

TecoApi

API pro PLC systémy Tecomat

**TXV 005 37.01
první vydání
leden 2018
změny vyhrazeny**

Historie změn

Datum	Vydání	Popis změn
Leden 2018	1	První vydání, popis odpovídá TecoApi v1.0.0

OBSAH

1 Úvod.....	3
2 Podmínky pro použití TecoApi.....	5
3 Služby TecoApi.....	7
3.1 Autentizace.....	7
3.2 Služba GetInfo.....	9
3.3 Služba GetList.....	11
3.4 Služba GetObject.....	13
3.5 Služba SetObject.....	15
3.6 Služba PutObject.....	17
3.7 Přehled chyb.....	19
4 Proměnné v PLC programu a formát JSON.....	22
5 Příklady.....	25
5.1 Strukturované proměnné.....	25
5.2 Proměnné typu pole.....	30
5.3 Komplexní proměnné.....	33

1 ÚVOD

Web server PLC systémů Teco je vybaven API rozhraním (`TecoApi`), kde poskytuje služby umožňující číst a zapisovat data z/do PLC systému pomocí HTTP protokolu. Data jsou poskytována ve formátu JSON (JavaScript Object Notation). JSON je textový, na jazyce zcela nezávislý formát, který se používá pro výměnu dat. Další podrobnosti o formátu JSON viz <http://www.json.org/json-cz.html>.

Přístup k PLC datům přes `TecoApi` lze použít na všech centrálních jednotkách řady Fox-trot od firmware v10.4. V systémech TC700 lze knihovnu použít na procesorech CP-7004 a CP-7007 od v10.4.

Pro naprogramování PLC je potřeba prostředí Mosaic v2018.1 nebo vyšší.

Princip činnosti

Komunikace přes `TecoApi` probíhá HTTP protokolem formou dotaz/odpověď. Komunikaci dotaz/odpověď lze charakterizovat následovně:

URI dotazu

URI je zkratka Uniform Resource Identifier (jednotný identifikátor zdroje) a skládá se z následujících částí:

```
{URI-scheme} :// {URI-host} / {resource-path} ? {query-string}
```

<code>{URI-scheme}</code>	udává protokol použitý pro vyslání dotazu (např. <code>http</code>)
<code>{URI-host}</code>	specifikace domainového jména nebo IP adresy PLC systému (např. <code>192.168.134.176</code>)
<code>{resource-path}</code>	definuje cestu ke konkrétnímu zdroji (službě) (např. <code>TecoApi/GetObject</code>)
<code>{query-string}</code>	parametry dotazu (např. specifikace požadovaného objektu) závisí na použité službě

Hlavička HTTP dotazu

Hlavička HTTP dotazu (HTTP request message header) obsahuje:

- požadovanou HTTP metodu – `TecoApi` podporuje metody `GET` a `PUT`
- další volitelné položky v závislosti na typu služby a způsobu použití například položka `{ Authorization: Basic am1lbm86aGVzbG8= }` slouží k autorizaci klienta na serveru nebo položka `Content-Type: application/json` udává, že tělo HTTP dotazu obsahuje data ve formátu JSON

Tělo HTTP dotazu

Tělo HTTP dotazu (HTTP request message body) – je využito u služeb používajících metodu `PUT` a přenáší se v něm parametry služby. `TecoApi` pracuje pouze s parametry ve formátu JSON.

Hlavička HTTP odpovědi

Hlavička HTTP odpovědi (HTTP response message header) obsahuje:

- HTTP status kód – kód 2xx pro úspěšně vykonanou službu nebo kódy 4xx resp. 5xx pro chybové odpovědi

- další volitelné položky v závislosti na typu služby – např. položka `Content-Type: application/json` udává, že tělo HTTP odpovědi obsahuje data ve formátu JSON

Tělo HTTP odpovědi

Tělo HTTP odpovědi (HTTP response message body) obsahuje data požadovaná metodou `GET`. `TecoApi` vrací data v odpovědi výhradně ve formátu JSON.

Příklad komunikace

```
GET /TecoApi/GetInfo HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (Fx2 CP2080I v1.1 PP 0004)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 428
Connection: close
```

```
{
  "api":
  {
    "version": "TecoApi v1.0.0"
  },
  "plc":
  {
    "family": "Fx2",
    "type": "CP2080",
    "serialNumber": "PP 0004",
    "version": "Fx2 CP2080I v1.1",
    "mac": "F8-DC-7A-11-F4-DC",
    "mode": "RUN ",
    "userProg":
    {
      "name": "DOC_TECOAPI v1.0",
      "compiled": "2018-01-18T16:39:38.000Z",
      "stamp": "E28AE29B"
    },
    "time": "2018-01-31T16:50:39.949Z"
  }
}
```

2 PODMÍNKY PRO POUŽITÍ TECOAPI

Web server PLC systémů Teco je vybaven API rozhraním (`TecoApi`), kde poskytuje služby umožňující číst a zapisovat objekty z/do PLC systému pomocí HTTP protokolu. Objektem, který je dostupný přes `TecoApi`, může být libovolná proměnná v PLC programu.

Pro přístup k objektům v PLC přes `TecoApi` rozhraní musí být splněny následující podmínky:

PLC program

- všechny proměnné, které mají být viditelné jako objekty přes `TecoApi` rozhraní, musí být označeny direktivou `{PUBLIC_API}`

PLC projekt

- v prostředí Mosaic musí být zapnuta volby:
 - Manažer projektu
 - Exportní soubory
 - Posílání souborů do PLC
 - Automaticky posílat novější soubory do PLC
- v projektu musí být alespoň jedna web stránka vytvořená nástrojem `WebMaker`
- v nástroji `WebMaker` musí být nastaveno alespoň jedno přístupové jméno+heslo pro službu `TecoApi`

PLC systém

- v PLC systémech TC700 a Foxtrot řady CP-1xxx musí být osazena SD karta
- PLC systémy TC700 a Foxtrot řady CP-1xxx musí mít verzi firmware procesorové jednotky alespoň v10.4

