Je nutné uvést alespoň jednu, maximálně však 3000 položek, jinak je dotaz považován za chybný.

## 1.2. Odpověď

Příklad výsledné odpovědi v JSON formátu

```
{
  "ico" : "01234567",
  "timestamp" : "2020-07-02T14:31:12.123+0100",
  "dataValidAsOf" : "2020-07-01",
  "status" : "WARNING",
  "statusMessagesCs" : ["Ve výsledcích se vyskytla minimálně jedna chyba služby",
    "Ve výsledcích se vyskytlo minimálně jedno zadání ve špatném
formátu"],
  "statusMessagesEn" : ["There is at least one service error in results",
    "There is at lest one wrong format error in results"],
  "registrationPlates" : [
    {
      "registrationPlate" : "ABC1234",
      "result" : "VALID"
    },
    {
      "registrationPlate" : "1A23456",
      "result" : "VALID"
    },
    {
      "registrationPlate" : "DEF5678",
      "result" : "NOT_VALID"
    },
    {
      "registrationPlate" : "1S34567",
      "result" : "SERVICE_ERROR"
    },
    {
      "registrationPlate" : "asda1421adawd",
      "result" : "WRONG_FORMAT"
    }
  ]
}
```

Strukturu odpovědi je možné vidět na následujícím diagramu:



Obrázek 2. Datová struktura odpovědi

V následujících tabulkách jsou uvedena jednotlivá pole odpovědi, jejich datové typy i povinnost výskytu.



### 1.2.1. Třída `AbstractResponse`

Pole	Typ	Povinný	Popis
<code>ico</code>	<code>TIco</code>	Ne	IČO organizace, která daný dotaz vyvolala
<code>timestamp</code>	<code>TIso8601DateAndTime</code>	Ano	Časová značka odpovědi
<code>dataValidAsOf</code>	<code>TIso8601Date</code>	Ano	K jakému datu jsou údaje platné. Typicky den dotazu minus 1 den
<code>status</code>	<code>EResponseStatus</code>	Ano	Stavový kód odpovědi. Konkrétní možné kódy vizte v kapitole Výčty.
<code>statusMessagesCs</code>	<code>TString255[]</code>	Ano	Pole zpráv ke stavu odpovědi v češtině. Může být prázdné.
<code>statusMessagesEn</code>	<code>TString255[]</code>	Ano	Pole zpráv ke stavu odpovědi v angličtině. Může být prázdné

### 1.2.2. Třída `CheckVehicleRegistrationResponse`

Pole	Typ	Povinný	Popis
<code>results</code>	<code>CheckVehicleRegistrationResult[]</code>	Ano	Pole výsledků prověření vozidel podle registračních značek.

### 1.2.3. Třída `CheckVehicleResult`

Pole	Typ	Povinný	Popis
<code>registrationPlate</code>	<code>TRegistrationPlate</code>	Ano	Registrační značka prověřovaného vozidla
<code>result</code>	<code>ECheckVehicleRegistrationResult</code>	Ano	Výsledek kontroly. Konkrétní možné kódy vizte v kapitole Výčty.

## 2. Služba Prověření průkazu řidiče taxi

Služba slouží k prověření řidiče, je-li evidován v RPSD jako řidič vozidla taxislužby.

Služba využívá architekturu REST, komunikačním protokolem je HTTP(S), k výměně dat je využit formát JSON a je dostupná na těchto URL:

Pro produkční prostředí:

<https://rpsd.mdcr.cz/services/taxiApi/v1/checkDriverCardValidity>

Pro testovací prostředí:

<https://test-rpsd.mdcr.cz/services/taxiApi/v1/checkDriverCardValidity>

Pro volání služby je nutné použít HTTP metodu POST.

## 2.1. Požadavek

Požadavek je nutné zaslat metodou POST na URL endpointu s tělem požadavku ve formátu JSON (viz příklady).

*Příklad těla požadavku v JSON formátu — dotaz na jeden průkaz*

```
{
  "cardNumbers" : ["012345"]
}
```

*Příklad těla požadavku v JSON formátu — dotaz na více průkazů*

```
{
  "cardNumbers" : ["012345", "6789", "967303", "erwr3w5wtrw"]
}
```

*Příklad těla požadavku v JSON formátu — dotaz s vyplněným IČO*

```
{
  "cardNumbers" : ["012345", "6789", "967303", "erwr3w5wtrw"],
  "ico" : "01234567"
}
```

Strukturu požadavku je možné zapsat takto:

<b>CheckDriverCardRequest</b>
⚙ ico : Tico
★ cardNumbers : TCardNumber[]

*Obrázek 3. Datová struktura požadavku*

V tabulce jsou uvedena jednotlivá pole požadavku, jejich datové typy i povinnost vyplnění.

