

API Design - Job Tracker

CS 497: Capstone Project

Winter 2023

Oregon State University

Last Update: Jan, 25, 2023

Data Model	4
Relationships	4
Users	5
Skills - shared amongst users	5
Contacts - shared amongst users	5
Applications - unique to user	6
User Routes	7
/users GET - Get Users	8
/users POST - Create User	9
/users PUT/PATCH - Edit Users	11
/users DELETE - Delete Users	11
/user/username GET - User	12
/users/username POST	13
/users/username PUT/PATCH - Edit User	13
/users/username DELETE - Delete User	14
Skill Routes	16
/skills GET - Get All Skills in DB	17
/skills POST	18
/skills PUT/PATCH	18
/skills DELETE	18
/skills/skill_id GET	18
/skills/skill_id POST	18
/skills/skill_id PUT/PATCH	18
/skills/skill_id DELETE	18
/users/username/skills GET - Get User's Skills	19
/users/username/skills POST - Tie Skill to User	20
/users/username/skills PUT/PATCH	21
/users/username/skills DELETE - Delete all User's Skills	21
/users/username/skills/skill_id GET - Get User's Skill	22
/users/username/skills/skill_id POST	23
/users/username/skills/skill_id PUT/PATCH - Update User's Skill	24
/users/username/skills/skill_id DELETE - Delete Skill from User	25
/users/username/applications/app_id/skills GET	26
/users/username/applications/app_id/skills POST	26
/users/username/applications/app_id/skills PUT/PATCH	26
/users/username/applications/app_id/skills DELETE	26
/users/username/applications/app_id/skills/skill_id GET	26

/users/username/applications/app_id/skills/skill_id POST	27
/users/username/applications/app_id/skills/skill_id PUT/PATCH	28
/users/username/applications/app_id/skills/skill_id DELETE	29
Contact Routes	30
Create a Contact	31
Delete Contact	32
Edit Contact	33
Get Contact	33
Application Routes	35
Create Application	35
Delete Application	36
Edit Application	36
Get Application	37
Get Applications	39

URL:

Index Route where app is deployed:

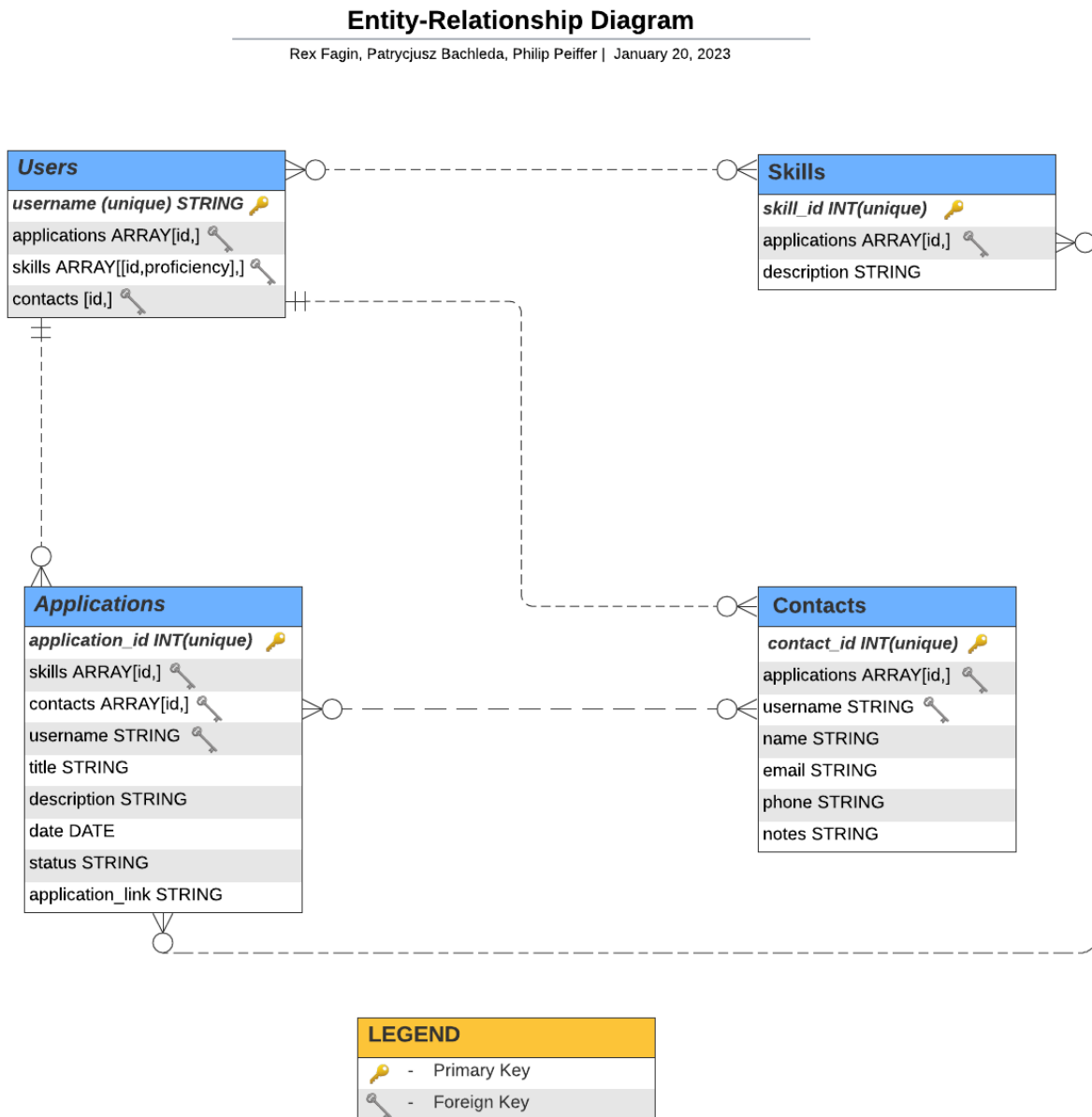
<https://>

This route is provides a button the user can click to login, logout, or create a new user

