500 can always be thrown and indicates a failure in the backend

If the response row is empty, it means that only a message is returned

"*" means that the body parameter is mandatory

Method	Url	Body	Codes	Credentials	Response	Description
POST	/register	address, tel, username*, password*, email*	400 409: Username already exists 201: Registration complete 200: Automatically logged in	Needed for auto login after register	-	Register a user
PUT	/login	username*, password*	200, 400, 401 409: Log out before trying to login	YES	user	Log in a user
GET	/logout		401, 200	YES	-	Log out a user
GET	/protected		401, 200	YES	-	Check if user is logged in: 200: User is logged in 401: User is not logged in
GET	/adminprotected		401, 200	YES	-	Check if user is an admin: 200: User is an admin 401: User is not an admin
GET	/user		401, 200	YES	User Data	Return data of logged in user
DELETE			401, 200	YES	-	Delete User
PUT	/user/edit	user details (except	401, 201 400: new user details	YES	-	Edit data of logged in user

Method	Url	Body	Codes	Credentials	Response	Description
		password)	rejected			
PUT	/user/ changePassword	password*, passwordNew*	200 401: Not logged in, wrong password	YES	-	Change password of logged in user. **password** is the old password, **passwordNew** the new one
GET	/offers			NO	All Offers	Get all public offers
POST	/offers/create	title*, description, price, category*, dateFrom, dateTo	201, 400, 401	YES	-	Create an offer
PUT	/offers/contact	id*	400, 401, 200	YES	provider	Get the **provider** details of an offer **id**.
PUT	/offers/edit	id, title, description, price, category, dateFrom, dateTo	400, 401, 201	YES	-	Update offer with the given **id**.
PUT	/offers/delete	id*	400, 401, 403, 200	User or admin with deleteOffers rights	-	Delete offer with the given **id**.
GET	/offers/myOffers		401, 200	YES	offers	Get all **offers** of the logged in user.
PUT	/offers/search	searchKey*, category, attributes	403, 200	Needed if the attributes array exists and has elements	offers	Get all **offers** that match the given searchKey. If a category is given, only return offers that belong to this category. If an attributes array is given, search in all attributes specified by this array (the default is to only search in the title)

Method	Url	Body	Codes	Credentials	Response	Description
				that begin with $		Useful values for attributes array: title, description, $provider.name$, $provider.address$
GET	/offers/ notApproved		200, 401, 403	ADMIN	offers	Get all non public offers that have 'validation in progress' as status
PATCH		approve* (bool), id*, message	200, 401, 403	ADMIN with setPublic rights	-	Approve / don't approve offer (id given in body) Approve: offer is set to public and the status to 'approved' Don't approve: status is set to 'Rejected reason: **message**'