### 2.1.1. Třída `CheckDriverCardRequest`

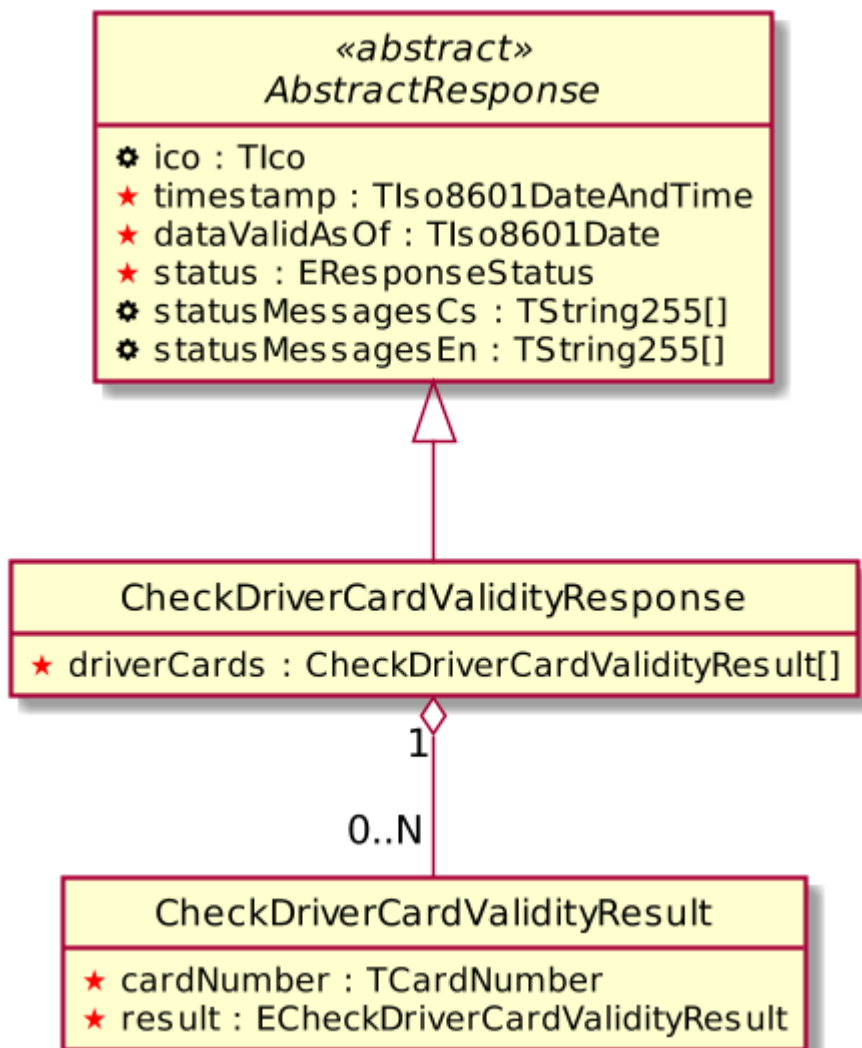
Pole	Datový typ	Povinné	Popis
<code>ico</code>	<code>TIco</code>	Ne	IČO subjektu, který se dotazuje. Pole nemusí být přítomné. Pokud je přítomné, jeho hodnota musí splňovat formát IČO včetně kontrolního součtu.
<code>cardNumbers</code>	<code>TCardNumber[]</code>	Ano	Pole s čísly průkazů, podle kterých se mají prověřit daní řidiči. Je nutné uvést alespoň jednu, maximálně však 3000 položek, jinak je dotaz považován za chybný.