Prostředí Mosaic

- prostředí Mosaic musí být alespoň v2018.1

Příklad

```
VAR_GLOBAL
plcName {PUBLIC_API} : STRING[10] := 'myFoxtrot';
plcType {PUBLIC_API} : UINT       := 2005;
plcDesc {PUBLIC_API} : STRING     := 'for testing only';
END_VAR

PROGRAM prgMain
VAR
END_VAR

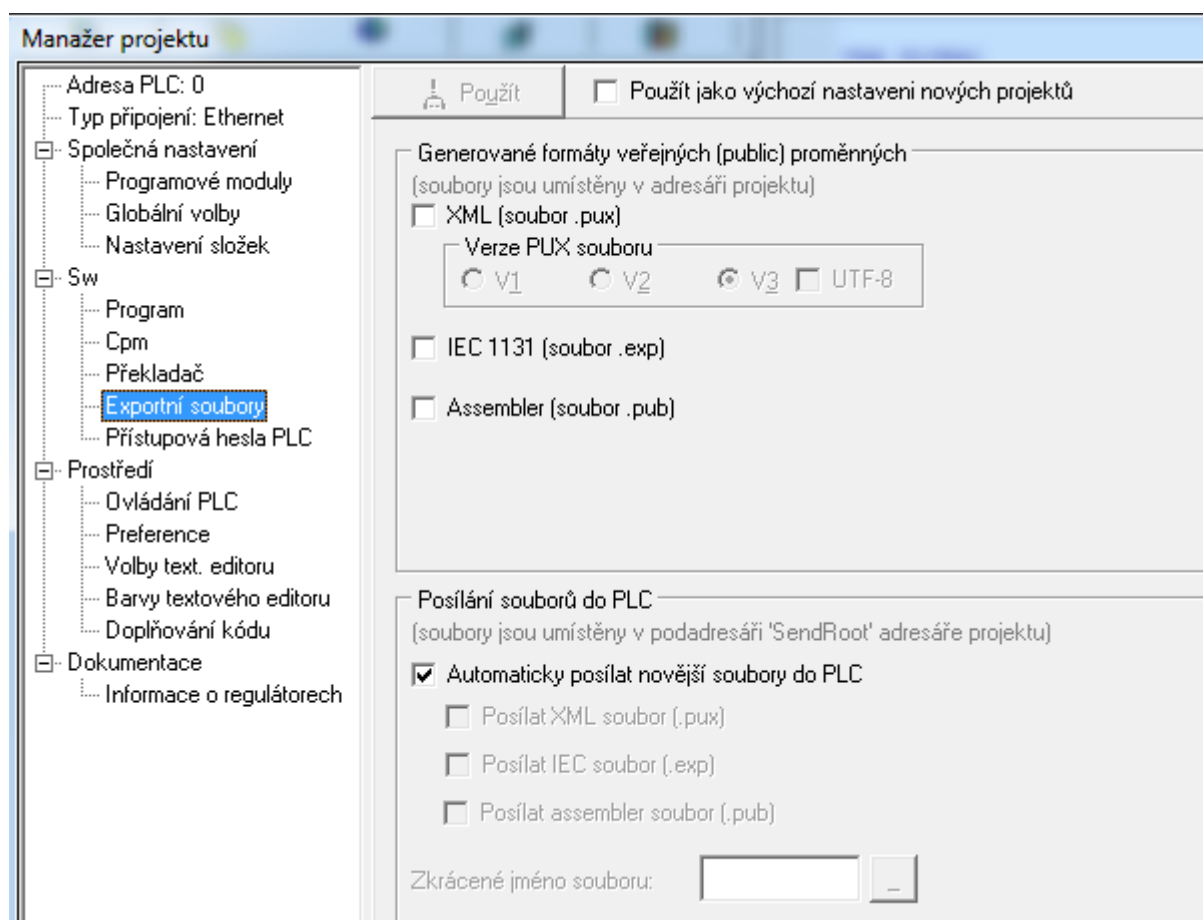
END_PROGRAM
```

Proměnné z PLC programu budou přes `TecoApi` vidět následovně:

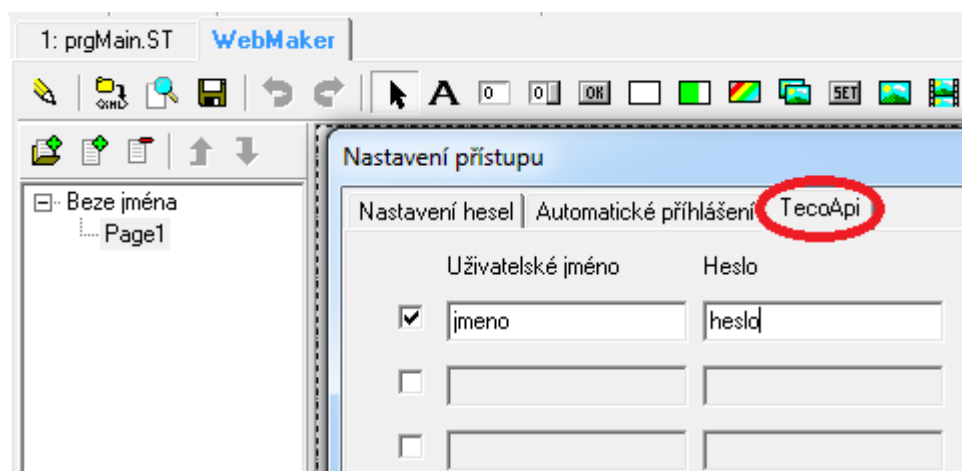
```
{
  "plcName": "myFoxtrot",
  "plcType": 2005,
  "plcDesc": "for testing only"
}
```

PLC projekt

- v prostředí Mosaic musí být zapnuty následující volby:



- v nástroji **WebMaker** musí být nastaveno alespoň jedno přístupové jméno+heslo pro službu **TecoApi**



3 SLUŽBY TECOAPI

Web server PLC systémů Teco je vybaven API rozhraním (`TecoApi`), kde poskytuje následující služby:

- `GetInfo` - získat informace o připojeném PLC systému
- `GetList` - načíst seznam dostupných objektů
- `GetObject` - načíst objekt z PLC
- `SetObject` - zapsat novou hodnotu do objektu
- `PutObject` - zapsat novou hodnotu do objektu

URI pro odeslání dotazu je konstruovaná následovně:

```
http://192.168.134.176/TecoApi/GetInfo
```

kde

<code>http</code>	Použitý protokol
<code>192.168.134.176</code>	Specifikace IP adresy PLC systému
<code>TecoApi</code>	Přístupový bod rozhraní <code>TecoApi</code>
<code>GetInfo</code>	Požadovaná služba

3.1 Autentizace

Všechny služby `TecoApi` vyžadují autentizaci, což je proces ověření identity uživatele nebo původce zprávy. Server poskytující služby `TecoApi` podporuje následující metody autentizace:

- Basic Access Authentication
- Digest Access Authentication

Obě autentizační metody jsou popsány v RFC 2617 (<https://tools.ietf.org/html/rfc2617>).

Basic access authentication

Basic access authentication je jednoduché ověření přístupu, kde se autentizační údaje (jméno a heslo) posílají v HTTP dotazu jako textový řetězec, který je zakódován metodou `Base64`. Smyslem kódování je umožnit přenos jmen a hesel obsahujících znaky, které nepatří do množiny povolených znaků pro HTTP protokol. Nejedná se tedy o kryptografické zabezpečení, dekodování lze provést velmi snadno a rychle. A to je důvod, proč by se tato metoda měla používat jen výjimečně a pouze pro testovací účely.

Tato metoda je použita ve většině příkladů v tomto dokumentu, protože komunikace jsou kratší a jednodušší než při použití Digest autentizace, což přispívá k čitelnosti a přehlednosti textu.

Digest Access Authentication

Digest Access Authentication je metoda, při které se heslo nezasílá v otevřené formě. Místo hesla se v HTTP dotazu zasílá `MD5` kontrolní součet jména, hesla a řetězce, který byl vydán serverem v rámci požadavku na autentizaci (zjednodušeně řečeno). Tato metoda by měla být při používání služeb `TecoApi` preferovaná.

