



Pràctica 11

Scape room API REST

Grup: Andreu, Lishi, Carol, Jordi

Lliuraments

L'entrega d'aquesta pràctica consta de dos documents:

- Un fitxer *.json que s'executi com a json server
- Un fitxer *.pdf on hi hagi la documentació de l'API i les solucions del joc

*** S'ha d'entregar l'enllaç del GIT al Moodle.**

Primera part: familiaritza't amb JsonServer

JsonServer¹ és una eina que et permet crear APIs de manera senzilla mitjançant documents JSON.

1. Una manera fàcil d'usar JsonServer és mitjançant Node.js. Per aquesta raó, el primer pas és instal·lar Node.js <https://nodejs.org/en> .
 - Tingues en compte d'instal·lar la **versió 20 o superior**. Pots comprovar la versió amb la comanda `node --version`

2. Instal·la JsonServer seguint els passos de la documentació: <https://www.npmjs.com/package/json-server>

`"npm install json-server"`

3. Crea un document JSON i llança'l com a JsonServer. Fes diferents proves amb GET, POST, PUT, DELETE...

Per crear el nostre API json server:

`"npx json-server db.json"`

4. Fes proves amb els condicionals per familiaritzar-los.

¹ <https://www.npmjs.com/package/json-server>

Segona part: pensa un joc

5. Pensa un ScapeRoom que es pugui implementar mitjançant un json-server. Abans d'implementar-lo, escriu en un document/paper les proves que tindrà el joc.

Hi han petites implementacions extres que ens ha calgut fer per acabar de donar-li lògica al nostre joc d'ScapeRoom, però les principals idees es corresponen en aquestes 8 proves:

1. Conjunt de preguntes que s'obtenen amb els **GETTERS**.
2. Aprofitant les respostes obtingudes tractant-los com a claus, obtenim d'aquestes claus els elements per ordenar i formar una paraula amb el que farem un **PATCH**.
3. Són portes màgiques com un laberint que vas fent **GETTERS** d'aquestes fins trobar la pista per accedir a la següent prova.
4. Fem una mica de Trivia, en que cada resposta correcta et porta a la clau següent per obtenir més preguntes amb els **GETTERS**.
5. La prova en que consisteix saber-se la cançó de la Macarena i omplint els camps buits amb un **PATCH**.
6. Joc d'objectes impostors que no pertanyen dins del grup de la llista relacionada amb el nou de la clau de la llista, en que els hi farem **DELETE**'s als impostors per obtenir la clau següent de la restant.
7. Joc en que fas del rol d'un lladre que té que escapar-se del crim comés i que per això té que assegurar-se d'eliminar totes les proves que el puguin incriminar, es tracta d'objectes amb una petita descripció a les que tens que eliminar fent **DELETE** els que puguin fer que sigui sospitos.
8. A partir dels objectes que no el poden incriminar, obtens uns codis que en conjunt farà de clau a la següent prova, que et demana fer un **PUT** a la prova anterior amb un id associat i la plantilla de la solució trobada.
9. El joc que continuà es tracta de corregir una frase coneguda per a que sigui la clau següent, aquí es podia haver fet un **PUT** o un **PATCH** per corregir, mentres especifiquem la id associada per aquests mètodes.
10. En la última prova es tracta de salvar un gat amb un **DELETE** per treure'l de dins del ScapeRoom amb la id associada, i posar una bomba a la sala de control per fer explotar tot amb un **POST**, d'una plantilla de l'existència de una bomba col·locada i per a que sigui una mica més emocionant, abans de sortir de la sala de control busquem els detalls de la sortida del ScapeRoom per escapar amb un **GETTER** sobre la mateixa sala de control.

Requisits

- Si vols fer servir **documents externs** com imatges, PDF, etc. Pots penjar-los al Google Drive i fer-los públics.
- El joc ha de tenir **3 proves**, com a mínim
- Per resoldre el joc, cal que el jugador executi, com a mínim una operació de cada tipus: **GET, POST, PUT, DELETE**.
- Per resoldre el joc, s'han d'utilitzar diferents **condicions**.