## 2.2. Odpověď

Příklad výsledné odpovědi v JSON formátu

```
{
  "ico" : "01234567",
  "timestamp" : "2020-07-02T14:31:12.123+0100",
  "dataValidAsOf" : "2020-07-01",
  "status" : "WARNING",
  "statusMessagesCs" : ["Ve výsledcích se vyskytla minimálně jedna chyba služby",
    "Ve výsledcích se vyskytlo minimálně jedno zadání ve špatném
formátu"],
  "statusMessagesEn" : ["There is at least one service error in results",
    "There is at lest one wrong format error in results"],
  "driverCards" : [
    {
      "cardNumber" : "012345",
      "result" : "VALID"
    },
    {
      "cardNumber" : "6789",
      "result" : "NOT_VALID"
    },
    {
      "cardNumber" : "967303",
      "result" : "WITHHELD"
    },
    {
      "cardNumber" : "653045",
      "result" : "SERVICE_ERROR"
    },
    {
      "cardNumber" : "erwr3w5wtrw",
      "result" : "WRONG_FORMAT"
    }
  ]
}
```

Strukturu odpovědi je možné vidět na následujícím diagramu:



Obrázek 4. Datová struktura odpovědi

V následujících tabulkách jsou uvedena jednotlivá pole odpovědi, jejich datové typy i povinnost výskytu.

### 2.2.1. Třída `AbstractResponse`

Pole	Typ	Povinný	Popis
<code>ico</code>	<code>TIco</code>	Ne	IČO organizace, která daný dotaz vyvolala
<code>timestamp</code>	<code>TIso8601DateAndTime</code>	Ano	Časová značka odpovědi
<code>dataValidAsOf</code>	<code>TIso8601Date</code>	Ano	K jakému datu jsou údaje platné. Typicky den dotazu minus 1 den
<code>status</code>	<code>EResponseStatus</code>	Ano	Stavový kód odpovědi. Konkrétní možné kódy vizte v kapitole Výčty.
<code>statusMessagesCs</code>	<code>TString255[]</code>	Ano	Pole zpráv ke stavu odpovědi v češtině. Může být prázdné.
<code>statusMessagesEn</code>	<code>TString255[]</code>	Ano	Pole zpráv ke stavu odpovědi v angličtině. Může být prázdné.

### 2.2.2. Třída `CheckDriverCardValidityResponse`

Pole	Typ	Povinný	Popis
<code>driverCards</code>	<code>CheckDriverCardValidityResult[]</code>	Ano	Pole výsledků prověření řidičů podle čísel průkazů řidiče.

### 2.2.3. Třída `CheckDriverCardValidityResult`

Pole	Typ	Povinný	Popis
<code>cardNumber</code>	<code>TCardNumber</code>	Ano	Číslo prověřovaného průkazu
<code>result</code>	<code>ECheckDriverCardValidityResult</code>	Ano	Výsledek prověření. Konkrétní možné kódy vizte v kapitole Výčty.

## 3. Společná datové typy

### 3.1. Třídy

#### 3.1.1. Třída `AbstractResponse`

Pole	Typ	Povinný	Popis
<code>ico</code>	<code>TIco</code>	Ne	IČO organizace, která daný dotaz vyvolala
<code>timestamp</code>	<code>TIso8601DateAndTime</code>	Ano	Časová značka odpovědi
<code>dataValidAsOf</code>	<code>TIso8601Date</code>	Ano	K jakému datu jsou údaje platné. Typicky den dotazu minus 1 den
<code>status</code>	<code>EResponseStatus</code>	Ano	Stavový kód odpovědi. Konkrétní možné kódy vizte v kapitole Výčty.
<code>statusMessagesCs</code>	<code>TString255[]</code>	Ano	Pole zpráv ke stavu odpovědi v češtině. Může být prázdné.
<code>statusMessagesEn</code>	<code>TString255[]</code>	Ano	Pole zpráv ke stavu odpovědi v angličtině. Může být prázdné

### 3.2. Typy polí

#### 3.2.1. Typ `TIco`

IČO neboli identifikační číslo organizace.

Pole je typu `String`, jeho hodnota může být maximálně 8 znaků dlouhá a obsahovat pouze číslice. Může, ale nemusí obsahovat doplnění úvodními nulami, pokud je číslo kratší než požadovaných 8 číslic. Kratší čísla budou nulami na začátku doplněna a bude provedena kontrola na správnost zadání IČO podle poslední kontrolní číslice na principu dělitelnosti váženého součtu číslic jedenácti. Kontrolní součet je podrobněji popsán v [\[1\]](#).

