11. Router でルーティング

Router とは

`Router` は `Express` のルーティングを簡単に作成できる機能です。今までは、ルーティングは、 `express()` で生成したオブジェクトの `HTTP` メソッドでアクセスしました。

```
const app = express();
app.get("/", (req, res) => {
    //処理
});

app.get("/profile", (req, res) => {
    //処理
});

app.post('/auth', (req, res) => {
    //処理
})
```

メインファイルにルーティングを記述すると、ルーティングが増えるためプログラムが見づらくなります。そこで `Router` を利用すると、別ファイルで管理することができます。

プロジェクトの作成

ファイル構成

```
├── .env
├── node_modules
├── package-lock.json
├── package.json
├── routes.js
└── server.js
```

Router によるルーティング

routes.js を作成し、ルーティングを記述します。内容は前のサンプルのルーティングと同じです。

```
const express = require('express')
const router = express.Router()

router.get('/', (req, res) => {
    res.send('Hello Express Router!!')
})

router.get("/profile", (req, res) => {
    res.send("This is Profile page.");
})

router.post('/auth', (req, res) => {
    const login_name = req.body.login_name;
    const password = req.body.password;

    let message = 'ログインできませんでした';
    if (login_name == process.env.LOGIN_NAME && password == process.env.PASSWORD) {
        message = 'ログインしました';
    }
    res.send(message);
})

module.exports = router
```

外部モジュールの読み込み

外部モジュールとして利用できるように、 module.exports を利用します。モジュール名は、express.Router() で作成した名前に合わせます。

```
module.exports = router
```

メインプログラム

server.js で routes.js を読み込み、ミドルウェアで処理します。もともと記述したルーティング処理は削除します。

```
const express = require('express')
// routes.js ファイルの読み込み
const routes = require('./routes')
```

```
require('dotenv').config()
const host = process.env.HOST
const port = process.env.PORT

const app = express()

app.use(express.urlencoded({ extended: true }))
app.use(express.static(__dirname + '/public'))

// Router をミドルウェアで処理
app.use(routes)

app.listen(port, host, () => {
    console.log(`Server listen: http://${host}:${port}`)
})
```

ルーティング確認

サーバを起動して、各 URL がルーティングできるか確認します。

```
% node server
```

ホーム

http://localhost:3000/

Hello Express Router!!

プロフィール

http://localhost:3000/profile

This is Profile page.

ログインページからポスト

http://localhost:3000/login.html

ログインしました

パラメータの利用

ルーティングでパラメータを利用するには、HTTPメソッド関数のパスを、パス:パラメータ名で記述します。

```
router.HTTPメソッド("パス:パラメータ名", (req, res) => {

}
```

パラメータの取得は `req.params` でパラメータ名を指定します。

```
値 = req.params.パラメータ名
```

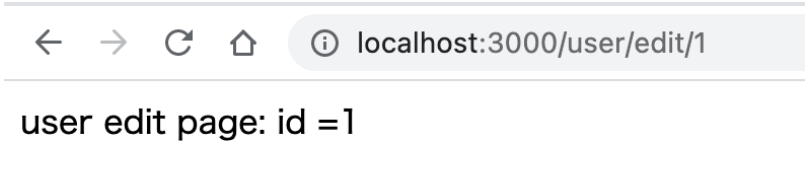
パラメータ id で処理

`routes.js` に `/user/edit/:id` ルーティングを追加します。:id の部分は任意の値で処理できます。

```
router.get("/user/edit/:id", (req, res) => {
    const id = req.params.id
    const message = 'ユーザID: ' + id
    res.send(message);
})
```

ルーティングの確認

サーバを再起動します。 http://localhost:3000/user/edit/1 にアクセスして確認してみましょう。※数値部分は任意



演習

問題1

`item.js` を作成して商品データを用意します。 `/item/:id` でルーティング（:id は任意の数値）を追加して、指定した ID で商品情報を表示してみましょう。

item.js

```
exports.values = {
    1: { name: 'コーヒー', price: 150 },
    2: { name: '紅茶', price: 180 },
    3: { name: 'ほうじ茶', price: 100 },
}
```