REST – Representational State Transfer

- REST è un paradigma di accesso a microservizi che si basa sui principi base del protocollo HTTP:
  - **E' orientato alle risorse:** I microservizi pertanto "espongono" risorse a cui si accede tramite REST
  - **E' client-server e request-response:** Il server si mette in attesa di ricevere richieste da parte dei client. Ad ogni richiesta segue una risposta. Il server non esegue azioni se non a seguito di una richiesta.
  - **E' stateless:** Il server non mantiene stati applicativi. Ogni richiesta viene quindi trattata in modo indipendente dalle richieste precedenti

REST – Representational State Transfer

- Le risorse sono identificate da URI
- Le azioni da eseguire sulle risorse sono specificate utilizzando i verbi HTTP (GET, POST, PUT, DELETE…)
- Lo scambio dati con le risorse può avvenire in vari formati standard, tra cui:
  - Testo
  - XML
  - Json

RESTful API

- I microservizi espongono la propria interfaccia (API – Application Programming Interface) in termini di risorse, specificandone:
  - URI identificativo
  - Verbi HTTP ammessi
  - Formato dello scambio dei dati e dei parametri
- Un servizio che espone le proprie risorse in modo conforme con il paradigma REST è detto Restful API

RESTful API

Le API RESTful devono rispettare una serie di requisiti:

- Le risorse sono identificate da nomi e non da verbi:
  - `/articoli/`
  - `/articoli/{id}`
- Le risorse possono essere innestate una dentro l'altra
  - `/articoli/{id}/commenti`
  - `/articoli/{id}/commenti/{id}`
  - `/articoli/{id}/commenti/{id}/autore`

RESTful API

- Le risorse possono essere raggiungibili in modi differenti
  - `/articoli/{id}/commenti/{id}/autore`
  - `/autori/{id}`
- Le azioni CRUD da eseguirsi sulle risorse sono specificate associando un significato preciso ai verbi HTTP: GET (lettura), POST (creazione), PUT (aggiornamento), DELETE (cancellazione)
  - GET /articoli/{id}
  - PUT /articoli/{id}

RESTful API

- Le azioni NON CRUD possono essere specificate inserendo un verbo alla fine dell'url. Nell'url non devono mai apparire metodi crud.
  - `/autori/{id}/sospendi-account`
  - `/playlists/{id}/play`
- Le azioni restituiscono codici standard: 2xx, 4xx, 5xx…
- I nomi di risorse e di azioni sono significativi ed utilizzano – per separare le parole e migliorare la leggibilità
- Tutti gli URL hanno una struttura uniforme e consistente

RESTful API - Vantaggi

- **Indipendenza:** in termini di piattaforma (bastano client e server che supportino HTTP) e di linguaggio (i formati XML/JSON sono facilmente interpretabili da qualsiasi linguaggio di programmazione)
- **Semplicità e flessibilità**: Agevola e semplifica la definizione delle interfacce dei sercizi
- **Scalabilità e performance:** il paradigma stateless consente ai microservizi di scalare orizzontalmente. Il formato richiesta/risposta è inoltre molto efficiente anche dove la larghezza di banda è limitata

RESTful API - Vantaggi

- **Caching:** le risposte fornite possono essere archiviate in qualsiasi livello di cache (browser, CDN o cache server interni) per migliorare ulteriormente le performance