Příklady korektních hodnot:

- 25596641
- 03066070 (s nevýznamovou nulou na začátku)
- 3066070 (bez nevýznamové nuly)

#### 3.2.2. Typ `TRegistrationPlate`

Registrační značka motorových vozidel v ČR. Pole je typu `String` s délkou 5 až 8 alfanumerických znaků, přičemž jsou povoleny jen tyto: A, B, C, D, E, F, H, I, J, K, L, M, N, O, P, R, S, T, U, V, X, Y, Z a čísla 0



– 9. V každé RZ musí být alespoň jedno číslo. Písmena G, Q, W jsou z řady vynechána, aby nedošlo k vizuální záměně s jinými znaky. [2]

Příklady korektních hodnot:

- LN5758 (starší SPZ)
- ALA4011 (starší SPZ)
- 4A23000 (nová RZ)
- A6738E (exportní)
- V7425 (veteránská)
- 11R0389 (závodní)
- F7498 (testovací)
- 023XX11 (diplomatická)
- DICTATOR (volitelná)

### 3.2.3. Typ `TCardNumber`

Číslo průkazu řidiče taxislužby. Pole je typu `String`, délky maximálně 6 znaků a může obsahovat pouze číslice. Číslo může ale nemusí být doplněno nevýznamovými nulami na začátku na požadovanou délku 6 znaků.

Příklady korektních hodnot:

- 1
- 001234
- 1234
- 104893
- 999999

### 3.2.4. Typ `TIso8601DateAndTime`

Datum a čas podle ISO 8601 ve formátu `YYYY-MM-DDThh:mm:ss.SSSX`

Příklady korektních hodnot:

- 2020-05-19T09:59:32.064+01:00
- 2020-12-31T08:59:32.064Z

### 3.2.5. Typ `TIso8601Date`

Datum podle ISO 8601 ve formátu `YYYY-MM-DD`.

Příklady korektních hodnot:

- 2020-01-01
- 2020-12-31

### 3.2.6. Typ `TString255`

Řetězec o maximální délce 255 znaků.

## 3.3. Výčty

### 3.3.1. Výčet `EResponseStatus`

Hodnota	HTTP Kód	Popis
<code>OK</code>	200	Požadavek a odpověď bez problému.
<code>WARNING</code>	200	V požadavku/odpovědi se vyskytly překonatelné problémy.
<code>TOO_MANY_ITEMS</code>	413	V požadavku je příliš mnoho dotazů k prověření
<code>TOO_MANY_REQUESTS</code>	429	Požadavek odeslán příliš brzy pro předchozím (byla překročena povolená zátěž API)
<code>TIMEOUT</code>	408	Odpověď se nepodařilo zformulovat v daném čase.
<code>SERVICE_ERROR</code>	500	Ve službě došlo k neočekávanému problému.

### 3.3.2. Výčet `ECheckVehicleRegistrationResult`

Hodnota	Popis
<code>VALID</code>	Vozidlo bylo podle dané RZ nalezeno v RPSD jakožto registrované vozidlo taxislužby.
<code>NOT_VALID</code>	Vozidlo není evidováno jako platné vozidlo taxislužby. Nebylo nalezeno v DB, nebo nemá platnou registraci.
<code>WRONG_FORMAT</code>	RZ vozidla se nepodařilo systémem přečíst. Je zadaná v chybném formátu.
<code>SERVICE_ERROR</code>	Ve službě došlo k neočekávanému problému.

### 3.3.3. Výčet ECheckDriverCardValidityResult

Hodnota	Popis
VALID	Průkaz byl nalezen a je platný.
NOT_VALID	Průkaz není platný. Nebyl nalezen v DB, nebo jeho platnost již vypršela.
WITHHELD	Průkaz byl zadržen.
WRONG_FORMAT	Číslo průkazu bylo zadáno v chybném formátu.
SERVICE_ERROR	Ve službě došlo k neočekávanému problému.

## 4. Zabezpečení a výkon

Služba je veřejně dostupná, tudíž se zatím neuvažuje o jakékoliv formě autentizace a autorizace pro její běžné použití.

Služba bude kontrolovat míru zatížení podle požadavků z jednotlivých IP adres a to následujícím způsobem:

- Z jedné IP adresy bude umožněno pouze zpracovávání 1 požadavku v daný okamžik. Nebude tedy možnost odeslat z jedné IP adresy několik požadavků naráz s předpokladem paralelního zpracování
- Dotaz na jedno vozidlo nebo jeden průkaz bude umožněn z jedné IP adresy v četnosti 1 dotaz s 1 položkou pro prověření za 1 sekundu a doba odezvy nepřesáhne 1 sekundu.
- Hromadný dotaz na až 3000 vozidel nebo průkazů bude umožněn z jedné IP adresy v četnosti 1 hromadný dotaz za 1 hodinu a doba odezvy nepřesáhne 10 sekund.



