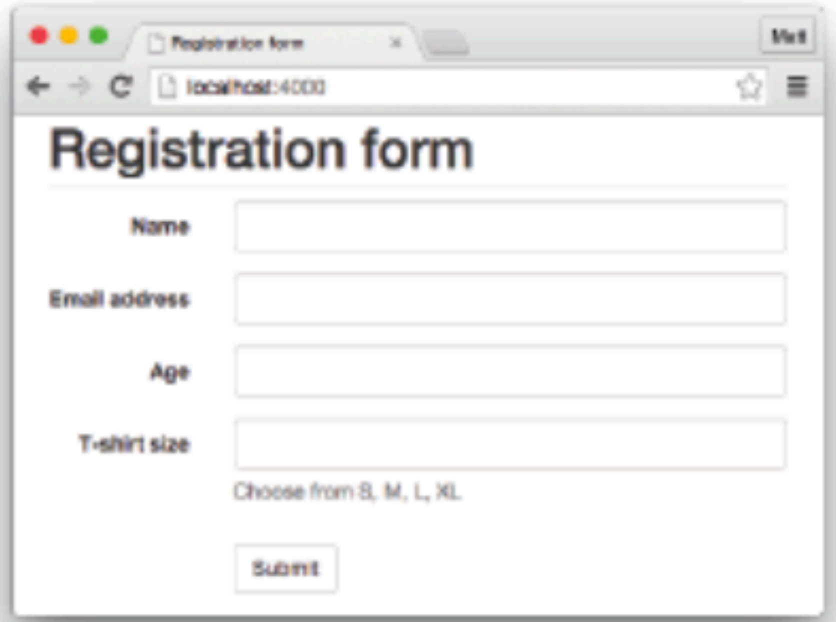
Joi + Hapi Validation

Agenda

- Error Handling UI Strategies
- Joi Installation
- Semantic Versioning of npm modules
- Joi in Donation

