

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	20122034	駒木根通元

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、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

・ゲーム『WARDAME』のプレイアブルキャラクターの入手方法と入手先や販売価格について一目でわかる。

no: フレームの登場順

name: フレームの名前

gender: フレームの性別

blueprint: フレームの本体設計図入手方法

quest: 関連クエスト

parts: フレームの各パーツ設計図入手方法

place: 関連する惑星

stage: 関連ステージ

mission type: 該当するステージのミッションタイプ

release date: フレームの登場日

primed date: プライムフレームの登場日

price(platinum): フレームの価格

(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。

- ・自分が入手したいキャラクターについてすぐに情報を得ることができる。
- ・自分のゲームの進行度でどのキャラクターを入手可能かわかる。
- ・ゲームの目標を立てやすい。このキャラクターを作りたいからこのステージまで今日は進めようとか。

(ウ) 作成した Google スプレッドシートの URL

<https://docs.google.com/spreadsheets/d/1l3xzuztJzYCh8V0U2TKVRnX6hh3TkDkVWnjVsJ4ImQk/edit#gid=0>

(エ) API エンドポイントの URL

<https://api.sssapi.app/CIYrEMY9egsA2dx0fVMIT>

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

- さまざまなルートに対応する Web アプリケーションを構築した。認証機能、承認機能がある。
- アクセス権を持つユーザーと持たないユーザーを区別することができる。
- アクセスしたユーザーのデータがわかる。

(イ) 自分が理解したこと

- URL の/から後ろの部分に対応する形に変化させると、表示されるページが対応したページに変化する。
- ユーザーがログインする仕組みが必要なときは、前処理、処理、後処理の三つのステップがある。

- すべてのルートに適用されるミドルウェア関数は一番上に、4つの引数を持つミドルウェア関数は最後に配置する必要がある。
- クライアントアプリケーションでユーザーデータの応答がわかる。
- `isAuthorized()` ミドルウェアが呼び出されると、特定の値を持つ `authorization` ヘッダーが検索される。このヘッダーに特定の値を指定しないとユーザーデータを応答しない。

(ウ) どんなことに役立つか

- アクセス権を持つユーザーとそうでないユーザーで表示したいページを分けたいときに役立つ。
- アクセスしたユーザーのデータを知りたいときに役立つ。

(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

```

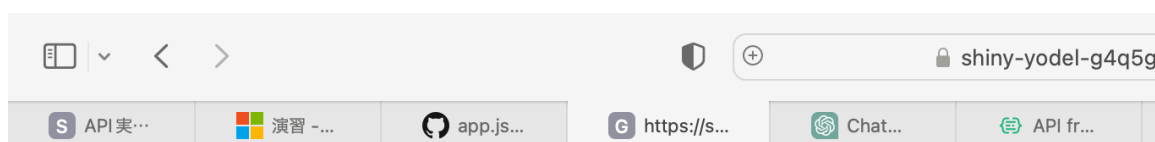
nodejs-http > exercise-express-middleware > JS client.js > headers > authorization
1  const http = require("http");
2
3  http.get(
4    {
5      port: 3000,
6      hostname: "localhost",
7      path: "/users",
8      headers: {
9        authorization: 'secretpassword'
10     },
11   },
12   (res) => {
13     console.log(`connected - statusCode: ${res.statusCode}`);
14     res.on("data", (chunk) => {
15       console.log("chunk", "" + chunk);
16     });
17     res.on("end", () => {
18       console.log("No more data");
19     });
20   });