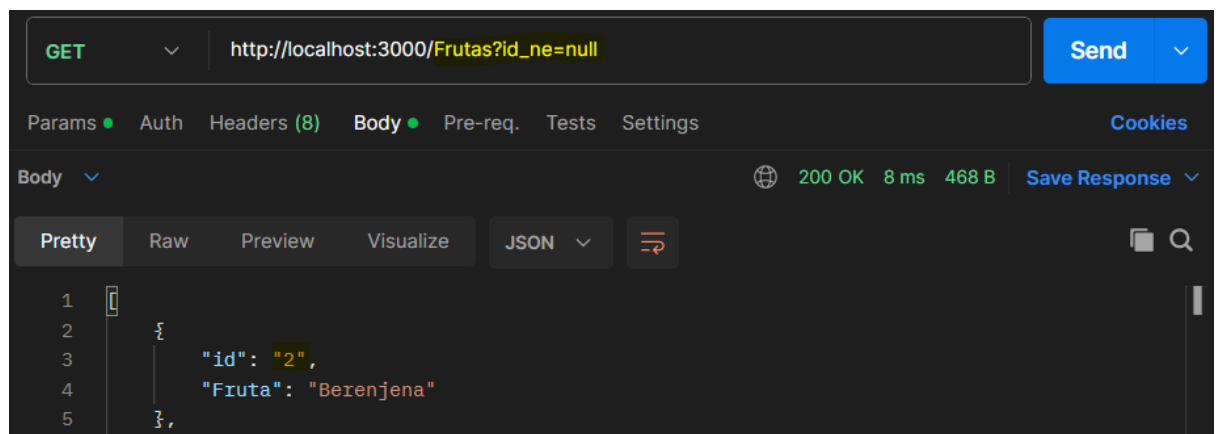
Per resoldre el joc podem utilitzar les condicions de manera més complexa utilitzant el Node.js amb un fitxer server.js, que caldrà executar amb “**node** server.js” o podem indicar-ho directament el filtratge de condicions directament a la URL.

Possibles condicions:

→ ==	lt→ <	lte→ <=	gt→ >	gte→ >=	ne→ !=
------	-------	---------	-------	---------	--------

Com ara, per mirar els objectes del grup de fruites que segueixen existint:

- 'URL/Frutas?id_ne=null'



O per obtenir la llista d'objectes associats a una id superior al valor 1:

- 'URL/Escapa?id_ne=1'

```
GET http://localhost:3000/Escapa Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Body 200 OK 11 ms 1.59 KB Save Response

Pretty Raw Preview Visualize JSON

1 {
2   "id": "1",
3   "escenario": "Te estás preparando para atracar una banco. Esta noche no has
4     dormido bien y estás torpe. Prepárate porque vas a tener que remendar tus
5     errores."
6 },
7 {
8   "id": "2",
9   "objeto": "Ganzua",
10  "descripcion": "La ganzua la has usado para abrir la puerta trasera del banco,
11    pero sin darte cuenta se te ha caído del bolsillo al salir corriendo
12    dirección a la caja fuerte.",
13  "codigo": "20"
14 }
```

Resultat d'utilitzar condicions, per obtenir només objectes per saber quins eliminar.

```
GET http://localhost:3000/Escapa?id_ne=1 Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Body 200 OK 11 ms 1.4 KB Save Response

Pretty Raw Preview Visualize JSON

1 {
2   "id": "2",
3   "objeto": "Ganzua",
4   "descripcion": "La ganzua la has usado para abrir la puerta trasera del banco,
5     pero sin darte cuenta se te ha caído del bolsillo al salir corriendo
6     dirección a la caja fuerte.",
7   "codigo": "20"
8 },
9 {
10  "id": "3",
11  "objeto": "Estetoscopio",
12  "descripcion": "El estetoscopio voló de dentro de tu mochila al quedarse
13    enganchado en la rueda de la puerta de la caja fuerte.",
14  "codigo": "12"
15 }
```

Com que no teniem gaires llistes d'objectes ni valors numèrics amb els que comparar i fer condicions, hem pogut aplicar pels 2 exemples anteriors.

