

HTTP method verb	endpoint	authorisation	description	constraints	payload	response	errors
GET	/api/useraccount	user	get all useraccount			200 { "ids": int, "created_at": "date", "username": "string", "following": array, "following_count": int, "followers": array, "followers_count": int, "likes": array, "likes_count": int, "profile_picture": url, "description": "string", "user_videos": array }	400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }
POST	/api/useraccount/ <id> eg. /api/useraccount@hejman s</id>	user	get 1 useraccount	param : id (which is the username)		200 { "ids": int, "created_at": "date", "username": "string", "following": array, "following_count": int, "followers": array, "followers_count": int, "likes": array, "likes_count": int, "profile_picture": url, "description": "string", "user_videos": array }, }	400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }
PATCH	/api/useraccount/ <id> eg. /api/useraccount@hejman s</id>	user	update a useraccount info	param: id	{ "username": "string", "profile_picture": url, "description": "string", "user_videos": array }	200 { "status": "okay", "msg": "profile updated" }	400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }
PUT	/api/useraccount	user	add a new useraccount	allow user to change their username, profile pic, description and videos	{ "ids": int, "created_at": "date", "username": "string", "following": array, "following_count": int, "followers": array, "followers_count": int, "likes": array, "likes_count": int, "profile_picture": url, "description": "string", "user_videos": array }	200 { "status": "ok", "msg": "profile added" }	400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }
DEL	/api/useraccount/ <id></id>	user	delete a user account	param: id		200 { "status": "okay", "msg": "user account deleted" }	401 { "status": "error", "msg": "not authorised" } 400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }
GET	/api/usersdes	user	get all user des			200 { "ids": int, "user_id": string, "user_d": string, "created_at": date, "messages": [{ "ids": int, "sender_id": string, "receiver_id": string, "content": string, "created_at": date, }] }, }	400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }
PUT	/api/usersdes	user	add a new de		200 { "ids": int, "user_d": string, "user_p": string, "created_at": date, "messages": [{ "ids": int, "sender_id": string, "receiver_id": string, "content": string, "created_at": date, }] }, }	400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }	
PATCH	/api/usersdes/ <id> eg. /api/usersdes@you</id>	user	update a particular de	param: id	[{ "ids": int, "user_d": string, "user_p": string, "created_at": date, "messages": [{ "ids": int, "sender_id": string, "receiver_id": string, "content": string, "created_at": date, }] }], }	200 { "status": "okay", "msg": "de updated" }	400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }

Method	URL	Description	Request Body	Status	Response Body
	/auth/register	register a new user	<pre>{ "email": "string", "username": "string", "password": "string" }</pre>		<pre>400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] } 400 { "status": "error", "msg": "duplicate email/username" }</pre>
POST	/auth/login		<pre>{ "email": "string(optional)", "username": "string (optional)", "password": "string" }</pre>	201	<pre>{ "access": "string", "refresh": "string" } 400 { "status": "error", "msg": "login failed" } 400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }</pre>
POST	/auth/refresh	refresh the access token	<pre>{ "refresh": "string" }</pre>	201	<pre>{ "access": "string" } 400 { "status": "error", "msg": "login failed" } 400 { "errors": [{ "value": "string", "msg": "string", "param": "string", "location": "string" }] }</pre>