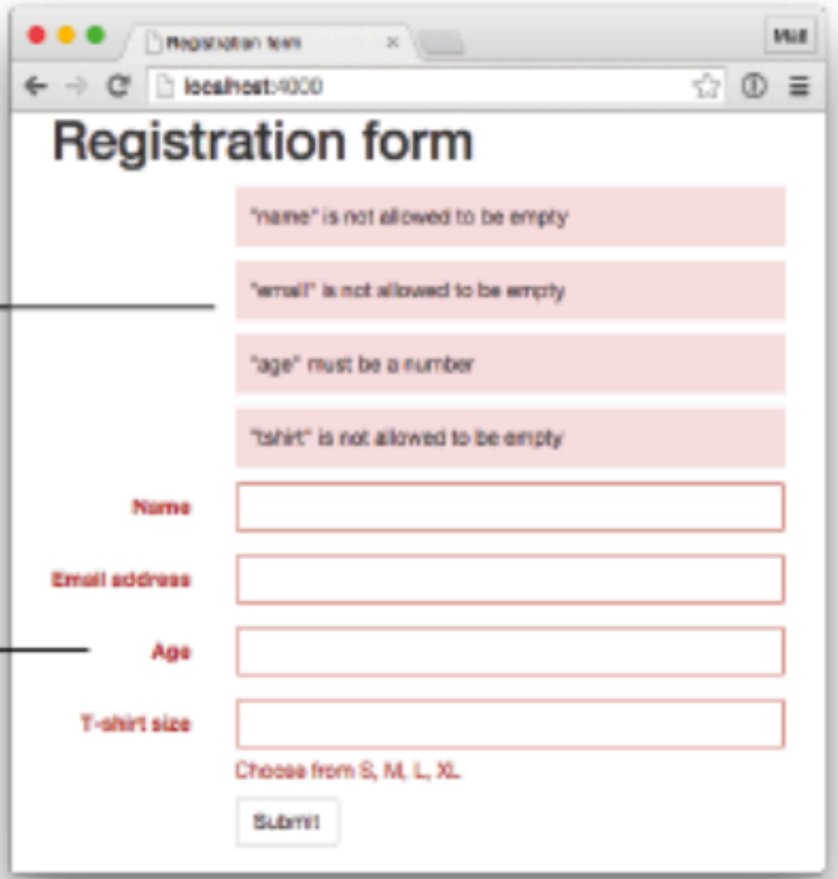
The basic form


localhost:4000

Registration form

Name	
Email address	
Age	
T-shirt size	
	Choose from S, M, L, XL
	Submit

The form with errors


localhost:4000

Registration form

"name" is not allowed to be empty

"email" is not allowed to be empty

"age" must be a number

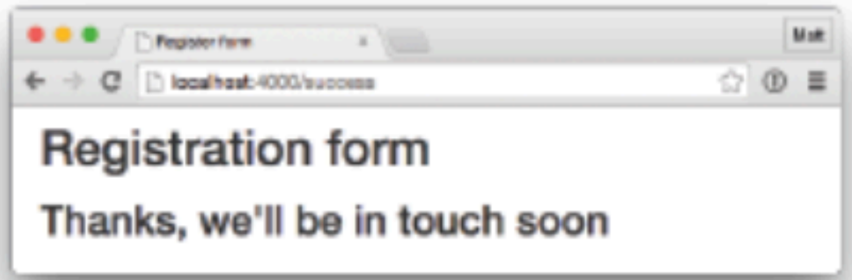
"tshirt" is not allowed to be empty

Name	
Email address	
Age	
T-shirt size	
	Choose from S, M, L, XL
	Submit

All errors are listed at the top

Fields with errors are highlighted in red

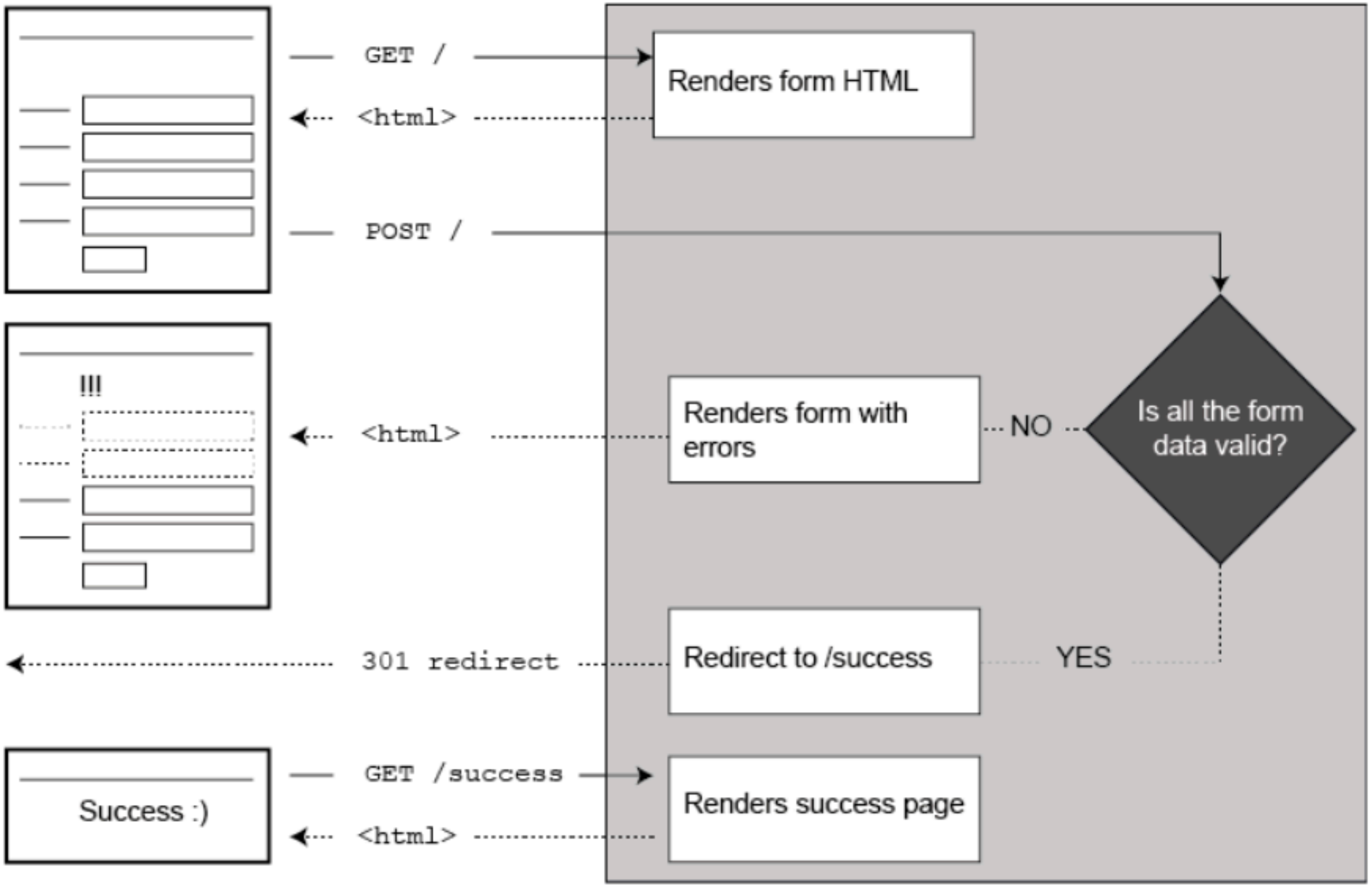
The success page


localhost:4000/success

Registration form

Thanks, we'll be in touch soon

- Potential Error Reporting Strategy


GET /
<html>
Renders form HTML
POST /
!!!
<html>
Renders form with errors
NO
Is all the form data valid?
YES
301 redirect
Redirect to /success
Success :)
GET /success
<html>
Renders success page

Install Joi

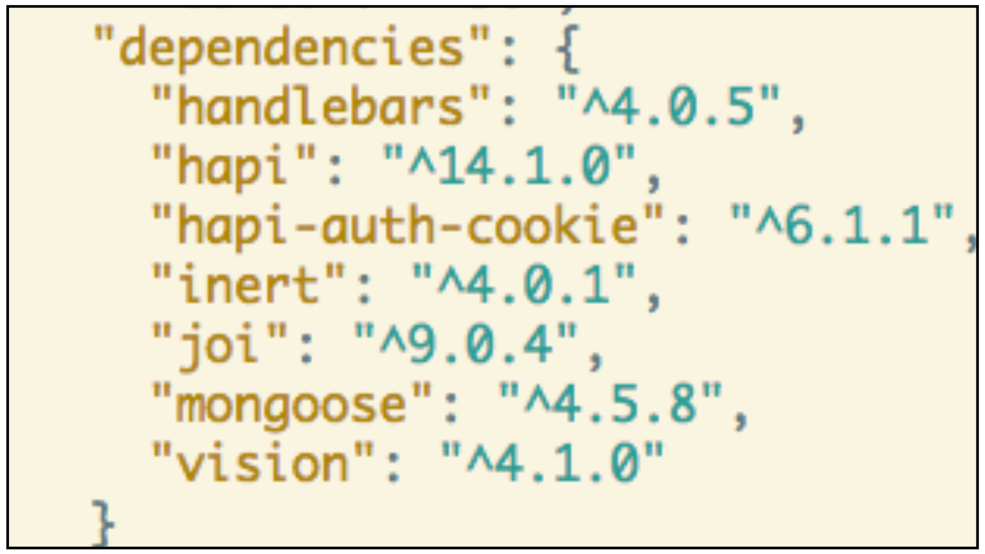
```
npm install joi -save
```

package.json

```
{
  "name": "donation-web",
  "version": "1.0.0",
  "description": "an application to host donations for candidates",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "handlebars": "^4.0.5",
    "hapi": "^14.1.0",
    "hapi-auth-cookie": "^6.1.1",
    "inert": "^4.0.1",
    "joi": "^9.0.4",
    "mongoose": "^4.5.8",
    "vision": "^4.1.0"
  }
}
```

Semantic Versioning of Packages

- Semantic versioning is a standard node projects use to communicate what kinds of changes are in a new release.
- Communicate what kinds of changes are in a release because sometimes those changes will break the code that depends on the package.

