UDACITY

‹ Return to Classroom

Image Processing API

REVIEW

CODE REVIEW 9

HISTORY

▸ src/file.ts 4

▾ src/processing.ts 2

```
import sharp from 'sharp';

// query segments
interface sharpResizeParams {
 source: string;
 target: string;
 width: number;
 height: number;
}
const imageProcess = async (params: sharpResizeParams): Promise <null | string> => {
  try {
    await sharp(params.source)
```

AWESOME

async/await syntax is clean and easy to understand when working with asynchronous programming. Gooc

```
      .resize(params.width, params.height)
      .toFormat('jpeg')
      .toFile(params.target);
    return null;
```

```
  } catch {
    return 'Image NOT Processed.';
```

AWESOME

I loved that you provided custom error messages to the user.

```
  }
};

export default imageProcess;
```

▶ README.md 1

▶ src/routes/api/images.ts 1

▶ src/routes/index.ts 1

▶ dist/file.js

▶ dist/index.js

▶ dist/processing.js

▶ dist/routes/api/images.js

▶ dist/routes/index.js

▶ dist/tests/fileSpec.js

▶ dist/tests/helpers/reporter.js

▶ dist/tests/indexSpec.js

▶ dist/tests/routes/api/imagesSpec.js

▶ dist/tests/routes/indexSpec.js

▶ src/index.ts

▶ src/tests/fileSpec.ts

▸ **src/tests/helpers/reporter.ts**

▸ **src/tests/indexSpec.ts**

▸ **src/tests/routes/api/imagesSpec.ts**

▸ **src/tests/routes/indexSpec.ts**

RETURN TO PATH