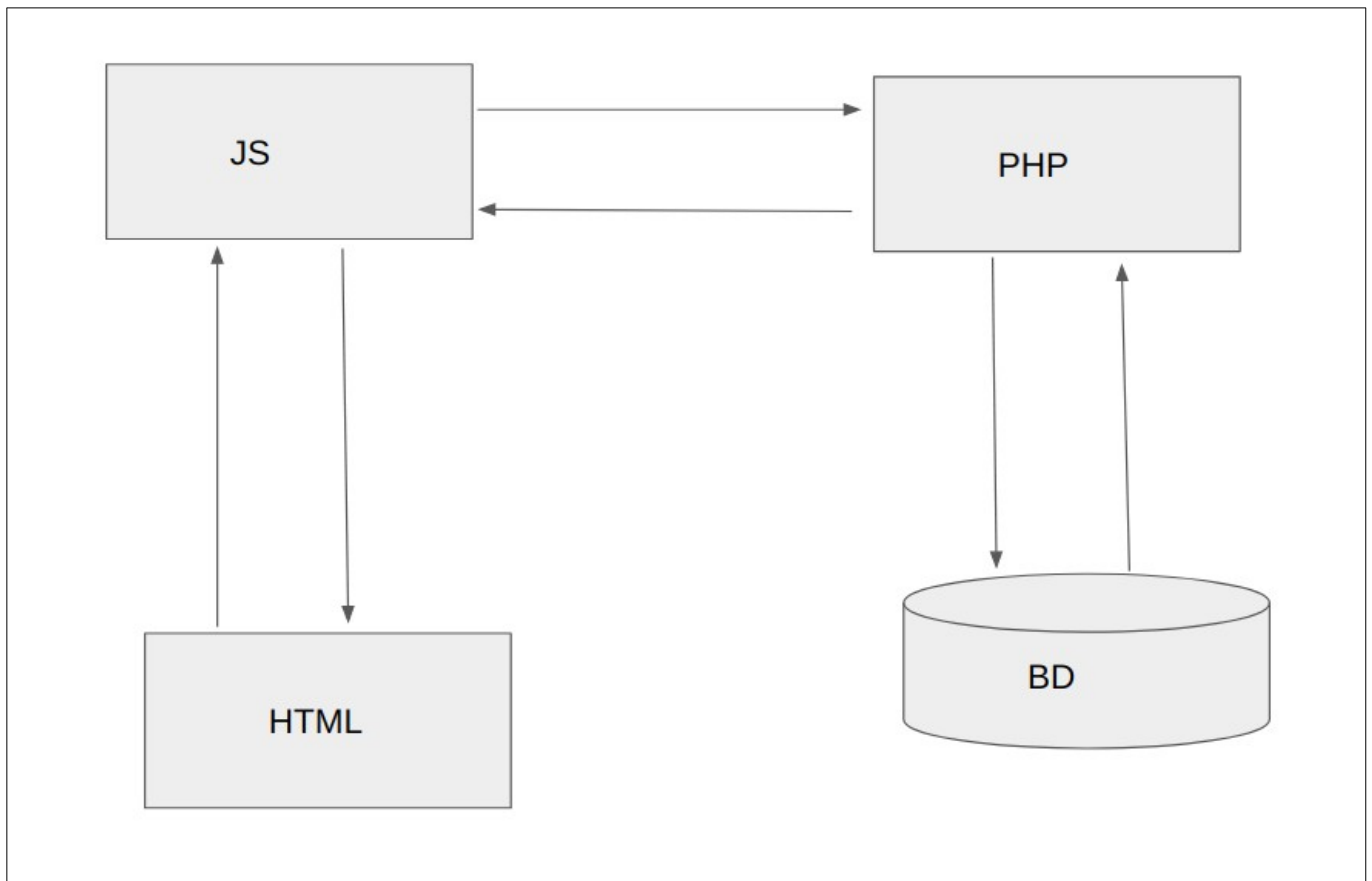


Pour avoir access à la BD (SQL) depuis JS, il faut passer par PHP de la maniere suivant:



**SAVE** (dans la BD)

**JS** (manager.js) →

**PHP** (database.php)  
(pour un méthode POST et envoie un fichier JSON)

```
fetch("/database.php", { // cette methode l'a
  method: "POST", // va a database.php
  body: JSON.stringify(data),
  mode: "same-origin",
  credentials: "same-origin",
  headers: {
    "Content-Type": "application/json" // on i
  }
}).then(function (response) {
  console.log('response', response);
})
.catch(function (error) {
  console.error(error);
});
```

## PHP reçoit le fichier JSON (

```
[Game][StartGame] Starting game                                     game.js:22
[Game][getDataPlayer] Getting data from server with params           logic.js:13
▼ {id: "1", name: "Player1", hp: "10", xp: "0", str: "1", ...}       logic.js:36
  end: "1"
  gold: "20"
  hp: "10"
  id: "1"
  inventory: Array(1)
    ► 0: {id: "1", name: "Epee", str: "1", end: "1", price: "0"}
      length: 1
      ► __proto__: Array(0)
  name: "Player1"
  str: "1"
  weapon: null
  xp: "0"
  ► __proto__: Object
```

name: Player1

hp: 10

xp: 0

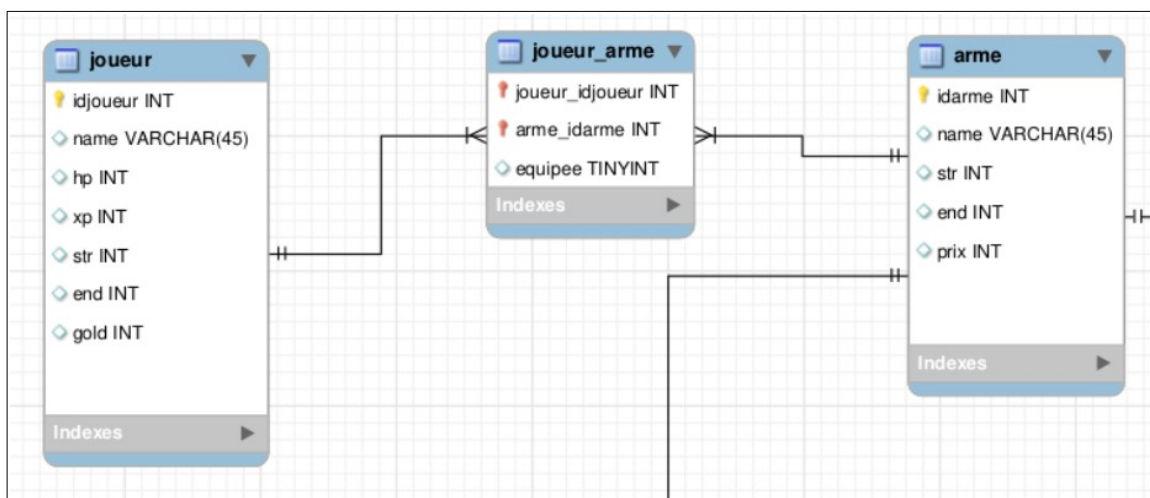
str: 1

end: 1

gold: 20

inventory: [ {id: 1, «Epee», str: 1, end: 1, price: 20}, {id: 2 «Granada», str: 3, end: 4, price:10}{id: 4, «Pistolet», str: 5, end: 6, price:30} /\* liste de weapons \*/

weapon: { id: 2 «Granada», str: 3, end: 4, price:10 }



Mais pour pouvoir faire les insertions dans la BD, **il faut le convertir a cette forme:**

`joueur.(`idjoueur`, `name`, `hp`, `xp`, `str`, `end`, `gold`)`

`arme.(`idarme`, `name`, `str`, `end`, `prix`)`

`joueur_arme(`joueur_idjoueur`, `arme_idarme`, `equipee`)`

**Pour après pour faire les insertions dans les 3 tables**

Par exemple:

Table Joueur

```
INSERT INTO `joueur`(`idjoueur`, `name`, `hp`, `xp`, `str`, `end`, `gold`) VALUES ("1","Player1","10","0","1","1","20")
```