Příklad

Komunikaci s Digest autentizací vyvoláme např. příkazem

```
curl --digest -u jmeno:heslo "http://192.168.134.178/TecoApi/GetObject?plcDesc"
```

Program `curl` pošle nejprve dotaz bez autentizačních údajů

```
GET /TecoApi/GetObject?plcDesc HTTP/1.1
Host: 192.168.134.178
User-Agent: curl/7.57.0
Accept: */*
```

na který odpoví `TecoApi` server zprávou s kódem `401 Unauthorized`

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest realm="TecoApi",qop="auth,auth-int",
nonce="009e441cfec37b7854fb2773",opaque="6e5eb49eff60eeae013dd1ca"
Server: TecoApi/1.0.0 (Fx2 CP2080I v1.1 PP 0004)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 133
Connection: keep-alive

{
  "error": {
    "code": "401.023",
    "message": "Authentication Requested",
    "time": "2018-01-29T18:29:39.730Z"
  }
}
```

Program `curl` zopakuje dotaz, tentokrát s autentizačními údaji

```
GET /TecoApi/GetObject?plcDesc HTTP/1.1
Host: 192.168.134.178
Authorization: Digest username="jmeno",realm="TecoApi",
nonce="009e441cfec37b7854fb2773",uri="/TecoApi/GetObject?plcDesc",
cnonce="f5ea8eeb141ee3580ef9909423bfccef",nc=00000001,
response="3b6179c5f7a93f1cb0d7435e1cedc812",qop="auth",
opaque="6e5eb49eff60eeae013dd1ca"
User-Agent: curl/7.57.0
Accept: */*
```

a server ověří zaslané údaje a vrátí požadovaná data

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (Fx2 CP2080I v1.1 PP 0004)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 39
Connection: close

{
  "plcDesc": "for testing only"
}
```


3.2 Služba GetInfo

Služba `GetInfo` vrátí informace o PLC systému, na kterém běží rozhraní `TecoApi`

Dotaz

Metoda	Požadovaná URI								
GET	<code>http://192.168.134.176/TecoApi/GetInfo</code>								
	<table> <tr> <td><code>http</code></td><td>Použitý protokol</td></tr> <tr> <td><code>192.168.134.176</code></td><td>Specifikace IP adresy PLC systému</td></tr> <tr> <td><code>TecoApi</code></td><td>Přístupový bod rozhraní <code>TecoApi</code></td></tr> <tr> <td><code>GetInfo</code></td><td>Požadovaná služba</td></tr> </table>	<code>http</code>	Použitý protokol	<code>192.168.134.176</code>	Specifikace IP adresy PLC systému	<code>TecoApi</code>	Přístupový bod rozhraní <code>TecoApi</code>	<code>GetInfo</code>	Požadovaná služba
<code>http</code>	Použitý protokol								
<code>192.168.134.176</code>	Specifikace IP adresy PLC systému								
<code>TecoApi</code>	Přístupový bod rozhraní <code>TecoApi</code>								
<code>GetInfo</code>	Požadovaná služba								

URI parametry

Žádné

Hlavička dotazu

Kromě níže uvedených položek může hlavička dotazu obsahovat další standardní položky odpovídající specifikaci protokolu HTTP/1.1

Položka	Popis
<code>Authorization</code>	Povinná položka.
<code>Accept</code>	Volitelná položka. Udává typ dat v odpovědi. Pokud je uvedena musí mít hodnotu <code>*/*</code> nebo <code>application/json</code> Rozhraní <code>TecoApi</code> podporuje pouze formát JSON

Tělo dotazu

Žádné

Příklad dotazu

```
GET /TecoApi/GetInfo HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

Odpověď

Odpověď obsahuje HTTP status kód, hlavičku odpovědi a tělo odpovědi.

Status kód

V případě úspěchu vrátí služba kód `200 OK`. V případě neúspěchu vrátí služba chybový kód `4xx` nebo `5xx` (viz kap.3.7 Přehled chyb).

Hlavička odpovědi

Kromě níže uvedených položek může hlavička odpovědi obsahovat další standardní položky odpovídající specifikaci protokolu HTTP/1.1

Položka	Popis
Content-Type	Typ dat v těle odpovědi (application/json)
Content-Length	Délka dat v těle odpovědi

Tělo odpovědi

Obsahuje informace o připojeném PLC systému:

Položka	Popis
api	Informace o API
.version	Název a verze API rozhraní
plc	Informace o PLC systému
.family	Rodina PLC systému
.type	Typ centrální jednotky PLC systému
.serialNumber	Sériové číslo centrální jednotky PLC systému
.version	Verze centrální jednotky PLC systému
.mac	MAC adresa PLC
.mode	Aktuální režim centrální jednotky PLC systému
	RUN PLC běží (provádí uživatelský program)
	HALT PLC stojí (řízení je zastaveno, program se nevykonává)
	ERR PLC hlásí závažnou chybu
.userProg	Informace o uživatelském programu
.name	Název uživatelského programu v PLC
.compiled	Datum a čas překladu uživatelského programu
.stamp	Identifikační razítko programu
.time	Aktuální datum a čas PLC systému

Příklad odpovědi

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 426

{
  "api":
  {
    "version": "TecoApi v1.0.0"
  },
  "plc":
  {
    "family": "sim",
    "type": "CP-W32",
    "serialNumber": "VN 1957",
    "version": "sim CP-W32I v1.1",
    "mac": "00-15-AF-CB-4E-7D",
    "mode": "RUN ",
    "userProg":
    {
      "name": "DOC_TECOAPI v1.0",
      "compiled": "2018-01-18T16:36:46.000Z",
      "stamp": "919E31FA"
    },
    "time": "2018-01-17T16:58:48.859Z"
  }
}
```

3.3 Služba GetList

Služba `GetList` vrátí seznam objektů dostupných přes rozhraní `TecoApi`. Objektem je každá proměnná v PLC programu, která má ve své deklaraci uvedenou direktivu `{PUBLIC_API}`.

Dotaz

Metoda	Požadovaná URI								
GET	<code>http://192.168.134.176/TecoApi/GetList?listType=struct</code>								
	<table> <tr> <td><code>http</code></td><td>Použitý protokol</td></tr> <tr> <td><code>192.168.134.176</code></td><td>Specifikace IP adresy PLC systému</td></tr> <tr> <td><code>TecoApi</code></td><td>Přístupový bod rozhraní <code>TecoApi</code></td></tr> <tr> <td><code>GetList</code></td><td>Požadovaná služba</td></tr> </table>	<code>http</code>	Použitý protokol	<code>192.168.134.176</code>	Specifikace IP adresy PLC systému	<code>TecoApi</code>	Přístupový bod rozhraní <code>TecoApi</code>	<code>GetList</code>	Požadovaná služba
<code>http</code>	Použitý protokol								
<code>192.168.134.176</code>	Specifikace IP adresy PLC systému								
<code>TecoApi</code>	Přístupový bod rozhraní <code>TecoApi</code>								
<code>GetList</code>	Požadovaná služba								

URI parametry

<code>listType</code>	Volitelný parametr Určuje formu v jaké je vrácen seznam objektů <code>listType=struct</code> seznam objektů je vrácen jako struktura <code>listType=array</code> seznam objektů je vrácen jako pole Pokud není <code>listType</code> uveden, seznam se vrátí jako struktura
-----------------------	---

Hlavička dotazu

Kromě níže uvedených položek může hlavička dotazu obsahovat další standardní položky odpovídající specifikaci protokolu HTTP/1.1

