

# ***LookDet***

## ***Dispečerský systém***

Programátorská příručka

*Verze 1.03*



AMiT Automation s.r.o. nepřijímá žádné záruky, pokud se týče obsahu této publikace a vyhrazuje si právo měnit obsah dokumentace bez závazku tyto změny oznámit jakékoli osobě či organizaci.

Tento dokument může být kopírován a rozšiřován za následujících podmínek:

1. Celý text musí být kopírován bez úprav a se zahrnutím všech stránek.
2. Všechny kopie musí obsahovat označení autorského práva společnosti AMiT Automation s.r.o. a veškerá další upozornění v dokumentu uvedená.
3. Tento dokument nesmí být distribuován za účelem dosažení zisku.

V publikaci použité názvy produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

AMiT je registrovaná ochranná známka.

**Copyright (c) 2025, AMiT Automation s.r.o.**

**Výrobce: AMiT Automation s.r.o.**

**Radlická 740/113c, 158 00 Praha 5**

**amitautomation.cz**

**Technická podpora: [support@amit-automation.cz](mailto:support@amit-automation.cz)**

## Obsah

|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
|          | Historie revizí .....                                      | 4         |
|          | Související dokumentace .....                              | 4         |
| <b>1</b> | <b>Úvod .....</b>                                          | <b>5</b>  |
| <b>2</b> | <b>Funkce v náhledu .....</b>                              | <b>6</b>  |
| 2.1      | callBacks .....                                            | 7         |
| 2.2      | Příklady použití .....                                     | 7         |
| 2.3      | Vložení funkce do náhledu .....                            | 9         |
| 2.3.1    | Sekce pro vložení JavaScriptu .....                        | 9         |
| 2.3.2    | Import souboru s JavaScriptem .....                        | 10        |
| <b>3</b> | <b>Pluginy .....</b>                                       | <b>11</b> |
| 3.1      | Správa pluginu .....                                       | 11        |
| 3.1.1    | Export a import existujících pluginů .....                 | 12        |
|          | Export .....                                               | 12        |
|          | Import .....                                               | 12        |
| 3.2      | Backend (serverové) pluginy .....                          | 13        |
| 3.2.1    | Struktura pluginu .....                                    | 13        |
| 3.2.2    | Hlavička pluginu .....                                     | 13        |
|          | Povinné parametry hlavičky .....                           | 13        |
|          | Nepovinné parametry hlavičky .....                         | 13        |
| 3.2.3    | Objekt \$interface .....                                   | 14        |
|          | Funkce pro práci s kanály v LookDetu .....                 | 14        |
|          | Funkce pro práci s provozním deníkem ve stanici AMiT ..... | 15        |
|          | Funkce pro práci s archívem v LookDetu .....               | 17        |
|          | Funkce pro práci se stanicemi a jejich daty .....          | 17        |
|          | Souborové operace .....                                    | 20        |
|          | Ostatní funkce .....                                       | 21        |
| 3.3      | Frontend (grafické) pluginy .....                          | 21        |
| 3.3.1    | Struktura pluginu .....                                    | 21        |
| 3.3.2    | Parametry v souboru info.txt .....                         | 21        |
|          | Povinné parametry .....                                    | 22        |
|          | Nepovinné parametry hlavičky .....                         | 22        |
| 3.3.3    | Design-time .....                                          | 22        |
|          | Podsložky .....                                            | 22        |
|          | Povinné funkce .....                                       | 22        |
|          | Dostupné funkce .....                                      | 23        |
| 3.3.4    | Runtime .....                                              | 27        |
|          | Podsložky .....                                            | 27        |
|          | Funkce typu callback .....                                 | 27        |
|          | Funkce typu api .....                                      | 28        |

## Historie revizí

Jméno dokumentu: lookdet\_ms\_cz\_103.pdf

| Verze | Datum        | Autor změny    | Změny                                                     |
|-------|--------------|----------------|-----------------------------------------------------------|
| 100   | 09. 10. 2018 | Říha Z.        | Nový dokument.                                            |
| 101   | 07. 01. 2021 | Říha Z.        | Aktualizace dokumentu dle chování <b>LookDet</b> v 4.1.x. |
| 102   | 03. 11. 2022 | Říha Z.        | Aktualizace dokumentu dle chování <b>LookDet</b> v 5.0.x. |
| 103   | 05. 06. 2025 | Kalendovský O. | Aktualizace dokumentu dle chování <b>LookDet</b> v 6.0.x. |
|       |              |                |                                                           |

## Související dokumentace

1. **LookDet** – Dispečerský systém – Návod na obsluhu – Administrátor  
soubor: lookdet\_a\_g\_cz\_xxx.pdf
2. Nápověda k části PseDet, vývojového prostředí **DetStudio**  
soubor: Psedet\_cs.chm
3. <https://jquery.com> – knihovna JavaScriptu
4. JavaScript Bible, 7th Edition – Autor Danny Goodman, Nakladatelství Wiley, 2010
5. PHP and MySQL Web Development, 5th Edition – Autor Luke Welling, Laura Thomson, Nakladatelství Addison-Wesley Professional

# 1 Úvod

---

Tento dokument je rozšířením návodu na obsluhu dispečerského systému **LookDet** pro administrátora (lookdet\_a\_g\_cz\_xxx.pdf). Popisuje možnosti tvorby uživatelských funkcí a pluginů v **LookDet** 6.0.x a vyšší.

**U uživatelů tohoto dokumentu se předpokládá pokročilá znalost programování v JavaScriptu a v PHP.**

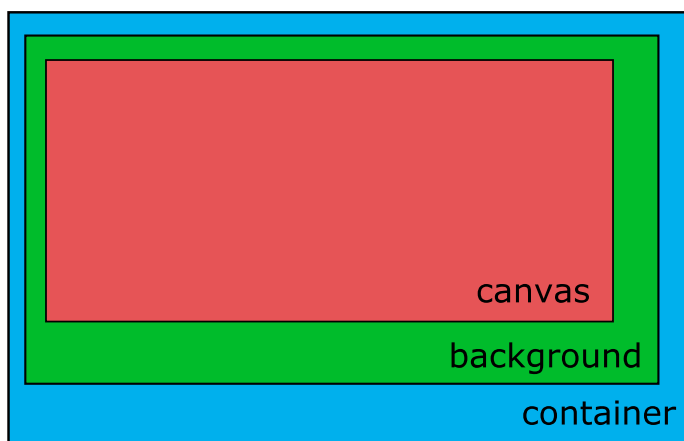
## 2 Funkce v náhledu

V sekci náhledy lze ke každému náhledu přiřadit funkci psanou v JavaScriptu. Funkce je volána po každém načtení náhledu (v JavaScriptu lze používat jQuery).

Funkce má jeden vstupní parametr „\$par“. Jedná se o objekt, který obsahuje:

- **context** – nastaven podle toho, kde je JavaScript spuštěn:
  - 0 – probíhá parametrizace SVG (v administraci),
  - 1 – v záložce,
  - 2 – v samostatném okně.
- **language** – kód jazyka uživatele (cze, eng, ...).
- **container** – DOM – DIV, ve kterém je uloženo SVG a pozadí – hlavní plocha.
- **background** – DOM – DIV, ve kterém je uloženo případné png (jpg) pozadí. Využít lze rovněž pro html5 canvas, zobrazení videa a podobně. Pozor na pořadí DOMu – viz náčrt níže.
- **canvas** – DOM – DIV, ve kterém je uložen svg náhled.
- **callbacks** – obsahuje funkce, pomocí kterých lze zaregistrovat zpětné volání při událostech:
  - **onContainerResize** – došlo ke změně velikosti náhledu,
  - **onRefresh** – byla obnovena data náhledu,
  - **onUnload** – náhled je odebírán, provést deaktivaci časovačů a podobně.
- **api** – obsahuje pomocné funkce, detailní popis funkcí viz kapitola 3.3.4 „Runtime“.
  - **getFromStation,**
  - **putToStation,**
  - **getSetDBData,**
  - **showWindow,**
  - **getSysText.**

### Struktura DOM



Obr. 1 – Struktura DOM

Z obrázku je patrné, že aby bylo možno využít DIV v pozadí (background) je třeba udělat SVG náhled TRANSPARENTNÍ (bez vyplněného pozadí), jinak pozadí překryje.

## 2.1 callBacks

Pokud chce uživatelská funkce dostávat upozornění na některou z událostí, musí si ji nejdříve registrovat. Provede to v hlavním těle JavaScriptu takto. Pokud se některá událost ve funkci nemá využívat, registrace se nemusí provádět.

```
/* při změně velikosti kontejneru */
$par.callBacks.onContainerResize({container:$par.container,callback:function(nS,nV)
{
    /* tělo funkce zpracující změnu velikosti */
    /* obsahuje 2 parametry nS a nV (nová šířka a výška náhledu) */
}});

/* při obnově dat */
$par.callBacks.onRefresh({container:$par.container,callback:function(){
    /* tělo funkce, která se provede při obnově dat náhledu */
    /* nedostává žádné parametry */
}});

/* při odstranění náhledu (zavření záložky, konec editace... */
$par.callBacks.onUnload({container:$par.container,callback:function(){
    /* tělo funkce, která se zavolá při odstranění náhledu */
    /* provedeme deaktivaci časovačů a podobně */
}});
```

## 2.2 Příklady použití

Při každé obnově náhledu se změní barva pozadí. JavaScript bude vypadat následovně:

```
/* při obnově dat změníme barvu pozadí */
$par.callBacks.onRefresh({container:$par.container,callback:function(){
    /* vygenerujeme náhodnou RGB barvu */
    var r=(Math.random()*255).toFixed(0);
    var g=(Math.random()*255).toFixed(0);
    var b=(Math.random()*255).toFixed(0);
    var nBarva="rgb("+r+","+g+","+b+") !important";
    /* pomoci jQuery změníme CSS styl pozadí na vygenerovanou barvu */
    $($par.background).css('background-color',nBarva);
}});
```

Každých 5 sekund se změní barva pozadí. JavaScript bude vypadat následovně:

```
/* aktivujeme interval, jeho identifikátor uložíme do proměnné timGenBP pro
deaktivaci */
/* interval vykovávání 5000 ms */
var timGenBP=window.setInterval(function(){
    /* vygenerujeme náhodnou RGB barvu */
    var r=(Math.random()*255).toFixed(0);
    var g=(Math.random()*255).toFixed(0);
    var b=(Math.random()*255).toFixed(0);
    var nBarva="rgb("+r+","+g+","+b+") !important";
    /* pomoci jQuery změníme CSS styl pozadí na vygenerovanou barvu */
    $($par.background).css('background-color',nBarva);
},5000);
/* protože jsme aktivovali časovač, musíme jej při deaktivaci náhledu vypnout! */
/* aktivujeme si událost při odstranění náhledu */
```

```
$par.callbacks.onUnload({container:$par.container,callback:function(){
    /* ruší se náhled - vypneme časovač změny pozadí */
    window.clearInterval(timGenBP);
}});
```

Při změně velikosti náhledu se změní barva pozadí. JavaScript bude vypadat následovně:

```
/* aktivujeme si událost při změně velikosti */
$par.callbacks.onContainerResize({container:$par.container,callback:function(sirka,
vyska){
    /* nastala změna velikosti plátna na rozměr sirka, výška */
    /* změníme barvu pozadí */
    /* vygenerujeme náhodnou RGB barvu */
    var r=(Math.random()*255).toFixed(0);
    var g=(Math.random()*255).toFixed(0);
    var b=(Math.random()*255).toFixed(0);
    var nBarva="rgb("+r+","+g+","+b+") !important";
    /* pomocí jQuery změníme CSS styl pozadí na vygenerovanou barvu */
    $($par.background).css('background-color',nBarva);
}});
```

Na pozadí bude vloženo video z YouTube. JavaScript bude vypadat následovně:

```
/* na pozadí vložíme video z YouTube */
$($par.background).html('<iframe class="youtubeVid"
width="'+$($par.background).width()+'px"
height="'+$($par.background).height()+'"src="https://www.youtube.com/embed/0fYL_qiD
Yf0?autoplay=1&html5=1" frameborder="0" allow="autoplay; encrypted-
media"></iframe>');

/* aktivujeme událost změna velikosti-upravíme velikost videa podle kontejneru */
$par.callbacks.onContainerResize({container:$par.container,callback:function(sirka,
vyska){
    /* změníme velikost videa */
    $(''.youtubeVid', $par.container).css('width',sirka).css('height',vyska);
}});
```

Každých 5 minut se uloží do periferie náhodné číslo, podle výsledku se probarví pozadí, výsledek zapíšeme do databáze **LookDetu** jako uživatelský parametr. JavaScript bude vypadat následovně:

```
/* aktivujeme interval, jeho identifikátor uložíme do proměnné timZapDoRS pro
deaktivaci */
/* interval vykonávání 300000 ms */
var timZapDoRS=window.setInterval(function(){
    /* pomocí API zapíšeme do stanice náhodné číslo, výsledek do konzoly */
    var zh=Math.random()*1000;
    /* cílem je 1 pole MF, ID stanice 7, wid 1082, typ 5 - MF, zapisujeme řádek
a sloupec 0 */
    $par.api.putToStation({id:'123','note':'Auto zapis do stanice',
req:[{idSt:7,req:[{name:'zapis 1',channel:1082,type:5,row:0,col:0,
rows:1,cols:1,old:[[-1]],new:[[zh]]}]}]},function(res){
    /* máme výsledek, probarvíme pozadí */
    /* výchozí stav OK */
    var nBarva='#aaffaa';
    /* globální výsledek zápisu - při chybě červené pozadí */
    if (res.OK!=1) { nBarva='#ffaaaa'; } else
        /* výsledek konkrétní položky - při chybě oranžové pozadí */
        if (res.data[0].OK!=1) { nBarva='#ffffaa'; }
```

```
/* pomoci jQuery změníme CSS styl pozadí na vygenerovanou barvu */
$(par.background).css('background-color',nBarva);
/* výsledek do databáze, identifikační klíč "autoWrite" */
/* do výsledku přidáme zapisovanou hodnotu */
res.val=zh;
/* a pomocí API uložíme k zvolenému uživateli */
$par.api.getSetDBData({req:{r1:{f:'putUserData',val:JSON.stringify(res),key:
"autoWrite"}}},function(vysl){
    console.log(vysl);
});
});
},300000);

/* protože jsme aktivovali časovač, musíme jej při deaktivaci náhledu vypnout! */
/* aktivujeme si událost při odstranění náhledu */
$par.callBacks.onUnload({container:$par.container,callback:function(){
    /* ruší se náhled - vypneme časovač zápisu */
    window.clearInterval(timZapDoRS);
}});
```

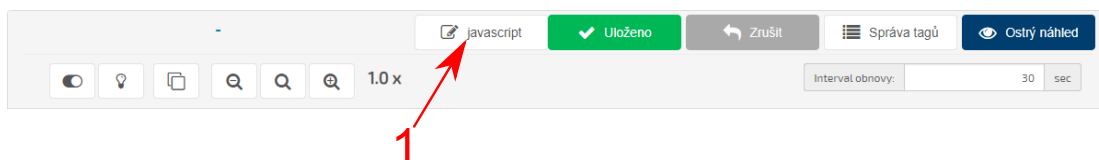
## 2.3 Vložení funkce do náhledu

Funkci lze do náhledu vložit dvojím způsobem:

- vepsáním JavaScriptu do sekce pro vložení JavaScriptu v náhledu,
- importem souboru s JavaScriptem.

### 2.3.1 Sekce pro vložení JavaScriptu

Samotný skript lze psát po kliknutí na tlačítko „**javascript**“, které se nachází v nástrojové liště náhledu. Nástrojová lišta náhledu se zobrazí společně s náhledem po kliknutí na vybraný náhled v seznamu náhledů.



Obr. 2 – Nástrojová lišta náhledu

Legenda

| Číslo | Význam                                        |
|-------|-----------------------------------------------|
| 1     | Tlačítko pro zadání uživatelského JavaScriptu |

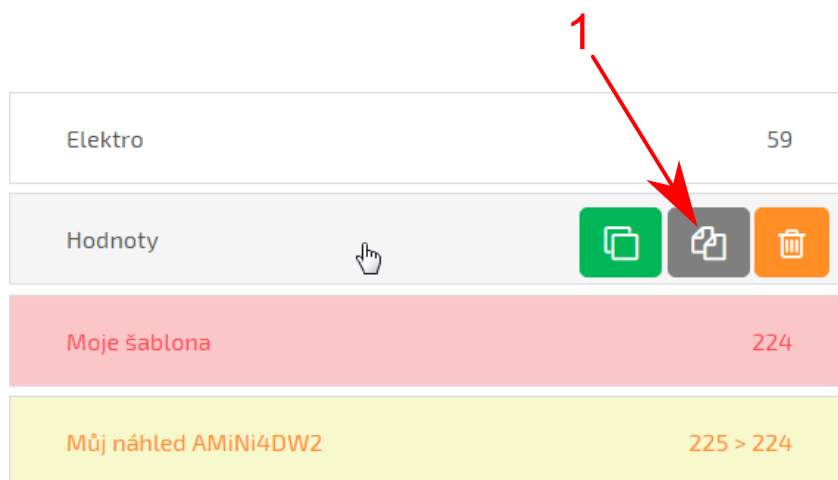
Po stisku tlačítka dojde k otevření sekce, pro psaní uživatelského JavaScriptu.

Nástrojová lišta sekce s JavaScriptem obsahuje tlačítka, jejichž význam je patrný z následující tabulky.

| Tlačítko                                                                            | Význam                                                                                                                                                                           |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Uložení změn, provedených v JavaScriptu. Tlačítko mění barvu v závislosti na tom, zda byla v JavaScriptu provedena změna (tlačítko je oranžové) či nikoliv (tlačítko je zelené). |
|  | Export JavaScriptu do souboru *.js.                                                                                                                                              |
|  | Zavření sekce pro vložení JavaScriptu.                                                                                                                                           |

## 2.3.2 Import souboru s JavaScriptem

Soubor s JavaScriptem (\*.JS) lze do náhledu importovat v sekci pro souborové operace náhledu.



Obr. 3 – Tlačítko „**Souborové operace**“

Legenda

| Číslo | Význam                                               |
|-------|------------------------------------------------------|
| 1     | Tlačítko pro otevření sekce se souborovými operacemi |

Po stisku tlačítka dojde k otevření sekce se souborovými operacemi.

## 3 Pluginy

Plugin je objekt, který rozšiřuje základní funkcionalitu systému **LookDet** podle potřeb uživatele. Pluginy se dělí na grafické/serverové a v rámci těchto dvou typů se dělí na lokální/globální/systémové.

Tlačítka **Grafické pluginy** **Serverové pluginy** přepínají mezi zobrazením grafických a serverových pluginů.

Zobrazení pluginů je ve formě tabulky.

| Hledání                 |                          | Grafické pluginy                    | Serverové pluginy                                                                                                                                                                                                                                           |
|-------------------------|--------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Název                   | Minimální verze LookDet  | Popis funkce                        | Aktivní                                                                                                                                                                                                                                                     |
| Lokální plugin          |                          |                                     |                                                                                       |
| new_plugin_175_1        | 6.0.15, BETA, 2025-04-22 |                                     | 0                                                                                                                                                                                                                                                           |
| new_plugin_6_1          | 6.0.15, BETA, 2025-04-22 |                                     | 0                                                                                                                                                                                                                                                           |
| new_plugin_6_2          | 6.0.15, BETA, 2025-04-22 |                                     |    |
| new_plugin_6_3          | 6.0.15, BETA, 2025-04-22 |                                     | 0                                                                                                                                                                                                                                                           |
| Globální plugin         |                          |                                     |                                                                                       |
| glob_average            | 5.0.11                   | Výpočet průměru za zadané období    | 1                                                                                                                                                                                                                                                           |
| glob_counting_previews  | 3.0.8                    | Zahrnout do počítání náhledů        | 0                                                                                                                                                                                                                                                           |
| glob_energie            | 4.0.0                    | Zpracování 15-cti minutových měření | 1                                                                                                                                                                                                                                                           |
| glob_show_count_preview | 3.0.8                    | Zobrazení počtu náhledů             | 0                                                                                                                                                                                                                                                           |
| glob_ZobrNumVal_v2      | 3.0.8                    | Zobrazení numerické hodnoty v.2     | 0                                                                                                                                                                                                                                                           |
| Systémový plugin        |                          |                                     |                                                                                                                                                                                                                                                             |
| sys_dataTable           | 4.6.10                   | Zobrazení dat v tabulce             | 1                                                                                                                                                                                                                                                           |
| sys_newWindow           | 3.0.8                    | Nový náhled v okně                  | 1                                                                                                                                                                                                                                                           |
| sys_prepniZalozku       | 6.0.0                    | Přepnutí na záložku s náhledem ...  | 1                                                                                                                                                                                                                                                           |

Obr. 4 – Tabulka pluginů

Grafické pluginy slouží k vytváření funkcí pro objekty v náhledech. Programují se v jazyce JavaScript.

Serverové pluginy slouží k vytváření algoritmů, které probíhají na pozadí. Programují se v jazyce PHP.

Oba typy pluginů je možné provázat tak, aby mezi sebou sdílely data.

Lokální pluginy jsou k dispozici pouze v rámci lokality, ve které jsou vytvořeny. Globální pluginy jsou k dispozici napříč celým serverem. Systémové pluginy jsou součástí instalace serveru, lze je využívat, ale nelze je editovat a odstranit.

### 3.1 Správa pluginu

Nový plugin lze založit stiskem tlačítka  v pravé části záhlaví jednotlivých typů pluginů – Grafické/Serverové a Globální/Lokální.

Po najetí myši na řádek s pluginem (mimo systémových pluginů) se zobrazí interaktivní tlačítka:

-    – pro grafické pluginy – export, import a odstranění pluginu,
-    – pro serverové pluginy – spuštění, import a odstranění pluginu.

Editaci pluginu (programového kódu) lze zahájit stiskem levého tlačítka myši na řádku s pluginem (mimo sloupec „Výsledek“ u serverových pluginů).

