JSON-RPC

A JSON-RPC (JavaScript Object Notation – Remote Procedure Call) egy egyszerű és könnyű súlyú távoli eljáráshívási (RPC) protokoll, amely az üzenetek formátumaként a JSON adatformátumot használja. A protokoll célja, hogy egy kliens egy távoli szerveren található függvényeket vagy metódusokat úgy tudjon meghívni, mintha azok helyben futó függvények lennének.

A REST-alapú webszolgáltatásokkal szemben, amelyek erőforrás-orientált szemléletet követnek, a JSON-RPC eljárás-orientált megközelítést alkalmaz. Ez azt jelenti, hogy a kliens közvetlenül egy megnevezett metódust hív meg, és átadja a szükséges paramétereket.

A JSON-RPC legfontosabb tulajdonságai a következők:

- **Transport-független** – A protokoll csak az üzenetek formátumát határozza meg, a kommunikációs csatornát nem. A JSON-RPC használható például HTTP, HTTPS, WebSocket vagy TCP felett, de akár üzenetsorok vagy beágyazott rendszerek soros kommunikációja felett is. Emiatt könnyen integrálható különböző rendszerarchitektúrákba.
- **Állapotmentes (stateless)** – Minden kérés tartalmaz minden szükséges információt a feldolgozáshoz: a metódus nevét, a paramétereket és a kérés azonosítóját. A szervernek nem kell állapotot vagy munkamenetadatokat tárolnia a hívások között. Ez egyszerűbbé teszi a skálázást és a terheléselosztást elosztott rendszerekben.
- **Könnyű súlyú (lightweight)** – Az üzenetstruktúra nagyon egyszerű, csak néhány kötelező mezőt tartalmaz. Nincsenek bonyolult erőforrás-leírások vagy URL-hierarchiák, ezért az üzenetek mérete kicsi, és a feldolgozás gyors.
- **Egyszerű implementáció** – A JSON feldolgozása szinte minden programozási nyelven támogatott. Egy JSON-RPC szerver implementálásához általában elegendő egy JSON-feldolgozó könyvtár, egy mechanizmus a metódusnevek függvényekhez rendelésére, valamint egy alapvető hibakezelés.
- **Nyelvfüggetlen** – Mivel a JSON egy általánosan használt adatcsere-formátum, a kliens és a szerver különböző programozási nyelveken is megvalósíthatók. Például egy Python kliens kommunikálhat egy Java vagy C++ szerverrel, ha mindkét fél követi a JSON-RPC 2.0 specifikációt.

A jelenleg legelterjedtebb verzió a **JSON-RPC 2.0**. https://www.jsonrpc.org/specification

JSON-RPC 2.0 üzenetstruktúra

Egy JSON-RPC kérés egy JSON objektum, amely a következő mezőket tartalmazza:

- **jsonrpc** – a protokoll verziója (kötelezően “2.0”)
- **method** – a meghívandó távoli metódus neve
- **params** – a metódus paraméterei (nem kötelező)
- **id** – a kérés azonosítója (a válasz párosításához)

Példa kérés:

```
{ "jsonrpc": "2.0", "method": "add", "params": [5, 7], "id": 1 }
```

Sikeres válasz esetén a szerver a következő formátumú üzenetet küldi:

```
{ "jsonrpc": "2.0", "result": 12, "id": 1 }
```

Ha hiba történik, a válasz egy **error** objektumot tartalmaz:

```
{ "jsonrpc": "2.0", "error": { "code": -32601, "message": "Method not found"
}, "id": 1 }
```

Notifications (értesítések)

Ha a kliens **id mező nélkül** küld egy kérést, azt a JSON-RPC **notification** üzenetnek tekinti.

Ebben az esetben a szerver **nem küld vissza választ**, mivel a kliens nem vár eredményt.

Példa:

```
{ "jsonrpc": "2.0", "method": "logEvent", "params": {"event": "started"} }
```

Batch kérések

A JSON-RPC lehetővé teszi, hogy a kliens egyszerre több kérést küldjön egyetlen üzenetben. Ilyenkor a kérések egy JSON-tömbben szerepelnek.

```
[
  {"jsonrpc": "2.0", "method": "add", "params": [1,2], "id": 1},
  {"jsonrpc": "2.0", "method": "subtract", "params": [5,3], "id": 2}
]
```

REST vs JSON-RPC

Jellemző	**REST**	**JSON-RPC**
Architektúra	Erőforrás-alapú	Eljárás-alapú
Végpontok	Több endpoint	Általában egy endpoint
HTTP metódusok használata	Igen	Nem szükséges
Szabvány típusa	Architektúrális elvek	Protokoll specifikáció
Tipikus használat	Publikus web API-k	Belső API-k, blockchain

A REST API-kat gyakran használják nyilvános webes szolgáltatásoknál, míg a JSON-RPC gyakrabban jelenik meg belső rendszerekben vagy blokklánc-interfészekben.

Gyakorlati példa Python nyelven

Ebben a példában egy egyszerű JSON-RPC szervert és klienst valósítunk meg Python nyelven.

A következő könyvtárakat használjuk:

- **Flask** – HTTP szerver létrehozásához
- **requests** – HTTP kliens megvalósításához

A szükséges csomagok telepítése:

```
pip install flask requests
```

JSON-RPC szerver (Python + Flask)

Hozzunk létre egy *server.py* nevű fájlt.

```
from flask import Flask, request, jsonify

app = Flask(__name__)

# elérhető RPC metódusok

def add(a, b):
    return a + b


def subtract(a, b):
    return a - b


methods = {
    "add": add,
    "subtract": subtract,
}


@app.route("/rpc", methods=["POST"])
def rpc():
    data = request.get_json()

    if data.get("jsonrpc") != "2.0":
        return jsonify(
            {
                "jsonrpc": "2.0",
                "error": {"code": -32600, "message": "Invalid Request"},
                "id": data.get("id"),
            }
        )

    method_name = data.get("method")
    params = data.get("params", [])
```

```
    request_id = data.get("id")

    if method_name not in methods:
        return jsonify(
            {
                "jsonrpc": "2.0",
                "error": {"code": -32601, "message": "Method not found"},
                "id": request_id,
            }
        )

    try:
        result = methods[method_name](*params)
        return jsonify(
            {
                "jsonrpc": "2.0",
                "result": result,
                "id": request_id,
            }
        )
    except Exception as e:
        return jsonify(
            {
                "jsonrpc": "2.0",
                "error": {"code": -32603, "message": str(e)},
                "id": request_id,
            }
        )


if __name__ == "__main__":
    app.run(port=5000)
```

A szerver indítása:

```
python server.py
```

JSON-RPC kliens (Python)

Hozzunk létre egy *client.py* nevű fájlt.

```
import requests
import json

url = "http://localhost:5000/rpc"
```

```
payload = {
  "jsonrpc": "2.0",
  "method": "add",
  "params": [10, 15],
  "id": 1
}

response = requests.post(url, json=payload)

print("Status:", response.status_code)
print("Response:", response.json())
```

Várható kimenet:

```
Status: 200
Response: {'jsonrpc': '2.0', 'result': 25, 'id': 1}
```

JSON-RPC 2.0 hibakódok

Code	Jelentés
-32700	Parse error – JSON feldolgozási hiba
-32600	Invalid Request – hibás kérésformátum
-32601	Method not found – a metódus nem létezik
-32602	Invalid params – hibás paraméterek
-32603	Internal error – szerver oldali hiba

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:informacios_rendszerek_integralasa:json-rpc

Last update: **2026/03/12 13:44**