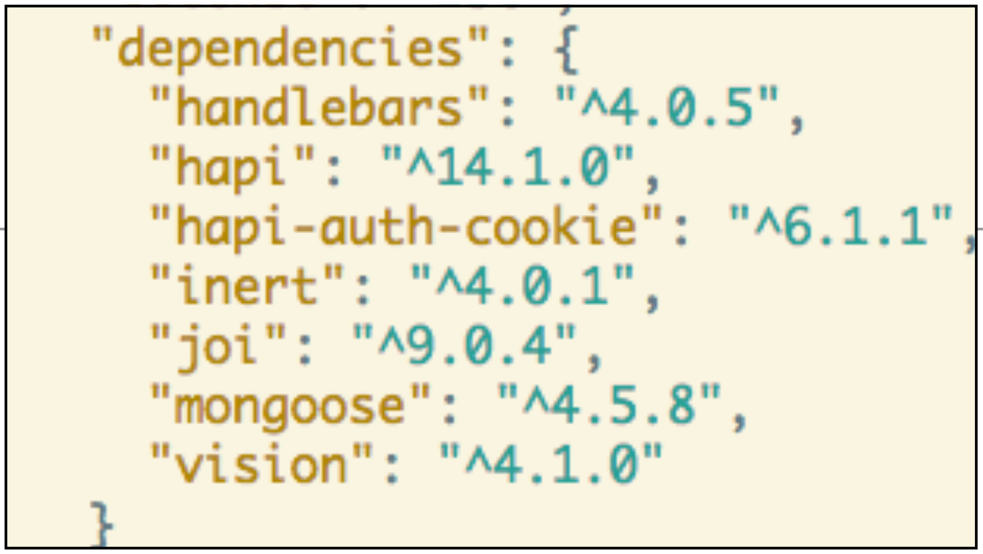


```
"dependencies": {
  "handlebars": "^4.0.5",
  "hapi": "^14.1.0",
  "hapi-auth-cookie": "^6.1.1",
  "inert": "^4.0.1",
  "joi": "^9.0.4",
  "mongoose": "^4.5.8",
  "vision": "^4.1.0"
}
```

- 3-component system in the format of x.y.z where:
  - x stands for a major version
  - y stands for a minor version
  - z stands for a patch
- Major.Minor.Patch.

^ Symbol

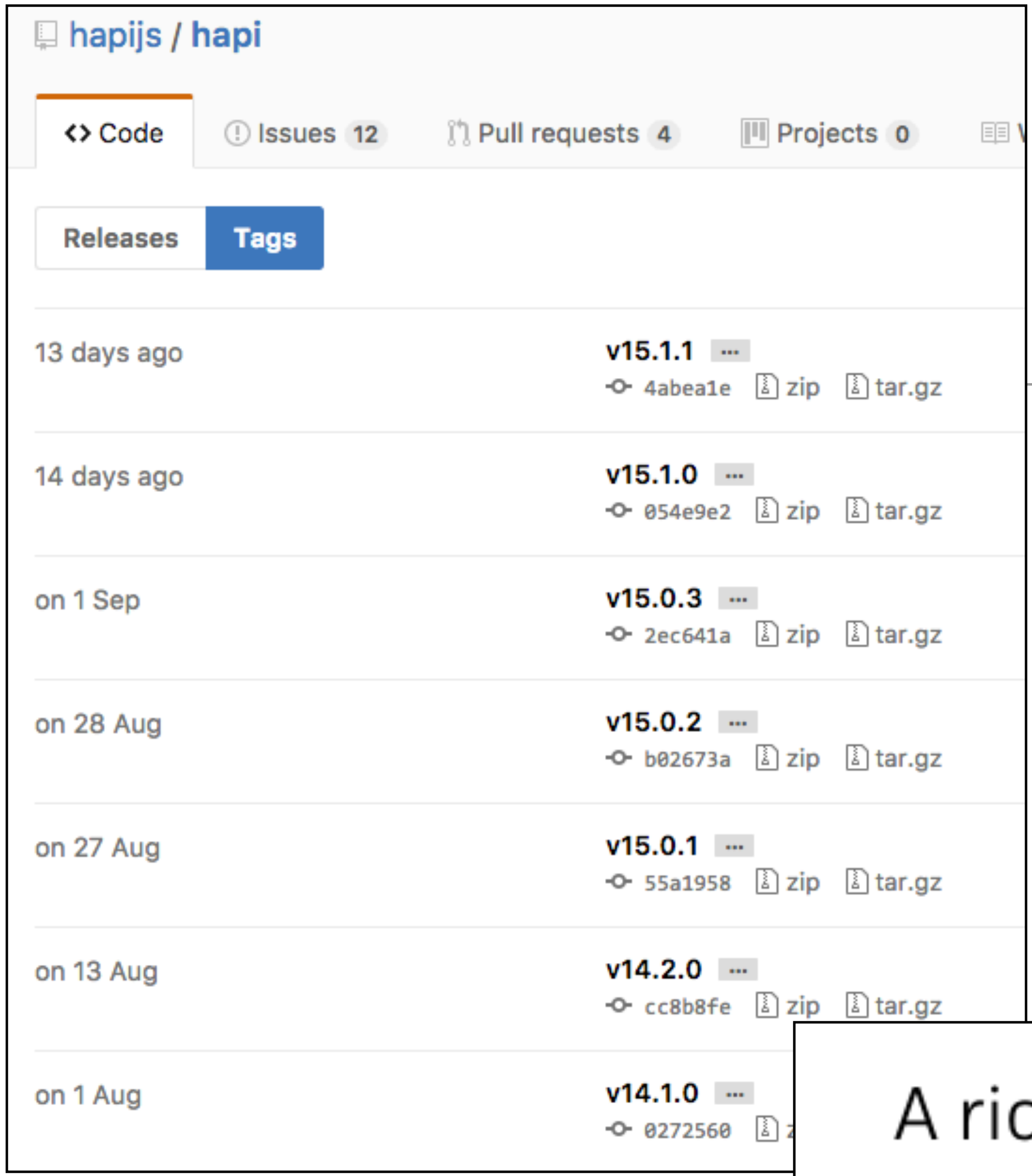
- The caret ^ range specifier permits automatic upgrades to minor version increments of a package
- So if 'npm install' is invoked, the actual version downloaded and installed may be more recent that the one enumerated in package.json
- For caret ranges, only major version must match. Any minor or patch version greater than or equal to the minimum is valid.



```
"dependencies": {
  "handlebars": "^4.0.5",
  "hapi": "^14.1.0",
  "hapi-auth-cookie": "^6.1.1",
  "inert": "^4.0.1",
  "joi": "^9.0.4",
  "mongoose": "^4.5.8",
  "vision": "^4.1.0"
}
```

- Example
  - ^1.2.3 permits versions from 1.2.3 all the way up to, but not including, the next major version, 2.0.0.