Po stisku levého tlačítka myši na řádku serverového pluginu ve sloupci „Výsledek“ se zobrazí textový výstup posledního volání pluginu. Opětovným stiskem levého tlačítka myši se tento textový výstup skryje.

Tlačítko  vyvolá ruční spuštění pluginu. Serverové pluginy se standardně volají automaticky podle časového plánu.

### 3.1.1 Export a import existujících pluginů

Existující pluginy lze ze systému **LookDet** exportovat. Data takto exportovaného pluginu lze editovat pomocí nástrojů osobní preference a následně importovat zpět do systému **LookDet**.

Grafické pluginy se exportují/importují zabalené v souboru formátu zip.

Serverové pluginy se exportují/importují v souboru formátu php.

#### Export

Export pluginu se provádí interaktivním tlačítkem  v pravé části řádku pluginu.

Export pluginu lze provádět také v rámci editace pluginu tlačítkem  v pravé horní části okna editoru programového kódu.

#### Import

Import pluginu lze provést třemi způsoby:

- interaktivním tlačítkem na řádku pluginu (pouze u grafických pluginů),
- tlačítkem v pravé části záhlaví sekce pluginů,
- přetažením (drag & drop).

Import pluginu interaktivním tlačítkem  (pouze u grafických pluginů) přepíše data stávajícího pluginu importovanými daty.

Import pluginu tlačítkem  v pravé části záhlaví sekce pluginů vytvoří v systému **LookDet** nový plugin podle importovaných dat.

Další možnost, jak importovat plugin, je přetáhnout (drag & drop) data na tlačítko  nebo  v pravé části záhlaví sekce pluginů. Tím dojde k vytvoření nového pluginu podle importovaných dat.

Poslední možnost, jak importovat plugin, je přetáhnout (drag & drop) data na řádek existujícího pluginu. Tím dojde k přepsání dat stávajícího pluginu importovanými daty.

## 3.2 Backend (serverové) pluginy

Seznam pluginů se zobrazí po kliknutí na tlačítko „**Serverové pluginy**“ v sekci „Pluginy“.

Význam sloupců v seznamu pluginů je stejný jako význam parametrů pluginů, zadávaných v jeho hlavičce.

### 3.2.1 Struktura pluginu

Každý plugin začíná řetězcem „<?php“ a končí řetězcem „?>“. V těle pluginu je hlavička s parametry (řádky hlavičky začínají řetězcem „//“). Za hlavičkou je vlastní tělo pluginu.

Vlastní plugin pak vypadá následovně:

```
<?php
// PARAMETR: HODNOTA
// PARAMETR: HODNOTA
```

```
Vlastní tělo pluginu
?>
```

### 3.2.2 Hlavička pluginu

Hlavička pluginu začíná vždy na prvním řádku za řetězcem „<?php“. Každý řádek hlavičky začíná řetězcem „//“. Na každém řádku může být pouze jeden parametr a na pořadí parametrů nezáleží. **Rozlišují se velká a malá písmena.**

Po založení nového pluginu jsou jména parametrů pluginu systémem **LookDet** automaticky předdefinována. Postačí tedy doplnit požadované hodnoty.

#### Povinné parametry hlavičky

Bez povinných parametrů nelze plugin uložit a aktivovat.

- **Plugin name** – název pluginu. Může obsahovat maximálně 20 znaků („a“ až „z“, „A“ až „Z“, „0“ až „9“ a „\_“ podtržítko).
- **Minimal LookDet version** – minimální vyžadovaná verze **LookDetu**.
- **Enabled** – aktivace / deaktivace pluginu (1 – aktivní, 0 – neaktivní). Výchozí hodnota je 1.
- **Maximum execution time** – maximální doba vykonávání v sekundách. Výchozí hodnota je 10 s.
- **Call interval** – interval volání pluginu v sekundách.
- **First call** – čas prvního startu pluginu (formát RRRR-MM-DD HH:MM:SS). Od zadaného času se bude plugin spouštět s periodou zadanou v parametru „Call interval“.
- **Call plan** – sekundový CRON plán (viz lookdet\_a\_g\_cz\_xxx.pdf) spouštění pluginu. Pokud je zadán, tak má přednost před parametrem „Call interval“.

#### Nepovinné parametry hlavičky

- **Description** – popis pluginu.
- **Description\_XXX** – lokalizovaný popis pluginu v jazyce XXX.
- **Version** – verze pluginu (výchozí hodnota je 0.0.1).

- **Version date** – datum vytvoření pluginu (formát RRRR-MM-DD).
- **Author** – jméno autora pluginu.
- **Plugin URL** – webová adresa zdroje pluginu (např. <https://amitautomation.cz>).
- **Last change** – jméno, datum a čas poslední změny pluginu. Parametr je automaticky aktualizován systémem **LookDet**.

### 3.2.3 Objekt \$interface

Při definici serverového pluginu lze využít objekt „\$interface“. Volání jeho funkcí lze provést následovně:

```
$interface->nazevFunkce (parametry)
```

V případě, že je některým z parametrů funkce také ID lokality a plugin je definován jako lokální, uvede se místo parametru idLokality hodnota „null“.

Jako výsledek se vrátí pole „OK“ obsahující informaci o vykonání funkce (0 – funkce se nevykonala, 1 – funkce se vykonala) a pole „data“ jehož obsah se liší v závislosti na volané funkci.

### Funkce pro práci s kanály v LookDetu

**getCurValue** Získání jedné hodnoty z tabulky aktuálních dat. Funkce vrátí čas získání hodnoty [time], hodnotu [value], dobu od poslední změny hodnoty [age] a čas poslední změny hodnoty v kanálu [lch].

Parametry:

- idLokality (pokud jde o lokální plugin, zadá se „null“),
- kanál,
- řádek,
- sloupec.

**getCurHandle** Získání celého obsahu kanálu aktuálních dat. Seřazeno řádek [row], sloupec [col]. Vrací pole obsahující řádek [row], sloupec [col], čas získání hodnoty [time], hodnotu [value], dobu od poslední změny hodnoty [age] a čas poslední změny hodnoty v kanálu [lch].

Parametry:

- idLokality (pokud jde o lokální plugin, zadá se „null“),
- kanál.

**getCurRow** Získání celého řádku kanálu. Seřazeno podle sloupce [col]. Vrací pole obsahující sloupec [col], čas získání hodnoty [time], hodnotu [value], dobu od poslední změny hodnoty [age] a čas poslední změny hodnoty v kanálu [lch].

Parametry:

- idLokality (pokud jde o lokální plugin, zadá se „null“),
- kanál,
- řádek.

**getCurCol** Získání celého sloupce kanálu. Seřazeno podle řádku [row]. Vrací pole obsahující řádek [row], čas získání hodnoty [time], hodnotu [value], dobu od poslední změny hodnoty [age] a čas poslední změny hodnoty v kanálu [lch].

Parametry:

- idLokality (pokud jde o lokální plugin, zadá se „null“),
- kanál,
- sloupec.

**getCurMatrix** Získání výřezu o určitých rozměrech. Seřazeno řádek [row], sloupec [col]. Vrací pole obsahující řádek [row], sloupec [col], čas získání hodnoty [time], hodnotu [value], dobu od poslední změny hodnoty [age] a čas poslední změny hodnoty v kanálu [lch].

Parametry:

- idLokality (pokud jde o lokální plugin, zadá se „null“),
- kanál,
- řádek,
- sloupec,
- řádků,
- sloupců.

**setCurValue** Zapiše hodnotu do kanálu. Vrací „null“ při úspěchu, textovou chybu při selhání.

Parametry:

- idLokality (pokud jde o lokální plugin, zadá se „null“),
- kanál,
- řádek,
- sloupec,
- typ dat (viz níže),
- hodnota.

Typ dat

| ID | Typ                            | Poznámka                                                   |
|----|--------------------------------|------------------------------------------------------------|
| 1  | Integer 8 bit                  | -128 až 127                                                |
| 2  | Unsigned integer 8 bit         | 0 až 255                                                   |
| 3  | Integer 16 bit                 | -32 768 až 32 767                                          |
| 4  | Unsigned integer 16 bit        | 0 až 65 535                                                |
| 5  | Long 32 bit                    | -2 147 483 648 až 2 147 483 647                            |
| 6  | Unsigned long 32 bit           | 0 až 4 294 967 295                                         |
| 7  | Double integer 64 bit          | -9 223 372 036 854 775 808<br>až 9 223 372 036 854 775 807 |
| 8  | Unsigned double integer 64 bit | 0 až 18 446 744 073 709 551 615                            |
| 9  | Real 32 bit                    | 1E <sup>-37</sup> až 1E <sup>+37</sup>                     |
| 10 | Double real 64 bit             | 1E <sup>-307</sup> až 1E <sup>+308</sup>                   |

## Funkce pro práci s provozním deníkem ve stanici AMiT

**getPDAMiTData** Vráť data provozního deníku ze stanice typu AMiT ve formátu [time], [code], [segment], [offset], [wid], [data]. Viz tabulka níže.

Parametry:

- idStanice,
- od – počáteční čas ve formátu RRRR-MM-DD HH:MM:SS,
- do – konečný čas ve formátu RRRR-MM-DD HH:MM:SS. Pokud jsou požadována data do aktuálního času, zadá se „null“.

| Formát    | Význam                                                                                                   |
|-----------|----------------------------------------------------------------------------------------------------------|
| [time]    | Čas vzniku hlášení v provozním deníku.                                                                   |
| [code]    | Kód hlášení v provozním deníku (viz nápověda k části PseDet vývojového prostředí DetStudio).             |
| [segment] | Místo vzniku hlášení v programu (viz nápověda k části PseDet vývojového prostředí DetStudio).            |
| [offset]  | Číslo procesu, kde došlo k hlášení / chybě (viz nápověda k části PseDet vývojového prostředí DetStudio). |
| [wid]     | WID proměnné / doplňkový kód (viz nápověda k části PseDet vývojového prostředí DetStudio).               |
| [data]    | Definiční hodnota (viz nápověda k části PseDet vývojového prostředí DetStudio).                          |

**getPDAMiThead** Vrátí parametry provozního deníku pro stanici AMiT ve formátu [lng], [id], [tabdw], [tabf], [prefix], [text], [text1]. Viz tabulka níže.

Parametry:

- idStanice.

| Formát   | Význam                                                                                                                                     |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------|
| [lng]    | Kód jazyka (cze, eng, ...).                                                                                                                |
| [id]     | Kód hlášení v provozním deníku (viz nápověda k části PseDet vývojového prostředí DetStudio).                                               |
| [tabdw]  | Textový identifikátor tabulky DW (viz nápověda k části PseDet vývojového prostředí DetStudio).                                             |
| [tabf]   | Textový identifikátor tabulky F (viz nápověda k části PseDet vývojového prostředí DetStudio).                                              |
| [prefix] | Řetězec s krátkým prefixem pro daný kód záznamu, platné jsou první tři znaky (viz nápověda k části PseDet vývojového prostředí DetStudio). |
| [text]   | Text hlášení.                                                                                                                              |
| [text1]  | Text komentáře.                                                                                                                            |

**getPDAMiTDW** Vrátí parametry tabulky DW provozního deníku pro stanici AMiT ve formátu [lng], [name], [from], [to], [text]. Viz tabulka níže.

