	on creation of user. A unique ID is automatically assigned by the DB. Error cases: must check that no user already exists with the given username. Must also verify that the admin is the one doing this (only admin can create users) Returns a JSON object with is_error (boolean), error_msg (possibly empty string). If is_error is false, also returns the user subobject that was created
GET /users/{user_id}	Retrieves the user with a given user_id, if such a user exists. Returns a JSON object with is_error (boolean), error_msg (possibly empty string). If is_error is false, also returns the user subobject.
GET /users/	Returns all users in the DB
DELETE /users/{user_id}	Deletes the user with the supplied user_d. Error Check: verify that the user with the given ID exists Returns a JSON object with is_error (boolean) and error_msg (possibly empty string)
GET /groups/{group_id}	Returns the group of users specified by the group_id, if such a group exists. Returns a JSON object with is_error (boolean), error_msg (possibly empty string). If is_error is false, also returns the group subobject.
GET /groups/	Returns all groups. Returns a JSON object with is_error (boolean), error_msg (possibly empty string). If is_error is false, also returns the group subobjects.
POST /groups	Creates a new group. Requires JSON input with "group_name" (string), "user_ids" (array of integers). Permissions are NOT specified via this endpoint. For instance: Returns a JSON object with is_error (boolean), error_msg (possibly empty string). If is_error is false, also returns the group subobject that was created.
PUT /groups/{group_id}	Updates a group with the specified group_id, so long as that group exists. Requires JSON input with "group_name" (string), "user_ids" (array of integers). Permissions are NOT specified via this endpoint. Permissions for a new group default to false for everything. For instance: