

Full Stack Project

Pet Adoption Platform

The backend portion of a pet adoption application will be responsible for providing all the data the frontend requires. All inputs received from the user should be validated. Keep your folder structures clean and scalable. Below is a list of API's that will need to be implemented for this application.

You can start your application by simply adding routes and functionality, or you can structure your application first (using an MVC approach or any other), or feel free to do a mashup of these options.

Notes:

- Make sure to protect any routes that need to be protected.
- Validate any incoming user input. Handle any errors.
- You are welcome to use any external libraries of your choosing.
- The route names are suggestions, feel free to call them whatever makes most sense to you.
- The implementation details are up to you, if you would like to use JWT for authentication, feel free to do so, whatever method you prefer.

Signup API

route: '/signup' [POST]

- The signup api is responsible for signing up a new user.
- Validate all the user input is valid
- Check that passwords match

- Make sure the email address is unique
- Store the user in your DB and log the user in
- Be sure not to save the users password as a plain string. (bcrypt is a great tool for this)

Fields:

- Email Address
- Password (twice to make sure passwords match)
- First and last name
- Phone number

Login API

route: '/login' [POST]

- The login api is responsible for logging in existing users
- Validate all the user input is valid
- Check the email and password match an existing user
- Retrieve the users data from the database and login the user.

Fields:

- Email address
- Password

Add Pet API

Route: '/pet' [POST] (Protected to admin only)

- The add pet api is responsible for adding new pets
- Validate all the user input is valid
- Handle photo upload
- Store pet information in the database

Fields:

- Type
- Name

- Adoption Status (Adopted, Fostered, Available)
- Picture (Picture location URL/Path)
- Height (number)
- Weight (Number)
- Color
- Bio
- Hypoallergenic (Boolean)
- Dietary restrictions
- Breed

Get Pet By ID API

Route: '/pet/:id' [GET]

- Get a pet by ID should take an id and return the corresponding pet from the database.

Edit Pet API

Route: '/pet/:id' [PUT] (protected to admin only)

- The add pet api is responsible for editing pets
- Validate all the user input is valid
- Handle photo upload
- Store pet information in the database

Fields: Same as Add Pet API

Get Pets API

Route: '/pet' [GET]

- The get pets API is responsible for retrieving pets that match the criteria given.
- Can receive query parameters to search the database
- Retrieve results to match query. If no parameters are passed it should return all the results.
- Should only return the fields necessary

Search Fields:

- Adoption Status
- Type
- Height
- Weight
- Name

Adopt/Foster API

Route '/pet/:id/adopt' [POST] (protected to logged in users)

- The Adopt/Foster API is responsible for adding the pet to the current users pets.
- This API also should change the pet's adoption status.

Field:

- Type (Adopt or foster)

Return Pet API

Route '/pet/:id/return' [POST] (protected to logged in users)

- The Return Pet API is responsible for returning the pet to the agency.
- The API should change the pets status back to available
- The API should remove the pet from the users pets.

Save Pet API

Route '/pet/:id/save' [POST] (protected to logged in users)

- The save PET api allows a user to save a pet for later
- The saved pet should be stored as saved in the users account

Delete Saved Pet API

Route '/pet/:id/save' [DELETE] (protected to logged in users)

- The save PET api allows a user to remove a saved pet.

Get Pets By User ID API

Route '/pet/user/:id' [GET]

- This api allows a user to get the pets owned by (or saved) by a user based on the user id.

Get User By ID API

Route '/user/:id' [GET]

- This api allows you to get a user based on the user's id.

Update User API

Route '/user/:id' [PUT] (protected to logged in user)

- This API allows you to change your settings while logged in.
- Validate user inputs
- Ensure that if the email is being changed it's not already in use

Fields:

- Password
- Email
- first name
- last name
- phone number
- bio

Get Users API

Route '/user' [GET] (protected to admin)

- The GET users API returns all users in the DB.
- The API should only return the information required

Get User By ID API

Route '/user/:id/full' [GET]

- This api allows you to get a user based on the user's id.
- The API should return all the user details (aside from password) and the users pets they own.