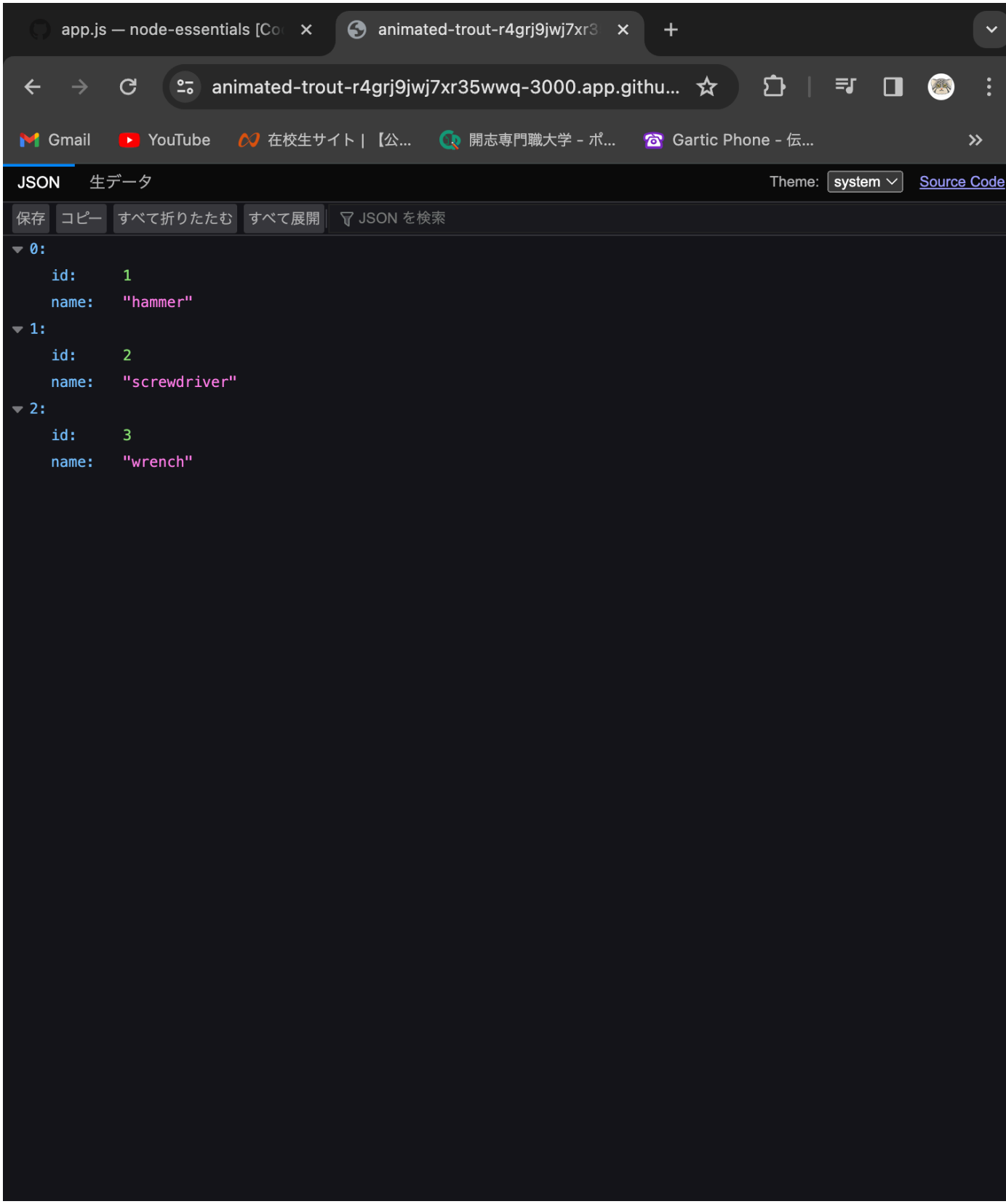
科目名	年度	レポート番号	クラス	学籍番号	名前
API 実習	2023	2	B	20122070	溝口将太