Položka	Popis
<code>Authorization</code>	Povinná položka.
<code>Accept</code>	Volitelná položka Pokud je uvedena musí mít hodnotu <code>*/*</code> nebo <code>application/json</code> Rozhraní <code>TecoApi</code> podporuje pouze formát JSON

Tělo dotazu

Žádné

Příklad dotazu

```
GET /TecoApi/GetList HTTP/1.1
Host: 192.168.134.176
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

Odpověď

Odpověď obsahuje HTTP status kód, hlavičku odpovědi a tělo odpovědi.

Status kód

V případě úspěchu vrátí služba kód `200 OK`. V případě neúspěchu vrátí služba chybový kód `4xx` nebo `5xx` (viz kap.3.7 Přehled chyb).

Hlavička odpovědi

Kromě níže uvedených položek může hlavička odpovědi obsahovat další standardní položky odpovídající specifikaci protokolu HTTP/1.1

Položka	Popis
Content-Type	Udává typ dat v těle odpovědi <code>application/json</code> .
Content-Length	Udává délku dat v těle odpovědi

Tělo odpovědi

Obsahuje seznam dostupných objektů ve formátu JSON. Seznam je organizován jako struktura nebo jako pole v závislosti na parametru `listType` .

Příklad PLC programu

```
VAR_GLOBAL
plcName {PUBLIC_API} : STRING[10] := 'myFoxtrot';
plcType {PUBLIC_API} : UINT      := 2005;
plcDesc {PUBLIC_API} : STRING    := 'for testing only';
END_VAR
```

Příklady odpovědi

```
GET /TecoApi/GetList HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 59
Connection: close

{
  "plcName": {},
  "plcType": {},
  "plcDesc": {}
}
```

```
GET /TecoApi/GetList?listType=array HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 71
Connection: close

[
  { "plcName": {} },
  { "plcType": {} },
  { "plcDesc": {} }
]
```

3.4 Služba GetObject

Služba `GetObject` načte objekt nebo několik objektů z PLC. Jména objektů se udávají jako parametr služby. Objektem může být libolná proměnná v PLC programu, která má ve své deklaraci uvedenou direktivu `{PUBLIC_API}`.

Dotaz

Metoda	Požadovaná URI								
GET	<code>http://192.168.134.176/TecoApi/GetObject?name1&name2&...</code>								
	<table> <tr> <td><code>http</code></td><td>Použitý protokol</td></tr> <tr> <td><code>192.168.134.176</code></td><td>Specifikace IP adresy PLC systému</td></tr> <tr> <td><code>TecoApi</code></td><td>Přístupový bod rozhraní <code>TecoApi</code></td></tr> <tr> <td><code>GetObject</code></td><td>Požadovaná služba</td></tr> </table>	<code>http</code>	Použitý protokol	<code>192.168.134.176</code>	Specifikace IP adresy PLC systému	<code>TecoApi</code>	Přístupový bod rozhraní <code>TecoApi</code>	<code>GetObject</code>	Požadovaná služba
<code>http</code>	Použitý protokol								
<code>192.168.134.176</code>	Specifikace IP adresy PLC systému								
<code>TecoApi</code>	Přístupový bod rozhraní <code>TecoApi</code>								
<code>GetObject</code>	Požadovaná služba								

URI parametry

`name1&...` Povinný parametr
Musí být uvedeno alespoň jedno jméno proměnné
Jména se oddělují znakem `&`
Celková délka dotazu může být max. 256 znaků
Velikost písmen ve jménech nehraje roli
`name1&name2&...` jsou jména proměnných v PLC

Hlavička dotazu

Položka	Popis
<code>Authorization</code>	Povinná položka.
<code>Accept</code>	Volitelná položka Pokud je uvedena musí mít hodnotu <code>*/*</code> nebo <code>application/json</code> Rozhraní <code>TecoApi</code> podporuje pouze formát JSON

Tělo dotazu

Žádné

Příklad dotazu

```
GET /TecoApi/GetObject?plcName&plcType HTTP/1.1
Host: 192.168.134.176
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

Odpověď

Odpověď obsahuje HTTP status kód, hlavičku odpovědi a tělo odpovědi.

Status kód

V případě úspěchu vrátí služba kód `200 OK`. V případě neúspěchu vrátí služba chybový kód `4xx` nebo `5xx` (viz kap.3.7 Přehled chyb).

Hlavička odpovědi

Kromě níže uvedených položek může hlavička odpovědi obsahovat další standardní položky odpovídající specifikaci protokolu HTTP/1.1

Položka	Popis
Content-Type	Typ dat v těle odpovědi (application/json)
Content-Length	Délka dat v těle odpovědi

Tělo odpovědi

Obsahuje seznam požadovaných objektů ve formátu JSON.

Příklad PLC programu

```
VAR_GLOBAL
  plcName {PUBLIC_API} : STRING[10] := 'myFoxtrot';
  plcType {PUBLIC_API} : UINT       := 2005;
  plcDesc {PUBLIC_API} : STRING      := 'for testing only';
END_VAR
```

Příklady odpovědi

```
GET /TecoApi/GetObject?plcName HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 32
Connection: close
```

```
{
  "plcName": "myFoxtrot"
}
```

```
GET /TecoApi/GetObject?plcName&plcType&plcDesc HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 86
Connection: close
```

```
{
  "plcName": "myFoxtrot",
  "plcType": 2005,
  "plcDesc": "for testing only"
}
```

3.5 Služba SetObject

Služba `SetObject` nastaví novou hodnotu objektu v PLC. Službu lze použít i pro nastavení několika hodnot současně. Nastavení ve tvaru `jméno=hodnota` se udávají jako parametry služby. Objektem může být libovolná proměnná v PLC programu, která má ve své deklaraci uvedenou direktivu `{PUBLIC_API}`. Novou hodnotou objektu se v tomto kontextu míní nová hodnota proměnné v PLC.

Dotaz

Metoda	Požadovaná URI
GET	<code>http://192.168.134.176/TecoApi/SetObject?name1=val1&...</code>
	<div> <div>http</div> <div>192.168.134.176</div> <div>TecoApi</div> <div>SetObject</div> </div> <div> <div>Použitý protokol</div> <div>Specifikace IP adresy PLC systému</div> <div>Přístupový bod rozhraní <code>TecoApi</code></div> <div>Požadovaná služba</div> </div>

URI parametry

<code>name1=val1</code>	<p>Povinný parametr</p> <p>Musí být uvedeno alespoň jedno nastavení <code>jméno=hodnota</code></p> <p>Jednotlivá nastavení jsou oddělená znakem <code>&</code></p> <p>Celková délka dotazu může být max. 256 znaků</p> <p>Velikost písmen ve jménech nehraje roli</p> <p><code>name1</code> je jméno proměnné v PLC</p> <p><code>val1</code> je hodnota, kterou chceme do proměnné zapsat</p>
-------------------------	---

Hlavička dotazu

Následující tabulka popisuje povinné a volitelné položky hlavičky dotazu.

