

picture-store implementation

assets
controllers/about.js
controllers/accounts.js
controllers/dashboard.js
controllers/start.js
models/json-store.js
models/picture-store.js
models/picture-store.json
models/user-store.js
models/user-store.json
utils/logger.js
views/layouts/main.hbs
views/partials/card.hbs
views/partials/mainpanel.hbs
views/partials/menu.hbs
views/partials/welcomemenu.hbs
views/about.hbs
views/dashboard.hbs
views/login.hbs
views/signup.hbs
views/start.hbs
⌕ .env
.gitignore
.jscsrc
README.md
package.json
routes.js
server.js
tempimage

ImageStore

Dashboard

About

Logout

Welcome homer simpson

Your Pictures:



Delete All

Picture Upload

First Name

Choose file No file chosen

Title

Title

Upload

 assets

controllers/about.js

controllers/accounts.js

controllers/dashboard.js

controllers/start.js

models/json-store.js

models/picture-store.js

models/picture-store.json

models/user-store.js

models/user-store.json

utils/logger.js

views/layouts/main.hbs

views/partials/card.hbs

views/partials/mainpanel.hbs

views/partials/menu.hbs

views/partials/welcomemenu.hbs

views/about.hbs

views/dashboard.hbs

views/login.hbs

views/signup.hbs

views/start.hbs

 .env

.gitignore

.jscsrc

README.md

package.json

routes.js

server.js

tempimage

Dashboard Controller

Wrapper for
cloudinary Service

Display Individual
Image

Dashboard View

Uploading Images

Upload Picture Form

File Name

Choose File

No file chosen

Title

Title

Upload

```
<form action="/dashboard/uploadpicture" method="post" enctype="multipart/form-data">
  <div class="two fields">
    <div class="field">
      <label>File Name</label>
      <input type="file" name="picture" required> </input>
    </div>
    <div class="field">
      <label>Title</label>
      <input placeholder="Title" type="text" name="title">
    </div>
  </div>
  <button class="ui mini blue submit button"> Upload </button>
</form>
```

Upload Picture Form + Route + Action

```
<form action="/dashboard/uploadpicture" method="post" enctype="multipart/form-data">
  <div class="two fields">
    <div class="field">
      <label>File Name</label>
      <input type="file" name="picture" required> </input>
    </div>
    <div class="field">
      <label>Title</label>
      <input placeholder="Title" type="text" name="title">
    </div>
  </div>
  <button class="ui mini blue submit button"> Upload </button>
</form>
```

```
router.post('/dashboard/uploadpicture', dashboard.uploadPicture);
```

```
const dashboard = {

  ...

  uploadPicture(request, response) {
    const loggedInUser = accounts.getCurrentUser(request);
    pictureStore.addPicture(loggedInUser.id, request.body.title, request.files.picture, function () {
      response.redirect('/dashboard');
    });
  },

  ...
};
```

picture-store module

Local copy of image
urls

Get all image urls
for a give user

Upload picture to
cloudinary

Delete specific
picture

Delete all pictures
for a given user

```
const cloudinary = require('cloudinary');
const env = require('../env.json');
console.log(env.cloudinary);

cloudinary.config(env.cloudinary);

const pictureStore = {

  store: new JsonStore('./models/picture-store.json', { pictures: [] }),
  collection: 'pictures',

  getAlbum(userid) {
    ...
  },

  addPicture(userId, title, imageFile, response) {
    ...
  },

  deletePicture(userId, image) {
    ...
  },

  deleteAllPictures(userId) {
    ...
  },
};
```

picture-store.json

```
{
  "pictures": [
    {
      "userid": "3ad52697-6d98-4d80-8273-084de55a86c0",
      "photos": [
        {
          "img": "http://res.cloudinary.com/edel020/image/upload/v1490596702/znivz9tsjbmqo2twf6ao.gif",
          "title": "Bart"
        },
        {
          "img": "http://res.cloudinary.com/edel020/image/upload/v1490596801/l1osts64u08outotg8kh.gif",
          "title": "Lisa"
        }
      ]
    }
  ]
}
```