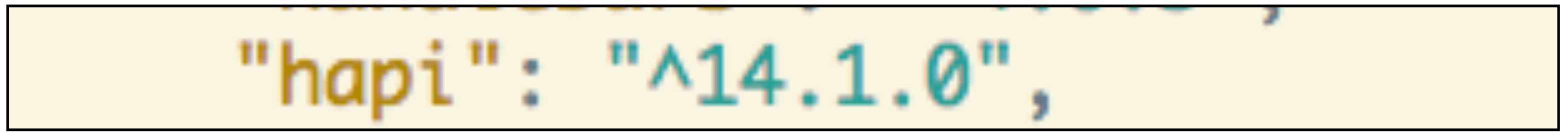
hapi.js versions

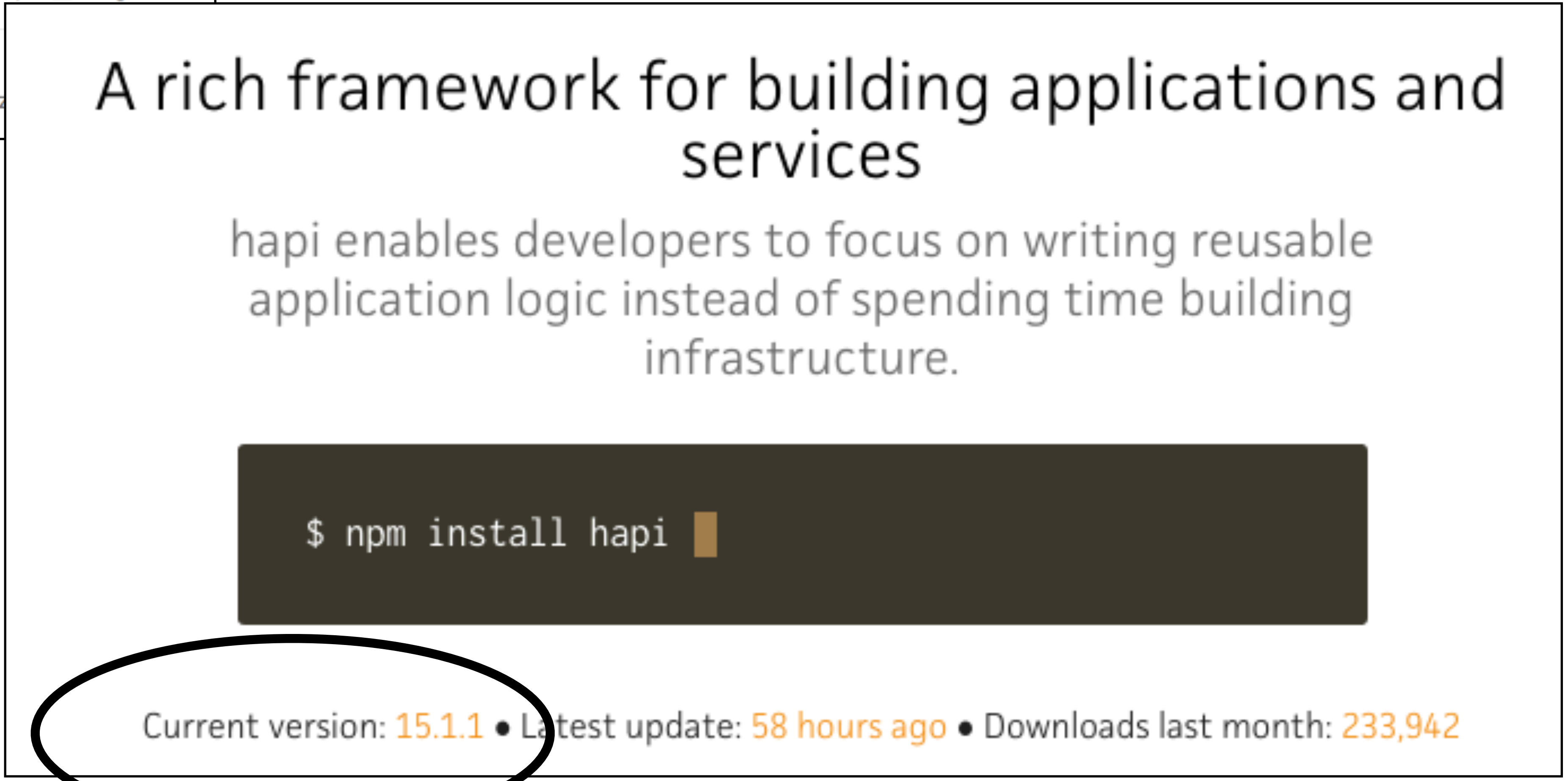

hapijs / hapi

Code | Issues 12 | Pull requests 4 | Projects 0

Releases | Tags

13 days ago	v15.1.1 · 4abea1e · zip · tar.gz
14 days ago	v15.1.0 · 054e9e2 · zip · tar.gz
on 1 Sep	v15.0.3 · 2ec641a · zip · tar.gz
on 28 Aug	v15.0.2 · b02673a · zip · tar.gz
on 27 Aug	v15.0.1 · 55a1958 · zip · tar.gz
on 13 Aug	v14.2.0 · cc8b8fe · zip · tar.gz
on 1 Aug	v14.1.0 · 0272560


```
"hapi": "^14.1.0",
```