Parametry:

- idStanice.

| Formát | Význam                                                                                      |
|--------|---------------------------------------------------------------------------------------------|
| [lng]  | Kód jazyka (cze, eng, ...).                                                                 |
| [name] | Jméno tabulky DW (viz nápověda k části PseDet vývojového prostředí DetStudio).              |
| [from] | Dolní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio). |
| [to]   | Horní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio). |
| [text] | Text z tabulky DW (viz nápověda k části PseDet vývojového prostředí DetStudio).             |

**getPDAMiTF** Vrábí parametry tabulky F provozního deníku pro stanici AMiT ve formátu [lng], [name], [from], [to], [text]. Viz tabulka níže.

Parametry:

- idStanice.

| Formát | Význam                                                                                      |
|--------|---------------------------------------------------------------------------------------------|
| [lng]  | Kód jazyka (cze, eng, ...).                                                                 |
| [name] | Jméno tabulky F (viz nápověda k části PseDet vývojového prostředí DetStudio).               |
| [from] | Dolní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio). |
| [to]   | Horní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio). |
| [text] | Text z tabulky F (viz nápověda k části PseDet vývojového prostředí DetStudio).              |

## Funkce pro práci s archívy v LookDetu

**getArchiveData** Vrábí archivované hodnoty ve formátu [time], [value] (čas, hodnota) seřazené podle času.

Parametry:

- idLokality,
- kanál,
- řádek,
- sloupec,
- od – počáteční čas ve formátu RRRR-MM-DD HH:MM:SS,
- do – konečný čas ve formátu RRRR-MM-DD HH:MM:SS. Pokud jsou požadována data do aktuálního času, zadá se „null“.

## Funkce pro práci se stanicemi a jejich daty

**getStations** Získá seznam stanic ve vybrané lokalitě a vrátí jako pole.

Parametry:

- idLokality (v lokálním pluginu se zadá „null“).

Vracená data obsahují parametry:

- id – jedinečné ID stanice,
- idl – ID lokality,
- t – typ stanice (viz tabulka níže),
- a – stanice je aktivní (true / false),
- cze – český popis stanice.

Možné typy stanic.

| Číselná hodnota | Typ stanice              |
|-----------------|--------------------------|
| 0               | AMiT TCP                 |
| 1               | AMiT UDP                 |
| 2               | AMiT dynamická IP adresa |
| 3               | M-Bus TCP                |
| 4               | M-Bus UDP                |
| 5               | MODBUS/TCP               |
| 6               | HTTP – POST              |
| 7               | LookDet                  |

**getStationsTr** Získá seznam lokalizací dané stanice a vrátí jako pole.

Parametry:

- idLokality (v lokálním pluginu se zadá „null“),
- idStanice.

Vrácená data obsahují parametry:

- l – kód jazyka (language),
- n – lokalizovaný název stanice.

**getFromStation** Načte hodnotu, nebo více hodnot ze stanice.

Parametry:

- idLokality (v lokálním pluginu se zadá „null“),
- idStanice,
- typ stanice (viz možné typy stanic ve funkci „getStations“),
- požadavky na čtení.

### Formát požadavku na čtení dat ze stanice AMiT

```
// skalární proměnná
["WID;Typ_Promenne;0;0;1;1"]
// matice
["WID;Typ_Promenne;Radek_Odkud;Sloupec_Odkud;Pocet_Radku;Pocet_Sloupce"]
```

Typy proměnných

| Hodnota | Typ dat        |
|---------|----------------|
| 0       | Integer        |
| 1       | Long           |
| 2       | Float          |
| 3       | Matice Integer |
| 4       | Matice Long    |
| 5       | Matice Float   |

### Formát požadavku na čtení dat ze stanice MODBUS

```
["Pocatecni Registr;Typ_Registru;;;Pocet_Registru;"]
```

Typy registrů

| Hodnota | Typ dat                 |
|---------|-------------------------|
| 0       | Holding 16 bit signed   |
| 1       | Holding 16 bit unsigned |
| 2       | Holding 32 bit signed   |
| 3       | Holding 32 bit unsigned |
| 4       | Holding 32 bit float    |
| 7       | Coils                   |

Odpověď je pole, obsahující výsledek a hodnotu (pole hodnot).

Význam výsledku komunikace pro stanice AMiT.

| Číselná hodnota | Význam           |
|-----------------|------------------|
| 0               | OK               |
| 1               | Chyba přenosu    |
| 2               | Špatné parametry |

| Číselná hodnota | Význam                                    |
|-----------------|-------------------------------------------|
| 3               | Špatná funkce                             |
| 8               | OK                                        |
| 100             | Neznámá chyba                             |
| 101             | Špatný kontrolní součet – poškozená data  |
| 102             | Chybné ID odpovědi                        |
| 103             | Stanice neodpověděla                      |
| 104             | Požadavek na nový klíč zaslán více jak 3x |
| 105             | Odpověď OK – nemá smysl při čtení dat     |
| 106             | Chyba při dešifrování dat                 |
| Ostatní         | Neznámý chybový kód                       |

Význam výsledku komunikace pro stanice MODBUS.

| Číselná hodnota | Význam                             |
|-----------------|------------------------------------|
| 0               | OK                                 |
| 1               | Nepodporovaná funkce               |
| 2               | Špatná adresa dat                  |
| 3               | Neplatná data                      |
| 4               | Selhání stanice                    |
| 5               | Potvrzení                          |
| 6               | Stanice je zaneprázdněna           |
| 8               | Chyba parity paměti                |
| 10              | Brána – přenosová cesta nedostupná |
| 11              | Brána – cílové zařízení neodpovídá |
| Ostatní         | Neznámý chybový kód                |

### Příklad:

Požadavek na data ze stanice č. 3 typu 1 (AMiT) v místní lokalitě:

```
// čtení jedné buňky matice [0,0] či proměnné:
// wid 2002, typ 2(F), radek 0, sloupec 0, radku 1, sloupcu 1
$interface->getFromStation(null,3,1,["2002;2;0;0;1;1"]);
// čtení více položek najednou:
// wid 2002 a 2003, typ 2(F), radek 0, sloupec 0, radku 1, sloupcu 1
$interface->getFromStation(null,3,1,["2002;2;0;0;1;1","2003;2;0;0;1;1"]);
```

**putToStation** Zapiše hodnotu, nebo více hodnot do stanice.

Parametry:

- idLokalita (v lokálním pluginu se zadá „null“),
- idStanice,
- typ stanice (viz možné typy stanic ve funkci „getStations“),
- požadavky na zápis.

Formát požadavků na zápis je stejný jako u funkce „getFromStation“, pouze se na konec doplní požadovaná hodnota (viz příklad níže).

Návratová hodnota funkce obsahuje výsledek komunikace.

| Hodnota | Význam                  |
|---------|-------------------------|
| 0       | Zápis skončil chybou    |
| 1       | Zápis proběhl v pořádku |

Odpověď je pole obsahující výsledek (viz funkce „getFromStation“). Pokud je návratová hodnota 0, obsahuje pole kód chyby. Pokud je návratová hodnota 1, obsahuje pole pouze hodnotu 1.

### Příklad:

Požadavek na zápis více hodnot do různých proměnných stanice č. 3 typu 1 (AMiT) v místní lokalitě:

```
// Zápis do jedné buňky matice / proměnné
// wid 2002, typ 2(F), radek 0, sloupec 0, radku 1, sloupce 1
$obj=new stdClass();
$obj->req="2002;2;0;0;1;1";
$obj->val=2019;
// zápis více buněk matice:
$obj1=new stdClass();
$obj1->req="1055;4;0;0;2;2";
$obj1->val=[[1,2],[3,4]];
$vysl=$interface->putToStation(null,3,1,[$obj,$obj1]);
```

\$vysl – obsahuje hodnotu/pole hodnot:

- 1 – zápis proběhl v pořádku,
- 0 – chyba zápisu (je uvedena včetně specifikace chyby).

## Souborové operace

**openFile** Otevře nebo založí soubor (případně cestu), vrací handle souboru, nebo false při chybě. Nejsou podporovány relativní cesty (../). Výchozí mód je „a+“ (otevři/založ pro čtení a zápis a jdi na konec souboru). Více viz php návod pro funkci fopen (<https://www.php.net/manual/en/function.fopen.php>)

Parametry:

- \$name,
- \$mode.

**addToFile** Otevře nebo založí soubor (případně cestu) a zapíše obsah \$content. Vrací true, nebo false při chybě. Nejsou podporovány relativní cesty (../).

Parametry:

- \$name,
- \$content.

**delFile** Vymaže soubor z disku. Nejsou podporovány relativní cesty (../).

Parametry:

- \$name.

**listDir** Seznam souborů ve složce (pole). Nejsou podporovány relativní cesty (../). Vrací false, nebo pole s obsahem souborů a dalších polí (podsložek).

Parametry:

- \$name.

**existsFile** Zjistí, zda existuje soubor zadaný v parametru \$name. Vrací true, pokud soubor existuje, jinak false.

Parametry:

- \$name.

## Ostatní funkce

**setOutput** Do výsledku pluginu dostupného z administrace přidá řádek textu.

Parametry:

- text.

**getPluginType** Vráť typ pluginu (0 – systémový, 1 – globální, 2 – lokální).  
Funkce je bez parametrů.

**getDBHandle** Vráť objekt PDO databáze – pouze pro systémové pluginy.  
Funkce je bez parametrů.

**writeSysLog** Zapiše text \$msg do systémového logu.

Parametry:

- \$msg

### **Příklad:**

Požadavek na zápis textu „Kritická chyba pluginu!“ do systémového logu.

```
$interface->writeSysLog('Kritická chyba pluginu!');
```

Log je dostupný v administraci **LookDet** na záložce „Systém“ v sekci „Logy“ pod názvem „pluginsBG-název pluginu“.

## 3.3 Frontend (grafické) pluginy

Ve srovnání se systémovými funkcemi, které lze přiřadit objektům v SVG obrázku pouze jedenkrát je výhodou grafických pluginů možnost definice více pluginů na jeden objekt v SVG.

Pro tvorbu frontend pluginů je nutné využít JavaScript.

Plugin se dělí na dvě části:

- runtime,
- designtime.

### 3.3.1 Struktura pluginu

Jak pro designtime tak pro runtime je třeba nadefinovat dva soubory:

- Main.js – JavaScript pluginu,
- \*.lang – texty (pro každou jazykovou mutaci musí být nadefinován jeden soubor).

Společný soubor pro obě části je info.txt. Je to povinný soubor s popisem pluginu (stejně s parametry backend pluginů, pouze bez řetězce „//“ na začátku řádku).

### 3.3.2 Parametry v souboru info.txt

Na každém řádku může být pouze jeden parametr a na pořadí parametrů nezáleží. **Rozlišují se velká a malá písmena.**

Po založení nového pluginu jsou jména parametrů pluginu systémem **LookDet** automaticky předdefinována. Postačí tedy doplnit požadované hodnoty.

