Exercises 7: NoSQL

1. What is the CAP theorem and how is it related to NoSQL systems? Which CAP requirement MongoDB does not implement?   
   (2 pts)
2. What are the different main types of NoSQL databases? Include a small description of each type.   
   (2 pts)
3. Create a web service API using the course [Creating APIs with Node.js](https://www.youtube.com/playlist?list=PLHlHvK2lnJndvvycjBqQAbgEDqXxKLoqn) (from the Node.js Module). You should implement all the services and features used in the course up to lesson 30, with the exceptions of Custom Validations (Lesson 23). You should use only MongoDB validations. Features from lessons 31 to 40 (for instance, Password encryption (lesson 31), Welcome emails (lesson 32), etc.) are not to be implemented.

**Tip:** In this course (lesson 13), it shows how to set up an online version of MongoDB. However, the site it uses ([https://cloud.mongodb.com](https://cloud.mongodb.com/)) changed how things are done. You should instead install MongoDB locally and connect using the Compass Tool (included with MongoDB) or the Studio 3T Tool (used in the course). The MongoDB server is localhost:27017.

(3 pts)

1. To the server created in the last exercise, you should add Pet Shop services. Services have an implementation similar to Products. They have the following operations:

* get
* getBySlug
* getById
* post
* put

In addition, you should implement a new service, called getPartnerHours, that given part of the name of a partner and a list of hours returns a list with title and partner name of the services offered in, at least, one of these hours. For instance, if the following Services exist:

{title: "Tosa radical", slug: "tosa-radical", description: "Tosa bem curta", partner: "Luis Braga", price: 120.45, hours: [10, 11, 14, 15, 16]}

{title: "Banho relaxante", slug: "banho-relaxante", description: "Banho quente demorado", partner: "José Braga Silva", price: 50.0, hours: [9, 11, 15, 17]}

And a getPartnerHours request has the following JSON data:

{"name": "braga", "hours": [11, 16]}

The following JSON result should be returned:

[{"title": "Tosa radical", "partner":"Luis Braga"}, {"title": "Banho relaxante", "partner":"José Braga Silva"} ]

The Service schema:

const schema = new Schema({

title: {

type: String,

required: [true, 'O título é obrigatório'],

trim: true

},

slug: {

type: String,

required: [true, 'O slug é obrigatório'],

trim: true,

index: true,

unique: true

},

description: {

type: String,

required: [true, 'A descrição é obrigatória']

},

partner: {

type:String,

Required: [true, 'O nome do profissional que presta o serviço é obrigatório']

},

price: {

type: Number,

required: [true, 'O preço é obrigatório']

},

hours: [{

type: String,

required: true

}]

});

(3 pts)

Good Work!