Exemple de varies condicions en una sola URL:

curl "http://localhost:3000/items?id=1&type=A&price_gte=100&price_lte=150"

- Juntament amb el joc s'ha d'entregar la solució
- El joc sempre inicia a l'endpoint /start/

```
GET localhost:3000/start/
```

Exemple de joc (1 prova)

```
{
  "correu_electronic": "Estimat/ada cap del departament de Ciberseguretat,

Em dirigeixo a tu com a director del CERN per sol·licitar-te ajuda urgent a
causa d'un recent ciberatac que ha compromès els nostres sistemes
informàtics. Estem buscant la col·laboració d'experts en seguretat
informàtica, com tu, per reforçar les nostres defenses i mitigar els riscos
futurs. La teva reputació i experiència en aquest camp ens fan considerar-te
un actiu insubstituïble en aquesta situació crítica. La teva participació no
només seria de gran ajuda per al CERN, sinó que també contribuiria a l'avanç
científic i tecnològic. Agrairíem molt el teu suport i estem oberts a
discutir els detalls de la teva col·laboració i qualsevol compensació
adequada pels teus serveis.

T'adjuntem els logs del servidor.

Atentament,
Direcció, CERN.",

  "logs": "s'han encriptat tots els arxius fent servir la clau 253",
  "algoritme_utilitzat": "RSA",
  "command_for_decrypt": "/arxius_{factoritza1}{factoritza2}"
}
```

La solució a aquesta prova seria:

```
GET localhost:3000/arxius_11_23
```

Tercera part: implementa el JSON Server

6. Implementa el json-server que et permeti jugar l'escape-room que has pensat.

Prèviament haurem creat el fitxer dockerfile amb la plantilla que s'ens ha facilitat pel Moodle, aquesta plantilla, se n'encarrega de generar una imatge a partir del nostre API.

El **dockerfile** que posarem al mateix directori que el nostre projecte API, db.json, té que estar sense format, cuidado que no sigui ~~dockerfile.txt~~

Ens assegurem de que el fitxer API, db.json existeixi dins del mateix directori de treball.

```
C:\Users\Lishi\p1>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 18A0-315B

Directorio de C:\Users\Lishi\p1

21/05/2024  18:44    <DIR>          .
21/05/2024  18:44    <DIR>          ..
21/05/2024  18:45             14.414 db.json
21/05/2024  12:09             391 Dockerfile
21/05/2024  18:45    <DIR>          eric
08/05/2024  11:56    <DIR>          node_modules
08/05/2024  11:56             22.407 package-lock.json
08/05/2024  11:56             65 package.json
                4 archivos             37.277 bytes
                4 dirs  140.017.782.784 bytes libres
```

Ens haurem creat un compte amb contrasenya del DockerHub que contindrà la imatge, i dins del nostre compte, creem un nou repositori de visibilitat **PUBLICA**, en el nostre cas ha sigut **gatoschrodinger** ubicat dins de la nostre sessió o compte ljl23.

Per generar aquesta imatge hem fet ús de la instrucció: "**docker build -t gatoschrodinger .**", en que crea una imatge en el directori actual.

```

C:\Users\Lishi\p1>docker build -t gatoschrodinger .
[+] Building 2.2s (10/10) FINISHED          docker:default
=> [internal] load build definition from Dockerfile      0.0s
=> => transferring dockerfile: 430B                      0.0s
=> [internal] load metadata for docker.io/library/node:latest 1.7s
=> [auth] library/node:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore                        0.0s
=> => transferring context: 2B                          0.0s
=> [1/4] FROM docker.io/library/node:latest@sha256:a8ba58f54e 0.0s
=> [internal] load build context                        0.1s
=> => transferring context: 14.64kB                      0.0s
=> CACHED [2/4] WORKDIR /app                          0.0s
=> CACHED [3/4] RUN npm install -g json-server          0.0s
=> [4/4] COPY db.json .                               0.1s
=> exporting to image                                  0.1s
=> => exporting layers                                    0.1s
=> => writing image sha256:3a3d11960e6bbb8d7e0c1e03c30457c1bb 0.0s
=> => naming to docker.io/library/gatoschrodinger        0.0s

View build details: docker-desktop://dashboard/build/default/default/r24u0s15aqlfcg8t0mw17vtjl

What's Next?
  View a summary of image vulnerabilities and recommendations → docke
r scout quickview

```

Després fem que aquesta imatge generada, s'executi en el nostre docker gràfic, amb els ports **3000:3000**: “**docker run -d -p 3000:3000 gatoschrodinger**”

```

C:\Users\Lishi\p1>docker run -d -p 3000:3000 gatoschrodinger
878f87528a737b19ee20a040e04b9d0ac786f52eff8c029fae2fe1bd2c6a7df4

```

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	gatoschrodinger 3a3d11960e6b	latest	In use	45 seconds ag	1.11 GB	▶ ⋮ 🗑

A continuació canviem aquest nom d'imatge, per a que coincideix amb el mateix que es troba en el DockerHub, nom d'usuari inclòs “**ljl23/gatoschrodinger**”.