Položka	Popis
<code>Authorization</code>	Povinná položka
<code>Accept</code>	Volitelná položka
	Pokud je uvedena musí mít hodnotu <code>*/*</code> nebo <code>application/json</code>
	Rozhraní <code>TecoApi</code> podporuje pouze formát JSON

Tělo dotazu

Žádné

Příklad dotazu

```
GET /TecoApi/SetObject?plcType=1005 HTTP/1.1
Host: 192.168.134.176
Authorization: Basic am1lbn86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

Odpověď

Odpověď obsahuje HTTP status kód a hlavičku odpovědi.

Status kód

V případě úspěchu vrátí služba kód `200 OK`. V případě neúspěchu vrátí služba chybový kód `4xx` nebo `5xx` (viz kap.3.7 Přehled chyb).

Hlavička odpovědi

Hlavička odpovědi může obsahovat standardní položky odpovídající specifikaci protokolu HTTP/1.1

Tělo odpovědi

Žádné

Příklad PLC programu

```
VAR_GLOBAL
  plcName   {PUBLIC_API} : STRING[10] := 'myFoxtrot';
  plcType   {PUBLIC_API} : UINT      := 2005;
  plcDesc   {PUBLIC_API} : STRING    := 'for testing only';
END_VAR
```

Příklady odpovědi

```
GET /TecoApi/SetObject?plcType=1005 HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
```

```
GET /TecoApi/SetObject?plcType=1005&plcDesc="testing is funny" HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
```


3.6 Služba PutObject

Služba `PutObject` nastaví novou hodnotu objektu v PLC. Službu lze použít i pro nastavení několika hodnot současně. Nastavení se udávají ve formátu JSON v těle dotazu. Objektem může být libovolná proměnná v PLC programu, která má ve své deklaraci uvedenou direktivu `{PUBLIC_API}`. Novou hodnotou objektu se v tomto kontextu míní nová hodnota proměnné v PLC.

Dotaz

Metoda	Požadovaná URI
PUT	<code>http://192.168.134.176/TecoApi/PutObject</code>
	<div> <div>http</div> <div>192.168.134.176</div> <div>TecoApi</div> <div>PutObject</div> </div> <div> Použitý protokol Specifikace IP adresy PLC systému Přístupový bod rozhraní <code>TecoApi</code> Požadovaná služba </div>

URI parametry

Žádné

Hlavička dotazu

Následující tabulka popisuje povinné a volitelné položky hlavičky dotazu.

Položka	Popis
Authorization	Povinná položka
Accept	Volitelná položka
	Pokud je uvedena musí mít hodnotu <code>*/*</code> nebo <code>application/json</code>
	Rozhraní <code>TecoApi</code> podporuje pouze formát JSON

Tělo dotazu

Nastavení nových hodnot ve formátu JSON.

Například nastavení hodnoty `1005` do proměnné `plcType` bude vypadat následovně:

```
{
  "plcType": 1005
}
```

JSON parser v `TecoApi` nepokrývá kompletně všechny možnosti zápisu JSON dat v těle dotazu. Omezení jsou následující:

- V objektech string jsou podporovány pouze ASCII znaky
- Není podporován zápis čísla ve vědecké notaci (např. `1.5E3`)
- Maximální délka názvu položky je 64 znaků
- Maximální délka hodnoty položky je 255 znaků
- Vnoření struktur je podporováno do úrovně 32 (Položka pole „spotřebuje“ 2 úrovně vnoření)

Příklad dotazu

```
PUT /TecoApi/PutObject HTTP/1.1
Host: 192.168.134.99
```

```
Authorization: Basic am1lbm86aGVzbG8=  
User-Agent: curl/7.57.0  
Accept: */*  
Content-Length: 19  
Content-Type: application/x-www-form-urlencoded  
  
{ "plcType": 1005 }
```

Odpověď

Odpověď obsahuje HTTP status kód a hlavičku odpovědi.

Status kód

V případě úspěchu vrátí služba kód `204 No Content`. V případě neúspěchu vrátí služba chybový kód `4xx` nebo `5xx` (viz kap.3.7 Přehled chyb).

Hlavička odpovědi

Hlavička odpovědi může obsahovat standardní položky odpovídající specifikaci protokolu HTTP/1.1

Tělo odpovědi

Žádné

Příklad PLC programu

```
VAR_GLOBAL  
  plcName {PUBLIC_API} : STRING[10] := 'myFoxtrot';  
  plcType {PUBLIC_API} : UINT       := 2005;  
  plcDesc {PUBLIC_API} : STRING      := 'for testing only';  
END_VAR
```

Příklady odpovědi

```
PUT /TecoApi/PutObject HTTP/1.1  
...  
{  
  "plcType": 1005  
}
```

```
HTTP/1.1 204 No Content  
Content-Length: 0  
Connection: close
```

```
PUT /TecoApi/PutObject HTTP/1.1  
...  
{  
  "plcType": 1005,  
  "plcDesc": "testing is funny"  
}
```

```
HTTP/1.1 204 No Content  
Content-Length: 0  
Connection: close
```

3.7 Přehled chyb

Všechny služby **TecoApi** vrací standardní HTTP status kód v hlavičce odpovědi (viz <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>).

Odpovědi na úspěšně provedený dotaz obsahují kód **200 OK** nebo **204 No Content**. Chybové odpovědi vrací kód **4xx** nebo **5xx**. Podrobnější informace o vzniklé chybě jsou uvedeny v těle odpovědi. Například:

```
{
  "error": {
    "code": "501.013",
    "message": "Unimplemented service 'HTTP GET | GetData'",
    "time": "2018-01-26T15:38:39.453Z"
  }
}
```

Jednotlivé položky v těle chybové odpovědi mají následující význam:

Položka	Popis
error	Chyba
.code	Kód chyby ve tvaru XXX.YYY XXX ...HTTP status kód YYY ...Teco chybový kód
.message	Textový popis chyby
.time	Datum a čas vzniku chyby

Příklad

Příkazem `curl -u jmeno:heslo "http://192.168.134.99/TecoApi/GetData"`

vyvoláme požadavek na neexistující **TecoApi** službu

```
GET /TecoApi/GetData HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

na který přijde odpověď obsahující informace o chybě

```
HTTP/1.1 501 Not Implemented
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Type: application/json
Content-Length: 151
Connection: close

{
  "error": {
    "code": "501.013",
    "message": "Unimplemented service 'HTTP GET | GetData'",
    "time": "2018-01-26T16:00:18.609Z"
  }
}
```