**Pozor** Změna názvu pluginu **ZNEFUNKČNÍ plugin všude**, kde byl použit a ve všech náhledech ho **bude třeba znovu nadefinovat!**

### Povinné parametry

Bez povinných parametrů nelze plugin uložit a aktivovat.

- **Plugin name** – název pluginu. Může obsahovat maximálně (20 znaků „a“ až „z“, „A“ až „Z“, „0“ až „9“ a „\_“ podtržítka).
- **Minimal LookDet version** – minimální vyžadovaná verze **LookDetu** (Výchozí hodnota je 3.0.8).
- **Enabled** – aktivace / deaktivace pluginu (1 – aktivní, 0 – neaktivní). Výchozí hodnota je 0.
- **Description** – popis pluginu, který se bude zobrazovat v administrátorské části při výběru funkcí pro SVG objekty.

### Nepovinné parametry hlavičky

- **Runtime only** – 1 znamená, že plugin se nebude nabízet v designu (nemá nastavení).
- **Last change** – jméno, datum a čas poslední změny pluginu. Parametr je automaticky aktualizován systémem **LookDet**.
- **Description\_XXX** – lokalizovaný popis pluginu v jazyce XXX.
- **Version** – verze pluginu (výchozí hodnota je 0.0.1).
- **Version date** – datum vytvoření pluginu (formát RRRR-MM-DD).
- **Plugin URL** – webová adresa zdroje pluginu (např. <https://amitautomation.cz>).
- **Author** – jméno autora pluginu.

### 3.3.3 Designu

V části designu se definuje seznam parametrů, které je třeba nastavit pluginu v administrátorské části.

- Po kliknutí na aktivní část SVG se spustí výběr pluginu (název/popis), po výběru se spustí pevná funkce v JavaScriptu, tzv. „pluginSetup“, která dostane okno (DIV) a je na ní, jaké parametry a zadávací políčka v něm vytvoří. Bude mít k dispozici JavaScriptové funkce typu „nalistujKanal“, „zadejBarvu“, „maskaBitu“, „Preložit“, apod.
- Ve výchozím stavu se vzhled formuláře připraví v html šabloně, která se do DIVu vloží.
- Html šablona může obsahovat tagy pro překlad.
- Parametry v SVG se ukládají přes API – doplní identifikaci pluginu.

### Podsložky

- css – styly, automaticky natažené s pluginem,
- js – javascripty, automaticky natažené s pluginem a zapouzdřené s main.js,
- lang – soubory s překlady textů pluginu,
- other – ostatní soubory, pro plugin dostupné (download) přes php funkci.

### Povinné funkce

Funkce jsou typu callback (cb).

**startEditDOM** Vytvoření zadávacího formuláře parametrů pluginu.

Parametry:

- dom – prvek, na který se kliklo a kde jsou uloženy jeho případné parametry.
- edit – informace zda se jedná o první volání (hodnota false), kdy by plugin měl zobrazit výchozí parametry nebo zda se jedná o změnu (hodnota true), kdy by měl plugin z DOM načíst dříve zadané parametry.
- container – cílová oblast (DOM), kde má plugin vytvořit zadávací formulář.

**saveToDOM** Uložení parametrů pluginu z vytvořeného formuláře.

Parametry:

- dom – prvek, kam mají být uloženy předané parametry.
- container – cílová oblast (DOM), kde má plugin zadávací formulář.

## Dostupné funkce

Funkce jsou typu application programming interface (api).

**getParams** Získá parametr(y) z DOM, pokud některý neexistuje, vrátí hodnotu “def” (viz příklad).

Parametry:

- dom – prvek, na který se kliklo a kde jsou uloženy jeho případné parametry.
- [požadavky] – viz příklad níže.

### **Příklad [požadavky]:**

Požadavek:

```
{name:"cil",def:""},{name:"sir",def:800},{name:"vys",def:600}
```

kde name je název parametru („a“ až „z“, „A“ až „Z“, „0“ až „9“ a „\_“). Neměl by být delší, než 10 znaků.

Odpověď:

```
{cil:"1",sir:"100",vys:"200"}
```

**setParams** Uloží parametr(y) do DOM.

Parametry:

- dom – prvek, na který se kliklo a kde jsou uloženy jeho případné parametry.
- [data] – viz příklad níže.

### **Příklad [data]:**

Požadavek:

```
{name:"cil",value:XXX},{name:"sir",value:YYY}
```

kde name je název parametru („a“ až „z“, „A“ až „Z“, „0“ až „9“ a „\_“). Neměl by být delší, než 10 znaků.

Odpověď:

Vrací hodnotu true/false.

**getText** Vrátí lokalizovaný text zadaného ID z pluginu v jazyce přihlášeného uživatele, nebo česky.

Parametry:

- id – id textu v pluginu.
- def – návratová hodnota v případě, že neexistuje ani lokalizovaný, ani český text.
- repl – Pole pro náhradu. Pokud je v přeloženém textu %1, nahradí se prvním textem z parametru **repl**, %2 druhým atd. Parametr je nepovinný.

**getSysText** Vrátí lokalizovaný text zadaného ID v jazyce přihlášeného uživatele.

Parametry:

- id – id textu.
- def – návratová hodnota v případě, že neexistuje lokalizovaný text.
- repl – Pole pro náhradu. Pokud je v přeloženém textu %1, nahradí se prvním textem z parametru **repl**, %2 druhým atd. Parametr je nepovinný.

**translate** Nahradí v textu klíčová slova #TRS:XXX# systémovými překlady. Kde XXX je ID systémového textu a #TRP:XXX# je překlad z pluginu.

Parametry:

- text.

**getFile** Načte soubor.

Parametry:

- name – název požadovaného souboru.
- onLoad – funkce, která bude zavolána po načtení, jako parametr je předán obsah souboru.

**getSetDBData** Popis viz kapitola 3.3.4 „Runtime“.

**getUserLanguage** Vrátí kód lokalizace (jazyka) uživatele (cze, eng, ...).  
Funkce je bez parametrů.

**getLanguages** Vrátí pole se seznamem dostupných lokalizací (jazyků).  
Funkce je bez parametrů.

**getPreviews** Získá seznam náhledů v lokalitě a vrátí jej jako parametr funkce „cBack“.

Parametry:

- cBack – funkce, která se volá při dokončení a obsahuje vrácená data. Odpověď je vrácena jako parametr předané funkce „cBack“.

Vrácená data obsahují parametry:

- id – jedinečné ID náhledu,
- name – lokalizovaný název,
- minWidth – minimální šířka náhledu,
- minHeight – minimální výška náhledu,
- maxWidth – minimální šířka náhledu,
- maxHeight – minimální výška náhledu.

### **Příklad vrácených dat:**

```
[{"id": "2", "name": "App info", "minWidth": "-1", "minHeight": "-1", "maxWidth": "-1", "maxHeight": "-1"}, {"id": "11", "name": "Bez grafiky", "minWidth": "200", "minHeight": "100", "maxWidth": "-1", "maxHeight": "-1"}]
```

**getStations** Získá seznam stanic ve vybrané lokalitě a vrátí jako parametr funkce „cBack“.

Parametry:

- cBack – funkce, která se volá při dokončení a obsahuje vrácená data. Odpověď je vrácena jako parametr předané funkce „cBack“.

Vrácená data obsahují parametry:

- id – jedinečné ID stanice,
- name – lokalizovaný název,
- type – číselně typ stanice (viz tabulka níže).

Možné typy stanic.

| Číselná hodnota | Typ stanice              |
|-----------------|--------------------------|
| 0               | AMiT TCP                 |
| 1               | AMiT UDP                 |
| 2               | AMiT dynamická IP adresa |
| 3               | M-Bus TCP                |
| 4               | M-Bus UDP                |
| 5               | MODBUS/TCP               |
| 6               | HTTP – POST              |
| 7               | <b>LookDet</b>           |

### **Příklad vrácených dat:**

```
[{"id": "3", "name": "AMiT", "type": "1"}, {"id": "6", "name": "MBUS", "type": "3"}]
```

**selectStation** Umožní nalistovat stanici z vybrané lokality.

Parametry:

- cBack – funkce, která se volá při dokončení a obsahuje vrácená data. Odpověď je vrácena jako parametr předané funkce „cBack“.

Vrácená data obsahují stejné parametry, jako u funkce `getStations`. Pokud uživatel stanici nevybere, bude mít `id` hodnotu -1.

### **Příklad vrácených dat:**

```
{"id": "8", "type": "1", "name": "stanice XYZ"}
```

**selectChannel** Umožní nalistovat kanál z vybrané lokality.

Parametry:

- id – id stanice. Pokud je zadána hodnota -1, zobrazí se veškeré kanály v lokalitě. Jinak se zobrazí pouze kanály od dané stanice.
- cBack – funkce, která se volá při dokončení a obsahuje vrácená data. Odpověď je vrácena jako parametr předané funkce „cBack“.

Vrácená data obsahují parametry:

- l – lokalita kanálu,
- ch – číslo kanálu **LookDetu**,

- r – řádek,
- c – sloupec,
- n – název,
- arch – archivovaný kanál.

### **Příklad vrácených dat:**

```
{1:7,ch:31500,r:0,c:0,n:"Teplota",arch:1}
```

**selectBits** Zadání bitové masky pomocí jednoho html input. Vyvolá dialog pro nastavení bitu a masky (viz [lookdet\\_a\\_g\\_cz\\_xxx.pdf](#)), po potvrzení uloží nastavenou hodnotu do předaného html inputu.

Parametry:

- input – html input pro uložení výsledku.

**initBits** Zadání bitové masky pomocí pole více html inputů. Na předaných html inputech aplikuje klávesovou zkratku pro vyvolání dialogu pro nastavení bitu a masky (viz [lookdet\\_a\\_g\\_cz\\_xxx.pdf](#)). Zbývající funkce je stejná jako u `selectBits`.

Parametry:

- dom – DOM (pole) input – při stisku tlačítka „m“ je umožněno zadání bitové masky, která je vrácena do příslušného inputu.

**initColorPicker** Umožní výběr barvy pomocí myši.

Parametry:

- dom – DOM (pole) DIVů umožňující následně výběr barvy pomocí myši. Barva je uložena v atributu „data-lookdet-color“ příslušného DIVu.

**checkInput** Ověří hodnotu zadanou v input boxu.

Parametry:

- input – input box s hodnotou pro ověření. Při chybě přidá input boxu třídu „check-error“ a nadřazenému DIVu „has-error“ – použití u bootstrap – zvýrazní špatně zadané pole. Požadavky na hodnotu jsou uvedeny v attributech input boxu.
- data-type – typ hodnoty. Možné varianty jsou:
  - int – celé číslo,
  - float – desetinné číslo,
  - text – text,
  - e-mail – e-mailová adresa.

Parametr „data-type“ je povinný – bez něj validace vždy skončí úspěšně.

- data-optional – 1 znamená, že hodnota není povinná (prázdný input je v pořádku), parametr není povinný.
- data-min – pro čísla, minimální povolená hodnota, nepoužití atributu znamená neomezenou spodní hranici.
- data-max – pro čísla, maximální povolená hodnota, nepoužití atributu znamená neomezenou horní hranici
- data-min-length – minimální délka textu, nepoužití parametru znamená bez omezení minimální délky.
- data-max-length – maximální délka textu, nepoužití parametru znamená bez omezení maximální délky.

