

API-Workshop

Eine Einführung in die
API-Entwicklung mit i-doit

Benjamin Heisig, IT Consultant bei der synetics GmbH
i-doit Anwenderkonferenz 2017 in Düsseldorf

Agenda

1. Motivation
2. Konfiguration
3. Technische Grundlagen
4. Methoden
5. Beispiele
6. Tipps und Tricks
7. Ressourcen
8. Q&A

Motivation



- Ressourcen einsparen, Arbeitserleichterung
- (Menschliche) Fehler vermeiden
- Robuste Schnittstellen

Technische Grundlagen

Create (Erstellen)

Read (Lesen)

Update (Aktualisieren)

Delete (Delete)

Technische Grundlagen

Was bedeutet **JSON-RPC**?

Technische Grundlagen

JSON: JavaScript Object Notation

- Schlankes Datenformat
- Ausdrucksstark
- Hauptsache kein XML ;-)

```
{  
  "attribute1": "string",  
  "attribute2": 123,  
  "attribute3": [1, 2, 3],  
  "attribute4": {  
    "key": "value"  
  }  
}
```

Technische Grundlagen

RPC: Remote Procedure Call

- Der Client sendet eine Anfrage an den Server, um eine Methode auszuführen
- Server sendet Ergebnis der Methode an Client zurück

Technische Grundlagen

JSON-RPC

- HTTP POST Request
- Content-Type: application/json

i-doit-spezifische Features

- API Key
- Zusätzliche Authentifizierung
- Logs
- Wahl der Sprache (en/de)
- Logbuch und Such-Index werden aktualisiert
- Bald™: Benutzerrechte werden berücksichtigt

Methoden

Methode	Beschreibung
<code>cmdb.objects</code>	Objekte auslesen
<code>cmdb.object.{create, read, update, delete}</code>	Ein Objekt erstellen, auslesen, aktualisieren, archivieren/als gelöscht markieren/unwiderruflich löschen

Methoden

Methode	Beschreibung
<code>cmdb.category.{create, read, update, delete}</code>	Einen Kategorie-Eintrag erstellen, lesen, aktualisieren, löschen
<code>cmdb.dialog.{create, read}</code>	Werte eines Dialog-Feldes erstellen oder auslesen

Methoden

Methode	Beschreibung
<code>cmdb.reports</code>	Einen Report ausführen
<code>cmdb.logbook.{create, read}</code>	Einen Logbuch-Eintrag erstellen, auslesen

Methoden

Methode	Beschreibung
<code>cmdb.objects_by_relation</code>	Objekte nach Beziehungsart auslesen
<code>cmdb.location_tree</code>	Standort-Pfad auslesen
<code>cmdb.workstation_components</code>	Arbeitsplatzkomponenten auslesen
<code>cmdb.impact</code>	Rekursive Abfrage aller Objektbeziehungen

Methoden

Methode	Beschreibung
<code>cmdb.object_types</code>	Objekttypen abfragen
<code>cmdb.category_info</code>	Attribute zu einer Kategorie auslesen
<code>cmdb.object_type_categories</code>	Auslesen, welche Kategorien zu welchen Objekttypen konfiguriert sind
<code>cmdb.object_type_groups</code>	Auslesen, welche Objekttypen welchen Objekttypgruppen zugeordnet sind

Methoden

Methode	Beschreibung
<code>idoit.version</code>	Version von i-doit abfragen
<code>idoit.login</code>	Login
<code>idoit.logout</code>	Logout
<code>idoit.constants</code>	Konstanten abfragen
<code>idoit.search</code>	In i-doit suchen

Beispiel

```
{
  "jsonrpc": "2.0",
  "method": "cmdb.object.create",
  "params": {
    "type": "C__OBJTYPE__SERVER",
    "title": "My little server",
    "apikey": "clia5q"
  },
  "id": 1
}
```

```
{
  "jsonrpc": "2.0",
  "result": {
    "id": "3351",
    "message": "Object was
              successfully created",
    "success": true
  },
  "id": 1
}
```

Beispiele

```
curl \  
--data '{"jsonrpc": "2.0", "method": "cmdb.object.create", "params": {"type":  
"C__OBJTYPE__SERVER", "title": "My little server", "apikey": "clia5q"}, "id":  
1}' \  
--header "Content-Type: application/json" \  
https://demo.i-doit.com/src/jsonrpc.php
```

Beispiele

- VMs provisionieren
- i-doit auf der Kommandozeile
- Komplexe Excel-Listen importieren
- Komplexe Reports erstellen
- IP-Adressen den richtigen Netzen zuordnen
- Datenaustausch mit Dritt-Applikationen
- u.v.m.

Tipps und Tricks

- API statt Datenbank
 - Zwei-Faktor-Authentifizierung nutzen
 - API Calls mitloggen
 - HTTPS
 - Login/Logout
 - Batch Requests
 - Kombination mit Events
- ⇒ Integrität gewährleisten
 - ⇒ Zusatz an Sicherheit
 - ⇒ Debugging
 - ⇒ Sensible Daten schützen
 - ⇒ Sessions wiederverwenden
 - ⇒ Rasend schnell

Ressourcen

- i-doit Knowledge Base:
 - Hauptartikel “[API \(JSON-RPC\)](#)”
 - [Beispiele zur Nutzung der API](#)
 - [Tipps und Tricks zur API](#)
 - [Kategorie-Felder für Datenarrays](#)
- Details für Wissensdurstige:
 - [JavaScript Object Notation \(JSON\)](#)
 - [Remote Procedure Call \(RPC\)](#)
 - [JSON-RPC](#)
- Libraries (Auswahl):
 - [i-doit API client](#) (PHP)
 - [i-doit-go-api](#) (Go)

Vielen Dank und weiterhin viel Spaß auf der i-doit Anwenderkonferenz 2017!

Kontakt

E-Mail:

bheisig@i-doit.com

Twitter:

[@bheisig](https://twitter.com/bheisig)

Meine Vorträge

- **Das Analyse-Add-on im Detail**
Sinn, Zweck und Einsatzmöglichkeiten
Tag 1, 15 Uhr, Lounge 2
- **Kabelmanagement und IPAM mit i-doit**
Funktionalitäten im praktischen Einsatz
Tag 2, 13 Uhr, Lounge 2
- **IT-Grundschutz mit i-doit**
Daten aus GSTOOL importieren und weiterverwenden
Tag 2, 14 Uhr, Lounge 3
- **Optimierung der Systemumgebung**
Tuning, Performance und Stabilität bei Webserver, Datenbank und PHP
Tag 2, 15 Uhr, Lounge 3