nodejs: json-server Fast API
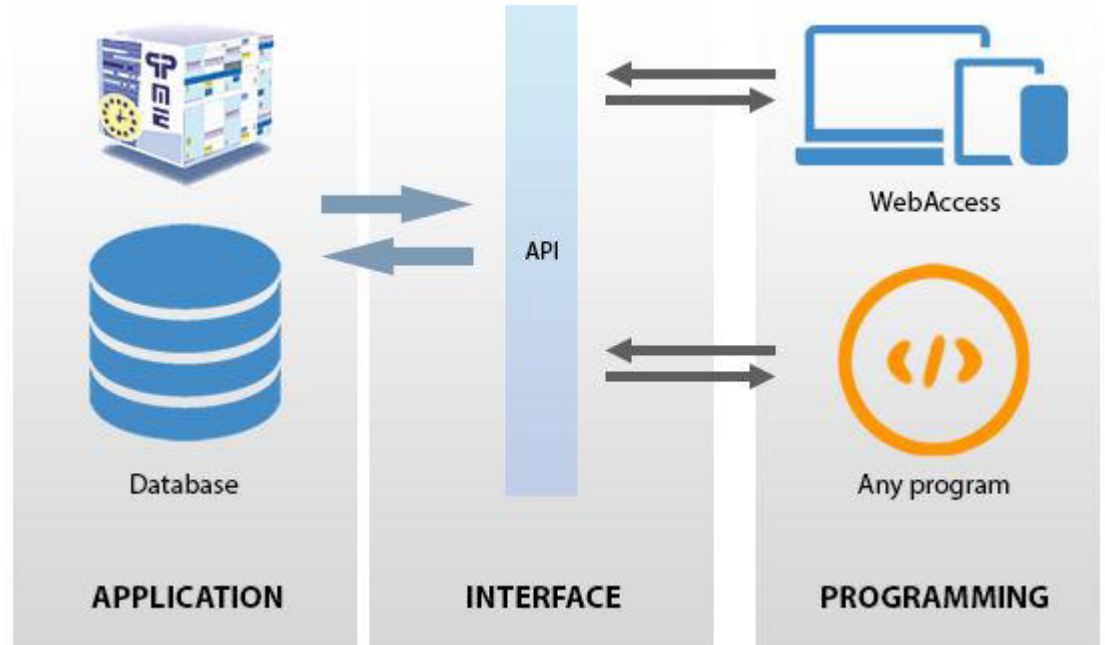
José Carlos Ramalho
2020-11-08

json-server

The json-server is a JavaScript library to create testing REST API.
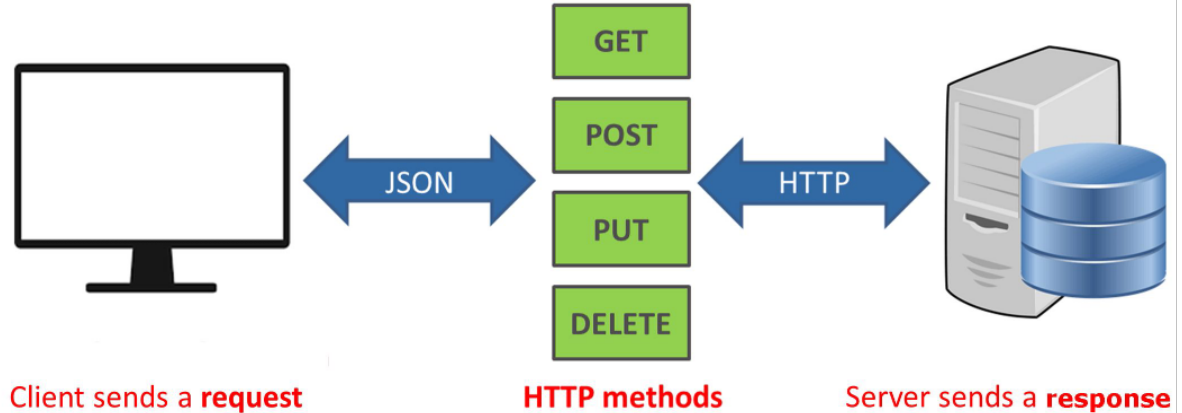
API: what is an API?