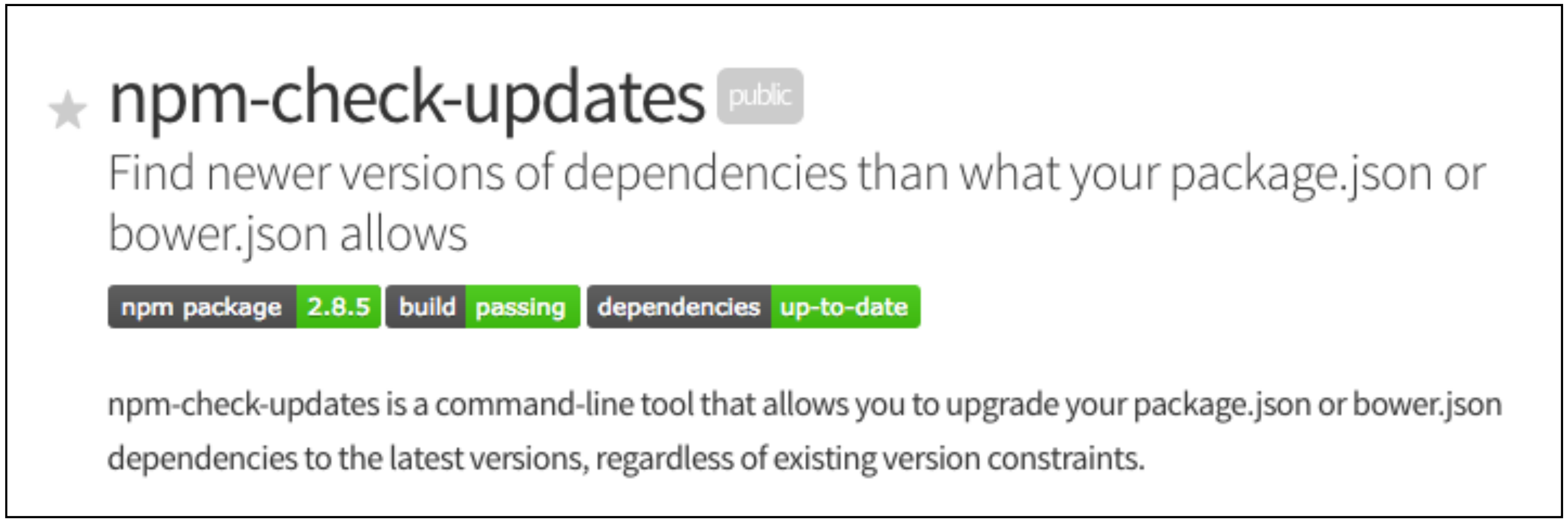

A rich framework for building applications and services

hapi enables developers to focus on writing reusable application logic instead of spending time building infrastructure.

```
$ npm install hapi
```

Current version: 15.1.1 • Latest update: 58 hours ago • Downloads last month: 233,942

Updating all Dependencies


npm-check-updates

Find newer versions of dependencies than what your package.json or bower.json allows

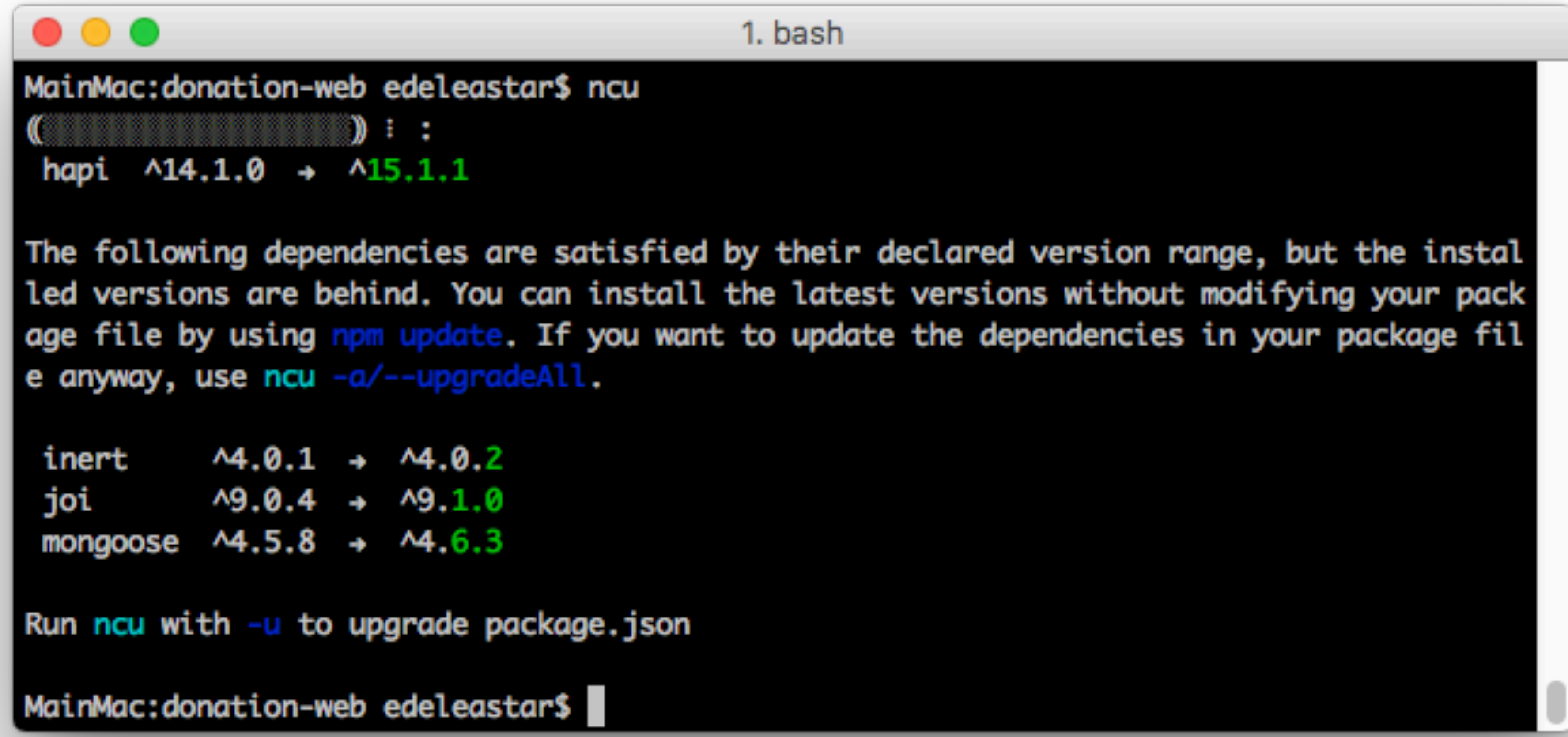
npm-check-updates is a command-line tool that allows you to upgrade your package.json or bower.json dependencies to the latest versions, regardless of existing version constraints.

```
$ npm install npm-check-updates -g
```

Report on Dependency Status (no change)

```
$ ncu
```

```
"dependencies": {
  "handlebars": "^4.0.5",
  "hapi": "^14.1.0",
  "hapi-auth-cookie": "^6.1.1",
  "inert": "^4.0.1",
  "joi": "^9.0.4",
  "mongoose": "^4.5.8",
  "vision": "^4.1.0"
}
```



```
MainMac:donation-web edeleastar$ ncu
 hapi  ^14.1.0  →  ^15.1.1

The following dependencies are satisfied by their declared version range, but the instal
led versions are behind. You can install the latest versions without modifying your pack
age file by using npm update. If you want to update the dependencies in your package fil
e anyway, use ncu -a/--upgradeAll.

 inert     ^4.0.1  →  ^4.0.2
 joi       ^9.0.4  →  ^9.1.0
 mongoose  ^4.5.8  →  ^4.6.3

Run ncu with -u to upgrade package.json

MainMac:donation-web edeleastar$
```

