DRIVEWAY

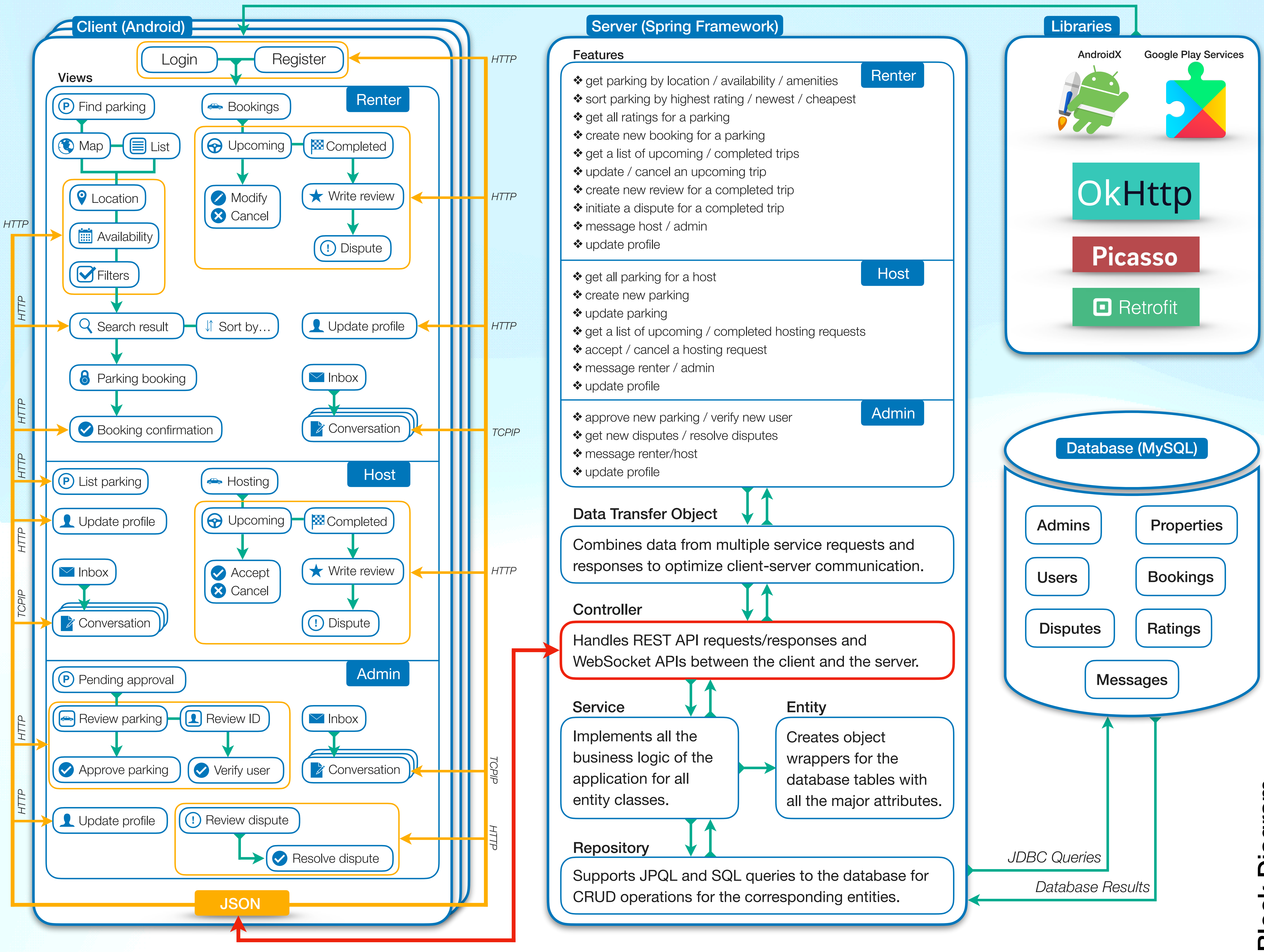
Parking

Block Diagram / Database Schema

1_CW_7

Member	Contribution
Varun Advani	25%
Jia Ze Chong	25%
Lindsay Timko	25%
Alex Volkov	25%

Block Diagram



Client (Android)

Login · Register

Views

Renter

- Find parking
  - Map · List
  - Location · Availability · Filters
  - Search result · Sort by…
  - Parking booking
  - Booking confirmation
- Bookings
  - Upcoming → Modify · Cancel
  - Completed → Write review → Dispute
- Update profile
- Inbox → Conversation

Host

- List parking
- Update profile
- Inbox → Conversation
- Hosting
  - Upcoming → Accept · Cancel
  - Completed → Write review → Dispute

Admin

- Pending approval
  - Review parking → Approve parking
  - Review ID → Verify user
- Inbox → Conversation
- Update profile
- Review dispute → Resolve dispute

HTTP · TCPIP · JSON

Server (Spring Framework)

Features

Renter

- ❖ get parking by location / availability / amenities
- ❖ sort parking by highest rating / newest / cheapest
- ❖ get all ratings for a parking
- ❖ create new booking for a parking
- ❖ get a list of upcoming / completed trips
- ❖ update / cancel an upcoming trip
- ❖ create new review for a completed trip
- ❖ initiate a dispute for a completed trip
- ❖ message host / admin
- ❖ update profile

Host

- ❖ get all parking for a host
- ❖ create new parking
- ❖ update parking
- ❖ get a list of upcoming / completed hosting requests
- ❖ accept / cancel a hosting request
- ❖ message renter / admin
- ❖ update profile

Admin

- ❖ approve new parking / verify new user
- ❖ get new disputes / resolve disputes
- ❖ message renter/host
- ❖ update profile

Data Transfer Object

Combines data from multiple service requests and responses to optimize client-server communication.

Controller

Handles REST API requests/responses and WebSocket APIs between the client and the server.

Service

Implements all the business logic of the application for all entity classes.

Entity

Creates object wrappers for the database tables with all the major attributes.

Repository

Supports JPQL and SQL queries to the database for CRUD operations for the corresponding entities.