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.

API: What is an API used for?

An **API** (Application Programming Interface) is a set of functions that allows applications to access data and interact with external software components, operating systems, or microservices. To simplify, an **API** delivers a user request to a system and sends the system's response back to a user.
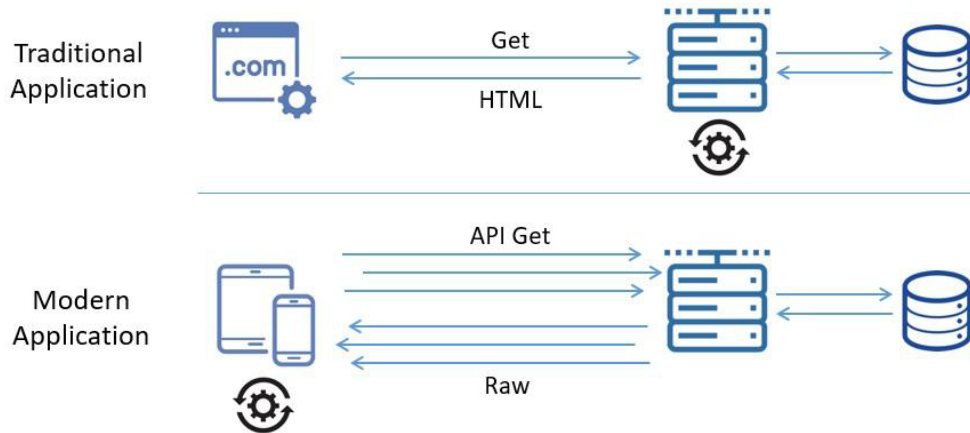
Restful API

| GET |
| POST |
| PUT |
| DELETE |

JSON   HTTP

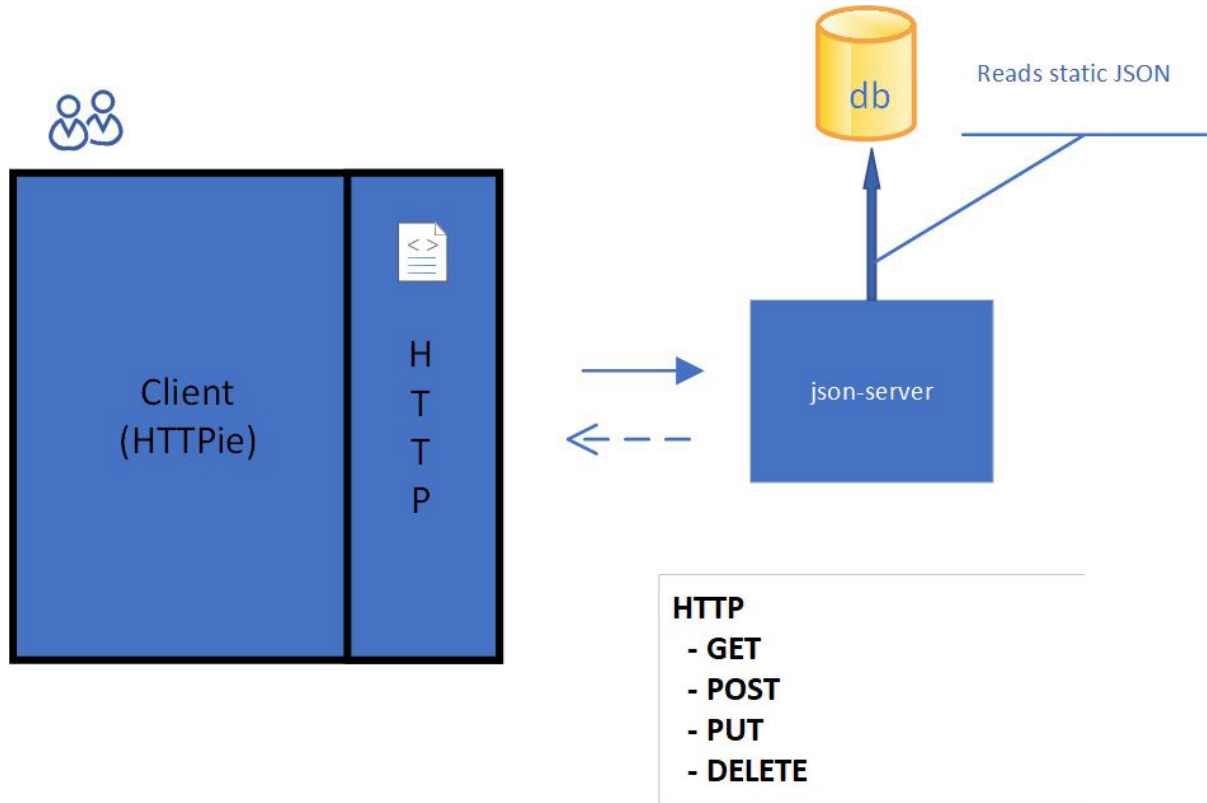Client sends a **request**   **HTTP methods**   Server sends a **response**