For each user, store the url + title of
each picture uploaded

Displaying Images

dashboard.index

Read image-store.json
for users 'album'

```
const dashboard = {  
  index(request, response) {  
    logger.info('dashboard rendering');  
    const loggedInUser = accounts.getCurrentUser(request);  
    const viewData = {  
      title: 'Pictures',  
      user: loggedInUser,  
      album: pictureStore.getAlbum(loggedInUser.id),  
    };  
    response.render('dashboard', viewData);  
  },  
  ...  
}
```

album

```
{  
  "userid": "3ad52697-6d98-4d80-8273-084de55a86c0",  
  "photos": [  
    {  
      "img": "http://res.cloudinary.com.....6ao.gif",  
      "title": "Bart"  
    },  
    {  
      "img": "http://res.cloudinary.com....otg8kh.gif",  
      "title": "Lisa"  
    }  
  ]  
}
```

dashboard.hbs

```
<section class="ui raised segment">
  <h3 class="ui header">
    Your Pictures:
  </h3>
  <div class="ui cards">
    {{#each album.photos}}
      {{> card }}
    {{/each}}
  </div>
</section>
```

```
"userid": "3ad52697-6d98-4d80-8273-084de55a86c0",
"photos": [
  {
    "img": "http://res.cloudinary.com.....6ao.gif",
    "title": "Bart"
  },
  {
    "img": "http://res.cloudinary.com....otg8kh.gif",
    "title": "Lisa"
  }
]
```

Your Pictures:



card.hbs

```
"userid": "3ad52697-6d98-4d80-8273-084de55a86c0",
"photos": [
  {
    "img": "http://res.cloudinary.com.....6ao.gif",
    "title": "Bart"
  },
  {
    "img": "http://res.cloudinary.com....otg8kh.gif",
    "title": "Lisa"
  }
]
```

```
<div class="ui card">
  <div class="ui small center aligned image">
    
  </div>
  <div class="content">
    <p class="center aligned header">{{title}}</p>
    <div class="center aligned meta">
      <a href="/dashboard/deletepicture?img={{img}}">
        <i class="red icon trash"></i>
      </a>
    </div>
  </div>
</div>
```



delete picture



```
<a href="/dashboard/deletepicture?img={{img}}">
  <i class="red icon trash"></i>
</a>
```

```
router.get('/dashboard/deletepicture', dashboard.deletePicture);
```

```
const dashboard = {
  ...

  deletePicture(request, response) {
    const loggedInUser = accounts.getCurrentUser(request);
    pictureStore.deletePicture(loggedInUser.id, request.query.img);
    response.redirect('/dashboard');
  },
};
```

Callbacks in picture-store

```
const dashboard = {  
  
  ...  
  
  uploadPicture(request, response) {  
    const loggedInUser = accounts.getCurrentUser(request);  
    pictureStore.addPicture(loggedInUser.id, request.body.title, request.files.picture, function () {  
      response.redirect('/dashboard');  
    });  
  },  
  
  ...  
};
```

```
uploadPicture(request, response) {  
  
    const loggedInUser = accounts.getCurrentUser(request);  
  
    pictureStore.addPicture(loggedInUser.id,  
  
        request.body.title,  
  
        request.files.picture,  
  
        function () {  
            response.redirect('/dashboard');  
        }  
  
        );  
  
},
```


First Name

Choose file

No file chosen

Title

Title

Upload

```
uploadPicture(request, response) {  
  
    const loggedInUser = accounts.getCurrentUser(request);  
    pictureStore.addPicture( loggedInUser.id,  
        request.body.title,  
        request.files.picture,  
  
        function () {  
            response.redirect('/dashboard');  
        }  
  
        );  
},
```

Id of
logged in
user

First Name

Choose file

No file chosen

Title

Title

Upload

```
uploadPicture(request, response) {  
  
  const loggedInUser = accounts.getCurrentUser(request);  
  
  pictureStore.addPicture(loggedInUser.id,  
    request.body.title,  
    request.files.picture,  
  
    function () {  
      response.redirect('/dashboard');  
    }  
  
  );  
},
```

Title
entered in
form

First Name

Choose file

No file chosen

Title

Title

Upload

```
uploadPicture(request, response) {  
  
  const loggedInUser = accounts.getCurrentUser(request);  
  
  pictureStore.addPicture(loggedInUser.id,  
  
    request.body.title,  
    request.files.picture,  
  
    function () {  
      response.redirect('/dashboard');  
    }  
  
  );  
},
```

The
picture
itself
(binary)

First Name

Choose file

No file chosen

Title

Title

Upload

```
uploadPicture(request, response) {  
  
  const loggedInUser = accounts.getCurrentUser(request);  
  
  pictureStore.addPicture(loggedInUser.id,  
    request.body.title,  
    request.files.picture,  
  
    function () {  
      response.redirect('/dashboard');  
    }  
  
  );  
},
```

A 'Callback' function - involved when the picture has been successfully uploaded

```
function () {  
  response.redirect('/dashboard');  
}
```



Callback triggers refresh
of dashboard



This will contain newly
uploaded images

[ImageStore](#)[Dashboard](#)[About](#)[Logout](#)

Welcome homer simpson

Your Pictures:





[Delete All](#)

Picture Upload

First Name

No file chosen

Title

[Upload](#)