Libraries

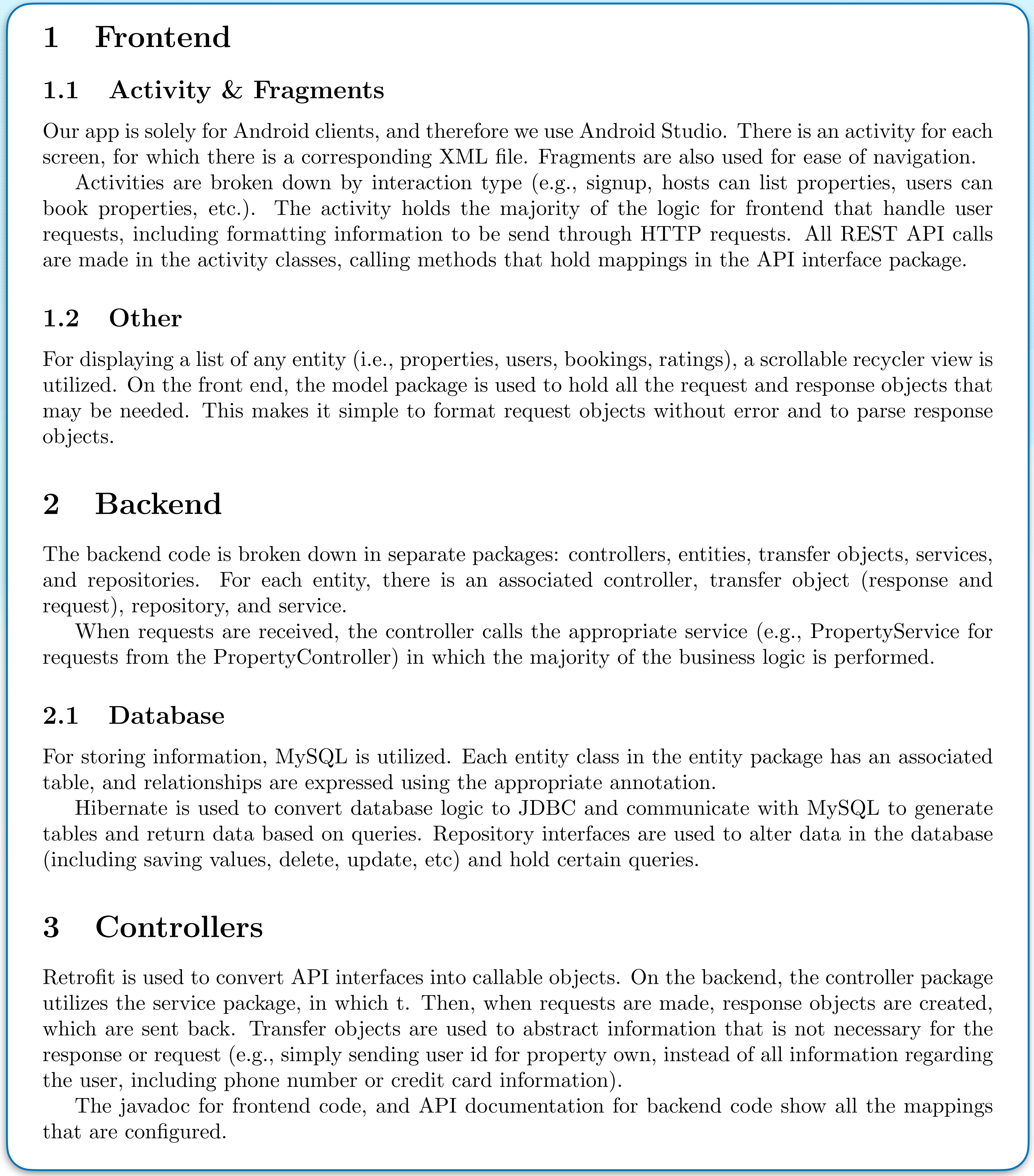
AndroidX · Google Play Services · OkHttp · Picasso · Retrofit

Database (MySQL)

Admins · Properties · Users · Bookings · Disputes · Ratings · Messages

JDBC Queries · Database Results

Design Description



1 Frontend

1.1 Activity & Fragments

Our app is solely for Android clients, and therefore we use Android Studio. There is an activity for each screen, for which there is a corresponding XML file. Fragments are also used for ease of navigation.

Activities are broken down by interaction type (e.g., signup, hosts can list properties, users can book properties, etc.). The activity holds the majority of the logic for frontend that handle user requests, including formatting information to be send through HTTP requests. All REST API calls are made in the activity classes, calling methods that hold mappings in the API interface package.

1.2 Other

For displaying a list of any entity (i.e., properties, users, bookings, ratings), a scrollable recycler view is utilized. On the front end, the model package is used to hold all the request and response objects that may be needed. This makes it simple to format request objects without error and to parse response objects.

2 Backend

The backend code is broken down in separate packages: controllers, entities, transfer objects, services, and repositories. For each entity, there is an associated controller, transfer object (response and request), repository, and service.

When requests are received, the controller calls the appropriate service (e.g., PropertyService for requests from the PropertyController) in which the majority of the business logic is performed.

2.1 Database

For storing information, MySQL is utilized. Each entity class in the entity package has an associated table, and relationships are expressed using the appropriate annotation.