“**docker image tag gatoschrodinger:latest ljl23/gatoschrodinger**”

```

C:\Users\Lishi\p1>docker image tag gatoschrodinger:latest ljl23/gatoschrodinger

```

Veurem com es genera la imatge que té el nom coincident “**ljl23/gatoschrodinger**” i es la que s'ens permet fer un “**Push to Hub**”.

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	gatoschrodinger 3a3d11960e6b	latest	In use	2 minutes ago	1.11 GB	
<input type="checkbox"/>	ljl23/gatoschrodinger 3a3d11960e6b	latest	In use	2 minutes ago	1.11 GB	
<input type="checkbox"/>	hello-world d2c94e258dcb	latest	Unused	1 year ago	13.25 MB	View packages and CVEs Pull Push to Hub
<input type="checkbox"/>	centos 5d0da3dc9764	latest	Unused	3 years ago	231.26 MB	

I ho confirmem desde el DockerHub que s'ha creat un nou tag amb el valor “**latest**” que hem indicat a la hora de modificar el nom de la imatge generada.

ljl23/gatoschrodinger

Updated less than a minute ago

This repository does not have a description INCOMPLETE

This repository does not have a category INCOMPLETE

Docker commands

To push a new tag to this repository:

```
docker push ljl23/gatoschrodinger:tagname
```

Public View

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	a few seconds ago

A partir d'ara, només ens cal copiar l'enllaç del repositori que conté la imatge per compartir-la amb els companys per a que es puguin descarregar la imatge i posteriorment crear un contenidor dins d'aquesta imatge desde el seu Docker gràfic amb un port personalitzat si pot ser.

hub.docker.com/repository/docker/ljl23/gatoschrodinger/general

Imatge descarregable per jugar

- Fes tests i juga** l'escape room tu mateix i dona'l perquè juguin companys per comprovar que funciona. Ves amb compte de guardar una còpia del fitxer original del joc per si es modifica mentre es juga.

<https://hub.docker.com/repository/docker/ljl23/gatoschrodinger/general>

8. Escribe una pequeña **documentació** (en pdf) del joc on s'indiqui per on començar a jugar. A més, dona les **solucions** a totes les proves per tal de tenir tot el joc resolt.

Documentació de l'ScapeRoom (gatodeschrodinger):

- a) Per iniciar el joc comença pel endpoint **/start/**, en que trobarem les instruccions a:
GET `http://localhost:3000/start`

- b) Prova 1: **ordenar**

GET `http://localhost:3000/ordenar`

```
"ordenar": [
  {
    "id": "save",
    "palabra": "xxxxxxx",
    "titulo": "ORDENAME",
    "instrucciones": "Deberas poner en la URL de cada fase el TITULO para acceder a la prueba. Ej este caso seria ORDENAME, lee la descripción para saber que hay que hacer",
    "descripcion": "Tendras que hacer un GET a partir de la URL/ORDENAME y apuntarte las partículas/elementos que vayas consiguiendo a partir de las respuestas de las preguntas, en que las respuestas son la clave que te ayudara a descubrir los elementos nuevos. Ves haciendo PATCH aquí con el 'save' como {:id}, desde la clave 'ordenar' e ir actualizando las XXXXX del campo 'palabra' por los elementos que consigas, el resultado de la palabra ordenada sera la nueva clave para acceder a la siguiente fase",
    "descripcion1": "Para saber la clave tendras que acertar las siguientes preguntas que te llevaran a saber la palabara ordenada correspondiente, suerte.",
    "regla": "A cada clave, clave o titulo que obtengas, corresponde hacerle un GET",
    "pista": "Actualizar con la palabra final en el campo 'palabra' con el método PATCH en la siguiente ruta: URL/save, ejemplo (http://localhost:3000/save) "
  }
]
```

Conjunt de solucions:

GET `http://localhost:3000/Jordi`

GET `http://localhost:3000/Peter Parker`

GET `http://localhost:3000/Si`

GET <http://localhost:3000/1789>

[Solució final](#)

