Javascript Test

The Challenge is creating 2 objects. First, Create an Event Manager Object, named "**EventsManager**". It is an Implementation of the **Publish/Subscribe Pattern**, which allows to bind and trigger custom events (along with its callbacks). Second, Create a Person object according to its use below.

Requirements for EventsManager:

1. Should be written in plain Javascript (no 3rd party allowed).

2. Should support only custom events (no DOM events)

3. Please fix the method "logFood" and suggest 3 solutions for it

The code must not include:

1. for loops (these):
    a. for (i=0;i<5;i++){....}
    b. for (var i in obj)
2. while loops

Guidelines

1. write your code above the "**NOTICE**" comment
2. Please read carefully the code and the 2 challenges for more guidance
3. the **result** should be seen in the console of the browser's developer tools
4. **BONUS**: if time allows, the **result** can be printed to the html file

Supported API

on (eventName, callback, *context):

- should allow to register a callback to event and allow **optional context** to be invoked to

off (eventName, *callback)

- should allow remove registered callbacks of the specified eventName
- BONUS: should allow unregister a previously registered event and callback

trigger (eventName, data)

- should allow triggering an eventName and send arguments (data object) that will be passed to a callback as an argument
- **BONUS**: allow to send several data arguments as: trigger('myevent', arg1, arg2, arg3);

This code should work:

```
/* CHALLENGE 1 A : Please write the source code of “EventsManager” Object

*/Your Code for EventsManager


var MyEventsManager = EventsManager();

/* CHALLENGE 1 B : Please write the source code of “Person”
*/

var Person = ? Write the Code


Person.prototype.waitToEat = function() {
    this.eventManager.on('breakfast:ready', this.eat, this);
}

Person.prototype.eat = function(foods) {
    console.log("i'm", this.name, "and i'm eating", foods.join(","));
    this.foods.length = 0;
    this.foods = foods;
    this.eventManager.trigger('eat:done', this);
};

Person.prototype.finishEat = function(time) {
console.log("i'm", this.name, "and i finished eating at", time);
this.eventManager.off("breakfast:ready", this.finishEat);

}

// logFood method has a task ahead

  Person.prototype.logFood = function() {
      this.foods.forEach(function(item){
     console.log("I'm " + this.name + "and I ate " + item); }); };

/* NOTICE: After you add your code of EventsManager you should run

   all the code and test your success with the code below.

   Meaning, the code below should work without any errors

*/// start the app

MyEventsManager.on('eat:done', function(person){

     console.log(person.name, "finished eating");

});

MyEventsManager.on('breakfast:ready', function(menu){

     console.log("breakfast is ready with:", menu);

});

var john = new Person('john', MyEventsManager);

john.waitToEat();

MyEventsManager.on('eat:done', function(person){

     person.finishEat(new Date());

});

var breakfast = ["scrambled eggs", "tomatoes", "bread", "butter"];

MyEventsManager.trigger('breakfast:ready', breakfast);

/* CHALLENGE 2: Please FIX the source code of “logFood” according to the

instructions:

     this “logFood” method throws an error.

     "this.name" doesn't print the Person's name

     Please suggest 2 different solutions (by adding the relevant fix code)
```

```
    so "this.name" will print the relevant name
 */
john.logFood();
```