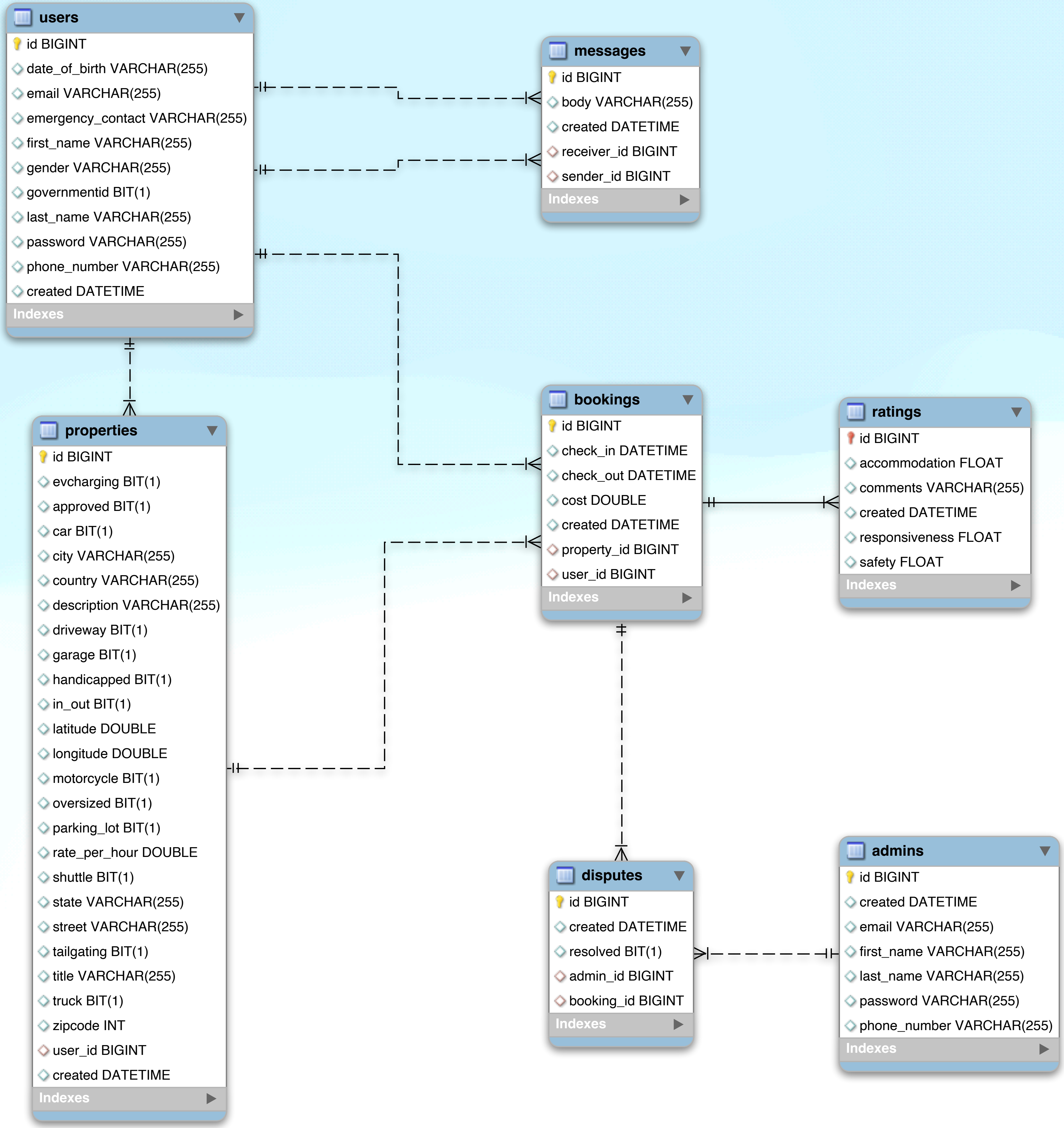
Hibernate is used to convert database logic to JDBC and communicate with MySQL to generate tables and return data based on queries. Repository interfaces are used to alter data in the database (including saving values, delete, update, etc) and hold certain queries.

3 Controllers

Retrofit is used to convert API interfaces into callable objects. On the backend, the controller package utilizes the service package, in which t. Then, when requests are made, response objects are created, which are sent back. Transfer objects are used to abstract information that is not necessary for the response or request (e.g., simply sending user id for property own, instead of all information regarding the user, including phone number or credit card information).

The javadoc for frontend code, and API documentation for backend code show all the mappings that are configured.

Database Schema



users
id BIGINT
date_of_birth VARCHAR(255)
email VARCHAR(255)
emergency_contact VARCHAR(255)
first_name VARCHAR(255)
gender VARCHAR(255)
governmentid BIT(1)
last_name VARCHAR(255)
password VARCHAR(255)
phone_number VARCHAR(255)
created DATETIME
Indexes

messages
id BIGINT
body VARCHAR(255)
created DATETIME
receiver_id BIGINT
sender_id BIGINT
Indexes

properties
id BIGINT
evcharging BIT(1)
approved BIT(1)
car BIT(1)
city VARCHAR(255)
country VARCHAR(255)
description VARCHAR(255)
driveway BIT(1)
garage BIT(1)
handicapped BIT(1)
in_out BIT(1)
latitude DOUBLE
longitude DOUBLE
motorcycle BIT(1)
oversized BIT(1)
parking_lot BIT(1)
rate_per_hour DOUBLE
shuttle BIT(1)
state VARCHAR(255)
street VARCHAR(255)
tailgating BIT(1)
title VARCHAR(255)
truck BIT(1)
zipcode INT
user_id BIGINT
created DATETIME
Indexes

bookings
id BIGINT
check_in DATETIME
check_out DATETIME
cost DOUBLE
created DATETIME
property_id BIGINT
user_id BIGINT
Indexes

ratings
id BIGINT
accommodation FLOAT
comments VARCHAR(255)
created DATETIME
responsiveness FLOAT
safety FLOAT
Indexes

disputes
id BIGINT
created DATETIME
resolved BIT(1)
admin_id BIGINT
booking_id BIGINT
Indexes

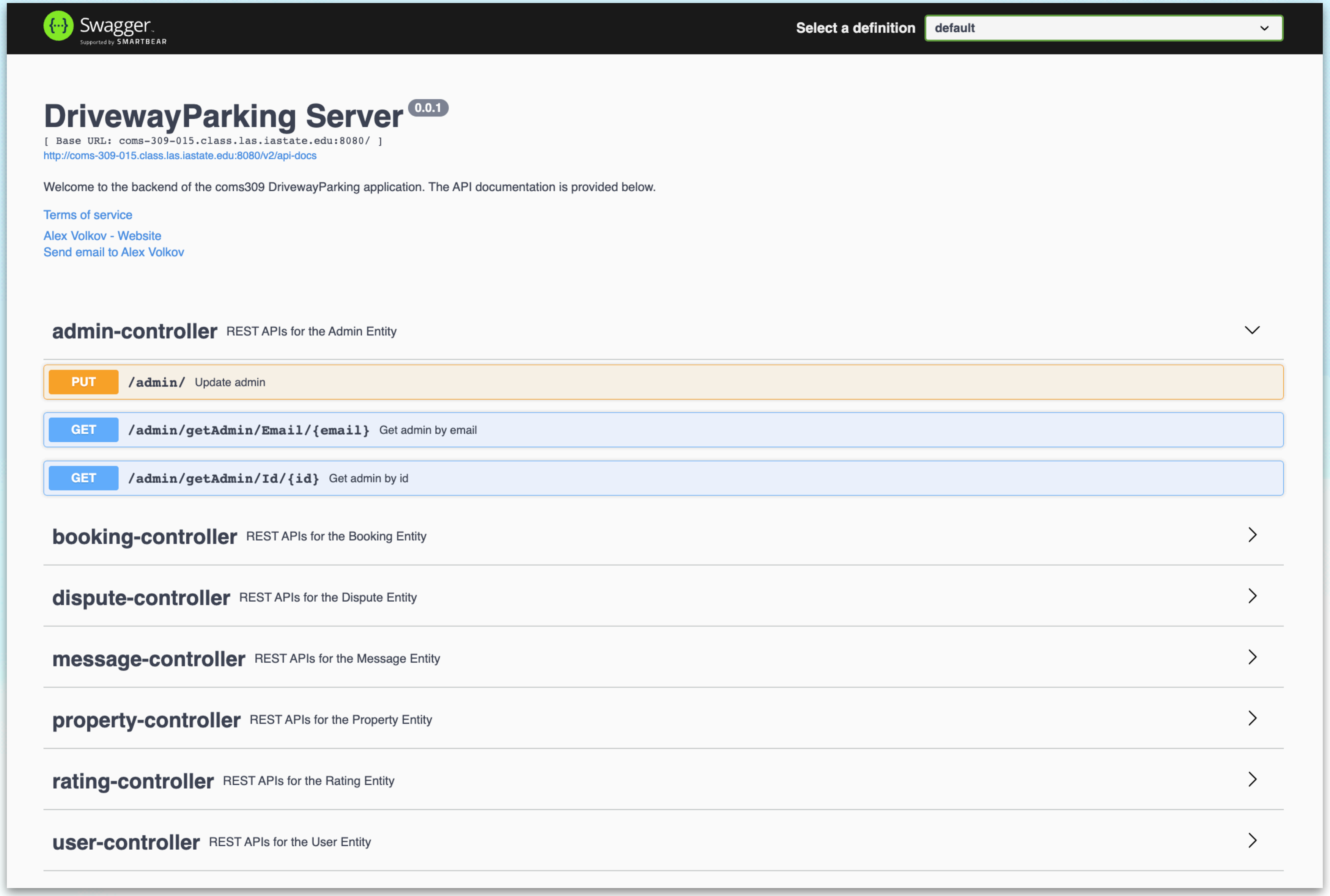
admins
id BIGINT
created DATETIME
email VARCHAR(255)
first_name VARCHAR(255)
last_name VARCHAR(255)
password VARCHAR(255)
phone_number VARCHAR(255)
Indexes

Java Docs API Documentation

driveway_parking_javadoc

View Online

Swagger API Documentation



Select a definition: default

DrivewayParking Server 0.0.1

`[ Base URL: coms-309-015.class.las.iastate.edu:8080/ ]`

http://coms-309-015.class.las.iastate.edu:8080/v2/api-docs

Welcome to the backend of the coms309 DrivewayParking application. The API documentation is provided below.

Terms of service

Alex Volkov - Website
Send email to Alex Volkov

admin-controller REST APIs for the Admin Entity

- PUT `/admin/` Update admin
- GET `/admin/getAdmin/Email/{email}` Get admin by email
- GET `/admin/getAdmin/Id/{id}` Get admin by id

booking-controller REST APIs for the Booking Entity

dispute-controller REST APIs for the Dispute Entity

message-controller REST APIs for the Message Entity

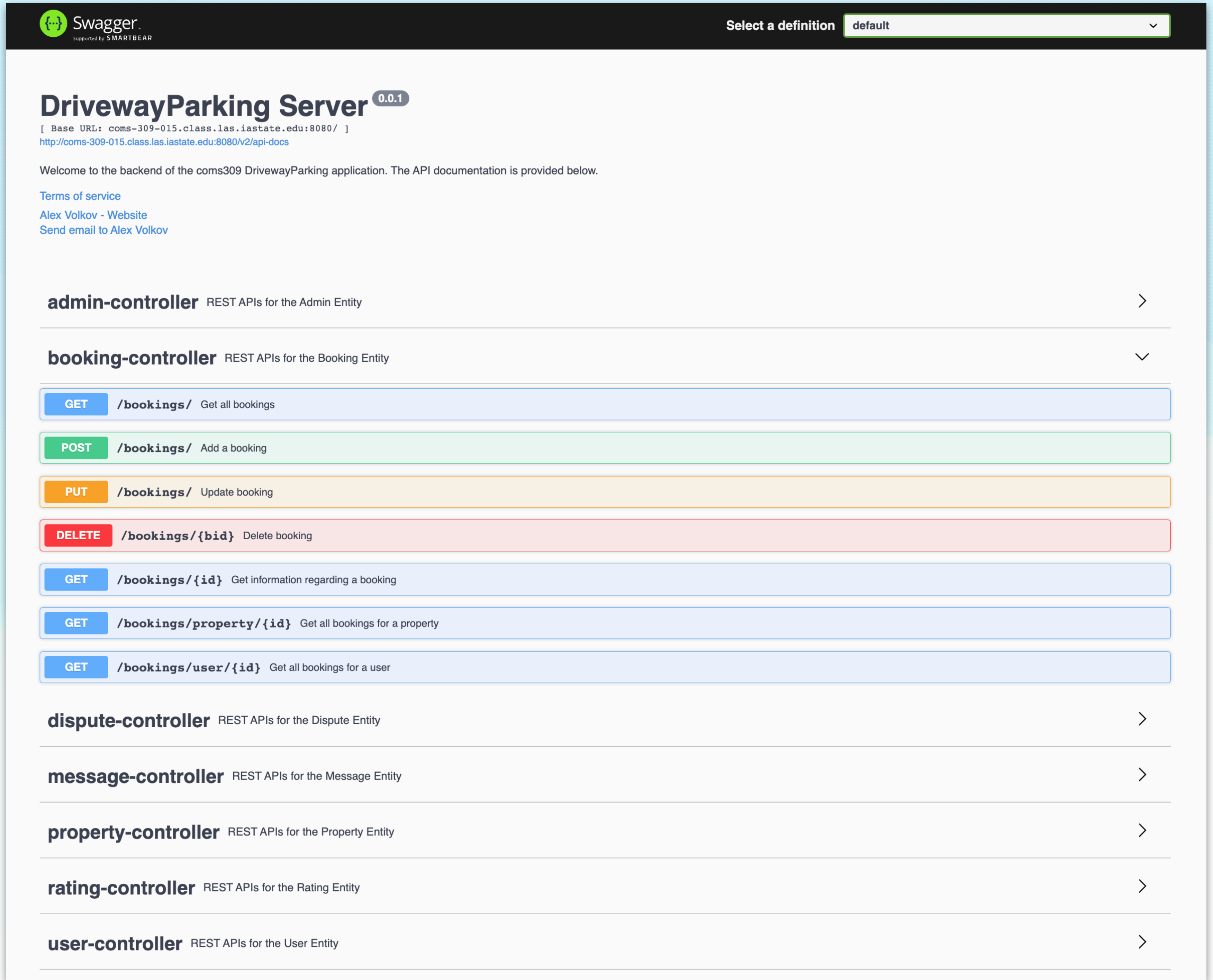
property-controller REST APIs for the Property Entity

rating-controller REST APIs for the Rating Entity

user-controller REST APIs for the User Entity

View Online

Swagger API Documentation



Select a definition: default

DrivewayParking Server 0.0.1

[Base URL: coms-309-015.class.las.iastate.edu:8080/]

http://coms-309-015.class.las.iastate.edu:8080/v2/api-docs

Welcome to the backend of the coms309 DrivewayParking application. The API documentation is provided below.

Terms of service

Alex Volkov - Website
Send email to Alex Volkov

admin-controller REST APIs for the Admin Entity

booking-controller REST APIs for the Booking Entity

- GET `/bookings/` Get all bookings
- POST `/bookings/` Add a booking
- PUT `/bookings/` Update booking
- DELETE `/bookings/{bid}` Delete booking
- GET `/bookings/{id}` Get information regarding a booking
- GET `/bookings/property/{id}` Get all bookings for a property
- GET `/bookings/user/{id}` Get all bookings for a user

dispute-controller REST APIs for the Dispute Entity

message-controller REST APIs for the Message Entity

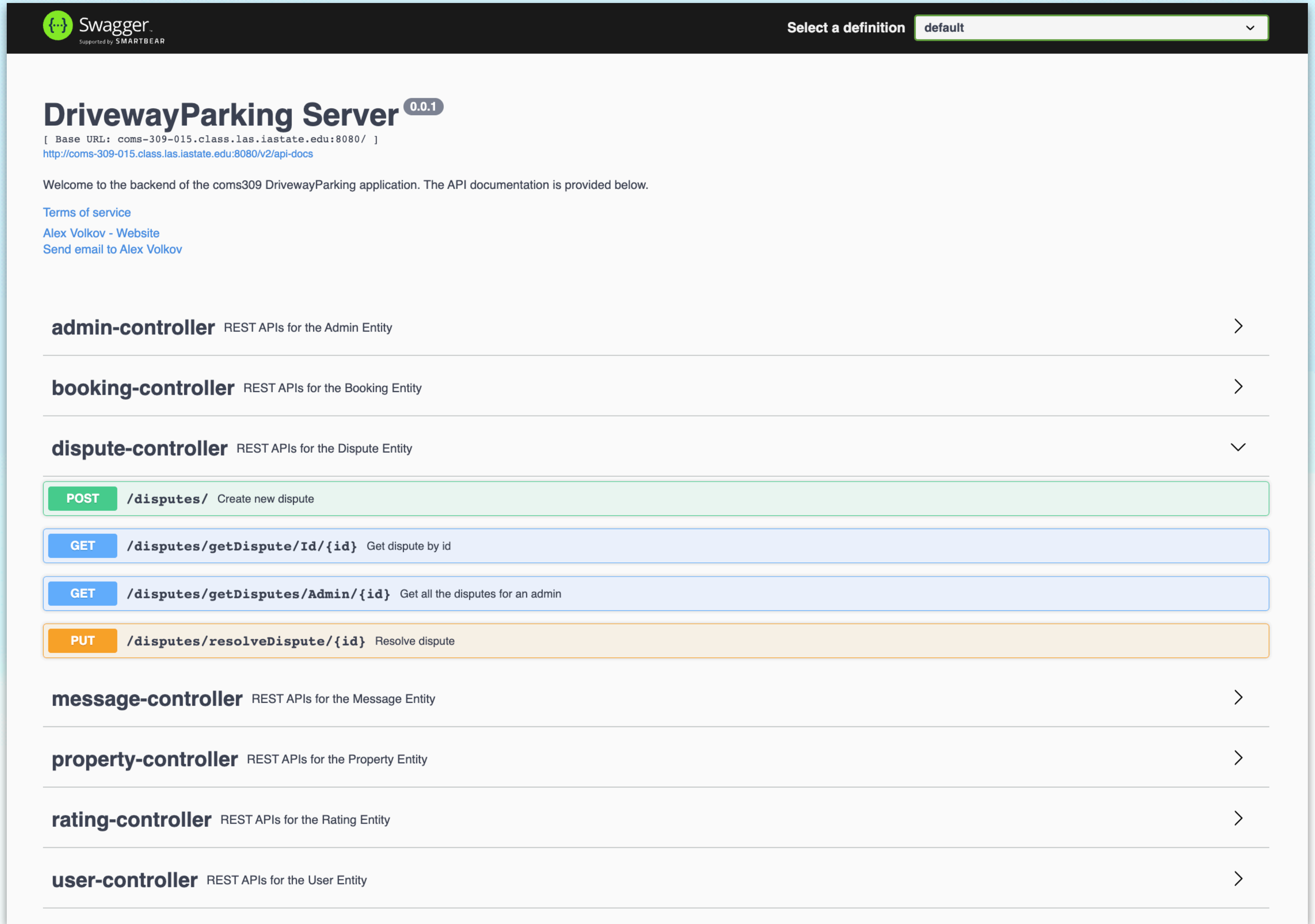
property-controller REST APIs for the Property Entity

rating-controller REST APIs for the Rating Entity

user-controller REST APIs for the User Entity

View Online

Swagger API Documentation



Select a definition default

DrivewayParking Server 0.0.1

[Base URL: coms-309-015.class.las.iastate.edu:8080/]

http://coms-309-015.class.las.iastate.edu:8080/v2/api-docs

Welcome to the backend of the coms309 DrivewayParking application. The API documentation is provided below.