Data Model

The app stores three kinds of entities in Datastore: Users, Applications, Skills, and Contacts.

Relationships



Users

Property	Data Type	Req	Constraints	Notes
username	String	Y	Must be unique	Obtained through Auth0
applications	Array	Y	List of IDs	Blank when user first created
skills	Array	Y	List of objects {skill_id: XXX, proficiency: XXX, description: XXX}	Blank when user first created
Contacts	Array	Y	List of IDs	Blank when user first created
Id	String	Y	Issued by Auth0 through the OAuth2.0 process and must match the JWT used to create the user. Must be unique.	This number will match the "sub" category in the JWT for protected requests. This number is also used as the unique ID for user entities.

Skills - shared amongst users

Property	Data Type	Req?	Constraints	Notes
id	int	Y	Must be unique	Generated by API when skill first created
description	string	Y		Name of skill: e.g. Javascript, Python, etc.

Contacts - unique to user

Property	Data Type	Req?	Constraints	Notes
id	int	Y	Must be unique	Generated by API when contact first created
user_id	String	Y	Must be unique	Taken from Auth Bearer Token
applications	array	Y	List of IDs contact is tied to	
first_name	string	Y	Max 30 characters	
last_name	string	Y	Max 30 characters	
email	string	N	Max 30 characters	
phone	string	N	Max 30 characters	
notes	string	N	Max 500 characters	

Applications - unique to user

Property	Data Type	Req?	Constraints	Notes
id	int	Y	Must be unique	Generated by API when application first created
skills	array	Y	List of IDs application is tied to	Can be blank array
contacts	array	Y	List of IDs application is tied to	Can be blank array
title	string	Y		
description	string	N		Long form description of job if desired

status	string	Y	Choice of: <ul style="list-style-type: none"> • Not started • Application In Progress • Applied - Awaiting Reply • Applied - Phone Interview Completed • Applied - On Site Interview Completed • No Response • Application Rejected • Offered • Offer Accepted 	
user_id	string	Y	Must be unique	Taken from Auth Bearer Token
link	string	N		Web address
last_modified	string	Y		
posting_date	string	Y		

User Routes

Needed Routes:

- /users
 - POST
- /users/username
 - DELETE
 - GET

Routes Not Needed:

- /users
 - GET
 - DELETE
 - PUT/PATCH
- /users/username
 - POST
 - PUT/PATCH

/users GET - Get Users

Method not allowed. No need to see all users.

/users POST - Create User

Request

Path Parameters

- None

Request Body

- Required? - N/A
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - Bearer Token, JWT from Auth0
- Header: "Authorization: Bearer <Token>"

Request JSON Attributes

Name	Description	Required?
N/A	All user information taken from JWT, so not body passed	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	201 - Created	
Fail	400 - Bad Request	User already exists
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	406 - Not Acceptable	Accept header is not application/json or */*

Response Examples

Success

201 - Created { "username": "philpeiffer", "id": 1234567890, "created": "01/25/2023", "skills": [], "contacts": [], "applications": [], }

Failure

400 - Bad Request

```
{  
  "400": "User already exists"  
}
```

401 - Unauthorized

```
{  
  "401": "JWT missing or invalid"  
}
```

406 - Not Acceptable

```
{  
  "406": "Accept header must be application/json or */*"  
}
```

/users PUT/PATCH - Edit Users

Method not allowed. No need to edit all users.

/users DELETE - Delete Users

Method not allowed. No need to delete all users.

/user/:user_id GET - User

Request

Path Parameters

- user_id - must match JWT provided

Request Body

- Required? - None
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: "Authorization: Bearer <JWT>"

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	200 - OK	
Fail	404 - Not Found	User does not exist
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match username

Response Examples

Success

<u>200 - OK</u> <pre>{ "username": "philpeiffer", "id": 12345, "skills": [{"id": 12345, "proficiency": 4, "description": "Python"}, {"id": 13456, "proficiency": 1, "description": "Javascript"}], "applications": [123, 234, 345], "contacts": [135, 186, 254] }</pre>
--

Failure

404 - Not Found

```
{
  "400": "User does not exist"
}
```

401 - Unauthorized

```
{
  "401": "JWT missing or invalid"
}
```

403 - Forbidden

```
{
  "403": "The owner of the JWT does not match the user resource requested"
}
```

/users/:user_id POST

This method is not allowed. Creating a user is handled through /users POST

/users/:user_id PUT/PATCH - Edit User

This method is not allowed. Updating of the user will be handled through relationship routes instead of trying to update a user directly.