Seznam chyb

Chybový kód Popis

400.001	Bad parameters (any PLC variable is not specified) v URI dotazu není specifikována žádná proměnná
413.002	Too many variables in one request (requested: XXX, allowed: YYY) v URI dotazu je specifikováno příliš mnoho proměnných XXX počet požadovaných proměnných YYY maximální možný počet proměnných v jedné odpovědi Foxtrot CP-1xxx max 128 proměnných, Foxtrot CP-2xxx max 256 proměnných
400.003	Name of variable is empty v URI dotazu je prázdné jméno proměnné
413.004	Too many variables (maximum is YYY) požadovaný počet položek překročil povolené maximum YYY maximální možný počet proměnných v jedné odpovědi Foxtrot CP-1xxx max 128 proměnných, Foxtrot CP-2xxx max 256 proměnných (např. proměnná myObject uvedená v kap.5.1 Strukturované proměnné obsahuje celkem 13 položek a lze jí tedy načíst jedním dotazem http://192.168.134.99/TecoApi/GetObject?myObject)
400.005	Variable 'NNN' is not found proměnnou specifikovanou v URI dotazu se nepodařilo najít NNN jméno požadované proměnné (např. proměnná NNN není v PLC programu definovaná nebo chybí direktiva {PUBLIC_API} v deklaraci proměnné)
400.006	No variable is found interní chyba serveru (prázdný seznam proměnných)
400.007	Name of variable 'NNN...' is too long Jméno proměnné je příliš dlouhé NNN... jméno požadované proměnné Foxtrot CP-1xxx max 128 znaků, Foxtrot CP-2xxx max 256 znaků
400.008	Name of item 'NNN...' is too long Jméno položky je příliš dlouhé NNN... jméno požadované proměnné Foxtrot CP-1xxx/CP-2xxx max 64 znaků
400.009	JSON syntax error (DESC) Chyba syntaxe JSON (služba PutObject) DESC specifikace chyby: JSON stack overflow Expected character '{' Expected character ':' Expected character '"' JSON stack underflow
500.010	Public file 'FFF' is not found Nenalezen public soubor FFF, který nezbytný pro služby TecoApi FFF jméno public souboru (public soubor je automaticky vytvářen při překladu programu PLC v prostředí Mosaic pokud je kdekoli v programu alespoň jedna deklarace proměnné s direktivou {PUBLIC_API}, při vyslání přeloženého programu se do PLC odešle také public soubor)

500.011 Public file 'FFF' is too big
Public soubor FFF je příliš velký (max. velikost souboru je 128 KB)
FFF jméno public souboru

Chybový kód Popis

500.012 Reading of public file 'FFF' failed
Public soubor FFF nelze přečíst
FFF jméno public souboru

501.013 Unimplemented service 'HTTP MMM | SSS'
Požadovaná služba TecoApi není implementovaná
MMM metoda (GET | PUT | ...)
SSS služba

400.014 Service is not specified
Služba TecoApi není specifikovaná

400.015 Value VVV was not written to variable 'NNN' (type of variable is TTT)
Hodnotu VVV nelze zapsat do proměnné NNN
VVV hodnota zapisovaná do proměnné NNN
NNN jméno proměnné
TTT datový typ proměnné v PLC
(např. pokud je proměnná typu USINT tak do ní lze zapsat pouze hodnoty v rozsahu od 0 do 255, zápis jiné hodnoty vyvolá tuto chybu)

400.016 Unknown value of parameter (param: 'PPP', value: 'VVV')
Nepřípustná hodnota parametru služby
PPP parametr
VVV hodnota parametru

400.017 Name of parameter is empty
v URI dotazu je prázdný název parametru

400.018 Unknown parameter (service: 'SSS', param: 'PPP')
Neznámý parametr služby
SSS služba
PPP parametr

400.019 Value of parameter is empty (service: 'SSS', param: 'PPP')
Chybí hodnota parametru služby
SSS služba
PPP parametr

401.020 New balls
Autentizační údaje jsou zastaralé, server služby TecoApi vydává nové údaje potřebné pro autentizaci
(Digest stale=true)

401.021 Unknown name or password
Chyba autentizace, neznámé jméno nebo heslo

403.022 Unsupported authentication method
Požadovaná metoda autentizace není podporovaná
(server TecoApi podporuje pouze metody Basic a Digest)

401.023 Authentication Request
Je požadovaná autentizace
(typicky při dotazu bez autentizačních údajů)

4 PROMĚNNÉ V PLC PROGRAMU A FORMÁT JSON

PLC systémy Tecomat vybavené rozhraním TecoApi se programují podle normy IEC 61131-3. Tím jsou dané datové typy, které lze v programech využívat pro deklaraci proměnných. Přehled základních datových typů uvádí následující tabulka:

Datový typ	Popis	Počet bitů	Rozsah hodnot	Hodnota v JSON
BOOL	Boolean	1	<0,1>	true, false
BYTE	Byte	8	<0,255>	number
WORD	Word	16	<0,65535>	number
DWORD	Double Word	32	<0,4294967295>	number
SINT	Short Integer	8	<-128,127>	number
INT	Integer	16	<-32768,32767>	number
DINT	Double Integer	32	<-2147483648,2147483647>	number
USINT	Unsigned Short Integer	8	<0,255>	number
UINT	Unsigned Integer	16	<0,65535>	number
UDINT	Unsigned Double Integer	32	<0,4294967295>	number
REAL	Real (single precision)	32	<±2.9E-39,±3.4E+38>	number
LREAL	Real (double precision)	64	viz IEC 60559	number
TIME	Time (duration)	32	<0,24d 20:31:23.647>	string
TOD	Time of day only	32	<0,24d 20:31:23.647>	string
DATE	Date only	64	od 1.1.1970 00:00:00.0	string
DT	Date and Time	64	od 1.1.1970 00:00:00.0	string
STRING	String	--	max. 255 ASCII znaků	string

Prezentaci hodnot proměnných ve formátu JSON ukazuje následující příklad:

PLC program

TYPE

```

TmySTRUCT : STRUCT
  varBool0      : BOOL   ;
  varBool1      : BOOL   ;
  varByte       : BYTE   ;
  varWord       : WORD   ;
  varDword      : DWORD  ;
  varUsint      : USINT  ;
  varUint       : UINT   ;
  varUdint      : UDINT  ;
  varSint       : SINT   ;
  varInt        : INT    ;
  varDint       : DINT   ;
  varReal       : REAL   ;
  varLreal      : LREAL  ;
  varTime       : TIME   ;
  varDate       : DATE   ;
  varDT         : DT     ;
  varString     : STRING ;
  varStringArray : ARRAY[0..5] OF STRING[8];
END_STRUCT;

```

END_TYPE

VAR_GLOBAL

```

plcData {PUBLIC_API} : TmySTRUCT := (
  varBool0      := FALSE,
  varBool1      := TRUE,

```

```
varByte      := 16#11,  
varWord      := 16#2222,  
varDword     := 16#33333333,  
varUsint     := 11,  
varUint      := 2222,  
varUdint     := 33333333,  
varSint      := -11,  
varInt       := -2222,  
varDint      := -33333333,  
varReal      := 1000.0,  
varLreal     := 1000000.0,  
varTime      := T#120h2m3.45s,  
varDate      := D#2015-06-17,  
varDT        := DT#2014-05-11-11:22:33.456,  
varString    := 'Hello everybody',  
varStringArray := ['zero', 'one', 'two', 'three', 'four', 'five']);
```

END_VAR

Dotaz

```
GET /TecoApi/GetObject?plcData HTTP/1.1  
Host: 192.168.134.99  
Authorization: Basic am1lbm86aGVzbG8=  
User-Agent: curl/7.57.0  
Accept: */*
```

Odpověď

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Cache-Control: no-cache  
Content-Length: 615  
Connection: close  
  
