Rozhraní REST systému REXYGEN

Referenční příručka

REX Controls s.r.o.

Verze 3.0.1
Plzeň
1.3.2024

Obsah

Kapitola 1

Úvod

Počínaje verzí 2.50, systém REXYGEN poskytuje jednoduché komunikační rozhraní využívající HTTP protokol a jiné webové technologie, souhrnně nazývané REST API. REST API je zabudované přímo do REXYGENu, a proto nejsou pro zprovoznění potřeba žádné speciální aplikace či nástroje. REST API umožňuje řízení systému REXYGEN buď pomocí vnějších nástrojů podporujících HTTP protokol, nebo přímo z webového prohlížeče, bez potřeby využití jakýchkoli specializovaných nástrojů.

Toto API zužitkovává jednoduchý vestavěný webový server, který byl zároveň také zabudován do REXYGENu, počínaje verzí 2.11. HTTP protokol běží na serveru při základním nastavení na portu 8008 a HTTPS (HTTP přes SSL vrstvu) na portu 8009. Oba dva porty mohou být přenastaveny a webový server může být zapnut či vypnut v nastaveních. Pro detailní informace viz [1]

Kapitola 2

Základní principy

REST API bylo vyvinuté se zřetelem k následujícím klíčovým faktorům: výkonu, jednoduchosti, možnosti procházení a bezpečnosti. Tyto principy tvoří finální strukturu a mechanismus API.

Výkon je zajištěn pomocí vysoce integrovaného a vestavěného webového serveru bez využití jakéhokoli podpůrného aplikačního rámce mezi serverem a jádrem REXYGEN systému. I přes přímou integraci serveru do jádra je server oddělen od jádra tenkým rozhraním, aby možnost přímé interakce s jádrem běžícím v reálném čase byla co nejmenší a bylo možno server případně oddělit do samostatné aplikace. Webový server běží na specifickém vlákně s upravitelnými prioritami využívajícím komunikace přes asynchronní soket.

Z hlediska struktury je výkon dosažen rozlišením *dat* a *metadat*, přičemž operací s již přenesenými daty je minimum. Rozlišuje se krátký a dlouhý formát odpovědi, takže v případě, že ke splnění účelu stačí krátký formát odpovědi, množství přenesených dat může být redukováno. Uživateli je doporučeno využívat data ve formátu JSON z důvodu maximalizace výkonu. Zároveň je formát JSON momentálně jediným formátem podporujícím zápisové (POST) požadavky, přestože získat data (požadavek GET) lze i ve formátech HTML a XML a ve formátu čistého textu.

Principy jednoduchosti a možnosti vyhledávání se projevují přímo ve struktuře API. Tato struktura je podrobněji popsaná v kapitole 4. API se snaží co nejpřesněji odrážet v reálném čase strukturu a vazby řídícího algoritmu. Téměř všechna pravidla a zvyklosti, které uživatel zná z využívání systémů REXYGEN, REXYGEN Studio a jeho diagnostiky, mohou být aplikována. Možnost vyhledávání zjednodušuje novému uživateli využívání REST API REXYGEN systému bez nutnosti detailního studia uživatelského manuálu a bez nutnosti použití specializovaných nástrojů. Každý se základními znalostmi HTTP protokolu a moderního webového prohlížeče může započít využívat REST API systému REXYGEN. Každý rozhraním poskytnutý zdroj může být nalezen, uveden a zkoumán pomocí standardního moderního webového prohlížeče.

Server byl také sestaven na základě respektu k bezpečí a soukromí. Přestože je webový server nedílnou součástí řídícího systému, tenké rozhraní mezi jádrem a serverem činí šanci vnějšího zásahu zanedbatelnou. Všechna data jsou důsledně zkontrolována, paměť

je nulována a množství přidělené paměti je minimalizováno. Bezpečnost je zajištěna využitím standardního HTTPS (HTTP přes SSL vrstvu) protokolu a poskytnutím zatím základní limitované uživatelské autentizace.

Kapitola 3

Umístění

REST API se nachází na URL adrese ve formátu `http://host:port/api`, pokud je využíván HTTP server, případně na URL adrese `https://host:port/api`, pokud je využíván HTTPS server. Viz [1] pro více informací o konfiguraci serveru. Základním nastavením je port 8008 přidělen pro HTTP připojení a port 8009 pro HTTPS připojení.

Uživatel má možnost okamžitě začít využívat a interagovat s REST API zadáním korespondující URL adresy do webového prohlížeče.

