

# REST API SPECIFICATION -- PART4

## Some additional endpoints

### Some changes in Login and Register endpoints

- Instead of returning just the jwt token (as I have specified in `api_specs_1`), I would like these endpoints to return a json with **both** the jwt token and an object with the fields of the user (name, etc...)

### Logged endpoint

- Method: POST
- Url: `/logged`
- Requires a bearer token in the auth header
- Returns
  - a json with the information about the user sending the request with a success status code (200) if JWT is valid
  - status 401 if JWT is invalid

### A small change on Suppliers endpoints

- When getting one or more suppliers (`/suppliers`, `/suppliers?start=a&end=b`, `/suppliers/id`, `/suppliers?category=cat`), I would like the json of each supplier, to contain, in addition to the fields specified in `api_specs_1`, a boolean field with the name `has_offers` (true if there are offers from the certain `supplier_id`, else false)

### Offers endpoints

- The `GET` endpoint can be accessed by all users (authenticated or not)
- The `POST` endpoint:
  - can be accessed only by suppliers
  - the `supplier_id` specified in the json of the new offer, should be the id of the supplier making the request (will be found from the jwt token)
  - of course, the product specified in the json of the new offer, should belong to the supplier specified in the json of the offer
- The following `PUT` and `DEL` endpoints can be accessed only by the supplier who has created the specified offer

Method	Url	Response	exceptions
GET	<code>/offers?start=a&amp;end=b</code>	returns all the offers sliced on indexes a to b	
GET	<code>/offers?supplier_id=id</code>	returns all the offers for the supplier with the specified id	status 400 if supplier with the specified id is not found

POST	/offers	posts a new offer and returns a json with its details, after is created	a status code like 400 if the creation leads to an error or 401 if unauthorized
PUT	/offers? supplier_id=supplier_id&product_id=product_id	updates the offer with the specified ids and returns a json with the updated offer on success	a status code like 400 if update leads to an error or 401 if unauthorized
DEL	/offers? supplier_id=supplier_id&product_id=product_id	deletes the offer with the specified ids	

### Clients endpoints

- All of these endpoints require authentication
- The client `id` specified in the URL, should be the `id` of the client sending the request, which will be found via the bearer token
- For example, client with `id=5` is not authorized to read or update the information of the client with `id=3`

Method	Url	Response	exceptions
GET	/clients/id	returns the client with <code>id=id</code>	status 404 if the client with this <code>id</code> is not found
PUT	/clients/id	updates the client with <code>id=id</code> and returns the updated object	status 404 if the client is not found or if the new information lead to error while updating
POST	/clients/id	deletes the client with <code>id=id</code> and returns status 200 if deleted successfully	an error status like 400 if could not be deleted

### Change password endpoint

- Of course can be accessed only by authorized users (this will be checked using the bearer token)
- The `user_id` in the request body should be the same with the `id` of the user sending the request (found from the jwt token in the header)

- For example, user with id=3 is not authorized to update the password of the user with id=5
- Method: POST
- Url: /password\_mod
- Body: {

```
user_id: ...,  
old_password: ...,  
new_password: ...,
```

```
}
```