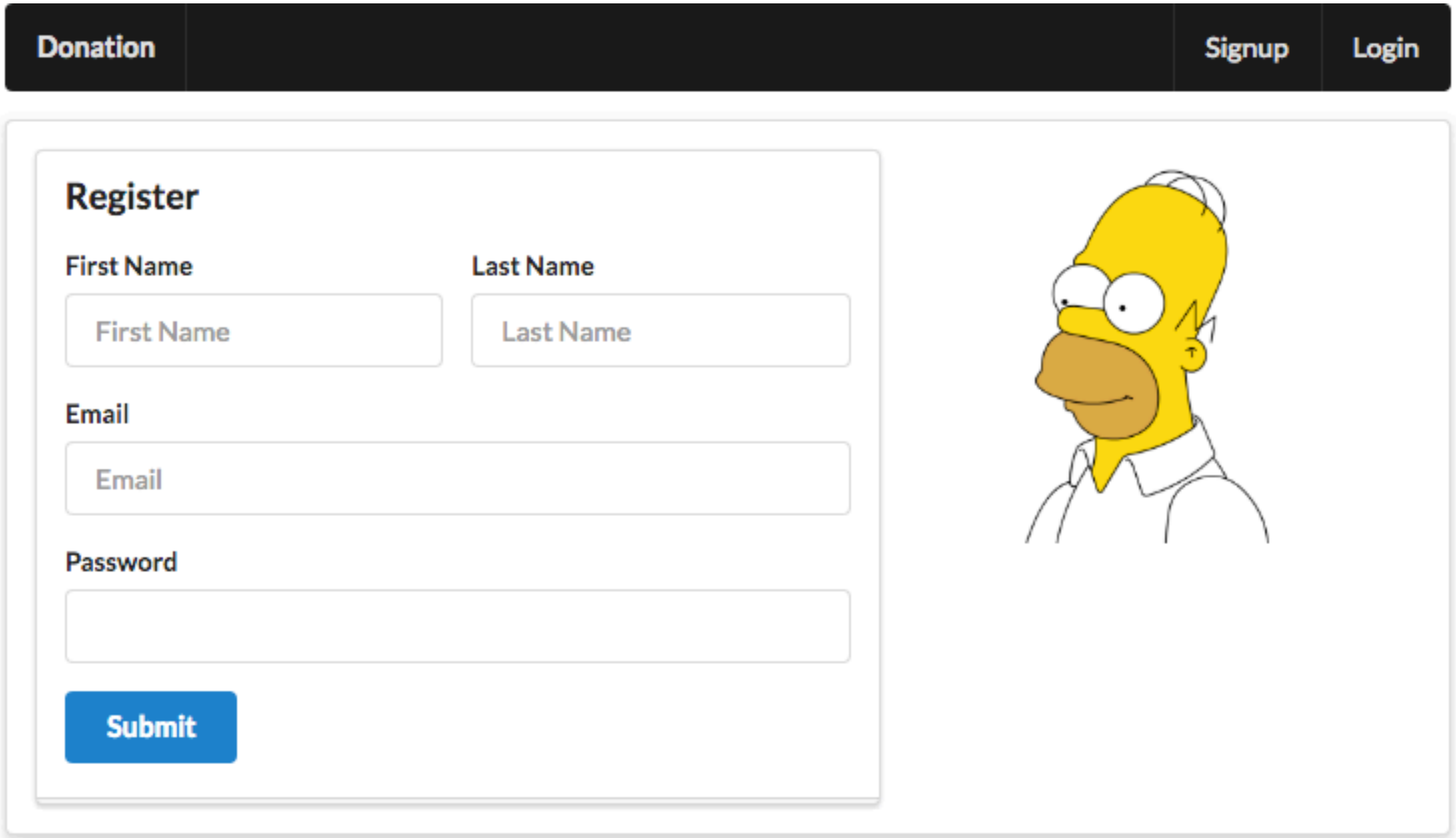
Force upgrade all dependencies

```
$ ncu -u
```