API: in our context

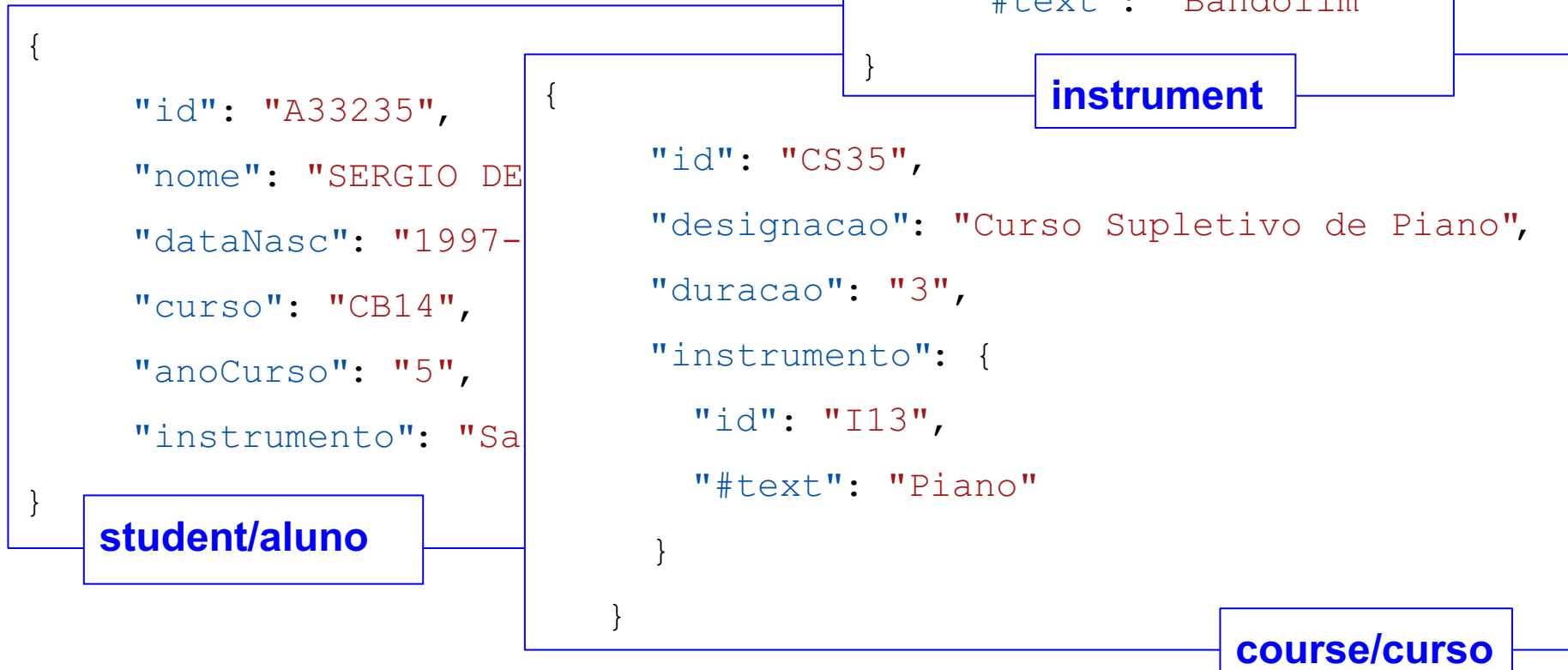An API is a way to expose a set of functionalities over a dataset to other systems.