GET <http://localhost:3000/Gatrenat>

c) Per obtenir la següent prova 2: **Puertas Magicas**

GET <http://localhost:3000/Puertas Magicas>

```
"Gatrenat": [
  {
    "id": "1",
    "FASE_2": "Enorabuena has adivinado la palabra. Vamos a por
el siguiente juego. Ha sido facil no?",
    "titulo": "Puertas Magicas",
    "nivel anterior": "ordenar",
    "descripcion": "Aquí deberas abrir cada puerta donde hay una
clave escondia(tranquilo que se vera facil cual es la clave). El
objetivo de esta FASE es ir abriendo las puertas hasta llegar a
la clave para pasar a la siguiente FASE."
  }
],
"Puertas Magicas": [
  {
    "id": "1",
    "instrucciones": "Adivina la puerta correcta. Ves haciendo
GET's hasta llegar a la parte de la puerta donde contenga la
palabra",
    "puertas que hay": "AZUL, VERDE, AMARILLA"
  }
]
```

[Solució:](#)

GET <http://localhost:3000/nobita>

d) Prova 3: **Cuanto Sabes**

GET <http://localhost:3000/Cuanto Sabes>

```
"Cuanto Sabes": [
  {
    "id": "1",
    "pregunta": "A cuánto equivale la x si  $2*(2 * x) = 2?$ "
  }
],
```

```

"0.5": [
  {
    "id": "1",
    "pregunta": "Quién escribió el Don Quijote?"
  }
],
"Miguel de Cervantes": [
  {
    "id": "1",
    "pregunta": "Los peces pueden tener pulmones?
(Verdad/Falso) "
  }
],
"Verdad": [
  {
    "id": "1",
    "pregunta": "Cuál es la unidad estructural del sistema
nervioso?"
  }
],
"Neurona": [
  {
    "id": "1",
    "FASE_4": "Increíble has logrado pasar la prueba, eres duro
de roer.",
    "titulo": "Canta",
    "nivel anterior": "ORDENAME, Puertas Magicas, Cuanto
Sabes",
    "descripcion": "A continuación deberas reemplazar las
cadenas de XX del siguiente texto con el método PATCH. El titulo
de la canción sera la clave para pasar a la siguiente FASE",
    "ruta": "URL/Canta"
  }
]

```

Conjunt de soluciones:

GET <http://localhost:3000/0.5>

GET <http://localhost:3000/Miguel de Cervantes>

GET <http://localhost:3000/Verdad>

GET <http://localhost:3000/Neurona>

e) Prova 4: **Canta**

GET <http://localhost:3000/Canta>

```
"Canta": [
```

```
{
  "id": "1",
  "instrucciones": "Deberas hacer un PATCH en la canción pero
sustituyendo las x rellenas por el número de letras
correspondiente en el campo 'cancion' en la id actual",
  "descripcion": "Deberas rellenar las cadenas de 'xxxx' y
decir de que canción se trata y el título de la canción sera la
clave para pasar al siguiente nivel",
  "cancion": "Dale a tu xxxxxx alegría xxxxxxxx que tu xxxxxx
es para darle alegría y cosa xxxxx"
}
]
```

Solució:

GET <http://localhost:3000/Macarena>

f) Prova 5: **Impostores**

GET <http://localhost:3000/Impostores>

```
"Impostores": [
  {
    "id": "1",
    "descripcion": "Eimina con DELETE las palabras incorrectas
de ese grupo, a partir de la id que le corresponde, ejemplo:
URL/Frutas/1",
    "descripcion1": "Elimina los objetos que no sean una
fruta",
    "grupos": "Pon esta ruta: URL/Frutas?id_ne=null, per
obtener el llistat d'objectes per cada grup"
  }
],
"Frutas": [
  {
    "id": "1",
    "Fruta": "Magdalena"
  },
  {
    "id": "2",
    "Fruta": "Berenjena"
  },
  {
    "id": "3",
    "Fruta": "Granada"
  },
]
```

```
{
  "id": "4",
  "Fruta": "Lechuga"
},
"Granada": [
  {
    "id": "1",
    "descripcion": "Elimina los ejercicios que no se practiquen en el gym de pesas, usando DELETE a partir del id asociado al ejercicio, ejemplo: URL/Gyms/1",
    "siguiente": "Pon en tu ruta Gyms, en esta parte tendras que dejar cual es el ejercicio de gym",
    "ruta": "URL/Gyms"
  },
  "Gyms": [
    {
      "id": "1",
      "Gym": "Front level"
    },
    {
      "id": "2",
      "Gym": "Boxing"
    },
    {
      "id": "3",
      "Gym": "Beisbol"
    },
    {
      "id": "4",
      "Gym": "Hip Trust"
    }
  ],
  "Hip Trust": [
    {
      "id": "1",
      "FASE_6": "Buen ojo, has visto las que eran incorrectas",
      "nivel anterior": "ORDENAME, Puertas Magicas, Cuanto Sabes, Canta, Impostores",
      "titulo": "Escapa",
      "descripcion": "En esta prueba vamos a ser un fugitivo buscado por la ley. Tendras que eliminar los objetos con su id
```

```

    asociado que creas que te pueden llegar a delatar para incubar
    tu crimen, lee la historia y luego decides cuales eliminar. Para
    pasar al siguiente nivel obtén la clave de unir los códigos que
    dan los objetos que no eliminas del escenario.",
    "instrucciones": "Usa GET's y DELETE's para jugar.",
    "ruta": "URL/Escapa"
  }
]