Kapitola 4

Struktura zdrojů

Všechny záznamy dostupné v REST API jsou umístěny ve stromové struktuře jako zdroje. Zdroje jsou identifikovány buď dle jejich cesty umístění, nebo, a to častěji, podle URL adresy. Zdroje mohou být čteny nebo aktualizovány pomocí standardních HTTP požadavků. Jsou dostupné požadavkem GET v HTML formátu a tím pádem mohou být zobrazeny a prohlíženy přímo ve webovém prohlížeči. Každý zdroj může nést jakýkoli z těchto typů informací:

- **data** – vyjadřuje skutečnou *hodnotu (value)*, *kvalitu (quality)* a *časovou značku (time-stamp)* parametru konfigurace nebo signálu, kterým může být například vstup, výstup, parametr atd.,
- **metadata** – vyjadřuje informaci o vlastnostech *dat (data)*, jako třeba druh (type), rozsah (range) atd.,
- **seznam podřízených zdrojů (children list)** – vyjadřuje informaci o ostatních zdrojích, které patří pod vybraný zdroj jako zdroje podřízené. Tento seznam je součástí *metadat (metadata)*.

4.1 Strom zdrojů

Tento strom se nachází na adrese `http://host/api`. První úroveň reprezentuje cílové zařízení a neposkytuje žádná data. Obsahuje následující podřízené zdroje:

- **moduly (modules)** – poskytuje seznam modulů načtených cílovou exekutivou,
- **archivy (archives)** – poskytuje seznam archivů používaných cílovou exekutivou,
- **drivery (drivers)** – poskytuje seznam driverů používaných cílovou exekutivou,
- **úrovně (levels)** – poskytuje seznam prioritních úrovní v cílové exekutivě,
- **úlohy (tasks)** – poskytuje seznam úloh v cílové exekutivě,

- **data** – poskytuje přístup k signálům v cílové exekutivě v cílově specifickém jmenném prostoru,
- **systém (system)** – poskytuje informace o cílovém zařízení,
- **informace (info)** – poskytuje informace o cílové exekutivě.

4.1.1 moduly (modules)

Zdroj *moduly (modules)* poskytuje pouze seznam načtených modulů. Žádné jiné informace nejsou v současné verzi dostupné.

4.1.2 archivy (archives)

Zdroj *archivy (archives)* poskytuje pouze seznam archivů používaných cílovou exekutivou. Žádné jiné informace nejsou v současné verzi dostupné.

4.1.3 drivery (drivers)

Zdroj *drivery (archives)* poskytuje pouze seznam driverů používaných cílovou exekutivou. Žádné jiné informace nejsou v současné verzi dostupné.

4.1.4 úrovně (levels)

Zdroj *úrovně (levels)* poskytuje pouze seznam prioritních úrovní používaných cílovou exekutivou. Žádné jiné informace nejsou v současné verzi dostupné.

4.1.5 úlohy (tasks)

Zdroj *úlohy (tasks)* poskytuje pouze seznam úloh v cílové exekutivě. Každá úloha je nadřazeným zdrojem se svými funkčními bloky jako podřízenými zdroji. Každý subsystém je zároveň také nadřazeným zdrojem s vlastními podřízenými funkčními bloky.

4.1.6 data

Zdroj *data* zpřístupňuje veškeré zdroje v cílovém jmenném prostoru, tj. pojmenování specifické pro cíl. V případě systému REXYGEN je použité standardní pojmenování signálů REXYGENu.

4.1.7 systém (system)

Zdroj *systém (system)* poskytuje různé informace o cílovém zařízení včetně verzí softwaru a hardwaru a identifikace platformy.

4.1.8 informace (info)

Zdroj *informace (info)* poskytuje informace o cílové exekutivě včetně názvu, popisu, jména autora, jména společnosti a několika kontrolních součtů a ID, které jednoznačně určují cílovou exekutivu.

Kapitola 5

Požadavky (requests)

Tato kapitola popisuje HTTP požadavky, které jsou podporované REST API.

5.1 Typy požadavků

V současné verzi jsou podporovány požadavky GET a POST. Požadavek GET je využíván pro čtení zdrojů a nemá dopad na stav cílového zařízení nebo běžící exekutivy. Stejná hodnota je vrácena ve chvíli, kdy je vysláno více požadavků GET a stav cílového zařízení nebo exekutivy není změněn. Požadavek POST je používán pro nastavení hodnoty zdroje. Pouze **data** mohou být tímto požadavkem upravena, není možné jakýmkoli způsobem upravit **metadata**. Modifikovaný zdroj musí být modifikovatelný a uživatel musí být autorizován a musí mít příslušná oprávnění při modifikaci zdroje pomocí požadavku POST.