### 3.3.4 Runtime

V části runtime se definuje chování pluginu v běžném (uživatelském) provozu **LookDetu**.

- po kliknutí na aktivní část SVG se zavolá příslušný plugin a předá se mu informace o tom, na co se kliklo, zbytek je na pluginu.
- plugin může mít funkci „onLoad“, která se zavolá při načtení SVG. Předá se odkaz na SVG, zbytek je na pluginu.
- plugin má k dispozici php funkce **LookDetu** (např. získej hodnotu kanálu apod.).

#### Podsložky

- css – styly, automaticky natažené s pluginem.
- js – JavaScripty, automaticky natažené s pluginem a zapouzdřené s main.js.
- lang – soubory s překlady textů pluginu.
- other – ostatní soubory, pro plugin dostupné (download) přes php funkci.

#### Funkce typu callback

**leftClick** Na objekt se kliklo levým tlačítkem (v designtime se nevolá).

Parametry:

- loc – ID lokality.
- dom – DOM, na který se kliklo.
- container – DIV, kterým je obalen náhled.

**rightClick** Na objekt se kliklo pravým tlačítkem. (v designtime se nevolá).

Parametry:

- loc – ID lokality.
- dom – DOM, na který se kliklo.
- container – DIV, kterým je obalen náhled.

**loadPreview** Funkce pluginu zavolaná po načtení náhledu do stránky.

Parametry:

- container – DIV, kterým je náhled obalen.
- items – položky na kterých je plugin aktivován.
- mode – mód. Nabývá hodnot:
  - 0 – designtime,
  - 1 – runtime (záložky),
  - 2 – runtime (náhled v novém okně).

**unloadPreview** Funkce pluginu zavolaná po zrušení náhledu ze stránky.

Parametry:

- container – DIV, kterým je náhled obalen.
- items – položky na kterých je plugin aktivován.
- mode – mód (viz popis `loadPreview`).

**resizePreview** Funkce pluginu zavolaná při změně velikosti náhledu.

Parametry:

- container – DIV, kterým je náhled obalen.
- width – nová šířka.
- height – nová výška.
- mode – mód (viz popis `loadPreview`).

**refresh** Funkce pluginu zavolaná po globální obnově dat náhledu.

Parametry:

- container – DIV, kterým je náhled obalen.
- id – DOM, ID kontejneru.
- idLoc – ID lokality.
- mode – mód (viz popis `loadPreview`).

## Funkce typu api

**getParams** Získá z DOM parametr(y). Pokud některý neexistuje, vrátí hodnotu „def“.

Parametry:

- dom – DOM.
- [požadavky] – viz příklad níže.

### **Příklad [požadavky]:**

Požadavek:

```
{name:"cil",def:""},{name:"sir",def:800},{name:"vys",def:600}
```

kde `name` je název parametru („a“ až „z“, „A“ až „Z“, „0“ až „9“ a „\_“). Neměl by být delší, než 10 znaků.

Odpověď:

```
{cil:"1",sir:"100",vys:"200"}
```

**getText** Vrátí lokalizovaný text pluginu v jazyce přihlášeného uživatele, nebo česky. Pokud není ani jeden, vrátí obsah parametru **def**.

Parametry:

- id – ID textu.
- def – návratová hodnota pro případ, že ID textu neexistuje.
- repl – parametr je nepovinný, jedná se o pole pro náhradu. Pokud je v přeloženém textu %1, nahradí se prvním textem z pole **repl**, %2 druhým atd.

**getSysText** Získá překlad textu pro aktuální jazyk. Pokud překlad neexistuje, vrátí obsah parametru **def**.

Parametry:

- id – ID textu.
- def – návratová hodnota pro případ, že ID textu neexistuje.
- repl – parametr je nepovinný, jedná se o pole pro náhradu. Pokud je v přeloženém textu %1, nahradí se prvním textem z pole **repl**, %2 druhým atd.

**translate** Nahradí v textu klíčová slova #TRS:XXX# systémovými překlady, kde XXX=ID systémového textu a #TRP:XXX# překlady z pluginu.

Parametry:

- text – viz výše.

**getFile** Načte soubor.

Parametry:

- name – název požadovaného souboru.
- onLoad – funkce, která bude zavolána po načtení, jako parametr je předán obsah souboru.

**getUserLanguage** Vrátí kód jazyka uživatele (cze, eng, ...).

**showWindow** Zobrazí okno.

Parametry:

- title – nadpis okna.
- html – obsah okna.
- buttons – tlačítka v zápatí.
- width – šířka okna v pixelech.
- height – výška okna v pixelech.
- onOpen – funkce, která se zavolá po otevření okna, nebo „null“. Do funkce je předán objekt okna a název dialogu.
- onClose – funkce, která se zavolá po zavření okna, nebo „null“. Do funkce je předán název dialogu.
- escClose – true / false – zavřít okno tlačítkem ESC ano / ne.
- modal – true / false okno zobrazit jako modální / běžné.

#### **Příklad:**

```
buttons:{"Ahoj":function(){ alert('AHOJ!');},
"Zavřít":function(){ $
(this).dialog("close");}
```

**popupMenu** Zobrazí menu na pozici x, y.

Parametry:

- x – pozice v ose X.
- y – pozice v ose Y.
- pol – Seznam položek menu. Jednotlivé položky jsou definovány v tagu <LI>.
- cBack – vrací zvolenou položku z tagu <LI> nebo hodnotu „false“.

#### **Příklad:**

Prvek s ID „idTlac“, menu na kliknutí.

```
$("#idTlac").click(function(ev){
    var sezPol=<li data-id="1">Volba 1</li>
    <li data-id="2">Volba 2</li>
    <li data-id="-1">"+$parPl.api.getSysText(5,"Zavřít")+"</li>";
    $parPl.api.popupMenu(ev.pageX,ev.pageY,sezPol,function(vyb){
        if(vyb===false){return;}var idVP=$(vyb).attr("data-id");
        // v proměnné idVP máme ID položku, na kterou se kliklo ...
    });
});
```

**getSetDBData** Práce s daty.

Parametry:

- data – různé typy objektů (req, ...).
- cBack – volá se při dokončení a obsahuje vrácená data.

Data v podobě  
objektu „req“

Parametry (počet se liší v závislosti na použité funkci – viz níže):

- tl – nepovinný parametr. Obsahuje maximální dobu zpracování na serveru v sekundách.
- l – id lokality.
- ch – kanál.
- r – řádek.
- c – sloupec.
- rc – počet řádků.
- cc – počet sloupců.
- t1 – počáteční čas (formát RRRR-MM-DD HH:MM:SS).
- t2 – konečný čas (pokud není zadán, bere se aktuální čas).
- id(St) – id stanice.

Parametry odpovědi (počet se liší v závislosti na použité funkci – viz níže):

- OK – hodnota 1 = v pořádku, hodnota 0 = chyba.
- err – popis chyby. Vyskytuje se pouze v případě, kdy OK = 0.
- data – obdržená data. Vyskytuje se pouze v případě, kdy OK = 1.
- dur(ation) – délka vykonávání požadavku na serveru v ms.
- t – typ dat.
- v – hodnota kanálu.
- age – stáří dat v sekundách.
- lch – čas poslední změny hodnoty v kanálu.
- par – původní požadavek.
- id – id stanice pro provozní deník / lokalita\_kanál\_řádek\_sloupec.
- name – jméno stanice.
- l – id lokality.
- ch – kanál.
- offl – offline po v sekundách (viz lookdet\_a\_g\_cz\_xxx.pdf).
- ur – práva uživatele na stanici.
- ga – příznak globálního administrátora.
- aac – aktuální počet aktivních poruch.

### **Příklad:**

Tři požadavky r1, r2 a r3. Název je v moci pluginu – podle něj bude zpracovaná odpověď.

Tvar souhrnného požadavku:

```
{req:{r1:{f:"getCurValue",l:0,ch:1,r:1,c:3},r2:{f:"getCurHandle",l:0,ch:3},r3:{f:"nesmysl"}},tl:1}
```

Popis požadavku:

```
r1:{f:"getCurValue",l:0,ch:1,r:1,c:3}
```

Interní název r1, požadovaná funkce `getCurValue` (viz níže), lokalita (l) 0, kanál (ch) 1, řádek (r) 1, sloupec (c) 3.

```
r2:{f:"getCurHandle",l:0,ch:3}
```

Interní název r2, požadovaná funkce `getCurHandle` (viz níže), lokalita (l) 0, kanál (ch) 3.

```
r3:{f:"nesmysl"}
```

Interní název r3, funkce `nesmysl` (viz níže), žádné další parametry.

Odpověď:

OK:1 – operace se zdařila

dur:3.151 – operace trvala 3,151 ms

ts:12345678 – unix čas serveru před čtením dat (lze využít pro čtení hodnot změněných od při příštím dotazu)

Parametr data obsahuje všechny požadavky (zde r1-r3)

```
data:{r1:{},r2:{},r3:{}}
```

Každý parametr pak znovu svůj výsledek OK a vlastní data, následované otiskem původního požadavku par.

```
r1:{OK:1,data:[{t:"2018-02-26 08:14:01",v:"0.500",age:"XXX",lch:"2018-02-26 07:10:00"},par:{f:"getCurValue",l:"0",ch:"1",r:"1",c:"3"}}
```

Jednotlivé položky odpovědi – část data obsahuje pole hodnot:

```
[{t:"2018-02-26 08:14:01",v:"0.500"}]
```

Požadavek r2 obsahuje pole více hodnot, proto jsou jednotlivé položky rozšířeny o parametry r(řádek) a c(sloupec)

```
r2:{OK:1,data:[{r:"1",c:"3",t:"2018-02-26 08:14:01",v:"0.500"}...],par:{f:"getCurHandle",l:"0",ch:"1"}}
```

Požadavek r3 je neplatný – výsledek OK = 0, `err` obsahuje popis chyby anglicky.

```
r3:{OK:0,err:"Wrong function name 'nesmysl'",par:{f:"nesmysl"}}}
```

Dostupné funkce a seznam parametrů:

*Funkce* `getCurValue` Získání jedné hodnoty.

Parametry:

- l – lokalita,
- ch – kanál,
- r – řádek,
- c – sloupec.

**Příklad:**

Dotaz:

```
{req:{r1:{f:"getCurValue",l:0,ch:1,r:1,c:3}},tl:1}
```

### Odpověď:

```
r1:{OK:1,data:[0:{t:"2018-10-03 11:09:15",age:"437118",v:"0.500",lch:"2018-10-02 13:19:59"}],par:{f:"getCurValue",l:0,ch:1,r:1,c:3}},dur:2.523}
```

*Funkce* **getCurValues** Získání pole hodnot, zapsaných do **Lookdetu** od určitého času.

#### Parametry:

- l – lokalita,
- ch – kanál,
- r – řádek,
- c – sloupec,
- t1 – počáteční čas (formát RRRR-MM-DD HH:MM:SS).

