Max points 100. All questions carry equal weight

1. True or False. nodejs allows for non-blocking request handling.

True

1. Write client-side code that asynchronously makes a request for to obtain json data from local host running at port 3456 at url “/books”. (Hint: I want a saga)

export async function\* getBooks() {

    const requestURL = await fetch('http://localhost:3456/books.json');

    try {

        const books = yield call(request, requestURL);

        yield put(booksLoaded(books));

    } catch (err){

        yield put(bookLoadingError(err));

    }

}

1. Write a nodejs server-side handler for requests at “/books”. In the handler, read a json file called books.json and send it to the client. Use only streams for this purpose.

const sendBooks = async () => {

    const ws1 = fs.createWriteStream('output.json');

    const books = await fetch('books.json');

    books.body.pipe(ws1);

    const rs1 = fs.createReadStream(`${\_\_dirname}/output.json`);

    console.log(JSON.stringify(rs1));

}

1. Write another nodejs server-side handler for requests at “/books”. In the handler, send a request to bookstore.com/books and after receiving the response, send it to the client. Use only streams for this purpose.

const sendBooks = async () => {

    const url1 = 'bookstore.com/books';

    const ws1 = fs.createWriteStream('output.json');

    const books = await fetch(url1);

    books.body.pipe(ws1);

    const rs1 = fs.createReadStream(`${\_\_dirname}/output.json`);

    console.log(JSON.stringify(rs1));

}

1. Write client-side code to obtain the price of a specific book from localhost:3456/books/title.

export async function\* getPrice() {

    const title = yield select(makeTitle());

    const requestURL = `http://localhost:3456/books/${title}`;

    try {

        const books = yield call(request, requestURL);

        yield put(booksLoaded(books, title));

    } catch (err){

        yield put(bookLoadingError(err));

    }

}

1. Write server-side code to send the price of a book. In the handler, use the length of the title times 3 as the price of the book and send it to the client in json format. Include the title in the json.

app.get('/books/:title',(req,res) =>{

    const {title} = req.params;

    fetch(`http://localhost/books/${title}`);

    const price = title.length \* 3;

    res.json({title: `${title}`, price: `${price}`})

     .then(json => res.send(json));

})

1. Write one or more regular expressions to find telephone numbers in a string.

let re = /(\d[\s-]?)?[\(\[\s-]{0,2}?\d{3}[\)\]\s-]{0,2}?\d{3}[\s-]?\d{4}/;

1. Write three different test strings to test the regular expression in 7.

The phone number is 601-123-4567.

6011234567 is my number.

Call (601) 123-4567 to reach me.

1. True or False. Expressjs can be used to make http requests to other servers.

True

1. Explain nodejs events with an example from your homework submissions.

Events are an action when something happens in code, mainly used to fire events.

ws3.on('close', () =>{

    messageEmitter.emit('ws3Close');

})