Traditional vs. Modern

json-server

Dataset: A Music School

```
{
    "id": "I22",
    "#text": "Bandolim"
}
```

instrument

```
{
    "id": "A33235",
    "nome": "SERGIO DE
    "dataNasc": "1997-
    "curso": "CB14",
    "anoCurso": "5",
    "instrumento": "Sa
}
```

student/aluno

```
{
    "id": "CS35",
    "designacao": "Curso Supletivo de Piano",
    "duracao": "3",
    "instrumento": {
        "id": "I13",
        "#text": "Piano"
    }
}
```

course/curso

Dataset: setting up

```json
{
    "alunos": [ …
    ],
    "cursos": [ …
    ],
    "instrumentos": [ …
    ]
}
```

db.json

https://epl.di.uminho.pt/~jcr/TRANSF/db.json

json-server: installation

```
npm install -g json-server
```

json-server: server start

```
json-server --watch db.json
```

Try the following routes:

GET http://localhost:3000/alunos
GET http://localhost:3000/alunos/A33199
GET http://localhost:3000/cursos
GET http://localhost:3000/cursos/CB1
GET http://localhost:3000/instrumentos
GET http://localhost:3000/instrumentos/I14
GET http://localhost:3000/db

POSTMAN

json-server: operators - filtering

Retrieve all students that study Guitar

    GET   http://localhost:3000/alunos?instrumento=Guitarra

Retrieve all students that attend Cello course

    GET   http://localhost:3000/alunos?curso=CS19

Deep fields...

    GET   http://localhost:3000/cursos?instrumento.id=I3

Combining...

    GET   http://localhost:3000/cursos?id=CB1&id=CB2

POSTMAN

curl

json-server: pagination

`_page` and `_limit` parameters

In the Link header you'll get first, prev, next and last links.

GET http://localhost:3000/alunos?_page=2

Access-Control-Expose-Headers ⓘ	X-Total-Count, Link
Link ⓘ	<http://localhost:3000/alunos?_page=1>; rel="first", <http://localhost:3000/alunos?_page=1>; rel="prev", <http://localhost:3000/alunos?_page=3>; rel="next", <http://localhost:3000/alunos?_page=37>; rel="last"
X-Content-Type-Options ⓘ	
Content-Type ⓘ	application/json; charset=utf-8

json-server: ordering

`_sort and _order parameters`

> GET http://localhost:3000/alunos?_sort=curso&_order=asc

Multiple criteria: ordering students by course and then by name

> GET http://localhost:3000/alunos?_sort=curso,nome&_order=asc,desc

json-server: slicing

`_start, _end or _limit parameters`

GET http://localhost:3000/alunos?_start=100&_limit=6

With these parameters you can create your own pagination...

json-server: full-text search

`q parameter`

      GET http://localhost:3000/alunos?q=MARTINS

json-server: get request

```
npm install axios
```

get_request.js

```javascript
const axios = require('axios');

axios.get('http://localhost:3000/alunos)
    .then(resp => {
        data = resp.data;
        data.forEach(a => {
            console.log(`${a.id}, ${a.nome}, ${a.instrumento}`);
        });
    })
    .catch(error => {
        console.log(error);
     });
```

json-server: post request

```
const axios = require('axios');


axios.post('http://localhost:3000/instrumentos, {
    "id": "I23",
    "#text": "Castanholas"
}).then(resp => {
    console.log(resp.data);
}).catch(error => {
    console.log(error);
});
```

Verifica se o registo foi inserido!

json-server: put request - to modify data

```
const axios = require('axios');


axios.put('http://localhost:3000/instrumentos/I23, {

    "#text": "Kazoo"

}).then(resp => {

    console.log(resp.data);

}).catch(error => {

    console.log(error);

});
```

Verifica se o registo foi alterado!

json-server: delete request - to delete data

```
const axios = require('axios');


axios.delete('http://localhost:3000/instrumentos/I23, {
}).then(resp => {
    console.log(resp.data);
}).catch(error => {
    console.log(error);
});
```

Verifica se o registo foi apagado!