```

```

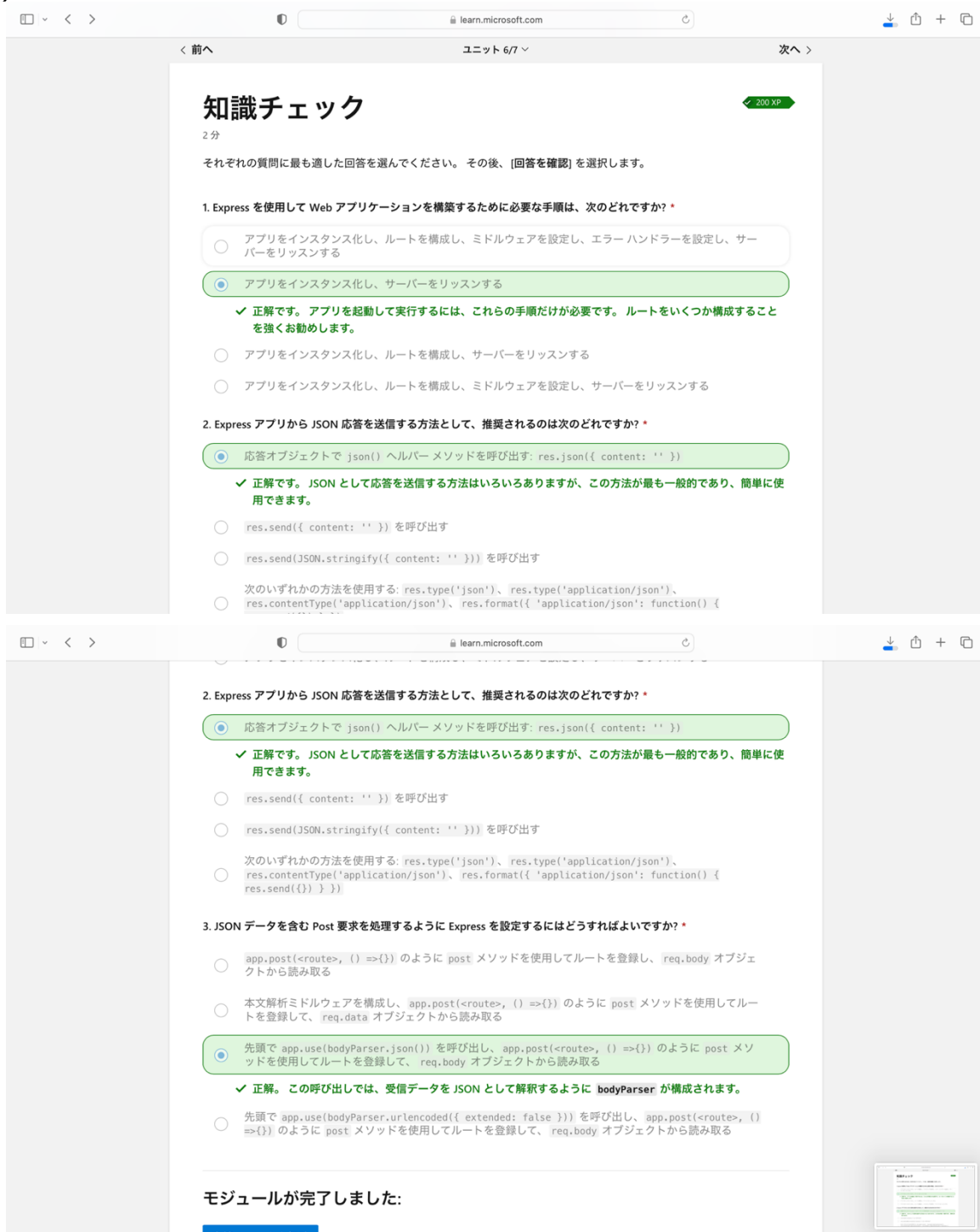
@Komasan-2525 + /workspaces/node-essentials/nodejs-http/exercise-express-middleware (main) $ node client.js
connected - statusCode: 200
chunk [{"id":1,"name":"User Useron"}]
No more data
Closing connection
@Komasan-2525 + /workspaces/node-essentials/nodejs-http/exercise-express-middleware (main) $ node client.js
connected - statusCode: 401
chunk Not permitted
No more data
Closing connection
@Komasan-2525 + /workspaces/node-essentials/nodejs-http/exercise-express-middleware (main) $ node client.js
connected - statusCode: 200
chunk [{"id":1,"name":"User Useron"}]
No more data
Closing connection
@Komasan-2525 + /workspaces/node-essentials/nodejs-http/exercise-express-middleware (main) $

```



```
[{"id":1,"name":"hammer"}, {"id":2,"name":"screwdriver"}, {"id":3,"name":"wrench"}]
```

(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



知識チェック 200 XP

2分

それぞれの質問に最も適した回答を選んでください。その後、[回答を確認] を選択します。

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、ルートを作成し、ミドルウェアを設定し、エラー ハンドラーを設定し、サーバーをリスンする

☒ アプリをインスタンス化し、サーバーをリスンする

✓ 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、ルートを作成し、サーバーをリスンする

☐ アプリをインスタンス化し、ルートを作成し、ミドルウェアを設定し、サーバーをリスンする

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くつもありますが、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({}) } })`

☐ `res.send({})` を呼び出す

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作成し、`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

モジュールが完了しました:

[実装のブロックを解除](#)