RESTful API - Esempio

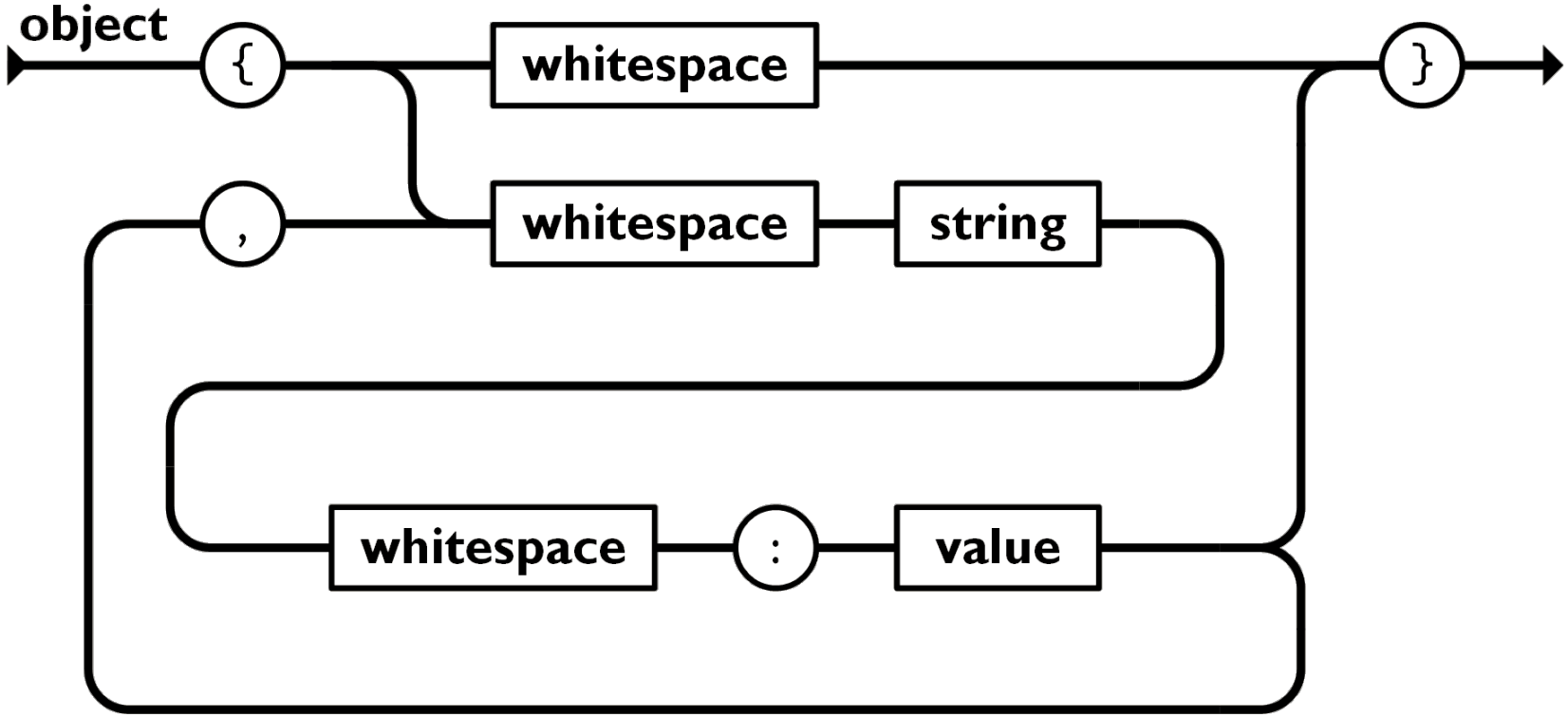
- `GET /students` → Recupera tutti gli studenti
- `POST /students` → Aggiunge un nuovo studente
- `GET /students/{id}` → Recupera uno studente dato il suo id
- `PUT /students/{id}` → Modifica uno studente identificato dal suo id
- `DELETE /students/{id}` → Cancella uno studente identificato dal suo id
- `GET /students/{id}/esami` → Recupera i voti di uno studente
- `GET /students/{id}/media-voti` → Recupera la media vori di uno studente

Json

- JSON (JavaScript Object Notation) è un formato di rappresentazione dei dati nato al fine di agevolarne lo scambio. E' un sottoinsieme del linguaggio di programmazione Javascript ed è fortemente utilizzato nelle API Restful
- Json è un formato molto semplice da leggere e da scrivere. I dati sono rappresentati come testo e scambiati come tali.
- Json si basa su due strutture dati principali:
  - Gli oggetti e Le collezioni (array)

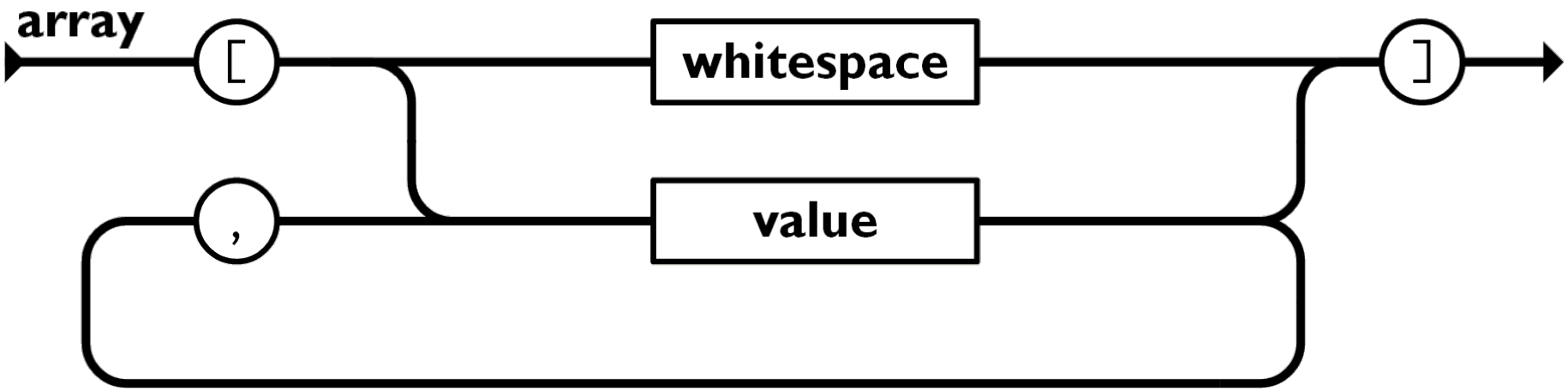
Json - Oggetti

Gli oggetti sono contenuti in una coppia di graffe e al loro interno contengono una serie di coppie chiave : valore separate da ;