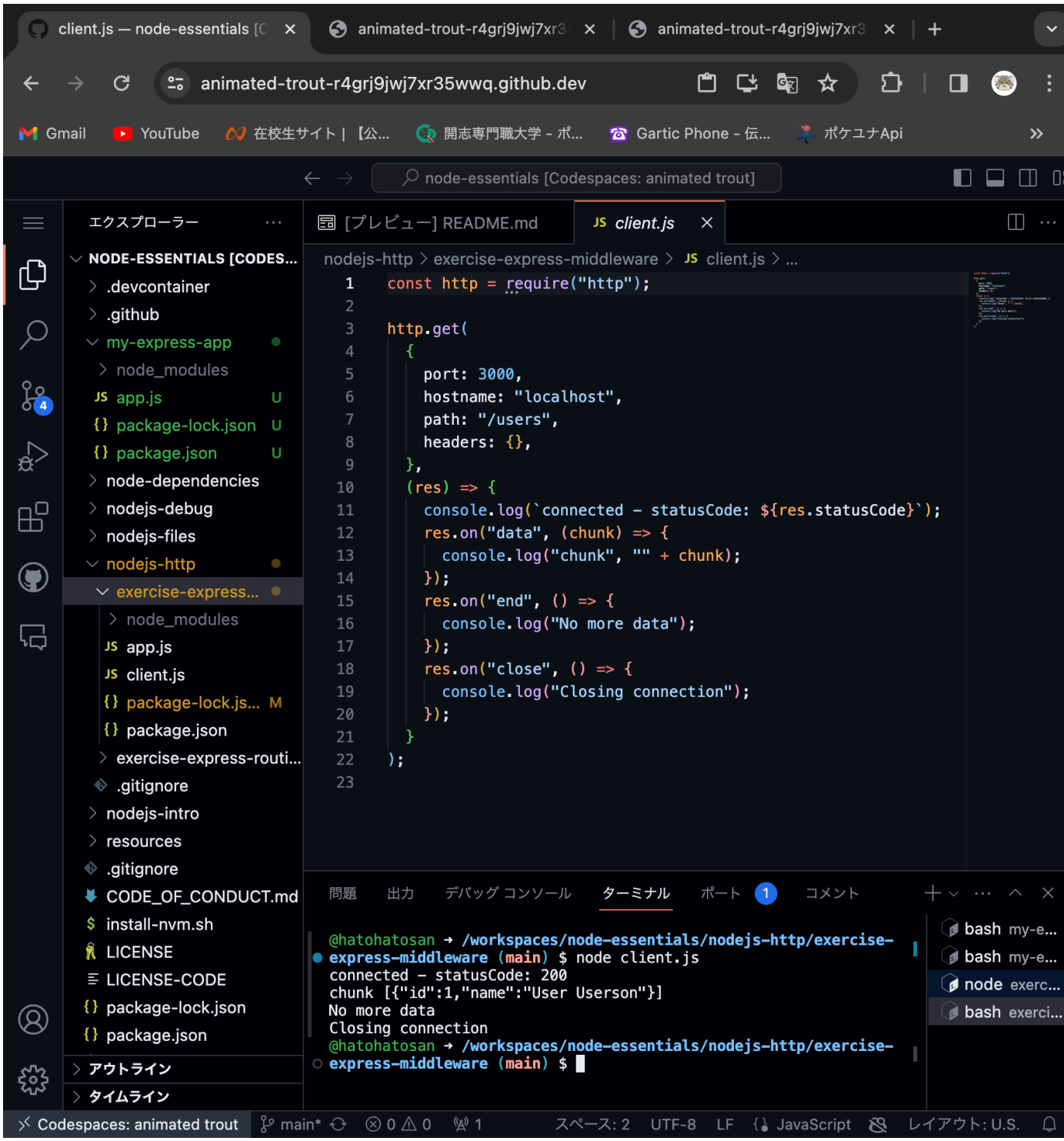
レポートは最大 5 ページ以内とします。ページ数や文字数よりも、わかりやすく書けているかどうかが、点数アップの分かれ目です。
Google スプレッドシートをもとに API を作成し、下記を行ってください。