5.2 Podporované datové formáty (MIME typy)

REST API podporuje MIME typy, které jsou uvedeny v následujícím seznamu. Podporovaný MIME typ může být specifikován v poli `Accept` záhlaví HTTP požadavku.

- `text/html` – HTML,
- `application/json` – JSON,
- `text/xml` – XML,
- `text/plain` – čistý text,
- `text/csv` – seznam hodnot oddělené čárkou.

Momentálně jsou v POST požadavcích podporovány pouze formáty `application/json` a `application/x-www-form-urlencoded`. Tyto formáty mohou být specifikovány v poli `Content-Type` záhlaví HTTP požadavku.

Kapitola 6

Dotazy (queries)

Tato kapitola obsahuje informace o dotazech v REST API.

6.1 O dotazech

Textový řetězec dotazu je přídatná informace připojená k URL adrese, která může ovlivnit typ a formát vrácených dat a jak jsou požadavky serverem zpracovány. URL adresa s takovýmto textovým řetězcem má následující formát:

```
http://host/api/resource?query
```

Dotaz může být parametrem s hodnotou, v tom případě má takovýto formát:

```
http://host/api/resource?parameter=value
```

V případě, že je přítomno více parametrů, je použit tento formát:

```
http://host/api/resource?parameter1=value1&parameter2=value2
```

6.2 Podporované dotazy

Tato sekce obsahuje informace o dotazech podporovaných REST API.

6.2.1 data

Dotaz `data` nemá hodnotu. Specifikuje, že mají být vrácena *data* daného zdroje.

6.2.2 meta

Dotaz `meta` nemá hodnotu. Specifikuje, že mají být vrácena *metadata* daného zdroje.

6.2.3 show

Dotaz `show` nemá hodnotu. Specifikuje, že data zdroje mají být zobrazena v uživatelsky přijatelném formátu. Tento dotaz je zatím podporován pouze u `TRND` funkčních bloků, kde dotaz `show` způsobí vrácení grafické reprezentace dat.

6.2.4 export

Dotaz `export` nemá hodnotu. Specifikuje, že data zdroje mají být vrácena ve formátu vhodném pro další operace. Tento dotaz je zatím podporován pouze u `TRND` funkčních bloků, kde dotaz `export` způsobí vrácení dat z vyrovnávací paměti ve specifikovaném souborovém formátu.

6.2.5 override

Dotaz `override` nemá hodnotu. Způsobuje, že zapisovaný signál je přepnut do *manuálního módu* neboli *módu lokálního přepsání*). Toto může být aplikováno například na vstupy funkčního bloku pro uvedení signálu do fixního stavu. Dotaz `override` je k dispozici pouze u POST požadavku.

6.2.6 format

Dotaz `format` má hodnotu, kterou může být buď `long` (dlouhý) nebo `short` (krátký), a která specifikuje detaily o vrácené informaci. U signálu je zobrazena pouze skutečná hodnota pokud `format=short`, ale kvalita (quality) signálu a časová značka (time-stamp) jsou zahrnuty navíc pokud `format=long`.

6.2.7 robot

Dotaz `robot` nemá hodnotu. Informuje server, že klientem není uživatel. Pokud je tento dotaz specifikován, server neodesílá odpovědi vyžadující interakci s uživatelem a předpokládá se základní autentizace. Tento dotaz by měl být specifikován vždy, pokud je pro interakci s REST API použit `wget` nebo `curl`.

6.2.8 mime

Dotaz `mime` má hodnotu specifikující MIME formát dat vrácených ze serveru, viz sekci 5.2 pro podporované MIME typy. Typ MIME specifikovaný dotazem `mime` má vždy přednost před polem `Accept` záhlaví HTTP požadavku.

Kapitola 7

Autentizace

Autentizace je další věcí podporovanou vestavěným webovým serverm REXYGEN systému. Viz [1] pro více informací jak zabezpečit cílové zařízení a zprovoznit autentizaci uživatele.

Rozlišujeme dva mechanismy uživatelské autentizace:

- **HTTP Cookie** – tento mechanismus je využit vestavěným HTML rozhraním,
- **HTTP Basic Authentication** – tento mechanismus by měl být využit při interakci s REST API za využití programu – viz sekci 6.2.7. URL adresa s uživatelskou autorizací má následující formát:

  ```
  http://user:password@host/api/resource
  ```

  Autentizační mechanismy by neměly být kombinovány. Požadavek selže, pokud je některý z obou využitých druhů autentizace nesprávný.

Literatura

[1] REX Controls s.r.o.. *RexCore – User manual*, 2020. →.