### **Příklad:**

#### Dotaz:

```
{req:{r1:{f:"getCurValues",list:[{l:7,ch:31500,r:0,c:0},{l:7,ch:31501,r:3,c:4}],from:"2018-09-20 00:10:10"}},tl:1}
```

### Odpověď:

```
r1:{OK:1,data:{r1:{OK:1,data:[0:{t:"2018-10-03 11:09:15",age:"439335",v:"301.000",lch:"2018-10-02 13:19:58,id:"7_31500_0_0"},1:{t:"2018-10-03 11:09:15",v:"300.000",lch:"2018-10-02 13:19:58,id:"7_31501_1_0"}],par:{f:"getCurValues",from:"2018-09-20 00:10:10",list:"list"}},dur:2.523}}
```

Id je lokalita\_kanal\_radek\_sloupec.

*Funkce* **getCurHandle** Získání celého kanálu.

#### Parametry:

- l – lokalita,
- ch – kanál.

### **Příklad:**

#### Dotaz:

```
{req:{r1:{f:"getCurHandle",l:0,ch:1}},tl:1}
```

### Odpověď:

```
r1:{OK:1,data:[0:{r:"0",c:"0",t:"2018-02-26 08:14:01",age:"440809",v:"0.500"}],par:{f:"getCurHandle",l:0,ch:1},dur:2.589}
```

*Funkce* **getCurRow** Získání celého řádku kanálu.

#### Parametry:

- l – lokalita,
- ch – kanál,
- r – řádek.

### **Příklad:**

#### Dotaz:

```
{req:{r1:{f:"getCurRow",l:0,ch:1,r:2}},tl:1}
```

### Odpověď:

```
r1:{OK:1,data:[0:{c:"0",t:"2018-02-26 08:14:01",age:"441555",v:"0.500",lch:"2018-02-26 07:10:00"}...],par:{f:"getCurRow",l:"0",ch:"1",r:"2"},dur:2.568}
```

*Funkce* **Získání celého sloupce kanálu.**  
*getCurCol*

#### Parametry:

- l – lokalita,
- ch – kanál,
- c – sloupec.

#### **Příklad:**

##### Dotaz:

```
{req:{r1:{f:"getCurCol",l:0,ch:1,c:0},tl:1}}
```

### Odpověď:

```
r1:{OK:1,data:[0:{r:"0",t:"2018-02-26 08:14:01",age:"441555",v:"0.500",lch:"2018-02-26 07:10:00"}...],par:{f:"getCurCol",l:"0",ch:"1",c:"0"},dur:2.568}
```

*Funkce* **Získání výřezu kanálu.**  
*getCurMatrix*

#### Parametry:

- l – lokalita,
- ch – kanál,
- r – řádek,
- c – sloupec,
- rc – počet řádků,
- cc – počet sloupců.

#### **Příklad:**

##### Dotaz:

```
{req:{r1:{f:"getCurMatrix",l:0,ch:1,r:2,c:3,rc:2,cc:2},tl:1}}
```

### Odpověď:

```
r1:{OK:1,data:[0:{r:"2",c:"3",t:"2018-02-26 08:14:01",age:"442215",v:"0.500"}...],par:{f:"getCurMatrix",l:"0",ch:"1",r:"2",c:"3",rc:"2",cc:"2"},dur:2.999}
```

*Funkce* **Získání hlavičky provozního deníku zařízení firmy AMiT.**  
*getPDAMiTHed*

#### Parametry:

- id – ID stanice s provozním deníkem.

#### data obsahují:

- lng – kód jazyka (cze, eng, ...).
- id – kód hlášení v provozním deníku.
- tabdw – textový identifikátor tabulky DW (viz nápověda k části PseDet vývojového prostředí DetStudio).
- tabf – textový identifikátor tabulky F (viz nápověda k části PseDet vývojového prostředí DetStudio).

- prefix – Řetězec s krátkým prefixem pro daný kód záznamu, platné jsou první tři znaky (viz nápověda k části PseDet vývojového prostředí DetStudio).
- text – text hlášení.
- text1 – text komentáře.

### **Příklad:**

Dotaz:

```
{req:{r1:{f:"getPDAMiTHead",id:485,t1:1}}
```

Odpověď:

```
r1:{OK:1,data:[0:{lng:"cze",id:"0",tabdw:"",tabf:"",prefix:"OK",text:"Žádná  
chyba",text1:"Žádná chyba"}...],par:{f:"getPDAMiTHead",id:485,dur:3.695}}
```

*Funkce  
getPDAMiTDW*

Získání tabulky DW provozního deníku zařízení firmy AMiT.

Parametry:

- id – ID stanice s provozním deníkem.

data obsahují:

- lng – kód jazyka (cze, eng, ...).
- name – jméno tabulky DW (viz nápověda k části PseDet vývojového prostředí DetStudio).
- from – dolní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio).
- to – horní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio).
- text – text z tabulky DW (viz nápověda k části PseDet vývojového prostředí DetStudio).

### **Příklad:**

Dotaz:

```
{req:{r1:{f:"getPDAMiTDW",id:485,t1:1}}
```

Odpověď:

```
r1:{OK:1,data:[0:{lng:"cze",name:"Buffer",from:"0",to:"0",text:"NOSu"}...],  
par:{f:"getPDAMiTDW",id:485,dur:30.683}}
```

*Funkce  
getPDAMiTF*

Získání tabulky F provozního deníku zařízení firmy AMiT.

Parametry:

- id – ID stanice pro provozní deník.

data obsahují:

- lng – kód jazyka (cze, eng, ...).
- name – jméno tabulky F (viz nápověda k části PseDet vývojového prostředí DetStudio).
- from – dolní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio).
- to – horní mez pro zobrazení textu (viz nápověda k části PseDet vývojového prostředí DetStudio).
- text – text z tabulky F (viz nápověda k části PseDet vývojového prostředí DetStudio).

### **Příklad:**

Viz funkce `getPDAMiTData` výše.

*Funkce* `getPDAMiTData` Získání dat z provozního deníku zařízení firmy AMiT.

Parametry:

- `id` – ID stanice pro provozní deník,
- `t1 (t2)` – počáteční čas (konečný čas).

data obsahují:

- `t` – čas vzniku hlášení v provozním deníku.
- `code` – kód hlášení v provozním deníku (viz nápověda k části `PseDet` vývojového prostředí `DetStudio`).
- `segment` – místo vzniku hlášení v programu (viz nápověda k části `PseDet` vývojového prostředí `DetStudio`).
- `offset` – číslo procesu, kde došlo k hlášení/chybě (viz nápověda k části `PseDet` vývojového prostředí `DetStudio`).
- `wid` – WID proměnné/doplňkový kód (viz nápověda k části `PseDet` vývojového prostředí `DetStudio`).
- `data` – WID proměnné/doplňkový kód (viz nápověda k části `PseDet` vývojového prostředí `DetStudio`).

Bližší popis viz nápověda k části `PseDet` vývojového prostředí `DetStudio`.

### **Příklad:**

Dotaz:

```
{req:{r1:{f:"getPDAMiTData",id:485,t1:"2018-09-25 12:55:30",t1:1}}
```

Odpověď:

```
r1:{OK:1,data:[0:{t:"2018-10-10 13:35:02",code:"20",segment:"0",offset:"0",wid:"-1"}...],par:{f:"getPDAMiTData",id:485,t1:"2018-09-25 12:55:30",t2:null},dur:3.673}
```

*Funkce* `getArchiveData` Získání archivních dat kanálu.

Parametry:

- `l` – lokalita,
- `ch` – kanál,
- `r` – řádek,
- `c` – sloupec,
- `t1 (t2)` – počáteční čas (konečný čas).

### **Příklad:**

Dotaz:

```
{req:{r1:{f:"getArchiveData",l:7,ch:31500,r:0,c:0,t1:"2018-10-01 00:10:10",t1:1}}
```

Odpověď:

```
r1:{OK:1,data:[0:{v:"15",t:"1538384656"}...],par:{f:"getArchiveData",l:"7",ch:31500,r:0,c:0,t1:"2018-10-01 00:10:10"},dur:2.999}
```

*Funkce* Získání záznamu nebo záznamů z archívu.  
*getArchiveInTime*

Parametry:

- l – lokalita,
- ch – kanál,
- r – řádek,
- c – sloupec,
- t – čas od kterého se získávají záznamy,
- age – maximální stáří dat ve dnech (výchozí 1).

Pozor operace může být časově velmi náročná (s rostoucím age – max. stářím dat)!

### **Příklad:**

Dotaz:

```
{req:{r1:{f:'getArchiveInTime',list:[{l:1,ch:1,r:2,c:3},{l:1,ch:2,r:3,c:4}],time:'2018-01-01 00:10:10',age:31}},tl:100}
```

Odpověď:

Totožná s „getCurValues“, pokud hodnota v archívu není ani po „age“ dní zpět, bude mít parametr „v“ a „time“ hodnotu „null“.

*Funkce* Zápis hodnoty do databáze.  
*setCurValue*

Parametry:

- l – lokalita,
- ch – kanál,
- r – řádek,
- c – sloupec,
- t – typ dat (viz níže),
- v – hodnota kanálu.

Typ dat

| ID | Typ                            | Poznámka                                                |
|----|--------------------------------|---------------------------------------------------------|
| 1  | Integer 8 bit                  | -128 až 127                                             |
| 2  | Unsigned integer 8 bit         | 0 až 255                                                |
| 3  | Integer 16 bit                 | -32 768 až 32 767                                       |
| 4  | Unsigned integer 16 bit        | 0 až 65 535                                             |
| 5  | Long 32 bit                    | -2 147 483 648 až 2 147 483 647                         |
| 6  | Unsigned long 32 bit           | 0 až 4 294 967 295                                      |
| 7  | Double integer 64 bit          | -9 223 372 036 854 775 808 až 9 223 372 036 854 775 807 |
| 8  | Unsigned double integer 64 bit | 0 až 18 446 744 073 709 551 615                         |
| 9  | Real 32 bit                    | $1E^{-37}$ až $1E^{+37}$                                |
| 10 | Double real 64 bit             | $1E^{-307}$ až $1E^{+308}$                              |

### **Příklad:**

Dotaz:

```
{req:{r1:{f:'setCurValue',l:7,ch:41235,r:0,c:0,t:9,v:123},tl:1}
```

Odpověď:

```
r1:{OK:1,},par:{f:'setCurValue',l:7,ch:41235,r:0,c:0,t:9,v:123},dur:2.999}
```

*Funkce* **putUserData** Uložení uživatelských dat, nebo nastavení do databáze **LookDetu** (maximální velikost je 64 kB).

Parametry:

- val – hodnota pro uložení (max. 64 kB),
- key – max. 10 znaků, identifikace proměnné pluginem,
- idPrev – nepovinný, pokud není nastavení vázáno ke konkrétnímu náhledu, je globální pro uživatele.

*Funkce* **getUserData** Získání dříve uložených uživatelských dat z databáze **LookDetu**.

Parametry:

- key – max. 10 znaků,
- idPrev – nepovinný, pokud není nastavení vázáno ke konkrétnímu náhledu, je globální pro uživatele.

*Funkce* **getAssignedStations** Získání jména stanic(e) přiřazené kanálu / kanálům.