Dokumentace vzniká paralelně s implementací, tudíž se předpokládá, že parametry využívání služby budou upraveny (rozvolněny) podle chování a výkonu služby při pilotním provozu. Uvažuje se o úpravě na "malý" a "velký" dotaz, kdy "malým" dotazem by se rozuměl požadavek nikoliv s jednou položkou, ale několika položkami (malé desítky). "Velký" dotaz by pak obsahoval počet položek od této hranice výše a platila by pro něj pravidla pro hromadný dotaz.

### 4.1. Předávání IP adresy

IP adresa protistrany bude zjišťována standardním způsobem z příchozích paketů. Je však třeba myslet i na speciální scénář, který představuje GUI rozhraní systému. To je koncipováno jako samostatná komponenta provozovaná odděleně od API na jiné infrastruktuře. Z pohledu API se tak jedná o další protistranu specifickou tím, že agreguje neomezené množství uživatelů. GUI je tedy nutné chápat jako protistranu určitým způsobem privilegovanou (v tom smyslu, že od ní bude přijímána větší než výše popsaná zátěž) a zároveň zavázanou (v tom smyslu, že bude povinna předávat IP adresy koncových uživatelů). Tím, že GUI předá s každým požadavkem IP adresu uživatele na API, bude možné garantovat jeho výkon tak, jak je výše popsáno.

Pro tento účel nechť je použita HTTP hlavička `X-Remote-Address`, ve které se bude nacházet zmíněná IP adresa. Hlavička může vypadat např. takto:

```
X-Remote-Address: 10.109.1.132
```

Je nutné zajistit, aby ji smělo použít jen GUI rozhraní. Proto bude na straně API podporována základní HTTP autentizace [3], po které bude povoleno zpracování `X-Remote-Address`. Autentizace bude provedena formou další HTTP hlavičky `Authorization`, která může vypadat např. takto:

```
Authorization: Basic QWxhZGRpbjppcGVuU2VzYW11
```

Hodnota **Basic** značí základní autentizaci a token **QWxhZGRpbjppcGVuU2VzYW11** je metodou Base64 zakódovaný řetězec **Aladdin:OpenSesame**, který zde reprezentuje autentizační údaje **<username>:<password>**. Jedná se jen o příklad, konkrétní přístupové údaje budou sděleny během implementace a testování.

# Odkazy

- [1] [https://cs.wikipedia.org/wiki/Identifika%C4%8Dn%C3%AD\\_%C4%8D%C3%ADslo\\_osoby](https://cs.wikipedia.org/wiki/Identifika%C4%8Dn%C3%AD_%C4%8D%C3%ADslo_osoby)
- [2] [https://cs.wikipedia.org/wiki/St%C3%A1tn%C3%AD\\_pozn%C3%A1vac%C3%AD\\_zna%C4%8Dky\\_v\\_%C4%8Cesku](https://cs.wikipedia.org/wiki/St%C3%A1tn%C3%AD_pozn%C3%A1vac%C3%AD_zna%C4%8Dky_v_%C4%8Cesku)
- [3] [https://en.wikipedia.org/wiki/Basic\\_access\\_authentication](https://en.wikipedia.org/wiki/Basic_access_authentication)

# Použité zkratky

## **API**

Application program interface. Aplikační programové rozhraní, slouží pro strojový přístup z jiných aplikací a systémů.

## **ČP**

Číslo průkazu řidiče taxislužby

## **DB**

Databáze

## **GUI**

Graphical user interface. Grafické uživatelské rozhraní, slouží pro přístup lidských uživatelů systému.

## **HTTP**

Hypertext transfer protocol. Internetový komunikační protokol mezi webovými servery

## **IČO**

Identifikační číslo osoby

## **RPSD**

Rejstřík podnikatelů v silniční dopravě

## **REST**

Representational state transfer. Architektura pro tvorbu webových služeb

## **RZ**

Registrační značka vozidla (dříve označována jako SPZ)

## **SPZ**

Státní poznávací značka vozidla (v dokumentu se nerozlišuje mezi SPZ a RZ, je ale použito novější terminologie RZ)

## **WS**

Webová služba

## **UC**

Use case. Případ užití