

PASOS PARA LA RESOLUCIÓN

Como el módulo ya existía, explicaré los pasos desde la edición de los códigos necesarios.

7. Edición de los archivos necesarios para la correcta implementación del módulo.

`__init__.py`

```
from . import controllers
```

`controllers.py`

```
from odoo import http
from odoo.http import request
import json

class SuscriptionAPI(http.Controller):
    @http.route("/api/suscription", type="http", auth="public")
    def get_suscriptions(self, **kwargs):
        status = kwargs.get("status")

        estados_validos = ["active", "expired", "pending", "cancelled"]
        domain = []

        if status:
            if status not in estados_validos:
                return request.make_response(
                    json.dumps({"error": "Status no válido"}),
                    status=400,
                    headers=[("Content-Type", "application/json")],
                )

        domain.append(("status", "=", status))

        registros =
request.env["subscription.subscription"].sudo().search(domain)

        data = []

        for r in registros:
            data.append({"name": r.name, "status": r.status, "price": r.price})
```

```

        return request.make_response(
            json.dumps(data), headers=[("Content-Type", "application/json")]
)

```

8. Agregación de registros para comprobar el correcto funcionamiento

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8069/api/suscription`. The response body is a JSON array:

```

[{"name": "Suscripción Premium", "status": "active", "price": 12}, {"name": "Suscripción básica", "status": "cancelled", "price": 0}, {"name": "Suscripción egipcia", "status": "expired", "price": 200}, {"name": "Suscripción mediana", "status": "active", "price": 50}

```

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:8069/api/suscription/Suscripción%20egipcia`. The response body is a JSON object:

```

{"name": "Suscripción egipcia", "status": "expired", "price": 200}

```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' selected, showing sections for Collections, Environments, History, Flows, and Files (Beta). The main area has tabs for 'Overview' and 'GET http://localhost:8069/api/subscription'. The 'Params' tab is active, showing a table with two rows: 'status' (Key: status, Value: hola) and 'Key' (Key: Key, Value: Value). Below this, the 'Body' tab shows a JSON response: { "error": "Status no válido" }. The status bar at the bottom indicates a '400 BAD REQUEST' response.

This screenshot shows the same Postman interface after changing the query parameter 'status' to 'active'. The 'Params' table now has one row: 'status' (Key: status, Value: active). The 'Body' tab shows a JSON response with two items: { "name": "Suscripción Premium", "status": "active", "price": 12 } and { "name": "Suscripción mediana", "status": "active", "price": 10 }. The status bar at the bottom indicates a '200 OK' response.

We're updating our plans and pricing on March 1. See [our blog](#) for more details.

My Workspace New Import Overview GET http://localhost:8069/api/subscription

http://localhost:8069/api/subscription

GET http://localhost:8069/api/subscription?status=cancelled

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> status	cancelled		
Key	Value	Description	

Body Cookies (1) Headers (4) Test Results

{ } JSON ▾ Preview Visualize

```
1 [  
2   [  
3     {  
4       "name": "Suscripción básica",  
5       "status": "cancelled",  
6       "price": 0  
7     }  
]
```

200 OK 70 ms 222 B

AI