Parametry:

- l – lokalita,
- ch – kanál.

**Příklad:**

Dotaz:

```
{req:{r1:{f:"getAssignedStations",list:[{l:1,ch:100},{l:2,ch:200}]}}}
```

Odpověď:

```
r1:{OK:1,data:[0:{id:"487",name:"XK_21",ch:"100",offl:"300",ur:65535}...],par:{f:"getAssignedStations",list:"list"},dur:2.864}
```

Seznam oprávnění na stanice

| Bit | Význam                                                                       |
|-----|------------------------------------------------------------------------------|
| 1   | Uživatel je oprávněn k zobrazení parametrů z řídicího systému.               |
| 2   | Uživatel je oprávněn ke změně parametrů v řídicím systému.                   |
| 3   | Uživatel je oprávněn k zobrazení poruch.                                     |
| 4   | Uživatel je oprávněn k převzetí poruchy.                                     |
| 5   | Zasílat informace o vzniku poruchy na uživatelův e-mail.                     |
| 6   | Zasílat vzniklé poruchy na uživatelské zařízení Android připojené k systému. |
| 7   | Uživatel smí nahlížet do provozního deníku stanice.                          |

*Funkce* **getAlarmStatus** Získání počtu poruch.

Parametry:

- id – ID stanice.

**Příklad:**

Dotaz:

```
{req:{r1:{f:"getAlarmStatus",list:[{487,600}]},t1:1}
```

Odpověď:

```
r1:{OK:1,data:[0:{aac:"19",idS:"487"}...],par:{f:"getAlarmStatus",list:[487,600]},dur:3.204}
```

**getFromStation** Získání dat ze stanice AMiT.

Parametry:

- id – identifikátor požadavku. Pro kontrolu je vrácen v odpovědi v původní podobě.
- params – různé typy objektů (req, ...). Začíná vždy id stanice, za kterým je uveden jeden či více požadavků (dotazů na data).
- cBack – funkce, která se volá při dokončení a obsahuje vrácená data. Odpověď je vrácena jako parametr předané funkce „cBack“. Pokud selže kontrola parametrů před odesláním na server, vrátí false, jinak true.

Struktura požadavku:

- idSt – id stanice,
- channel – WID proměnné ve stanici AMiT,
- type – typ proměnné ve stanici AMiT (viz tabulka níže),
- row – počáteční řádek matice,
- col – počáteční sloupec matice,
- rows – počet řádků matice,
- cols – počet sloupců matice.

Typy proměnných

| Hodnota | Typ dat        |
|---------|----------------|
| 0       | Integer        |
| 1       | Long           |
| 2       | Float          |
| 3       | Matice Integer |
| 4       | Matice Long    |
| 5       | Matice Float   |

**Příklad:**

Pro kontrolu je nastaveno id čtení na 123. Ze stanice s idSt 7 čteme matici Float (type=5) s WIDem (channel) 1082, od řádku 2(row), od sloupce 2(col), počet řádků 4(rows), počet sloupců 4(cols). Ze stejné stanice potom ještě kanál 1083, typ Long (1), řádky, sloupce se nemusejí uvádět. Ve stejném dotazu načteme ještě ze stanice 9 kanál 1000 typu Long(1).

Dotaz:

```
{id:"123",req:[{idSt:7,req:[{channel: 1082,type:5,row:2,col:2,rows:4,cols:4},{channel:1083,type:1}]],{idSt:9,req:[{channel:1000,type:1}]]}
```

Odpověď:

```
{id:123,OK:1,data:[{OK:1,data:[0,0,0,0],[1,1,1,1],[2,2,2,2],[3,3,3,3]},{OK:0,err:-12,errd:"Chyba od AMiT"},{OK:0,err:-999,errd:"Nenalezeno v db..."}];
```

Odpověď byla načtena ze serveru v pořádku. 1. část dat je v pořádku, data obsahují pole načtených hodnot – 4 řádky a 4 sloupce, 2 a 3. část obsahuje

chyby (`OK = 0`, `err` obsahuje kód a `errrd` popis chyby). V případě nematcové proměnné je vráceno přímo číslo bez pole (ne `[100]` ale `100`).

**putToStation** Zápis dat do stanice AMiT.

Parametry:

- `id` – identifikátor požadavku. Pro kontrolu je vrácen v odpovědi v původní podobě.
- `note` – poznámka uložená v logu při zápisu (viz `lookdet_a_g_cz_xxx.pdf`)
- `params` – různé typy objektů (`req`, ...). Začíná vždy `id` stanice, za kterým je uveden jeden či více požadavků (hodnot pro zápis).
- `onResult` – funkce, do které se předá výsledek zápisu. Pokud selže kontrola parametrů před odesláním na server, vrátí `false`, jinak `true`.

Struktura požadavku:

- `idSt` – `id` stanice,
- `name` – popis parametru,
- `channel` – WID proměnné ve stanici AMiT,
- `type` – typ proměnné ve stanici AMiT (viz tabulka níže),
- `row` – počáteční řádek matice,
- `col` – počáteční sloupec matice,
- `rows` – počet řádků matice,
- `cols` – počet sloupců matice,
- `old` – původní hodnoty načtené ze stanice (pro zápis do logu – viz `lookdet_a_g_cz_xxx.pdf`),
- `new` – nové hodnoty, které budou zapsány do stanice.

### Příklad:

Příklad zápisu do 2 stanic:

Pro kontrolu je nastaveno `id` zápisu na 123. Do logu se uloží text: „Poznámka k zápisu“. Pole `req` obsahuje jednotlivé požadavky na stanice. Do stanice s `idSt` 7 se zapíše matice 4×4. Do stanice s `idSt` 3 se zapíše proměnná typu Long. Parametr `name` obsahuje popis „zapis X“.

Dotaz:

```
{id:"123","note":"Poznámka k zápisu",req:[{idSt:7,req:[{name:"zapis 1",channel:1082,type:5,row:2,col:2,rows:4,cols:4,old:[[0,0,0,0],[1,1,1,1],[2,2,2,2],[3,3,3,3]],new:[[1,2,3,4],[5,6,7,8],[9,8,7,6],[5,4,3,2]]}],{idSt:3,req:[{name:"zapis 2",channel:1000,type:0,old:12345,new:12346}]}]}
```

Odpověď:

```
{id:"123",OK:1,data:[{OK:1},{OK:0,err:-12,errrd:"Spatne parametry"}]}
```

Zápis do stanice 7 byl úspěšně proveden, do stanice 3 selhal s chybou -12 (špatné parametry).

**showAlarms** Zobrazení poruch.  
Funkce je bez parametrů.

**pluginPHP** Zavolání vlastní PHP funkce na serveru. Funkce jsou definovány v PHP souborech a jsou součástí pluginu.

**Pozor!** PHP musí začínat „<?php namespace LookDetPlugin;“.

Parametry:

- func – název volané PHP funkce.
- params – (pole, nebo „null“) – jsou předány PHP funkci.
- cBack – funkce, která je zavolána po vykonání na PHP serveru a je do ní předána odpověď v JSON tvaru.

Odpověď v JSON tvaru může být např.

- při chybě:

```
{OK:0,err:"POPIS CHYBY",dur:0.14}
```

OK = 0 – znamená, že došlo k chybě.

err – obsahuje bližší popis chyby.

dur – informuje o době vykonávání na straně serveru.

- při správném vykonání:

```
{OK:1,res:ODPOVED_PHP_FUNKCE,dur:0.14}
```

OK = 1 – znamená, že vše proběhlo v pořádku.

res – obsahuje odpověď volané PHP funkce.

dur – informuje o době vykonávání na straně serveru.

### **Příklad 1:**

Funkci „sectiAaB“ je předán jeden parametr v podobě objektu.

javascript:

```
$parPl.api.pluginPHP('sectiAaB', [{a:1,b:2}],function(data){
    console.log(data);
});
```

php:

```
<?php namespace LookDetPlugin;
```

```
function sectiAaB($par){
    return $par->a+$par->b;
}
```

### **Příklad 2:**

Funkci „sectiAaB“ jsou předány 2 parametry.

javascript:

```
$parPl.api.pluginPHP('sectiAaB', [1,2],function(data){
    console.log(data);
});
```

php:

```
<?php namespace LookDetPlugin;
```

```
function sectiAaB($a,$b){
    return $a+$b;
}
```

**Příklad 3:**

Funkce vrátí, co se jí předá – jeden parametr.

javascript:

```
$parPl.api.pluginPHP('myEcho',['ahoj'],function(data){  
    console.log(data);  
});
```

php:

```
<?php namespace LookDetPlugin;  
  
function myEcho($co){  
    return $co;  
}
```

**Příklad 4:**

Funkce bez parametru.

javascript:

```
$parPl.api.pluginPHP('vratDatum',null,function(data){  
    console.log(data);  
});
```

php:

```
<?php namespace LookDetPlugin;  
  
function vratDatum(){  
    return date();  
}
```

**addButton** Doplní tlačítko na pravou stranu navigační lišty.

Parametry:

- title – text, který se zobrazí při najetí myši (hint).
- icon – ikona tlačítka. Více zde: (<https://fontawesome.com/v4/icons/> ). Název ikony se zadává tak jak je uveden v seznamu dostupných (např.: 'edit', 'cog' a pod.).
- onClick – funkce, která se vyvolá po kliknutí na tlačítko.

Funkce vrátí DOM tlačítka – při úspěchu, nebo false, pokud se nezdaří tlačítko přidat. Tlačítko doporučujeme přidávat ve funkci „\$parPl.cb.loadPreview“.

**Příklad:**

```
{Var tl=$parPl.api.addButton('Jsem tlačítko!','cog',function(){console.log('KLIK  
NA TLACITKO!')}};
```

**convertTime  
ToShow** Vrátí čas stanice v časovém pásmu, do kterého je stanice přiřazena, nebo v časovém pásmu webového prohlížeče. Funkce zohledňuje nastavení LookDetu (podle času webového prohlížeče / podle času stanice).

Parametry:

- idSt – id stanice,
- engNow – převáděný čas ve formátu RRRR-MM-DD HH:MM:SS,
- retInLocalTime volba, zda bude čas vrácen v časovém pásmu stanice (false) nebo v časovém pásmu webového prohlížeče (true).

**convertTime** Vrátí místní čas webového prohlížeče odpovídající zadanému času serveru  
**ServerToLocal LookDet.**

Parametry:

- engNow – převáděný čas ve formátu RRRR-MM-DD HH:MM:SS,
- retInLocalTime – volba, zda bude čas vrácen v časovém pásmu stanice (false) nebo v časovém pásmu webového prohlížeče (true).

Grafické naznačení funkce: čas serveru -> čas prohlížeče / čas serveru -> čas stanice.

**convertTime** Vrátí čas serveru **LookDet** odpovídající zadanému místnímu času prohlížeče.  
**LocalToServer**

Parametry:

- engNow – převáděný čas ve formátu RRRR-MM-DD HH:MM:SS,
- retInLocalTime – volba, zda bude čas vrácen v časovém pásmu stanice (false) nebo v časovém pásmu webového prohlížeče (true).

Grafické naznačení funkce: čas stanice -> čas prohlížeče / čas serveru -> čas serveru.