Terms of service

Alex Volkov - Website
Send email to Alex Volkov

admin-controller REST APIs for the Admin Entity

booking-controller REST APIs for the Booking Entity

dispute-controller REST APIs for the Dispute Entity

- POST `/disputes/` Create new dispute
- GET `/disputes/getDispute/Id/{id}` Get dispute by id
- GET `/disputes/getDisputes/Admin/{id}` Get all the disputes for an admin
- PUT `/disputes/resolveDispute/{id}` Resolve dispute

message-controller REST APIs for the Message Entity

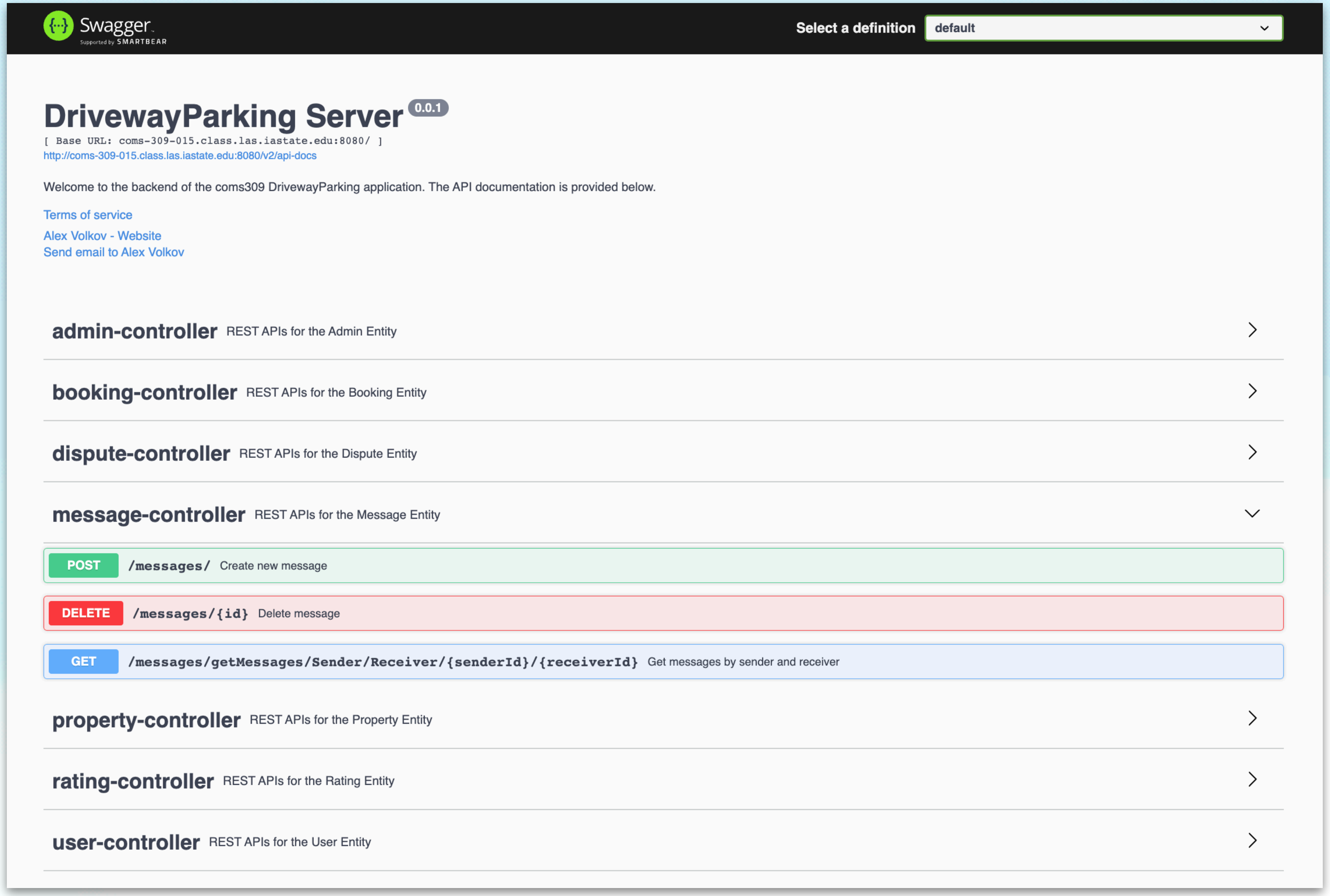
property-controller REST APIs for the Property Entity

rating-controller REST APIs for the Rating Entity

user-controller REST APIs for the User Entity

[View Online]

Swagger API Documentation



Select a definition: default

DrivewayParking Server 0.0.1

[Base URL: coms-309-015.class.las.iastate.edu:8080/]
http://coms-309-015.class.las.iastate.edu:8080/v2/api-docs

Welcome to the backend of the coms309 DrivewayParking application. The API documentation is provided below.

Terms of service

Alex Volkov - Website
Send email to Alex Volkov

admin-controller REST APIs for the Admin Entity

booking-controller REST APIs for the Booking Entity

dispute-controller REST APIs for the Dispute Entity

message-controller REST APIs for the Message Entity

- POST `/messages/` Create new message
- DELETE `/messages/{id}` Delete message
- GET `/messages/getMessages/Sender/Receiver/{senderId}/{receiverId}` Get messages by sender and receiver

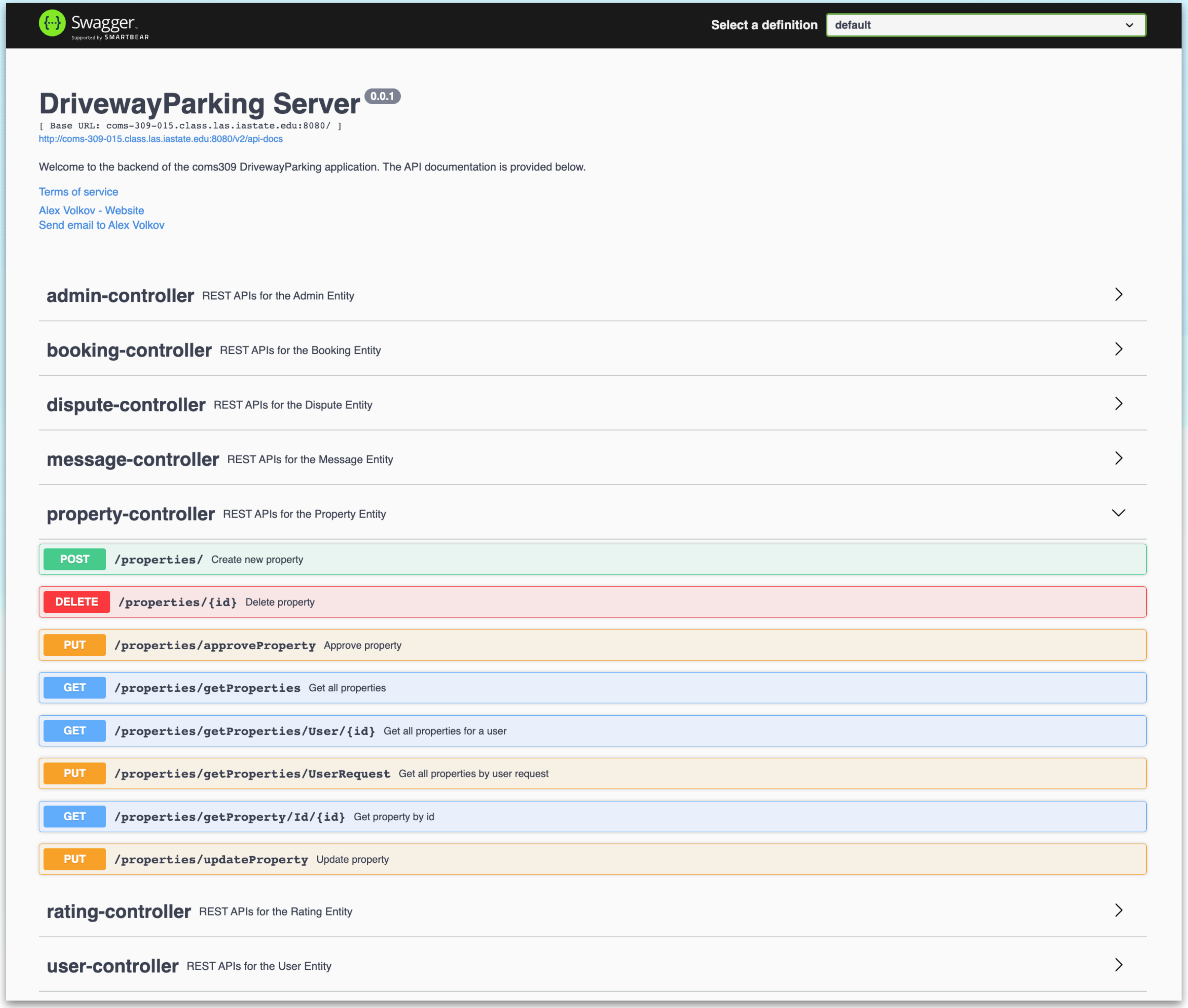
property-controller REST APIs for the Property Entity