/users/:user_id DELETE - Delete User

Request

Path Parameters

- user_id - must match JWT username sent with request

Request Body

- Required? - No
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: "Authorization: Bearer <Token>"

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – None
- Failure – JSON

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	204 - No Content	
Fail	404 - Not Found	User does not exist
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match username

Response Examples

Success

204 - No Content

Failure

404 - Not Found
<pre>{ "400": "User does not exist" }</pre>

401 - Unauthorized

```
{  
  "401": "JWT missing or invalid"  
}
```

403 - Forbidden

```
{  
  "403": "The owner of the JWT does not match the user resource requested"  
}
```


Skill Routes

Needed Routes:

- /skills ← this is referring to all skills in database (not user specific), so POST is not allowed. Refer to routes below for creating a new skill tied to user. When a user creates a new skill, it will add it to the skills table
 - GET
- /skills/skill_id
 - None
- /users/username/skills
 - GET
 - POST ← posts to the shared skills table AND user skills array
- /users/username/skills/skill_id
 - GET
 - PUT/PATCH
 - DELETE
- /applications/app_id/skills
 - None
- /applications/app_id/skills/skill_id
 - POST, DELETE

Routes Not Needed:

- /skills
 - DELETE
 - PUT/PATCH
 - POST
- /skills/skill_id
 - GET
 - POST
 - PUT/PATCH
 - DELETE
- /users/username/skills
 - DELETE
 - PUT/PATCH
- /users/username/skills/skill_id
 - POST
- /applications/app_id/skills
 - GET, PUT/PATCH, POST, DELETE
- /applications/app_id/skills/skill_id
 - GET, PUT/PATCH

/skills GET - Get All Skills in DB

Request

Path Parameters

- None

Request Body

- Required? - N/A
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: "Bearer <Token>"

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	406 - Not Acceptable	Accept header is not application/json or */*

Response Examples

Success

<u>200 - OK</u> [{ "id": 12345, "description": "Python", }, { "id": 45678, "description": "DevOps", },]

Failure

401 - Unauthorized

```
{  
  "401": "JWT missing or invalid"  
}
```

406 - Not Acceptable

```
{  
  "406": "Accept header must be application/json or */*"  
}
```

/skills POST - Create New Skill

Request

Path Parameters

- None

Request Body

- Required? - Yes
- Format? - JSON

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: Bearer <Token>

Request JSON Attributes

Name	Description	Required?
description	String. Name of skill.	Yes

Request Body Example

```
{
  "description": "MS Office",
}
```

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	New skill is created in shared skills table - also returned if skill is existing in database
Fail	400 - Bad Request	Request body contains missing or extra information than what's required.
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match user in user database
Fail	406 - Not Acceptable	Accept header is not application/json or */*
Fail	415 - Unsupported Media	Request body is not JSON

Response Examples

Success

```
201 Created
{
  "id": 12345,
```

```
}    "description": "Python",  
}
```

Failure

400 - Bad Request

```
{  
    "400": "Bad request body. Contains unrecognized keys."  
}
```

401 - Unauthorized

```
{  
    "401": "JWT missing or invalid"  
}
```

403 - Forbidden

```
{  
    "403": "JWT doesn't belong to recognized user"  
}
```

406 - Not Acceptable

```
{  
    "406": "Accept header must be application/json or */*"  
}
```

415 - Unsupported Media

```
{  
    "415": "Request body must be JSON"  
}
```

/skills PUT/PATCH

Method not allowed.

/skills DELETE

Method not allowed.

/skills/:skill_id GET

Method not allowed.

/skills/:skill_id POST

Method not allowed.

/skills/:skill_id PUT/PATCH

Method not allowed.

/skills/:skill_id DELETE

Method not allowed.

/users/:user_id/skills GET - Get User's Skills

Request

Path Parameters

- user_id

Request Body

- Required? - None
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: Bearer <Token>

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success –
- Failure –

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match user
Fail	404 - Not Found	User doesn't exist
Fail	406 - Not Acceptable	Accept header is not application/json or */*

Response Examples

Success

<u>200 - OK</u> [{ "id": 12345, "description": "Python", "self": "url.com/users/:user_id/skills/12345" "proficiency": 4, "applications": [{ "id": 1234,

```
        "title": "Google SD 1",
        "self": "url.com/users/:user_id/applications/1234"
      }
    ],
    {
      "id": 45678,
      "description": "DevOps",
      "self": "url.com/users/:user_id/skills/45678"
      "proficiency": 1,
      "applications": []
    },
  ]
}
```

Failure

401 - Unauthorized

```
{
  "401": "JWT missing or invalid"
}
```

403 - Forbidden

```
{
  "403": "JWT and user_id don't match"
}
```

404 - Not Found

```
{
  "404": "User does not exist"
}
```

406 - Not Acceptable

```
{
  "406": "Accept header must be application/json or */*"
}
```


/users/:user_id/skills POST

Method not allowed

/users/:user_id/skills PUT/PATCH

Method not allowed.

/users/:user_id/skills DELETE - Delete all User's Skills

Method not allowed.

