Notes:

1. Output fields marked with * can be passed into the HTTP request header to filter results.

2. Parameters in red are required.

3. The following HTTP methods signify the following:

    (a) GET is used to retrieve data from the database.
    (b) POST is used to insert new data into the database.
    (c) PUT is used to update existing data in the database.
    (d) DELETE is used to remove existing data from the database.

◇ **GET /entity/{entity}/events/**

Description	Gets all events that a specified entity is either the owner of or a guest of, and the roles they have for those events. Multiple records will be returned for the same event if the entity has more than one role for that event.	
HTTP Header Params	None.	
HTTP Body Params	None.	
Output	`*entity`	The PK of the guest.
	`*event`	The PK of the event.
	`*event_entity_role`	Record PK.
	`*role`	The PK of the role.
	`event_name`	The name of the event.
	`role_name`	The name of the role.

◇ **GET /event/{event}/guests/**

Description	Returns all guests attending a specific event and their roles. The result set does NOT include the owner of the event. Multiple records will be returned for the same guest if they have more than one role.	
HTTP Header Params	None.	
HTTP Body Params	None.	
Output	`*entity`	The PK of the guest.
	`*event`	The PK of the event.
	`*event_entity_role`	Record PK.
	`*role`	The PK of the role.
	`event_name`	The name of the event.
	`role_name`	The name of the role.

◇ **GET /event/{event}/guests/{guest}**

Description	Returns a specific guest attending a specific event and their roles. No results will be returned if the specified guest is the owner. Multiple records will be returned for the same guest if they have more than one role. This is identical to calling `GET /event/{event}/guests/?entity=guest`.
HTTP Header Params	None.
HTTP Body Params	None.

Output	`*entity`	The PK of the guest.
	`*event`	The PK of the event.
	`*event_entity_role`	Record PK.
	`*role`	The PK of the role.
	`event_name`	The name of the event.
	`role_name`	The name of the role.

◇ **GET /event/{event}/owner/**

Description	Gets the owner of a specific event and their roles. Multiple records will be returned for the owner if they have more than one role.
HTTP Header Params	None.
HTTP Body Params	None.

Output	`*entity`	The PK of the guest.
	`*event`	The PK of the event.
	`*event_entity_role`	Record PK.
	`*role`	The PK of the role.
	`event_name`	The name of the event.
	`role_name`	The name of the role.

◇ **GET /event/{event}/roles/**

Description	Returns all roles needed for a specific event.
HTTP Header Params	None.
HTTP Body Params	None.

Output	`*description`	The event-level description, or default description if an event-level description doesn't exist, of the role needed by the event.
	`*estimated_budget`	The amount of money that should be dedicated to this role.
	`*event_needed_role`	Record PK.
	`*needed_role`	The PK of the role needed by the event.
	`*needed_role_name`	The name of the role needed by the event.
	`*quantity_needed`	The number of people needed to perform this role.
	`event`	The PK of the event.

◇ **GET /roles/**

Description	Gets all roles from the database.	
HTTP Header Params	None.	
HTTP Body Params	None.	
Output	*description	The default description of the role.
	*name	The name of the role.
	*role	Record PK.

◇ **GET /roles/{role}**

Description	Gets a specific role from the database. This is equivalent to calling `GET /roles/?role={role}`.	
HTTP Header Params	None.	
HTTP Body Params	None.	
Output	*description	The default description of the role.
	*name	The name of the role.
	*role	Record PK.

◇ **POST /event/{event}/entities/{entity}/roles/**

Description	Adds a role to a specific entity attending a specific event. Note that the specified entity can be either the owner or a guest. Additionally, because this creates a new `event_entity_role` record, if an entity is not already associated with an event, calling this route will associate them with the event.	
HTTP Header Params	None.	
HTTP Body Params	estimated_budget	The budget that the entity should allocate to performing the role.
	role	The role that should be given to the specified entity attending the specified event.
Output	event_entity_role	The PK of the record that was created.

◇ **POST /event/{event}/guests/**

Description	Adds a guest to a specific event. This creates a new `event_entity_role` record with the specific event, specified entity, and the Guest role.	
HTTP Header Params	None.	
HTTP Body Params	entity	The PK of the entity who should be added to the specified event as a guest.
Output	event_entity_role	The PK of the record that was created.

◇ **POST /event/{event}/roles/**

Description	Adds a needed role to a specific event.	
HTTP Header Params	None.	
HTTP Body Params	`description`	The event-level description of the role needed by this event.
	`estimated_budget`	The amount of money that should be dedicated to this role.
	`needed_role`	The PK of the role needed by the event.
	`quantity_needed`	The number of people needed to perform this role.
Output	`event_needed_role`	The PK of the record that was created.

◇ **POST /roles/**

Description	Adds a new role to the database. This does not add a role to any specific event, nor does it add a role to any specific entity.	
HTTP Header Params	None.	
HTTP Body Params	`description`	The default description of the role to create.
	`name`	The name of the role to create.
Output	`role`	The PK of the role that was created.

◇ **PUT /event/{event}/roles/{role}**

Description	Updates a specific needed role for a specific event.	
HTTP Header Params	None.	
HTTP Body Params	`description`	The new event-level description of the role needed by this event.
	`estimated_budget`	The new amount of money that should be dedicated to this role.
	`quantity_needed`	The new number of people needed to perform this role.
Output	`event_needed_role`	The PK of the recorded that was updated.

◇ **PUT /roles/{role}**

Description	Updates an existing role in the database.	
HTTP Header Params	None.	
HTTP Body Params	`description`	The new default description the role should have.
	`name`	The new name the role should have.
Output	`role`	The PK of the role that was updated.

◇ **DELETE /event/{event}/guests/{guest}**

Description	Completely removes a specified guest from an event. This removes every `event_entity_role` record that has the specified guest as the entity.	
HTTP Header Params	None.	
HTTP Body Params	None.	
Output	`event_entity_role`	The PK of the record that was deleted.

◇ **DELETE /roles/{role}**

Description	Deletes an existing role from the database.	
HTTP Header Params	None.	
HTTP Body Params	None.	
Output	`role`	The PK of the role that was deleted.