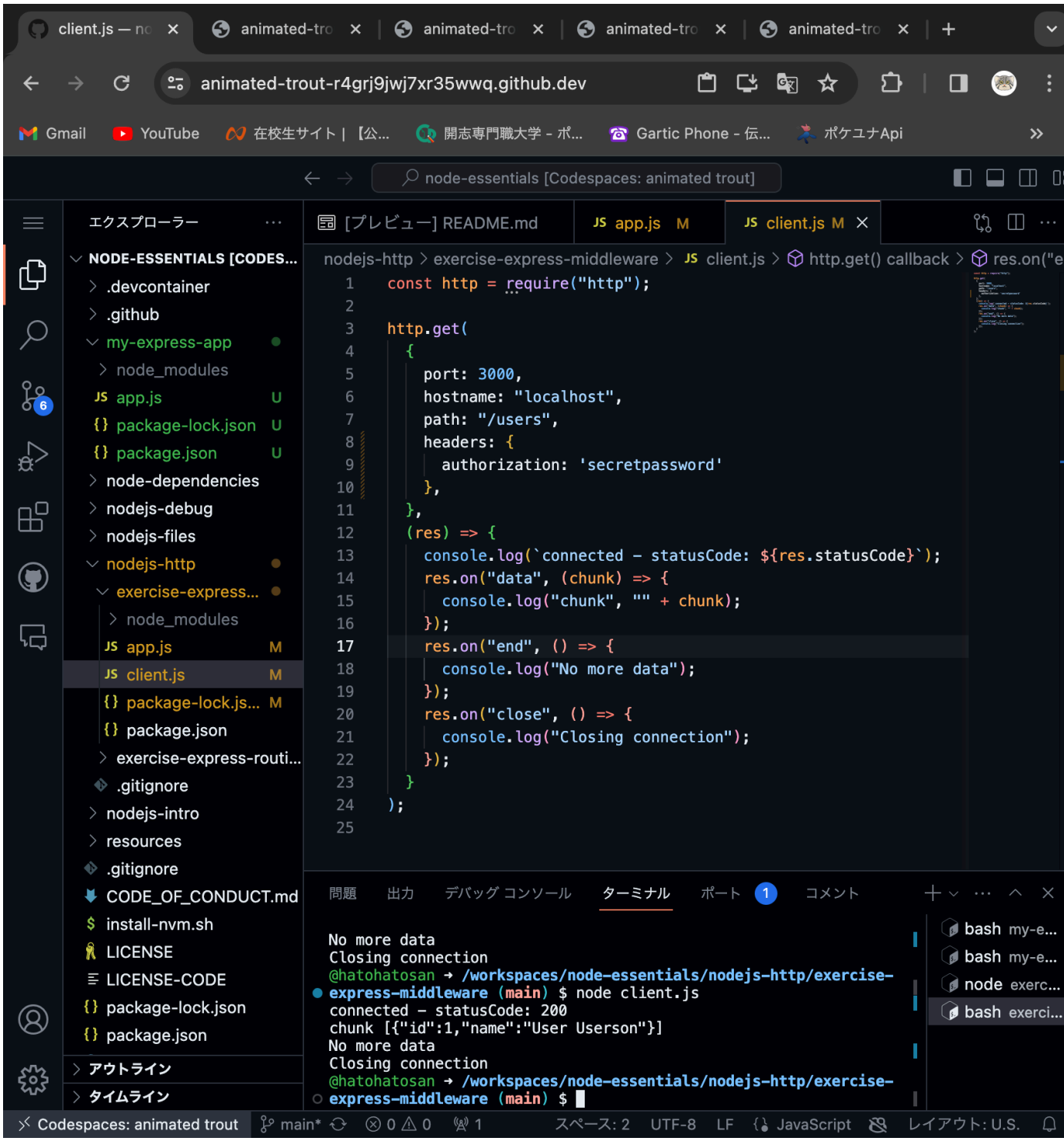
1. Google スプレッドシートをもとに作成した API について、以下を報告すること。
   (ア) 作成した API の概要
   おすすめのラーメン
   (イ) どんなことに役立つかの説明 << 読んだ人が具体的なイメージを思い浮かべるように。
   とにかくラーメンが好きな人に向けたもの
   (ウ) 作成した Google スプレッドシートの URL
   https://docs.google.com/spreadsheets/d/13b23bTyhDTkc1pa0ddvWLjuPL_lT7ctzLbFatzidjfo/edit#gid=0
   (エ) API エンドポイントの URL
   https://api.sssapi.app/Ym_SJNuqioMwOy_9-DTXu

2. Microsoft Learn の「Node.js と Express を使用して Web API を構築する」に取り組み、以下を報告する。
   (ア) 作成したものの説明
   Node.js 環境で動作する Web アプリケーションを開発するためのプロジェクト設定を提供しており、Express と呼ばれる Web アプリケーションフレームワークを使用するようになっている。
   (イ) 自分が理解したこと
   Package.json のファイル内で行われていること。
   Name・・・プロジェクトの名前を指定
   Version・・・プロジェクトのバージョンを指定
   Main・・・index.js をエントリーポイントとして設定
   Scripts・・・test というスクリプトを定義しており、実行時にエラーメッセージを表示
   License・・・ISC というライセンスを指定
   Dependencies・・・プロジェクトが依存しているパッケージを指定