/users/:user_id/skills/:skill_id GET - Get User's Skill

Request

Path Parameters

- User_id, skill_id

Request Body

- Required? - N/A
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: Bearer <Token>

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match user or skill not tied to user
Fail	404 - Not Found	User doesn't exist or skill doesn't exist
Fail	406 - Not Acceptable	Accept header is not application/json or */*

Response Examples

Success

<pre>200 OK { "id": 12345, "description": "Python", "proficiency": 4, "applications": [{"id": 123, "title": "Google SD 1", "self": "url.com/applications/123"}, {"id": 123, "title": "Amazon SD 1", "self": "url.com/applications/123"},] }</pre>

Failure

401 - Unauthorized

```
{  
  "401": "JWT missing or invalid"  
}
```

403 - Forbidden

```
{  
  "403": "JWT and user_id don't match"  
}
```

404 - Not Found

```
{  
  "404": "User does not exist"  
}
```

406 - Not Acceptable

```
{  
  "406": "Accept header must be application/json or */*"  
}
```

/users/:user_id/skills/:skill_id POST

This method is not allowed.

/users/:user_id/skills/:skill_id PUT - Tie Existing Skill to User OR Update Skill Proficiency

Request

Path Parameters

- User_id, skill_id

Request Body

- Required? - Yes
- Format? - JSON

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: Bearer <Token>

Request JSON Attributes

Name	Description	Required?
proficiency	Number between 1 and 5 (5 being highest proficiency). Defaults to Null.	No

Request Body Example

<pre>{ "proficiency": 5, }</pre>
----- OR -----
<pre>{}</pre>

Response

Response Body Format

- Success – None
- Failure – JSON

Response Body Notes

- On success, no response body is returned, just status code.

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Existing skill is tied to user
Fail	400 - Bad Request	Request body contains missing or extra information than what's required OR skill is already tied to user.
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match user or skill not tied to user
Fail	404 - Not Found	User doesn't exist or skill doesn't exist
Fail	415 - Unsupported Media	Request body is not JSON

Response Examples

Success

204 No Content

Failure

400 - Bad Request

```
{  
  "400": "Bad request body. Contains unrecognized keys."  
}
```

401 - Unauthorized

```
{  
  "401": "JWT missing or invalid"  
}
```

403 - Forbidden

```
{  
  "403": "JWT and user_id don't match"  
}
```

404 - Not Found

```
{  
  "404": "User does not exist"  
}
```

415 - Unsupported Media

```
{  
  "415": "Request body must be JSON"  
}
```

/users/:user_id/skills/:skill_id PATCH

method not allowed

/users/:user_id/skills/:skill_id DELETE - Delete Skill from User

Request

Path Parameters

- User_id, skill_Id

Request Body

- Required? - N/A
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: Bearer <Token>

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – None
- Failure – JSON

Response Body Notes

-

Response Statuses

Outcome	Status Code	Notes
Success	204 - No Content	
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match user or skill not tied to user
Fail	404 - Not Found	User doesn't exist or skill doesn't exist

Response Examples

Success

<u>204 No Content</u>

Failure

<u>401 - Unauthorized</u> { "401": "JWT missing or invalid" }
<u>403 - Forbidden</u> { "403": "JWT and user_id don't match"


```
}
```

404 - Not Found

```
{
```

```
    "404": "User does not exist"
```

```
}
```

/users/:user_id/applications/:app_id/skills GET

Method not allowed.

/users/:user_id/applications/:app_id/skills POST

Method not allowed.

/users/:user_id/applications/:app_id/skills PUT/PATCH

Method not allowed.

/users/:user_id/applications/:app_id/skills DELETE

Method not allowed.

/users/:user_id/applications/:app_id/skills/:skill_id GET

Method not allowed.

/users/:user_id/applications/:app_id/skills/:skill_id POST

Method not allowed.

/users/:user_id/applications/:app_id/skills/:skill_id PUT - Tie a skill to a user's application

Request

Path Parameters

- User_id, app_id, skill_id

Request Body

- Required? - N/A
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: Bearer <Token>

Request Notes

- Skill ID and App ID must exist and app ID must be tied to the user. This route does not create skills and apps on the fly, and it does not tie apps to users on the fly. It does tie skills to users on the fly.

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- If the app is already tied to the skill, you'll still get a 204 status (not a 404)

Response Statuses

Outcome	Status Code	Notes
Success	204 - No Content	
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match user or app not tied to user
Fail	404 - Not Found	User, skill, or app don't exist

Response Examples

Success

<u>204 No Content</u>

Failure

<u>401 - Unauthorized</u>

```
{
  "401": "JWT missing or invalid"
}

403 - Forbidden
{
  "403": "JWT and user_id don't match"
}

404 - Not Found
{
  "404": "User does not exist"
}
```

/users/:user_id/applications/:app_id/skills/:skill_id PATCH

Method not allowed.

/users/:user_id/applications/:app_id/skills/:skill_id DELETE - Delete skill from a user's application

Request

Path Parameters

- User_id, app_id, skill_id

Request Body

- Required? - N/A
- Format? - N/A

Request Authorization

- Required? - Yes
- Type? - JWT
- Header: Authorization: Bearer <Token>

Request Notes

- Skill ID and App ID must exist and be tied to the user already.

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- If the app does not contain the skill already, you'll still get a 204 status (not a 404). We quietly allow this error.

Response Statuses

Outcome	Status Code	Notes
Success	204 - No Content	
Fail	401 - Unauthorized	JWT is missing or invalid
Fail	403 - Forbidden	JWT doesn't match user or skill not tied to user or app not tied to user
Fail	404 - Not Found	User, skill, or app don't exist

Response Examples

Success

<u>204 No Content</u>

Failure

<u>401 - Unauthorized</u>

```
{
  "401": "JWT missing or invalid"
}

403 - Forbidden
{
  "403": "JWT and user_id don't match"
}

404 - Not Found
{
  "404": "User does not exist"
}
```

Contact Routes

Needed Routes:

- /users/:user_id/contacts
 - POST
 - GET
- /users/:user_id/contacts/:contact_id
 - GET
 - PUT
 - PATCH
 - DELETE

Routes Not Allowed:

- /users/:user_id/contacts
 - DELETE
 - PUT
 - PATCH
- /users/:user_id/contacts/:contact_id
 - POST

Create a Contact

Allows to create a new contact.

Request

Path Parameters

- POST users/:user_id/contacts

Request Body

- Required
- Format: JSON

Request Authorization

- Required

Request JSON Attributes

Name	Description	Required?
first_name	Contact's first name	yes
last_name	Contact's last name	yes
email	Contact's email	no
phone	Contact's phone number	no
notes	Info about the contact	no
contact_at_app_id	Applications related to the contact	no

Request Body Example

```
{  
  "last_name": "Smith",  
  "first_name": "Rob",  
  "email": "Smith@Smith.com",  
  "phone": "888-8888-888",  
  "notes": "Added by user 1",  
  "contact_at_app_id": []  
}
```

Response

Response Body Format

- Success: JSON
- Failure: JSON

Response Body Notes

- Success: JSON with ID of the posted object, the ID is used when testing the app with Postman, otherwise not required.

Response Statuses

Outcome	Status Code	Notes
Success	201	
Failure #1	400	The user sends invalid object keys.

Failure #2	415	The user sends unsupported Content-Type.
Failure #3	400	Problem with JSON format in body.
Failure #4	401	JWT is Invalid.

Response Examples

Success

201 CREATED

"5769252946575360"

Failure #1

400 Bad request

```
{
  "Error": "The user sends invalid object keys."
}
```

Failure #2

415 Unsupported Media Type

```
{
  "Error": "The user sends unsupported Content-Type."
}
```

Failure #3

400 Bad request

```
{
  "Error": "Problem with JSON format in body."
}
```

Failure #4

401 Unauthorized

```
{
  "401": "JWT is Invalid!"
}
```

Delete a Contact

Allows to delete a contact.

Request

Path Parameters

- DELETE users/:user_id/contacts/:contact_id

Request Body

- None

Request Authorization

- Required

Request JSON Attributes

- None

Request Body Example

- None

Response

Response Body Format

- Success: None
- Failure: JSON, error message

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	204	
Failure #1	404	No contact exists with this id.
Failure #2	401	JWT is Invalid.

Response Examples

Success

204 No Content

Failure #1

404 Not Found

```
{
  "Error": "No contact exists with this id."
}
```

Failure #2

401 Unauthorized

```
{
  "401": "JWT is Invalid!"
}
```

Edit a Contact with PUT

Allows to replace all old contact's properties with new contact's properties.

Request

Path Parameters

- PUT users/:user_id/contacts/:contact_id

Request Body

- Required
- Format: JSON

Request Authorization

- Required

Request JSON Attributes

Name	Description	Required?
first_name	Contact's first name	yes
last_name	Contact's last name	yes
email	Contact's email	no
phone	Contact's phone number	no
notes	Info about the contact	no
contact_at_app_id	Applications related to the contact	no

Request Body Example

```
{
  "last_name": "Smith",
  "first_name": "Rob",
  "email": "Smith@Smith.com",
  "phone": "888-8888-888",
  "notes": "Put request by user 1",
  "contact_at_app_id": []
}
```

Response

Response Body Format

- Success: None
- Failure: JSON, error message

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	200	
Failure #1	403	User does not own resource.
Failure #2	404	No contact exists with this id.
Failure #3	400	The user sends invalid object keys.

Failure #4	415	The user sends unsupported Content-Type.
Failure #5	401	JWT is Invalid.

Response Examples

Success

200 OK

Failure #1

403 Forbidden

```
{
  "403": "User does not own resource"
}
```

Failure #2

404 Not Found

```
{
  "Error": "No contact exists with this id."
}
```

Failure #3

400 Bad Request

```
{
  "Error": "The user sends invalid object keys."
}
```

Failure #4

415 Unsupported Media Type

```
{
  "Error": "The user sends unsupported Content-Type."
}
```

Failure #5

401 Unauthorized

```
{
  "401": "JWT is Invalid!"
}
```

Edit a Contact with PATCH

Allows to update contact's properties.

Request

Path Parameters

- PATCH users/:user_id/contacts/:contact_id

Request Body

- Required
- Format: JSON

Request Authorization

- Required

Request JSON Attributes

Name	Description	Required?
first_name	Contact's first name	yes
last_name	Contact's last name	yes
email	Contact's email	no
phone	Contact's phone number	no
notes	Info about the contact	no
contact_at_app_id	Applications related to the contact	no

Request Body Example

<pre>{ "last_name": "patch last_name", "first_name": "patch first_name" }</pre>

Response

Response Body Format

- Success: None
- Failure: JSON, error message

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	200	
Failure #1	403	User does not own resource.
Failure #2	400	Required keys must have value.
Failure #3	404	No contact exists with this id.
Failure #4	415	The user sends unsupported Content-Type.
Failure #5	401	JWT is Invalid.

Response Examples

Success

200 OK

Failure #1

403 Forbidden

```
{  
  "403": "User does not own resource"  
}
```

Failure #2

400 Bad Request

```
{  
  "Error": "Required keys must have value."  
}
```

Failure #3

404 Not Found

```
{  
  "Error": "No contact exists with this id."  
}
```

Failure #4

415 Unsupported Media Type

```
{  
  "Error": "The user sends unsupported Content-Type."  
}
```

Failure #5

401 Unauthorized

```
{  
  "401": "JWT is Invalid!"  
}
```

Get a Contact

Allows to get an existing contact.

Request

Path Parameters

- GET users/:user_id/contacts/:contact_id

Request Body

- None

Request Authorization

- Required

Request JSON Attributes

- None

Request Body Example

- None

Response

Response Body Format

- Success: JSON
- Failure: JSON, error message

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	200	
Failure #1	403	User ID in JWT does not match URL.
Failure #2	403	User does not own resource.
Failure #3	406	The user sends unsupported Accept header.
Failure #4	404	No contact exists with this id.
Failure #5	401	JWT is Invalid.

Response Examples

Success

```
200 OK
{
  "notes": "Added by user 1",
  "last_name": "Smith",
  "user": "64076c422bdf99d93b8e74f2",
  "phone": "888-8888-888",
  "email": "Smith@Smith.com",
  "first_name": "Rob",
  "contact_at_app_id": [],
```

```
"id": "5707748519968768"
}
```

Failure #1

```
403 Forbidden
{
  "Error": "User ID in JWT does not match URL"
}
```

Failure #2

```
403 Forbidden
{
  "403": "User does not own resource"
}
```

Failure #3

```
406 Not Acceptable
{
  "Error": "The user sends unsupported Accept header."
}
```

Failure #4

```
404 Not Found
{
  "Error": "No contact exists with this id."
}
```

Failure #5

```
401 Unauthorized
{
  "401": "JWT is Invalid!"
}
```


Get all Contacts

Allows to get all contacts that belong to the user.

Request

Path Parameters

- GET users/:user_id/contacts/

Request Body

- None

Request Authorization

- Required

Request JSON Attributes

- None

Request Body Example

- None

Response

Response Body Format

- Success: JSON
- Failure: JSON, error message

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	200	
Failure #1	403	User ID in JWT does not match URL.
Failure #2	406	The user sends unsupported Accept header.
Failure #3	401	JWT is Invalid.

Response Examples

Success

200 OK
<pre>{ "firstName": "Rob L.", "phone": "111-2222-333", "notes": "Contact at Amazon", "email": "rob@rob.com", "lastName": "Smith", "id": "5764518391054336" }</pre>

Failure #1

403 Forbidden

```
{  
  "Error": "User ID in JWT does not match URL"  
}
```

Failure #2

406 Not Acceptable

```
{  
  "Error": "The user sends unsupported Accept header."  
}
```

Failure #3

401 Unauthorized

```
{  
  "401": "JWT is Invalid!"  
}
```

Application Routes

Create an application

Allows user to create a new application

POST /user/:user_id/applications

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource

Request Headers

- Header: Authorization: "Bearer <Token>"

Request Body

- Required

Request Body Format

- JSON

Request Authorization

- Required: Yes
- Type: JWT

Request JSON Attributes

Name	Description	Required?
user_id	Name of the user. Only a-z, A-Z and spaces are allowed as input. 1-50 character length	Yes
skills	Skills associated with the application.	No
contacts	Contacts associated with the application	No
title	A title for the application	Yes
description	A description of what the application entails	Yes
posting_date	The application's posting date	No
status	The user's action status related to the application	Yes
link	The external link to the application	No

Request Body Example

```
{
  "username": "Jsmith",
  "skills": [],
  "contacts": [],
```

```
"title": "Netflix Database Administrator",
"description": "Senior level database position...",
"posting_date": "1/25/23",
"status": "Applied",
"link": "https://www.glassdoor.com/Job/software-developer-jobs-ABCD_A09,1.htm"
}
```

Response

Response Body Format

- JSON

Response Body Notes

- Datastore will automatically generate an ID and store it with the entity being created.
- The response will generate a “self” attribute that links to the specific url where the individual application resource can be accessed.

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If any required attributes are missing the application is not created an a 400 status code is returned
Failure	403 Forbidden	Bearer Token does not match resource owner

Response Examples

Success

Status: 201 Created

```
{
  "id": 123,
  "username": "Jsmith",
  "skills": [],
  "contacts": [],
  "title": "Google Junior Developer",
  "description": "Entry level developer position...",
  "posting_date": "1/25/23",
  "last_modified": "1/25/23",
  "status": "Applied",
  "link": "https://www.glassdoor.com/Job/software-developer-jobs-ABCD_A09,1.htm"
  "self": "https://jobseeker.com/applications/123"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 403 Forbidden

```
{
  "Error": "Bearer Token does not match this owner of this resource"
}
```

List all applications

List all applications owned by a user

GET /users/:user_id/applications

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource

Request Body

- None

Request Headers

- Header: Authorization: "Bearer <Token>"

Response

Response Body Format

- JSON

Request Authorization

- Required? - Yes
- Type? - JWT

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	application/json supported
Failure	403 Forbidden	Bearer Token does not match resource owner

Response Body Notes

- The response will generate a "self" attribute that links to the specific url where the individual application resource can be accessed.

Response Examples

Success

Status: 200 OK

```
{
  "applications": [
    {
      "id": 123,
```

```

"username": "Jsmith",
"skills": [],
"contacts": [],
"title": "Google Junior Developer",
"description": "Entry level developer position...",
"date": "1/25/23",
"status": "Applied",
"link": "https://www.glassdoor.com/Job/software-developer-jobs-ABCD_A09,1.htm",
"self": "https://jobseeker.com/applications/123"
},

{
  "id": 456,
  "username": "MJohnson",
  "skills": [{ "id": 1234,
    "description": "Python",
    "self": "users/:user_id/skills/:skill_id" }],
  "contacts": [],
  "title": "Netflix Database Administrator",
  "description": "Senior level database position...",
  "date": "1/25/23",
  "status": "Applied",
  "link": "https://www.glassdoor.com/Job/software-developer-jobs-BCDE_A00,1.htm",
  "self": "https://jobseeker.com/applications/456"
}
],
"collection size":2

```

Failure

Status: 403 Forbidden

```

{
  "Error": "Bearer Token does not match this owner of this resource"
}

```

Get an Application

Get an existing application by id

GET /user/:user_id/applications/:application_id

Request

Path Parameters

Name	Description
username	Unique user id that owns the resource

application_id	ID of the application
----------------	-----------------------

Request Body

- None

Request Headers

- Header: Authorization: "Bearer <Token>"
- Accept: application/json(accepted format)

Request Authorization

- Required: Yes
- Type: JWT

Response

Response Body Format

- JSON

Response Body Notes

- The response will generate a "self" attribute that links to the specific url where the individual application resource can be accessed.

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	application/json supported
Failure	403 Forbidden	Bearer Token does not match resource owner
Failure	404 Not Found	No application with this application_id exists
Failure	406 Unsupported	Client requests unsupported MIME type from route via header

Response Examples

Success

<p>Status: 200 OK</p> <pre>{ { "id": 123, "username": "Jsmith", "skills": [], "contacts": [], "title": "Google Junior Developer", "description": "Entry level developer position...", "date": "1/25/23", "status": "Applied", "link": "https://www.glassdoor.com/Job/software-developer-jobs-ABCD_A09,1.htm", "self": "https://jobseeker.com/applications/123" } }</pre>
--

Failure

Status: 403 Forbidden

```
{
  "Error": "Bearer Token does not match this owner of this resource"
}
```

Status: 404 Not Found

```
{
  "Error": "No application with this application_id exists"
}
```

Status 406 Unsupported

```
{
  "Error": "Not Acceptable, only application/json is accepted"
}
```

Edit an Application

Modify one or more allowed attributes of an application by application id

PATCH /user/:user_id/applications/:application_id

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource
application_id	ID of the application

Request Headers

- Header: Authorization: "Bearer <Token>"

Request Body

- Required

Request Body Format

- JSON

Request Authorization

- Required: Yes
- Type: JWT

Request JSON Attributes

Name	Description	Required?
------	-------------	-----------

title	A title for the application	No
description	A description of what the application entails	No
date	The application's posting date	No
status	The user's action status related to the application	No
link	The external link to the application	No

Request Body Example

```
{
  "title": "Amazon Data Scientist",
  "status": "Not Applied"
}
```

Response

Response Body Format

- JSON

Response Body Notes

- The response will generate a “self” attribute that links to the specific url where the individual application resource can be accessed.

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request does not include at least one allowed attribute, the application is not modified and a 400 status code is returned
Failure	403 Forbidden	Editing of protected attributes is forbidden on this route
Failure	403 Forbidden	Bearer Token does not match resource owner
Failure	404 Not Found	No application with this application id exists
Failure	415 Unsupported	Client sends unsupported MIIME type to route. (Only application/JSON is supported)

Response Examples

Success

Status: 201 Created

```
{
  "id": 123,
  "username": "Jsmith",
  "skills": [],
  "contacts": [],
  "title": "Amazon Data Scientist",
  "description": "Entry level developer position...",
  "date": "1/25/23",
  "status": "Not Applied",
  "link": "https://www.glassdoor.com/Job/software-developer-jobs-ABCD_A09,1.htm",
  "self": "https://jobseeker.com/applications/123"
}
```

Failure

Status: 400 Bad Request

```
{
```

```
"Error": "At least one allowed attribute must be specified"
}

Status 403 Forbidden

{
  "Error": "Editing id, username, skills, and contacts is not allowed"
}

Status: 403 Forbidden

{
  "Error": "Bearer Token does not match this owner of this resource"
}

Status 404 Not Found

{
  "Error": "No application with this application id exists"
}

Status 415 Unsupported

{
  "Error": "Server only accepts application/json requests"
}
```

Delete an Application

Delete an existing application by id

```
DELETE /user/:user_id/applications/:application_id
```

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource
application_id	ID of the application

Request Body

- None

Request Headers

- Header: Authorization: "Bearer <Token>"

Request Authorization

- Required: Yes
- Type: JWT

Request JSON Attributes

Name	Description	Required?

Request Body Example

Response

Response Body Format

- Success – None (no body)
- Failure – JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	403 Forbidden	Bearer Token does not match resource owner
Failure	404 Not Found	No application with this application_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden


```
{
  "Error": "Bearer Token does not match this owner of this resource"
}
```


Status: 404 Not Found


```
{
  "Error": "No application with this application_id exists"
}
```

Add Skill to Application

Add a skill to the skills attribute of an application

POST /user/:user_id/applications/:application_id/skills

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource
application_id	ID of the application

Request Headers

- Header: Authorization: "Bearer <Token>"

Request Body

- Required

Request Body Format

- JSON

Request Authorization

- Required: Yes
- Type: JWT

Request JSON Attributes

Name	Description	Required?
skill	Id of the skill	Yes

Request Body Example

{ "skill": "123", }

Response

Response Body Format

- JSON

Response Body Notes

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	
Failure	400 Bad Request	If the request does not include at least one allowed attribute, the application is not modified and a 400 status code is returned
Failure	403 Forbidden	Bearer Token does not match resource owner
Failure	404 Not Found	No application with this application id exists
Failure	415 Unsupported	Client sends unsupported MIIME type to route. (Only application/JSON is supported)

Response Examples

Success

Status: 204 OK

```
{
  "id": 123
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "At least one allowed attribute must be specified"
}
```

Status: 403 Forbidden

```
{
  "Error": "Bearer Token does not match this owner of this resource"
}
```

Status 404 Not Found

```
{
  "Error": "No application with this application id exists"
}
```

Status 415 Unsupported

```
{
  "Error": "Server only accepts application/json requests"
}
```

Remove Skill from Application

Delete a skill from the skills attribute of an application

```
DELETE /user/:user_id/applications/:application_id/skills/:skill_id
```

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource
application_id	ID of the application
skill_id	ID of the skill

Request Headers

- Header: Authorization: "Bearer <Token>"

Request Body

- None

Request Authorization

- Required: Yes
- Type: JWT

Response

Response Body Format

- None

Response Body Notes

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	
Failure	400 Bad Request	If the request does not include at least one allowed attribute, the application is not modified and a 400 status code is returned
Failure	403 Forbidden	Bearer Token does not match resource owner
Failure	404 Not Found	No application with this application id exists

Response Examples

Success

Status: 204 OK

Failure

Status: 400 Bad Request

```
{
  "Error": "At least one allowed attribute must be specified"
}
```

Status: 403 Forbidden

```
{
  "Error": "Bearer Token does not match this owner of this resource"
}
```

Status 404 Not Found

```
{
  "Error": "No application with this application id exists"
}
```

Add Contact to Application

Add a contact to the skills attribute of an application

POST /user/:user_id/applications/:application_id/contacts

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource
application_id	ID of the application

Request Headers

- Header: Authorization: "Bearer <Token>"

Request Body

- Required

Request Body Format

- JSON

Request Authorization

- Required: Yes
- Type: JWT

Request JSON Attributes

Name	Description	Required?
contact	Id of the contact	Yes

Request Body Example

<pre>{ "contact": "123", }</pre>

Response

Response Body Format

- JSON

Response Body Notes

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	
Failure	400 Bad Request	If the request does not include at least one allowed attribute, the application is not modified and a 400 status code is returned
Failure	403 Forbidden	Bearer Token does not match resource owner
Failure	404 Not Found	No application with this application id exists
Failure	415 Unsupported	Client sends unsupported MIIME type to route. (Only application/JSON is supported)

Response Examples

Success

Status: 204 OK

```
{
  "id": 123
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "At least one allowed attribute must be specified"
}
```

Status: 403 Forbidden

```
{
  "Error": "Bearer Token does not match this owner of this resource"
}
```

Status 404 Not Found

```
{
  "Error": "No application with this application id exists"
}
```

Status 415 Unsupported

```
{
  "Error": "Server only accepts application/json requests"
}
```

Remove Contact from Application

Delete a contact from the contacts attribute of an application

```
DELETE /user/:user_id/applications/:application_id/contacts/:contact_id
```

Request

Path Parameters

Name	Description
user_id	Unique user id that owns the resource
application_id	ID of the application
contact_id	ID of the contact

Request Headers

- Header: Authorization: "Bearer <Token>"

Request Body

- None

Request Authorization

- Required: Yes
- Type: JWT

Response

Response Body Format

- None

Response Body Notes

Response Statuses

Outcome	Status Code	Notes
Success	204 OK	
Failure	400 Bad Request	If the request does not include at least one allowed attribute, the application is not modified and a 400 status code is returned
Failure	403 Forbidden	Bearer Token does not match resource owner
Failure	404 Not Found	No application with this application id exists

Response Examples

Success

Status: 204 OK

Failure

Status: 400 Bad Request

```
{
  "Error": "At least one allowed attribute must be specified"
}
```

Status: 403 Forbidden

```
{
  "Error": "Bearer Token does not match this owner of this resource"
}
```

Status 404 Not Found

```
{
  "Error": "No application with this application id exists"
}
```

Application Methods Not Allowed

Methods not allowed on applications routes

PATCH /user/:user_id/applications
PUT /users/:user_id/applications
DELETE /user/:user_id/applications

Response Statuses

Outcome	Status Code	Notes
Failure	405 Not Allowed	Only GET and POST requests allowed for all applications route

Response Examples

Failure

Status: 405 Not Allowed

```
{
  "Error": "Only GET and POST requests allowed for all applications route"
}
```