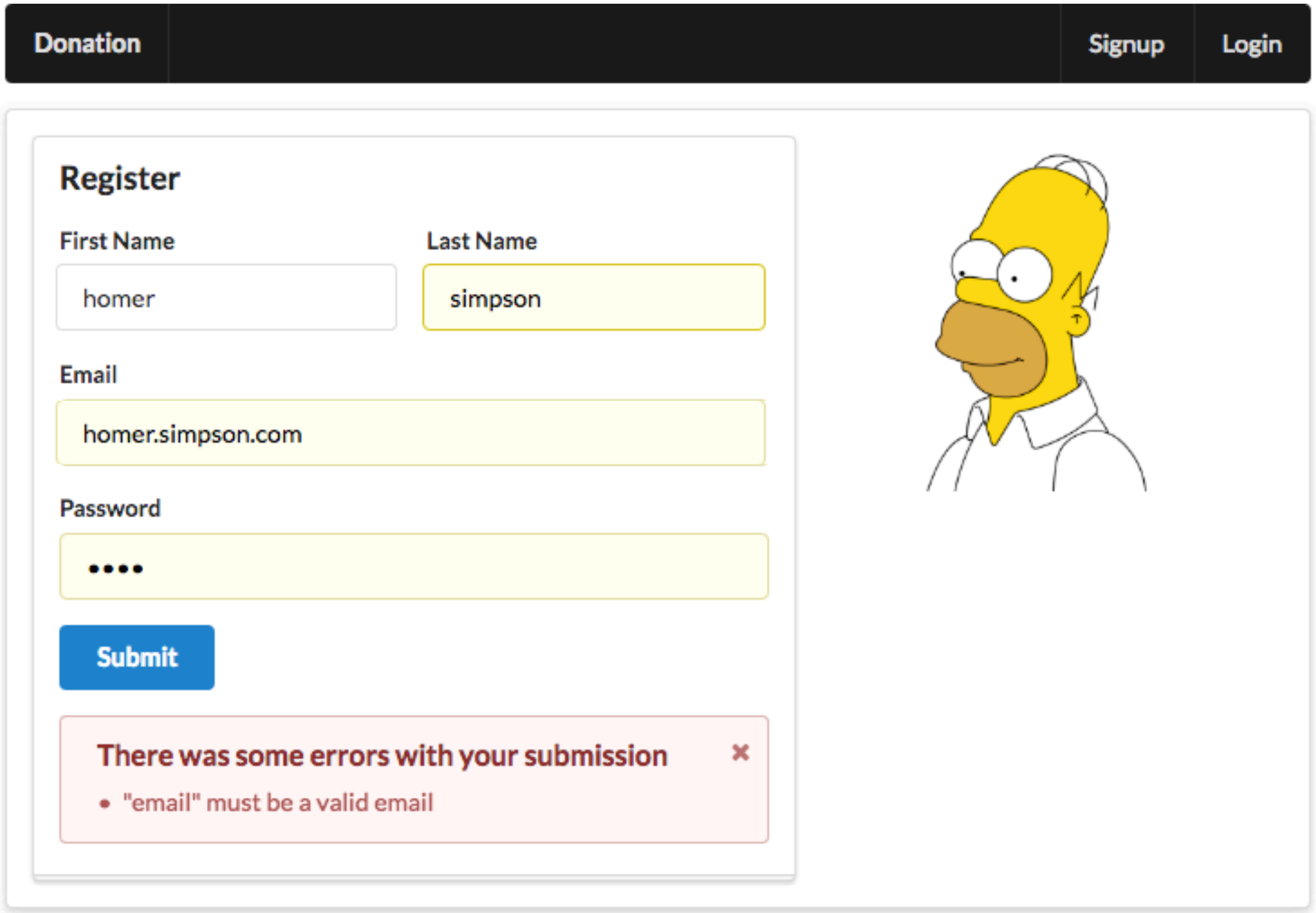
```
"dependencies": {
  "handlebars": "^4.0.5",
  "hapi": "^14.1.0",
  "hapi-auth-cookie": "^6.1.1",
  "inert": "^4.0.1",
  "joi": "^9.0.4",
  "mongoose": "^4.5.8",
  "vision": "^4.1.0"
}
```

```
"dependencies": {
  "handlebars": "^4.0.5",
  "hapi": "^15.1.1",
  "hapi-auth-cookie": "^6.1.1",
  "inert": "^4.0.1",
  "joi": "^9.0.4",
  "mongoose": "^4.5.8",
  "vision": "^4.1.0"
}
```

Joi in Donation



```
const schema = {
  firstName: Joi.string().required(),
  lastName: Joi.string().required(),
  email: Joi.string().email().required(),
  password: Joi.string().required(),
},
```