rating-controller REST APIs for the Rating Entity

user-controller REST APIs for the User Entity

View Online

Swagger API Documentation



DrivewayParking Server 0.0.1

`[ Base URL: coms-309-015.class.las.iastate.edu:8080/ ]`

http://coms-309-015.class.las.iastate.edu:8080/v2/api-docs

Welcome to the backend of the coms309 DrivewayParking application. The API documentation is provided below.

Terms of service

Alex Volkov - Website

Send email to Alex Volkov

admin-controller REST APIs for the Admin Entity

booking-controller REST APIs for the Booking Entity

dispute-controller REST APIs for the Dispute Entity

message-controller REST APIs for the Message Entity

property-controller REST APIs for the Property Entity

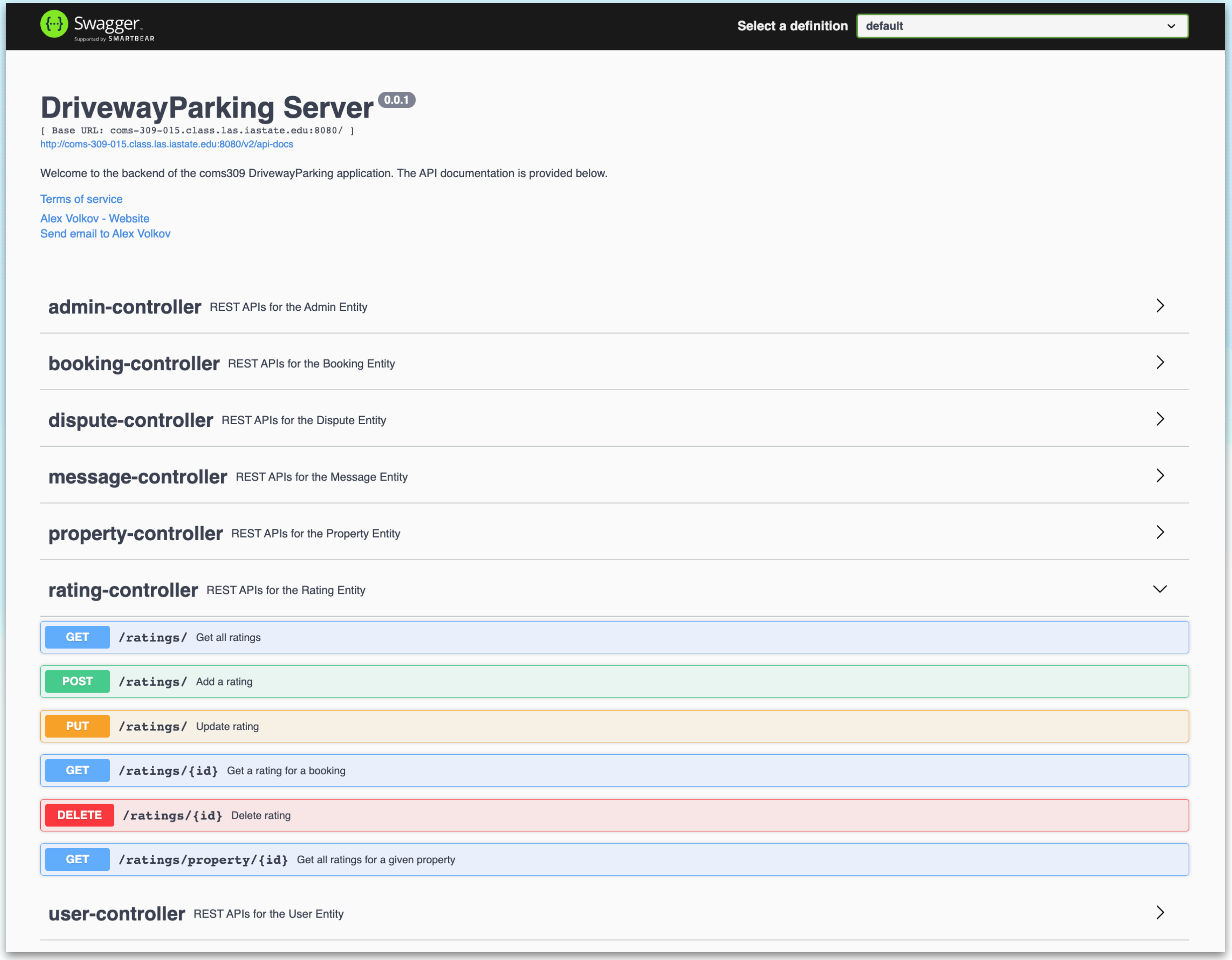
- POST `/properties/` Create new property
- DELETE `/properties/{id}` Delete property
- PUT `/properties/approveProperty` Approve property
- GET `/properties/getProperties` Get all properties
- GET `/properties/getProperties/User/{id}` Get all properties for a user
- PUT `/properties/getProperties/UserRequest` Get all properties by user request
- GET `/properties/getProperty/Id/{id}` Get property by id
- PUT `/properties/updateProperty` Update property

rating-controller REST APIs for the Rating Entity

user-controller REST APIs for the User Entity

View Online

Swagger API Documentation



DrivewayParking Server 0.0.1

[Base URL: coms-309-015.class.las.iastate.edu:8080/]
http://coms-309-015.class.las.iastate.edu:8080/v2/api-docs

Welcome to the backend of the coms309 DrivewayParking application. The API documentation is provided below.