```

Conjunt de solucions:

GET <http://localhost:3000/Granada>

GET [http://localhost:3000/Hip Trust](http://localhost:3000/Hip%20Trust)

Solució:

GET <http://localhost:3000/Macarena>

g) Prova 6: **Escapa**

GET <http://localhost:3000/Escapa>

```

"Escapa": [
  {
    "id": "1",
    "escenario": "Te estás preparando para atracar una banco.
Esta noche no has dormido bien y estás torpe. Prepárate porque
vas a tener que remendar tus errores.",
    "consulta llista de objectes": "URL/Escapa?id_ne=1, només
han de quedar 2 objectes, usa la ruta per saber quins queden"
  },
  {
    "id": "2",
    "objeto": "Ganzua",
    "descripcion": "La ganzua la has usado para abrir la puerta
trasera del banco, pero sin darte cuenta se te ha caído del
bolsillo al salir corriendo dirección a la caja fuerte.",
    "codigo": "20"
  },
  {
    "id": "3",
    "objeto": "Estetoscopio",
    "descripcion": "El estetoscopio voló de dentro de tu
mochila al quedarse enganchado en la rueda de la puerta de la
caja fuerte.",
    "codigo": "12"
  },
]

```

```

{
  "id": "4",
  "objeto": "Movil",
  "descripcion": "El móvil lo llevabas colgado del cuello con una cuerda, y su utilidad era que tu compinche, que vigilaba desde fuera del banco, te avisara cuando llegara la policia para que te diera tiempo a escapar.",
  "codigo": "58"
},
{
  "id": "5",
  "objeto": "Boligrafo",
  "descripcion": "Con el bolígrafo anotabas en tu mano los números de la caja fuerte que iban desbloqueando la cerradura hasta tener el código completo.",
  "codigo": "64"
},
{
  "id": "6",
  "objeto": "Guantes",
  "descripcion": "Los llevabas puestos CASI todo el tiempo",
  "codigo": "64"
}
]

```

Solució:

GET <http://localhost:3000/2058>

h) Prova 7: **Corrige**

GET <http://localhost:3000/Corrige>

```

"Corrige": [
  {
    "id": "1",
    "frase": "Lentejas, comida de viejas, si las quieres bien y si no, se las das al perro"
  }
]

```

Solució:

GET <http://localhost:3000/Lentejas, comida de viejas, si las quieres bien y si no, las dejas>

i) Prova 8: **Bomba**

GET <http://localhost:3000/Bomba>

```
"Bomba": [
  {
    "id": "1",
    "descripcion": "Tienes 2 salas, en una de ellas esta el
gatito, y en la otra tendras que activar la bomba para poder
salir del lugar.",
    "instrucciones": "Tendras que hacer un DELETE para recoger
al gato, desde su id asociado 'URL/sala gato/1' para sacar el
gato de ahí, haz un POST a la 'sala control' para poner la bomba
'URL/sala control', y después haz un GET rápidamente a la 'sala
control' antes de que explote la bomba para obtener los detalles
de la salida.",
    "bomba": "Bomba que colocar, para hacer el POST usa la
siguiente plantilla en el Body y pon el campo bomba entre
comillas dobles: { bomba: true }",
    "titulos de salas": "sala gato, sala control"
  }
]
```

Conjunt de soluciones:

GET *http://localhost:3000/sala control*

GET *http://localhost:3000/exit*