Register

First Name: homer

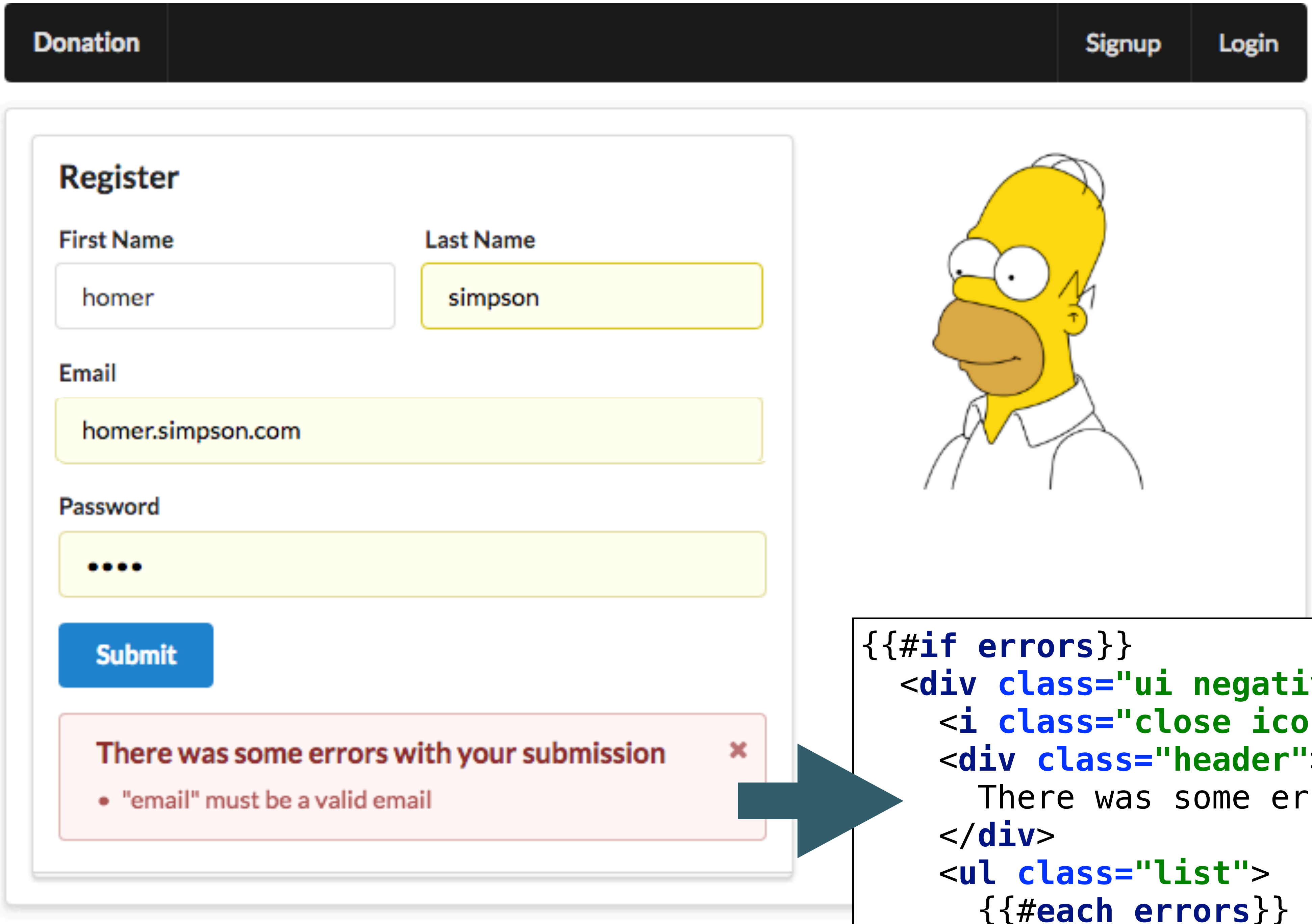
Last Name: simpson

Email: homer.simpson.com

Password: ••••

There was some errors with your submission

- "email" must be a valid email



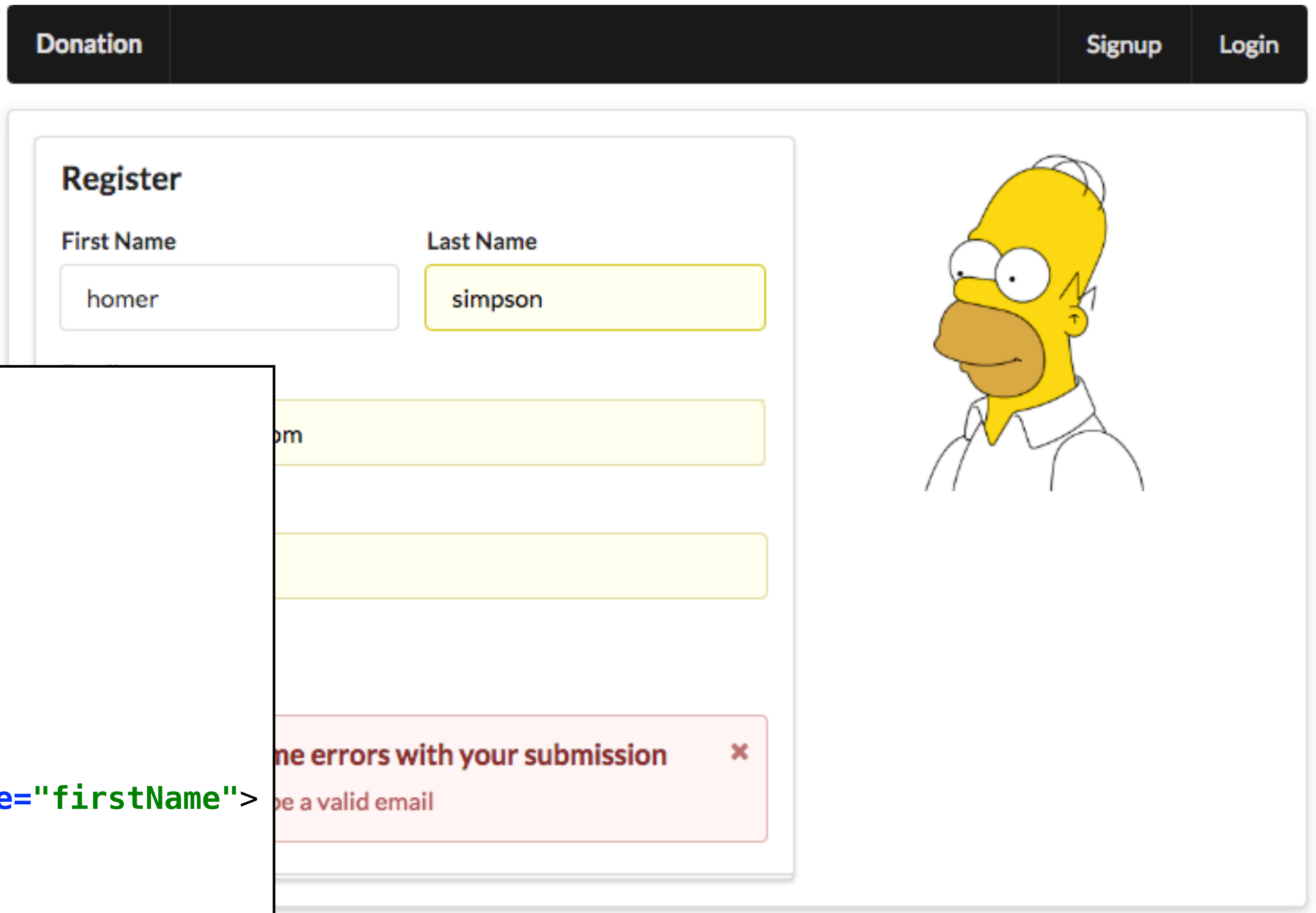
There was some errors with your submission

- "email" must be a valid email

formerror.hbs

```
{{#if errors}}
  <div class="ui negative message transition">
    <i class="close icon"></i>
    <div class="header">
      There was some errors with your submission
    </div>
    <ul class="list">
      {{#each errors}}
        <li>{{message}}</li>
      {{/each}}
    </ul>
  </div>
{{/if}}
```

signup.hbs



Register

First Name
homer

Last Name
simpson

me errors with your submission
be a valid email

```
{{> welcomemenu }}

<section class="ui raised segment">
  <div class="ui grid">
    <div class="ui ten wide column">
      <div class="ui stacked fluid form segment">
        <form action="/register" method="POST">
          <h3 class="ui header">Register</h3>
          <div class="two fields">
            <div class="field">
              <label>First Name</label>
              <input placeholder="First Name" type="text" name="firstName">
            </div>
            <div class="field">
              <label>Last Name</label>
              <input placeholder="Last Name" type="text" name="lastName">
            </div>
          </div>
          <div class="field">
            <label>Email</label>
            <input placeholder="Email" type="text" name="email">
          </div>
          <div class="field">
            <label>Password</label>
            <input type="password" name="password">
          </div>
          <button class="ui blue submit button">Submit</button>
        </form>
        {{> formerror }}
      </div>
    </div>
    <aside class="ui five wide column">
      <img src="images/homer3.png" class="ui medium image">
    </aside>
  </div>
</section>
```

```
{{#if errors}}
  <div class="ui negative message transition">
    <i class="close icon"></i>
    <div class="header">
      There was some errors with your submission
    </div>
    <ul class="list">
      {{#each errors}}
        <li>{{message}}</li>
      {{/each}}
    </ul>
  </div>
{{/if}}
```

Register handler

enable validation for this handler

the joi schema - tied to payload from request

handler if validation fails

the main handler for register, now validated

```
exports.register = {

  validate: {

    payload: {
      firstName: Joi.string().required(),
      lastName: Joi.string().required(),
      email: Joi.string().email().required(),
      password: Joi.string().required(),
    },

    failAction: function (request, reply, source, error) {
      reply.view('signup', {
        title: 'Sign up error',
        errors: error.data.details,
      }).code(400);
    },

  },

  handler: function (request, reply) {
    const user = new User(request.payload);

    user.save().then(newUser => {
      reply.redirect('/login');
    }).catch(err => {
      reply.redirect('/');
    });
  },

};
```