Terms of service

Alex Volkov - Website
Send email to Alex Volkov

admin-controller REST APIs for the Admin Entity

booking-controller REST APIs for the Booking Entity

dispute-controller REST APIs for the Dispute Entity

message-controller REST APIs for the Message Entity

property-controller REST APIs for the Property Entity

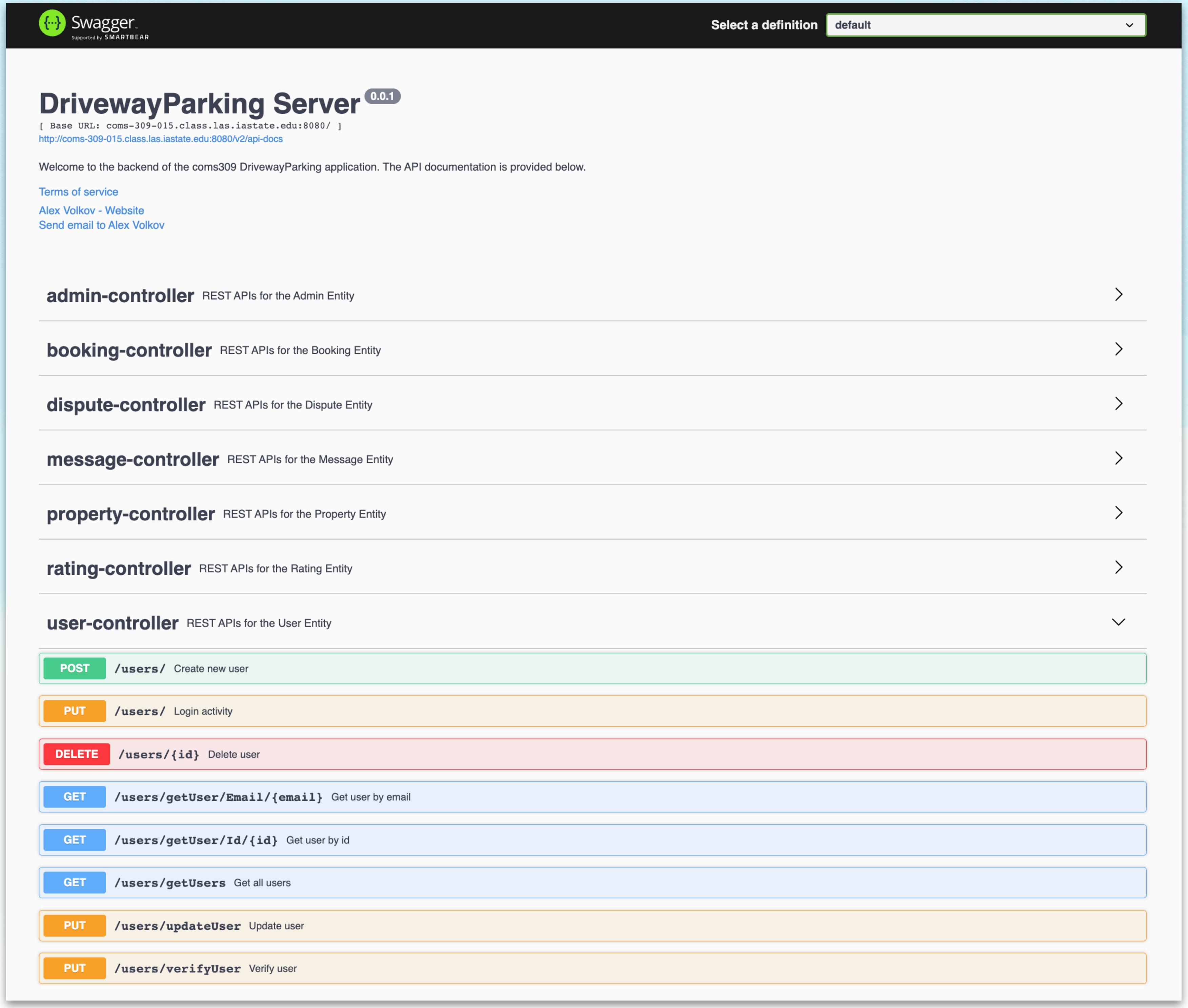
rating-controller REST APIs for the Rating Entity

- GET `/ratings/` Get all ratings
- POST `/ratings/` Add a rating
- PUT `/ratings/` Update rating
- GET `/ratings/{id}` Get a rating for a booking
- DELETE `/ratings/{id}` Delete rating
- GET `/ratings/property/{id}` Get all ratings for a given property

user-controller REST APIs for the User Entity

View Online

Swagger API Documentation



DriveawayParking Server 0.0.1

[Base URL: coms-309-015.class.las.iastate.edu:8080/]
http://coms-309-015.class.las.iastate.edu:8080/v2/api-docs

Welcome to the backend of the coms309 DrivewayParking application. The API documentation is provided below.

Terms of service

Alex Volkov - Website
Send email to Alex Volkov

admin-controller REST APIs for the Admin Entity

booking-controller REST APIs for the Booking Entity

dispute-controller REST APIs for the Dispute Entity

message-controller REST APIs for the Message Entity

property-controller REST APIs for the Property Entity

rating-controller REST APIs for the Rating Entity

user-controller REST APIs for the User Entity

- POST `/users/` Create new user
- PUT `/users/` Login activity
- DELETE `/users/{id}` Delete user
- GET `/users/getUser/Email/{email}` Get user by email
- GET `/users/getUser/Id/{id}` Get user by id
- GET `/users/getUsers` Get all users
- PUT `/users/updateUser` Update user
- PUT `/users/verifyUser` Verify user

View Online