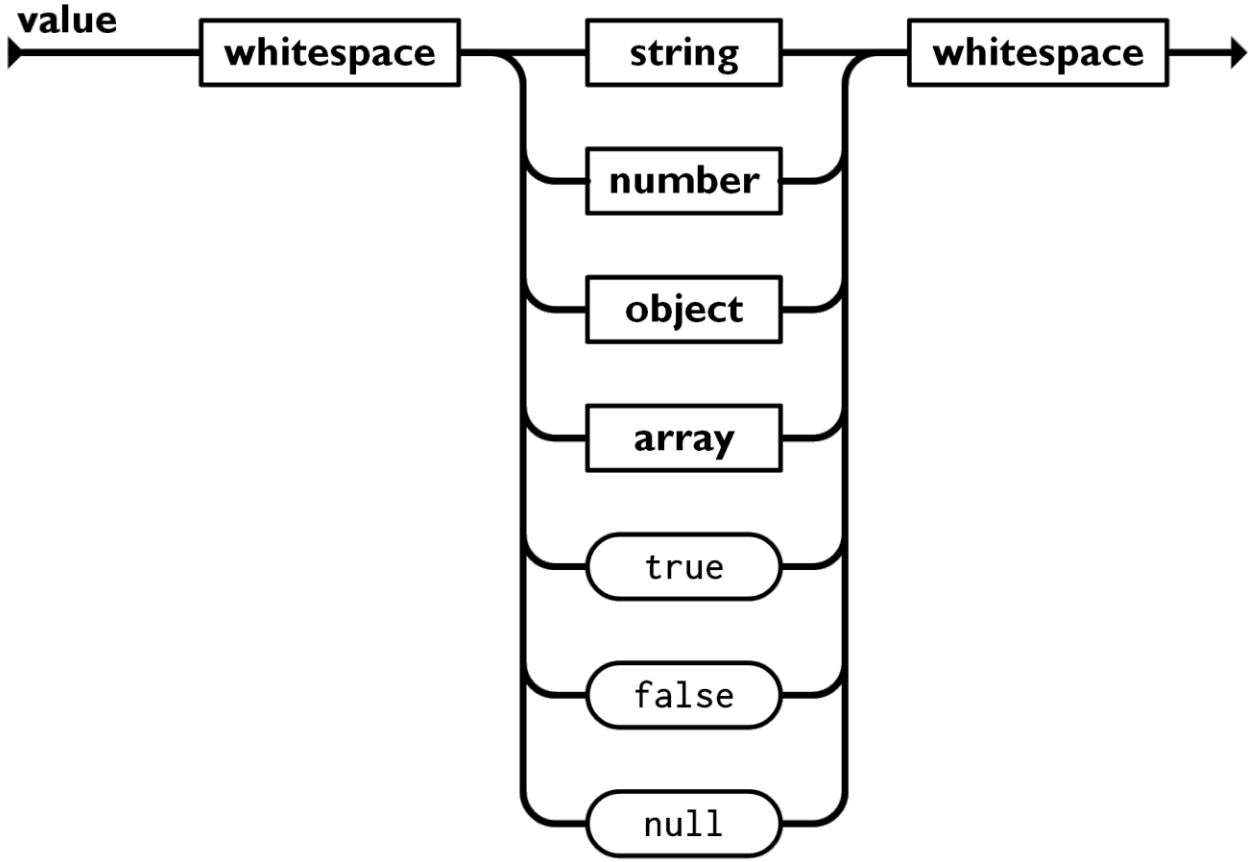
Json - Array

Gli array sono contenuti in una coppia di parentesi quadre e contengono una serie di valori separati da virgola



Json - Valori

I valori possono essere elementari (stringhe, numeri…) oppure array o oggetti



Json - Esempio


```
{
    "orders": [
        {
            "orderno": "748745375",
            "date": "June 30, 2088 1:54:23 AM",
            "trackingno": "TN0039291",
            "custid": "11045",
            "customer": [
                {
                    "custid": "11045",
                    "fname": "Sue",
                    "lname": "Hatfield",
                    "address": "1409 Silver Street",
                    "city": "Ashland",
                    "state": "NE",
                    "zip": "68003"
                }
            ]
        }
    ]
}
```