Table arme

```
INSERT INTO `arme`(`idarme`, `name`, `str`, `end`, `prix`) VALUES ("2","Granada",3,4,10)
```

Table relation entre el arma y el jugador

```
INSERT INTO `joueur_arme`(`joueur_idjoueur`, `arme_idarme`, `equipee`) VALUES (1,2,1)
```

idjoueur	name	hp	xp	str	end	gold
1	Player1	10	0	1	1	20

joueur_idjoueur	arme_idarme	equipee
1	1	0
1	2	1
1	4	0

idarme	name	str	end	prix
1	Epee	1	1	20
2	Granada	3	4	10
3	Couteau	4	2	4
4	Pistolet	5	6	30

---

## LOAD (depuis la BD)

**JS** (manager.js)



**PHP**

(database.php) pour chercher un jouer dans la BD  
(pour un méthode GET et n'envoie rien )

```
return fetch("database.php", { //
  method: "GET",
  mode: "same-origin",
  credentials: "same-origin",
  headers: {
  }
}).then(function(response) { //
  if (response.status === 200) { //
    return response.json();
  } else {
    return null
  }
}).then(function(jsonData) { //
  console.log(jsonData)
```

PHP va a chercher dans la BD le jouer

Si il ne le trouve pas —> Retourn un fichero JSON vide

Si il le trouve —> Retourn un fichier JSON avec tout ce qui a le joueur

### **Pour faire ça, il doit**

1- Chercher le idjoueur dans la table joueur

idjoueur	name	hp	xp	str	end	gold
1	Player1	10	0	1	1	20

2- Chercher le idjouer dans la table joueur\_arme pour prendre toutes les armes qu'il a

joueur_idjoueur	arme_idarme	equipee
1	1	0
1	2	1
1	4	0

3- Prendre le id de chaque arme (id\_arme) pour les chercher avec SELECT dans la table arme

idarme	name	str	end	prix
1	Epee	1	1	20
2	Granada	3	4	10
3	Couteau	4	2	4
4	Pistolet	5	6	30

### **En php on va a créer un object json avec ce qu'il a trouve dans les tables**

name: Player1

hp: 10

xp: 0

str: 1

end: 1

gold: 20

inventory: [ {id: 1, «Epee», str: 1, end: 1, price: 20}, {id: 2 «Granada», str: 3, end: 4, price:10}{id: 4, «Pistolet», str: 5, end: 6, price:30} /\* liste de weapons \*/

weapon: { id: 2 «Granada», str: 3, end: 4, price:10 }

### **JS recoie le fichier JSON**

si le fichier est vide —> Il n'y avait pas de jouer DONC il va lo creer le créer avec la class Player

```

class Player extends Participant {
  constructor(name) {
    const initialWeapon = new Weapon(1, "Epee", 0, 1, 1);
    super(1, name, 10, 0, 1, 1, initialWeapon, 20); // créer
    this._inventory = [initialWeapon]
  }
}

```

```

class Participant {
  constructor(id, name, hp, xp, str, end, wpn, gold) {
    this._id = id;
    this._name = name;
    this._hp = hp;
    this._xp = xp;
    this._str = str;
    this._end = end;
    this._wpns = wpn;
    this._gold = gold;
  }
}

```

```

class Weapon {
  constructor([id=0, name, price, str = 0, end = 0]) {
    this._id = id;
    this._name = name;
    this._str = str;
    this._end = end;
    this._price = price;
  }
}

```

A la fin, JS va avoir le fichier (soit il était dans la DB, soit on l'a créé)

name: Player1

hp: 10

xp: 0

str: 1

end: 1

gold: 20

inventory: [ {id: 1, «Epee», str: 1, end: 1, price: 20}, {id: 2 «Granada», str: 3, end: 4, price:10}{id: 4, «Pistolet», str: 5, end: 6, price:30} /\* liste de weapons \*/

weapon: { id: 2 «Granada», str: 3, end: 4, price:10 }