API Documentation

1- Convert the image to base64
then convert the result to JSON with `JSON.stringify({base64})`
to have the result like this

{"base64":"
.
.
.
FjWP/9k="}

2- Send post request with the JSON format of base64 in the body and `"X-CSRFToken": csrftoken` header

Example of the response:
{"success": true,
"total_number": 7,
"confidence": {"L 1l N": 0.42857142857142855, "Y N": 0.2857142857142857, "L 180 ml N": 0.14285714285714285, "M 500ml N": 0.14285714285714285, "L 180ml low": 0.0, "L 180ml str": 0.0, "M 180ml N": 0.0, "M 1L": 0.0, "M 2l": 0.0, "M 360ml": 0.0, "Y low": 0.0, "Y no fat": 0.0, "eran 180ml": 0.0},
"number": {"L 1l N": 3.0, "Y N": 2.0, "L 180 ml N": 1.0, "M 500ml N": 1.0, "L 180ml low": 0.0, "L 180ml str": 0.0, "M 180ml N": 0.0, "M 1L": 0.0, "M 2l": 0.0, "M 360ml": 0.0, "Y low": 0.0, "Y no fat": 0.0, "eran 180ml": 0.0},
"image": "iVBORw0KGgo . . . . AASUVORK5CYII="}

3- Send the response data to the internal api with the fridge id and the time of this process that has been chosen from the html form

Post request > `body: JSON.stringify({data, fridge, time_created})`

```
function getCookie(name) {
    let cookieValue = null;
    if (document.cookie && document.cookie !== '') {
        const cookies = document.cookie.split(';');
        for (let i = 0; i < cookies.length; i++) {
            const cookie = cookies[i].trim();
            // Does this cookie string begin with the name we want?
            if (cookie.substring(0, name.length + 1) === (name + '=')) {
                cookieValue = decodeURIComponent(cookie.substring(name.length + 1));
                break;
            }
        }
    }
    return cookieValue;
}
const csrftoken = getCookie('csrftoken');

let url = 'http://3.138.151.91/classify_image/classify/api/'

$(document).on('submit', function(e){
    e.preventDefault();
})
```

```
$("form#encode").on("submit", function () {
    $('body').loadingModal({
        position: 'auto',
        text: 'The picture is being analyzed...',
        color: '#fff',
        opacity: '0.7',
        backgroundColor: 'rgb(0,0,0)',
        animation: 'doubleBounce'
    });
    console.log("text2: ", "http://localhost:8000/fridges/" + $(".fridge-name option:selected").text())
    let image = $("input.image-upload").prop('files')[0];
    const reader = new FileReader();
    reader.addEventListener('load', function () {
        base64 = reader.result
        fetch(url, {
                method: 'POST',
                headers: {
                    "X-CSRFToken": csrftoken,
                },
                body: JSON.stringify({
                    base64
                })
            })
            .then(response => response.json())
            .then(data => {
                let fridge = $("#id_fridge_name").text();
                let time_created = $("#id_time_ceated").val();
                console.log(time_created)

                fetch('/api/stats/', {
                    method: "POST",
                    headers: {
                        "X-CSRFToken": csrftoken,
                        'Content-Type': 'application/json'
                    },
                    body: JSON.stringify({
                        data,
                        fridge,
                        time_created
                    })
                }).then(window.location.href =
                    "http://localhost:8000/fridges/" + $(".fridge-name option:selected").text())
            })
    }, false);
    if (image) {
        reader.readAsDataURL(image)
    }
})
```

4- In the internal api views.py take the fields we want and save them
products names from “number”
“image” > after decoding it
“fridge”
“time_created”

```python
class StatsViewSet(viewsets.ModelViewSet):
    queryset = models.Stats.objects.all()
    serializer_class = serializers.StatsSerializer
    lookup_field = 'slug'
    filterset_fields = ('slug')

    def create(self, serializer):
        products = self.request.data['data']['number']
        image = self.request.data['data']['image']
        fridge = self.request.data['fridge']
        time_created = self.request.data['time_created']
        print('products: ', products)

        file_name = "myphoto"+str(random.randint(1, 1000000000000))+".jpg"
        decoded = ContentFile(base64.b64decode(image), name=file_name)

        serializer = serializers.StatsSerializer(data={
            "photo":decoded,
            "fridge":fridge,
            "time_created":time_created,
            "l_180ml_n":products["L 180 ml N"],
            "l_180ml_low":products["L 180ml low"],
            "l_180ml_str":products["L 180ml str"],
            "l_1l_n":products["L 1l N"],
            "m_180ml_n":products["M 180ml N"],
            "m_1l":products["M 1L"],
            "m_2l":products["M 2l"],
            "m_360ml":products["M 360ml"],
            "m_500ml_n":products["M 500ml N"],
            "y_n":products["Y N"],
            "y_low":products["Y low"],
            "y_no_fat":products["Y no fat"],
            "eran":products['eran 180ml'],
        })
        serializer.is_valid()
        serializer.save()
        return HttpResponse(status=201)
```