Yelp camp notes

Notes:

You can download the code and use “npm install” to install all the dependencies.

Start the work:

npm init -y

This will create the npm file to record dependencies.

Set up view engine, here we use EJS. EJS is extended HTML.

Set up absolute path for views folder.

Set up routers:

## 增添新加campground 的页面：

Remember the order of router matters!

*// page to add new campground*

app.get('/campgrounds/new', (req, res) => {

res.render('campgrounds/new')

})

*//Show 1 campground:*

app.get('/campgrounds/:id', async (req, res) => {

const campID = req.params.id;

const campground = *await* Campground.findById(campID);

res.render('campgrounds/show', { campID, campground })

})

比如说你不可以把 /new 放在 /id 后面，因为这样会让server 以为new是一个id。

## 这是用来把数据写入DB的代码：

*// this line of coder allow you to see the req.body.*

app.use(express.urlencoded({ extended: true }))

app.post('/campgrounds', async (req, res) => {

const campground = new Campground(req.body.campground);

*await* campground.save();

res.redirect(`/campgrounds/${campground.\_id}`)

*// res.send(req.body)*

})

注意精彩的地方是req.body.campground

## Update a campground:

Use app.put to update.

<https://stackoverflow.com/questions/630453/what-is-the-difference-between-post-and-put-in-http>

On the client side, the “campground/edit” page can not have HTML method like PUT or DELETE, hence you should install the npm package “method override”

### override using a query value

ref: <https://www.npmjs.com/package/method-override>

## about Bootstrap

1. Follow the doc: add the CSS Popper and JS URL.
2. Add nav bar

## About bootstrap grid

There must be a **container** class in order to use the Grid.

In Bootstrap, one row has 12 units.

<https://getbootstrap.com/docs/5.2/layout/grid/>

You can specify how many units each column should take.

## About responsive bootstrap:

*<!-- responsive display -->*

<h2 class="diplay-2">

<div class="row">

<div class="col-md-6">

this line of context

this line of context should change

</div>

<div class="col-md-6">

this line of context should change as well

</div>

</div>

</h2>

以上代码就是当内容小于medium的时候，就不再平分一行的空间。

# Error handler

Aysnc error handler：

用catchAsync的module 来替换所有的try and catch。

# 440. Joi Schema validation (server side validation)

app.post('/campgrounds', catchAsync(async (*req*, *res*, *next*) => {

*// 1. this try and catch error handler should pass the error to the error handler at the bottom of the code.*

*// 2. The try catch structure has been replaced by the catchAsync().*

*// if (!req.body.campground) {*

*// throw new ExpressError("Invalid Campground data", "400 ")*

*// }*

const campgroundSchema = Joi.object({

campground: Joi.object({

title: Joi.string().required(),

price: Joi.number().required().min(0),

location: Joi.string().required,

image: Joi.string().required(),

description: Joi.string().required()

}).required()

})

const { error } = campgroundSchema.validate(*req*.body);

*if* (error) {

const msg = error.details.map(*el* => *el*.message).join(',')

*throw* new ExpressError(msg, 400)

}

console.log(result);

const campground = new Campground(*req*.body.campground);

*await* campground.save();

*res*.redirect(`/campgrounds/${campground.\_id}`)

*// res.send(req.body)*

}))

If(error) 逻辑里有一个复杂的数据操作：Error.details 是一个array，so we need to map() it, and turn it into a single string and join them.