Kevin Miller

SWDV 665

03/28/2021

Re-Design: Service Layer(s)

**EDIT:** This has not changed from week 2. Although I thought about making changes I decided this was the best route to take.

**Identifies all service endpoints required for your MVP.**

The user sends a GET request to the rest api which corresponds with node and express CRUD operations will take place and a POST response will be sent back to the user.

**Describes the purpose of each endpoint based on the goals of a user persona as it relates to their data (i.e., not their intentions with respect the feedback they want, but how what they want to do will change their data).**

Icon

Description automatically generated

Michael uses Leafable with his job as an moderator for an online game. He has been alerted that someone was caught botting and the account needs to be deleted. In this case, Michael is the user he sends the GET request to the api for player records. Once the account has been found a POST response is sent to Michael and he then deletes the account.

**Provides example requests and responses, including erroneous requests (e.g., error message responses for when bad data is input).**

Ill provide the code I have so far for my project here as I feel its the best method to use.

const axios = require('axios');

exports.homeRoutes = (req, res) => {

// Make a get request to /api/users

axios.get('http://localhost:3000/api/users')

.then(function (response) {

res.render('index', { users: response.data });

})

.catch(err => {

res.send(err);

})

}

exports.add\_user = (req, res) => {

res.render('add\_user');

}

exports.update\_user = (req, res) => {

axios.get('http://localhost:3000/api/users', { params: { id: req.query.id } })

.then(function (userdata) {

res.render("update\_user", { user: userdata.data })

})

.catch(err => {

res.send(err);

})

}

axios.post('/user', {

id: A-000212

userName: 'Finn',

})

.then((response) => {

console.log(response);

}, (error) => {

console.log(error);

});

**Diagrams how communication will go from your user interface pages to service endpoints.**

**Diagram

Description automatically generated**