   (ウ) どんなことに役立つか
   Express を使用して API を構築したことで、バックエンド開発やフルスタック開発などのスキルを上達させることができた。
   そして、将来 Web アプリケーションを開発する際に効率よく行うことができると思う。
   (エ) 作成した WebAPI が動いていることがわかる画面ショットを貼り付けること


app.js — node-essentials [Co
animated-trout-r4grj9jwj7xr3
animated-trout-r4grj9jwj7xr35wwq-3000.app.githu...
Gmail
YouTube
在校生サイト | 【公...
開志専門職大学 - ポ...
Gartic Phone - 伝...
JSON
生データ
Theme: system
Source Code
保存
コピー
すべて折りたたむ
すべて展開
JSON を検索
0:
id: 1
name: "hammer"
1:
id: 2
name: "screwdriver"
2:
id: 3
name: "wrench"


client.js — node-essentials [C
animated-trout-r4grj9jwj7xr3
animated-trout-r4grj9jwj7xr3
animated-trout-r4grj9jwj7xr35wwq.github.dev
Gmail
YouTube
在校生サイト | 【公...
開志専門職大学 - ポ...
Gartic Phone - 伝...
ポケユナApi
node-essentials [Codespaces: animated trout]
エクスプローラー
NODE-ESSENTIALS [CODES...
.devcontainer
.github
my-express-app
node_modules
app.js U
package-lock.json U
package.json U
node-dependencies
nodejs-debug
nodejs-files
nodejs-http
exercise-express...
node_modules
app.js
client.js
package-lock.js... M
package.json
exercise-express-routi...
.gitignore
nodejs-intro
resources
.gitignore
CODE_OF_CONDUCT.md
install-nvm.sh
LICENSE
LICENSE-CODE
package-lock.json
package.json
アウトライン
タイムライン
[プレビュー] README.md
client.js
nodejs-http > exercise-express-middleware > client.js > ...
1 const http = require("http");
2
3 http.get(
4 {
5 port: 3000,
6 hostname: "localhost",
7 path: "/users",
8 headers: {},
9 },
10 (res) => {
11 console.log(`connected - statusCode: ${res.statusCode}`);
12 res.on("data", (chunk) => {
13 console.log("chunk", "" + chunk);
14 });
15 res.on("end", () => {
16 console.log("No more data");
17 });
18 res.on("close", () => {
19 console.log("Closing connection");
20 });
21 }
22);
23
問題
出力
デバッグ コンソール
ターミナル
ポート 1
コメント
@hatohatosan → /workspaces/node-essentials/nodejs-http/exercise-express-middleware (main) $ node client.js
connected - statusCode: 200
chunk [{"id":1,"name":"User Userson"}]
No more data
Closing connection
@hatohatosan → /workspaces/node-essentials/nodejs-http/exercise-express-middleware (main) $
bash my-e...
bash my-e...
node exerc...
bash exerci...
Codespaces: animated trout
main*
0 0
1
スペース: 2
UTF-8
LF
JavaScript
レイアウト: U.S.



(オ)「知識チェック」の結果について、画面ショットを貼り付けること

1. Express を使用して Web アプリケーションを構築するために必要な手順は、次のどれですか? *

- ○ アプリをインスタンス化し、ルートを構成し、ミドルウェアを設定し、エラー ハンドラーを設定し、サーバーをリッスンする
- ◉ アプリをインスタンス化し、サーバーをリッスンする

✔ **正解です。 アプリを起動して実行するには、これらの手順だけが必要です。 ルートをいくつか構成することを強くお勧めします。**

- ○ アプリをインスタンス化し、ルートを構成し、サーバーをリッスンする
- ○ アプリをインスタンス化し、ルートを構成し、ミドルウェアを設定し、サーバーをリッスンする

2. Express アプリから JSON 応答を送信する方法として、推奨されるのは次のどれですか? *

- ◉ 応答オブジェクトで `json()` ヘルパー メソッドを呼び出す: `res.json({ content: '' })`

✔ **正解です。 JSON として応答を送信する方法はいろいろありますが、この方法が最も一般的であり、簡単に使用できます。**

- ○ `res.send({ content: '' })` を呼び出す
- ○ `res.send(JSON.stringify({ content: '' }))` を呼び出す
- ○ 次のいずれかの方法を使用する: `res.type('json')`、`res.type('application/json')`、`res.contentType('application/json')`、`res.format({ 'application/json': function() { res.send({}) } })`

3. JSON データを含む Post 要求を処理するように Express を設定するにはどうすればよいですか? *

- ○ `app.post(<route>, () =>{})` のように `post` メソッドを使用してルートを登録し、`req.body` オブジェクトから読み取る
- ○ 本文解析ミドルウェアを構成し、`app.post(<route>, () =>{})` のように `post` メソッドを使用してルートを登録して、`req.data` オブジェクトから読み取る
- ◉ 先頭で `app.use(bodyParser.json())` を呼び出し、`app.post(<route>, () =>{})` のように `post` メソッドを使用してルートを登録して、`req.body` オブジェクトから読み取る

✔ **正解。 この呼び出しでは、受信データを JSON として解釈するように `bodyParser` が構成されます。**

- ○ 先頭で `app.use(bodyParser.urlencoded({ extended: false }))` を呼び出し、`app.post(<route>, () =>{})` のように `post` メソッドを使用してルートを登録して、`req.body` オブジェクトから読み取る