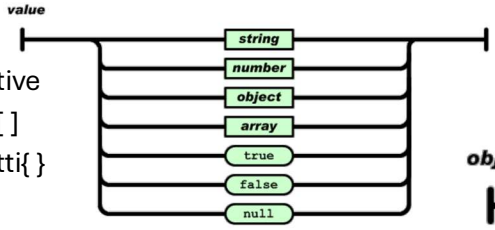
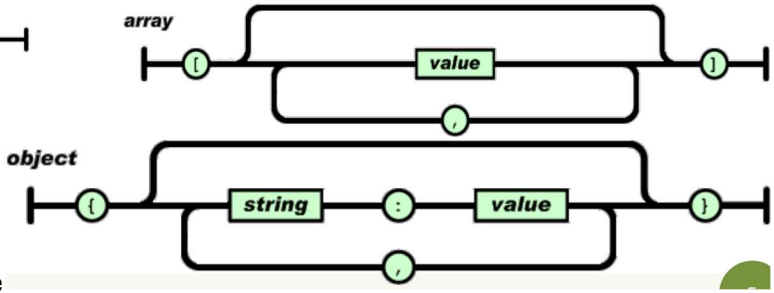
# 02_02 API JSON

File ha:

- Primitive
- Array[ ]
- Oggetti{ }




**NOTA**: le chiavi degli oggetti sono sempre stringhe
**NOTA**: le stringhe devono avere doppi apici
**NOTA**: non c'è un limite al nr di sotto oggetti che un oggetto può avere

```
{
    "firstName": "John",
    "lastName": "Smith",
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        "212 555-1234",
        "646 555-4567"
    ]
}
```

Name/Value Pairs
Child properties
String Array
Number data type

Esistono due metodi che permettono la comunicazione **json-javascript**:

- **Stringify**: converte un oggetto javascript in una stringa contente la versione json dell'oggetto. (lavora anche ricorsivamente escludendo però metodi è cose undefined)
- **Parse** : da una stringa json → crea oggetto javascript

### *Due tipi di url:*

- **Collezione** → insieme o lista di oggetti/risorse
  - Nome della collezione al plurale
- **Elemento** → singolo item e le sue proprietà
  - Nome della collezione + identificativo

- `http://api.polito.it/students`
- `http://api.polito.it/courses`
- `http://api.polito.it/students/s123456`
- `http://api.polito.it/courses/01zqp`

### *Metodi associati:*

- **GET** : recuperare un'intera collezione/singolo elemento
- **POST** : creare una nuova risorsa → aggiungere un elemento alla collezione
- **PUT** : aggiornare un elemento esistente di una collezione
- **DELETE** : eliminare un elemento

| Resource | GET | POST | PUT | DELETE |
|---|---|---|---|---|
| /dogs | List dogs | Create a new dog | Bulk update dogs (avoid) | Delete all dogs (avoid) |
| /dogs/1234 | Show info about the dog with id 1234 | ERROR | If exists, update the info about dog #1234 | Delete the dog #1234 |

Come li vede google?

| Standard Method | HTTP Mapping | HTTP Request Body | HTTP Response Body |
|---|---|---|---|
| List | GET <collection URL> | N/A | Resource* list |
| Get | GET <resource URL> | N/A | Resource* |
| Create | POST <collection URL> | Resource | Resource* |
| Update | PUT or PATCH <resource URL> | Resource | Resource* |
| Delete | DELETE <resource URL> | N/A | google.protobuf.Empty** |

**Relazioni tra le collezioni** → collezione/identificatore/relazione

`http://api.polito.it/students/s123456/courses` → lista dei corsi seguiti da s123456

**Richieste più complesse:**

- Use *?parameter=value* for more advanced resource filtering (or search)
  - E.g., https://api.twitter.com/1.1/statuses/user_timeline.json?screen_name=twitterapi&count=2

**ERRORI**: è bene comunicarli tramite gli status code di http, il corpo della risposta può contenere altre informazioni contenenti l'errore

```
{
   "developerMessage" : "Verbose, plain language description of the problem for the app developer with hints about how to fix it.",
   "userMessage":"Pass this message on to the app user if needed.",
   "errorCode" : 12345,
   "more info": "http://dev.teachdogrest.com/errors/12345"
}
```

## IMPLEMENTAZIONE API:

- HTTP API endpoints are just regular HTTP requests
- Request URL contains the Element Identifiers (`/dogs/1234`)
  - Extensive usage of parametric paths (`/dogs/:dogId`)
- Request/response Body contains the Element Representation (in JSON)
  - **Request:** `req.body` populated by the `express.json()` middleware
  - **Response:** `res.json()` to send the response
- Always validate input parameters
- Always validate input parameters
- Really, always validate input parameters

| | Collections | Elements |
|---|---|---|
| GET | `app.get('/answers', (req, res) => {`<br>`  dao.listAnswers().then((answers) => {`<br>`  res.json(answers);`<br>`  });`<br>`});` | `app.get('/answers/:id', (req, res) => {`<br>`  // TODO: validation of req.params.id`<br>`  dao.readAnswer(req.params.id)`<br>`  .then((answer)=>res.json(answer));`<br>`});` |
| POST | `app.use(express.json());`<br><br>`app.post('/answers', (req, res) => {`<br>`  const answer = req.body;`<br>`  // TODO: validation of answer`<br>`  dao.createAnswer(answer);`<br>`  res.end();`<br>`});` | |