{  
  "plcData":  
  {  
    "varBool0": false,  
    "varBool1": true,  
    "varByte": 17,  
    "varWord": 8738,  
    "varDword": 858993459,  
    "varUsint": 11,  
    "varUint": 2222,  
    "varUdint": 33333333,  
    "varSint": -11,  
    "varInt": -2222,  
    "varDint": -33333333,  
    "varReal": 1000.000000,  
    "varLreal": 1000000.000000,  
    "varTime": "01:02:03.450",  
    "varDate": "2015-06-17",  
    "varDT": "2018-05-11T11:22:33.456Z",  
    "varString": "Hello everybody",  
    "varStringArray":  
    [  
      "zero", "one", "two", "three", "four", "five"  
    ]  
  }  
}
```

Formát datumu a času

Pro datum a čas se v PLC používají datové typy `DATE`, `TIME`, `TIME_OF_DAY (TOD)` a `DATE_AND_TIME (DT)`. Hodnoty proměnných těchto datových typů se ve formátu JSON předávají jako `STRING` následovně:

Datový typ	Formát v JSON	Příklad
<code>DATE</code>	<code>"YYYY-MM-DD"</code>	<code>"2015-06-17"</code>
<code>TIME</code>	<code>"hh:mm:ss.zzz"</code>	<code>"01:02:03.450"</code>
<code>TIME_OF_DAY</code>	<code>"hh:mm:ss.zzz"</code>	<code>"01:02:03.450"</code>
<code>DATE_AND_TIME</code>	<code>"YYYY-MM-DDThh:mm:ss.zzzZ"</code>	<code>"2018-05-11T11:22:33.456Z"</code>

Jména proměnných

Jména proměnných v PLC programu musí odpovídat normě IEC 61 131-3. Jméno proměnné je řetězec písmen (malých nebo velkých), číslic a podtrhovacích znaků. Jméno musí začínat písmenem nebo podtrhovacím znakem a nesmí obsahovat mezery. Znaký národních abeced (písmena s háčky a čárkami) nejsou ve jménech proměnných povoleny. Umístění podtrhovacího znaku je významné, tedy např. „BF_LM“ a „BFL_M“ jsou jména pro dvě různé proměnné. Více podtrhovacích znaků za sebou není povoleno, velikost písmen nehraje roli. Maximální délka jména proměnné je 64 znaků.

5 PŘÍKLADY

Pro testování TecoApi lze použít webový prohlížeč nebo např. program curl (viz <https://curl.haxx.se/>).

5.1 Strukturované proměnné

Program v PLC

```
TYPE
  TmyFavorites : STRUCT
    color      : STRING[16];
    sport      : STRING[24];
    food       : STRING[32];
  END_STRUCT;

  TmyObject    : STRUCT
    first      : STRING[32];
    last       : STRING[64];
    age        : USINT;
    sex        : STRING[1];
    salary     : UDINT;
    married    : BOOL;
    registered  : BOOL;
    interests  : ARRAY[0..2] OF STRING[24];
    favorites  : TmyFavorites;
  END_STRUCT;
END_TYPE

VAR_GLOBAL
  myObject {PUBLIC_API} : TmyObject;
END_VAR

PROGRAM prgTestApi
  VAR
    init      : BOOL;
  END_VAR

  IF NOT init THEN
    myObject.first      := 'John';
    myObject.last       := 'Doe';
    myObject.age        := 39;
    myObject.sex        := 'M';
    myObject.salary     := 70000;
    myObject.married    := true;
    myObject.registered := false;
    myObject.interests[0] := 'Reading';
    myObject.interests[1] := 'Mountain Biking';
    myObject.interests[2] := 'Hacking';
    myObject.favorites.color := '"Blue"';
    myObject.favorites.sport := 'Soccer';
    myObject.favorites.food  := 'Spaghetti';
    init := true;
  END_IF;
END_PROGRAM
```

Dotaz na celou strukturu

Příkazem `curl --user jmeno:heslo "http://192.168.134.99/TecoApi/GetObject?myObject"`

vyvoláme dotaz

```
GET /TecoApi/GetObject?myObject HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

na který přijde odpověď obsahující všechny položky proměnné `myObject`.

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 360
Connection: close

{
  "myObject":
  {
    "first": "John",
    "last": "Doe",
    "age": 39,
    "sex": "M",
    "salary": 70000,
    "registered": true,
    "interests":
    [
      "Reading",
      "Mountain Biking",
      "Hacking"
    ],
    "favorites":
    {
      "color": "\"Blue\"",
      "sport": "Soccer",
      "food": "Spaghetti"
    }
  }
}
```

Dotaz na část struktury

Příkaz

`curl --user jmeno:heslo "http://192.168.134.99/TecoApi/GetObject?myObject.favorites"`

vyvolá dotaz na proměnnou `myObject.favorites`

```
GET /TecoApi/GetObject?myObject.favorites HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

na který přijde odpověď obsahující pouze požadovanou část dat

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 144
Connection: close

{
  "myObject":
  {
    "favorites":
    {
      "color": "\"Blue\"",
      "sport": "Soccer",
      "food": "Spaghetti"
    }
  }
}
```

Dotaz na vybrané položky struktury

Příkaz

```
curl --user jmeno:heslo "http://192.168.134.99/TecoApi/GetObject?myObject.first&myObject.favorites.sport"
```

vyvolá dotaz na proměnné `myObject.first` a `myObject.favorites.sport`.

```
GET /TecoApi/GetObject?myObject.first&myObject.favorites.sport HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

Odpověď vypadá následovně

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 110
Connection: close

{
  "myObject":
  {
    "first": "John",
    "favorites":
    {
      "sport": "Soccer"
    }
  }
}
```

Nastavení hodnoty jedné položky ve struktuře

Předpokládejme, že chceme nastavit proměnnou `myObject.registered` na hodnotu `true`. Pokud použijeme službu `SetObject` tak bude příkaz vypadat následovně

```
curl -u jmeno:heslo "http://192.168.134.99/TecoApi/SetObject?myObject.registered=true"
```

Program `curl` vyšle dotaz

```
GET /TecoApi/SetObject?myObject.registered=true HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

PLC potvrdí zápis hodnoty následovně

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Length: 0
Connection: close
```

Nastavení celé struktury

Pro nastavení celé struktury `myObject` použijeme službu `PutObject`. Tu vyvoláme příkazem

```
curl -T myObject.json -u jmeno:heslo "http://192.168.134.99/TecoApi/PutObject?myObject"
```

kde `myObject.json` je soubor s novým nastavením, který vypadá následovně

```
{
  "myObject":
  {
    "first": "Mia",
    "last": "Miller",
    "age": 25,
    "sex": "F",
    "salary": 51000,
    "married": false,
    "registered": true,
    "interests":
    [
      "Reading",
      "Cooking",
      "Traveling"
    ],
    "favorites":
    {
      "color": "Red",
      "sport": "Swimming",
      "food": "Salad"
    }
  }
}
```

Výsledkem je dotaz

```
PUT /TecoApi/PutObject?myObject HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
Content-Length: 370
```

```
{
  "myObject":
  {
    "first": "Mia",
    "last": "Miller",
    "age": 25,
    "sex": "F",
    "salary": 51000,
    "married": false,
    "registered": true,
    "interests":
    [
      "Reading",
      "Cooking",
      "Traveling"
    ],
    "favorites":
    {
      "color": "Red",
      "sport": "Swimming",
      "food": "Salad"
    }
  }
}
```

Odpověď na tento dotaz je

```
HTTP/1.1 204 No Content
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Length: 0
Connection: close
```

5.2 Proměnné typu pole

Proměnné typu pole deklarované v PLC programu s direktivou `PUBLIC_API` musí být indexované od 0. U vícerozměrných polí platí tato podmínka pro všechny rozměry pole.

Příklad deklarace polí:

```
VAR_GLOBAL
arr1 {PUBLIC_API} : ARRAY [0..5] OF STRING[10];
arr2 {PUBLIC_API} : ARRAY [0..5, 0..9] OF REAL;
arr3 {PUBLIC_API} : ARRAY [0..1, 0..2, 0..3] OF BYTE;
END_VAR
```

V PLC programu předpokládejme následující deklarace

```
VAR_GLOBAL
arr1 {PUBLIC_API} : ARRAY [0..5] OF STRING [10] :=
    ['zero', 'one', 'two', 'three', 'four'];

arr2 {PUBLIC_API} : ARRAY [0..2, 0..3] OF REAL :=
    [ 0.0, 0.1, 0.2, 0.3,
      1.0, 1.1, 1.2, 1.3,
      2.0, 2.1, 2.2, 2.3];

arr3 {PUBLIC_API} : ARRAY [0..1, 0..2, 0..3] OF BYTE :=
    [ 000, 001, 002, 003,
      010, 011, 012, 013,
      020, 021, 022, 023,
      100, 101, 102, 103,
      110, 111, 112, 113,
      120, 121, 122, 123 ];
END_VAR
```

Příkazem

```
curl -u jmeno:heslo "http://192.168.134.99/TecoApi/GetObject?arr1"
```

vyvoláme dotaz na první pole s názvem `arr1`

```
GET /TecoApi/GetObject?arr1 HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

Odpověď vypadá následovně

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 99
Connection: close
```

```
{
  "arr1":
  [
    "zero",
    "one",
    "two",
    "three",
    "four",
    ""
  ]
}
```

Pokud chceme nastavit do `arr1[5]` hodnotu `"five"` použijeme příkaz `SetObject`

```
curl -u jmeno:heslo "http://192.168.134.99/TecoApi/SetObject?arr1[5]=five"
```

Program `curl` pošle následující dotaz

```
GET /TecoApi/SetObject?arr1[5]=five HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

a PLC potvrdí zápis nové hodnoty

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Length: 0
Connection: close
```

Příkaz pro načtení dvourozměrného pole `arr2` opět použije službu `GetObject`

```
curl -u jmeno:heslo "http://192.168.134.99/TecoApi/GetObject?arr2"
```

Do PLC je odeslán následující dotaz

```
GET /TecoApi/GetObject?arr2 HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
```

Odpověď vypadá následovně

```
HTTP/1.1 200 OK
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Type: application/json
Cache-Control: no-cache
Content-Length: 272
Connection: close

{
  "arr2":
  [
    [
```

```

    0.000000,
    0.100000,
    0.200000,
    0.300000
  ],
  [
    1.000000,
    1.100000,
    1.200000,
    1.300000
  ],
  [
    2.000000,
    2.100000,
    2.200000,
    2.300000
  ]
]
}

```

Pro zápis všech hodnot v poli `arr2` použijeme službu `PutObject`. Příkaz

```
curl -T arr2.json -u jmeno:heslo "http://192.168.134.99/TecoApi/PutObject"
```

použije pro nastavení hodnot soubor `arr2.json` s následujícím obsahem

```

{
  "arr2": [
    [ 0.00, 0.11, 0.22, 0.33 ],
    [ 11.00, 11.11, 11.22, 11.33 ],
    [ 22.00, 22.11, 22.22, 22.33 ]
  ]
}

```

Zaslaný dotaz vypadá následovně

```

PUT /TecoApi/PutObject HTTP/1.1
Host: 192.168.134.99
Authorization: Basic am1lbm86aGVzbG8=
User-Agent: curl/7.57.0
Accept: */*
Content-Length: 211

{
  "arr2": [
    [ 0.00, 0.11, 0.22, 0.33 ],
    [ 11.00, 11.11, 11.22, 11.33 ],
    [ 22.00, 22.11, 22.22, 22.33 ]
  ]
}

```

Kladná odpověď má HTTP status `204 No Content`

```

HTTP/1.1 204 No Content
Server: TecoApi/1.0.0 (sim CP-W32I v1.1 VN 1957)
Content-Length: 0
Connection: close

```


5.3 Komplexní proměnné

Předpokládejme, že aplikace využívající `TecoApi`, potřebuje z PLC získat nejen informaci o aktuální hodnotě proměnné, ale i další informace o tom, jak lze s proměnnou zacházet. Například jestli je daná proměnná pouze pro čtení, pokud je možno do proměnné i zapisovat tak jaký je přípustný rozsah zapisovaných hodnot, jaké inženýrské jednotky mají být zobrazeny, atd. V tomto případě je vhodné založit pro každou proměnnou dostupnou přes `TecoApi` komplexní typ, který bude obsahovat vše potřebné. Ten může vypadat např. následovně:

```

TYPE
  T_ApiObjectAttr : STRUCT
    enable       : BOOL;
    visibility    : BOOL;
    readOnly     : BOOL;
    title        : STRING[32];
    minValue     : REAL;
    maxValue     : REAL;
    units        : STRING[16];
  END_STRUCT

  T_ApiObject : STRUCT
    value       : REAL;
    attr        : T_ApiObjectAttr;
  END_STRUCT
END_TYPE

```

Hodnota proměnné bude v položce `value`, všechny ostatní vlastnosti budou uvedeny ve struktuře `attr`, kterou zpravidla stačí jednou nastavit v deklaraci proměnné:

```

PROGRAM prgMain
  VAR
    myObject1 {PUBLIC_API} : T_ApiObject :=
      ( attr := ( enable       := true,
                  visibility    := true,
                  readOnly     := true,
                  title        := 'outside temperature',
                  units        := '°C' ));

    myObject2 {PUBLIC_API} : T_ApiObject :=
      ( attr := ( enable       := true,
                  visibility    := true,
                  readOnly     := true,
                  title        := 'inside temperature',
                  units        := '°C' ));

    myObject3 {PUBLIC_API} : T_ApiObject :=
      ( attr := ( enable       := true,
                  visibility    := true,
                  readOnly     := false,
                  title        := 'requested temperature',
                  minValue     := 10.0,
                  maxValue     := 30.0,
                  units        := '°C' ));
  END_VAR
END_PROGRAM

```

Načtení uvedených proměnných službou `GetObject` získá aplikace následující JSON

```
{
  "Main":
  {
    "myObject1":
    {
      "value": -0.500000,
      "attr":
      {
        "enable": true,
        "visibility": true,
        "readOnly": true,
        "title": "outside temperature",
        "minValue": 0.000000,
        "maxValue": 0.000000,
        "units": "°C"
      }
    },
    "myObject2":
    {
      "value": 22.100000,
      "attr":
      {
        "enable": true,
        "visibility": true,
        "readOnly": true,
        "title": "inside temperature",
        "minValue": 0.000000,
        "maxValue": 0.000000,
        "units": "°C"
      }
    },
    "myObject3":
    {
      "value": 23.000000,
      "attr":
      {
        "enable": true,
        "visibility": true,
        "readOnly": false,
        "title": "requested temperature",
        "minValue": 10.000000,
        "maxValue": 30.000000,
        "units": "°C"
      }
    }
  }
}
```

Ten se pak použije pro vlastní zobrazení hodnot proměnných a jejich případnou editaci.