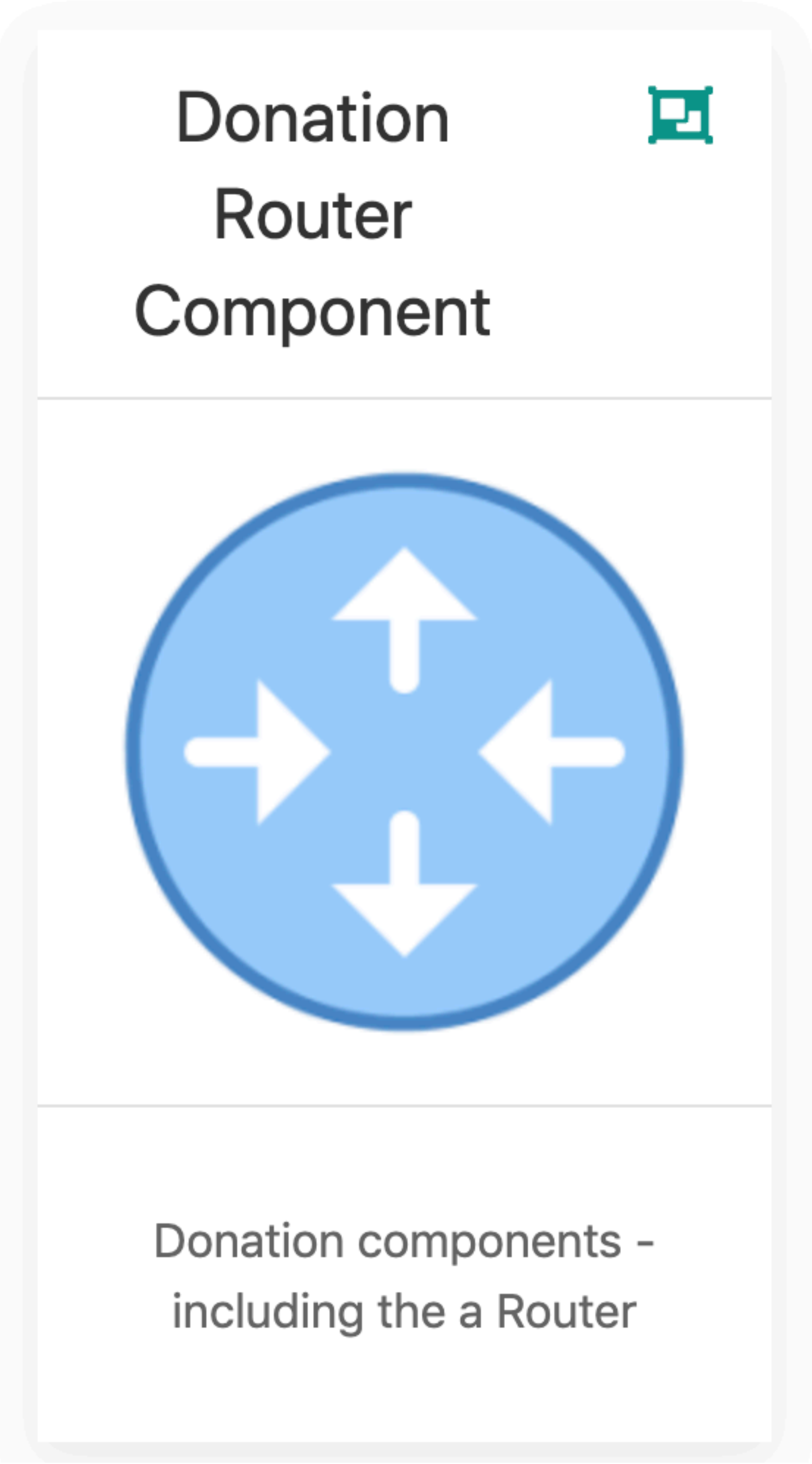
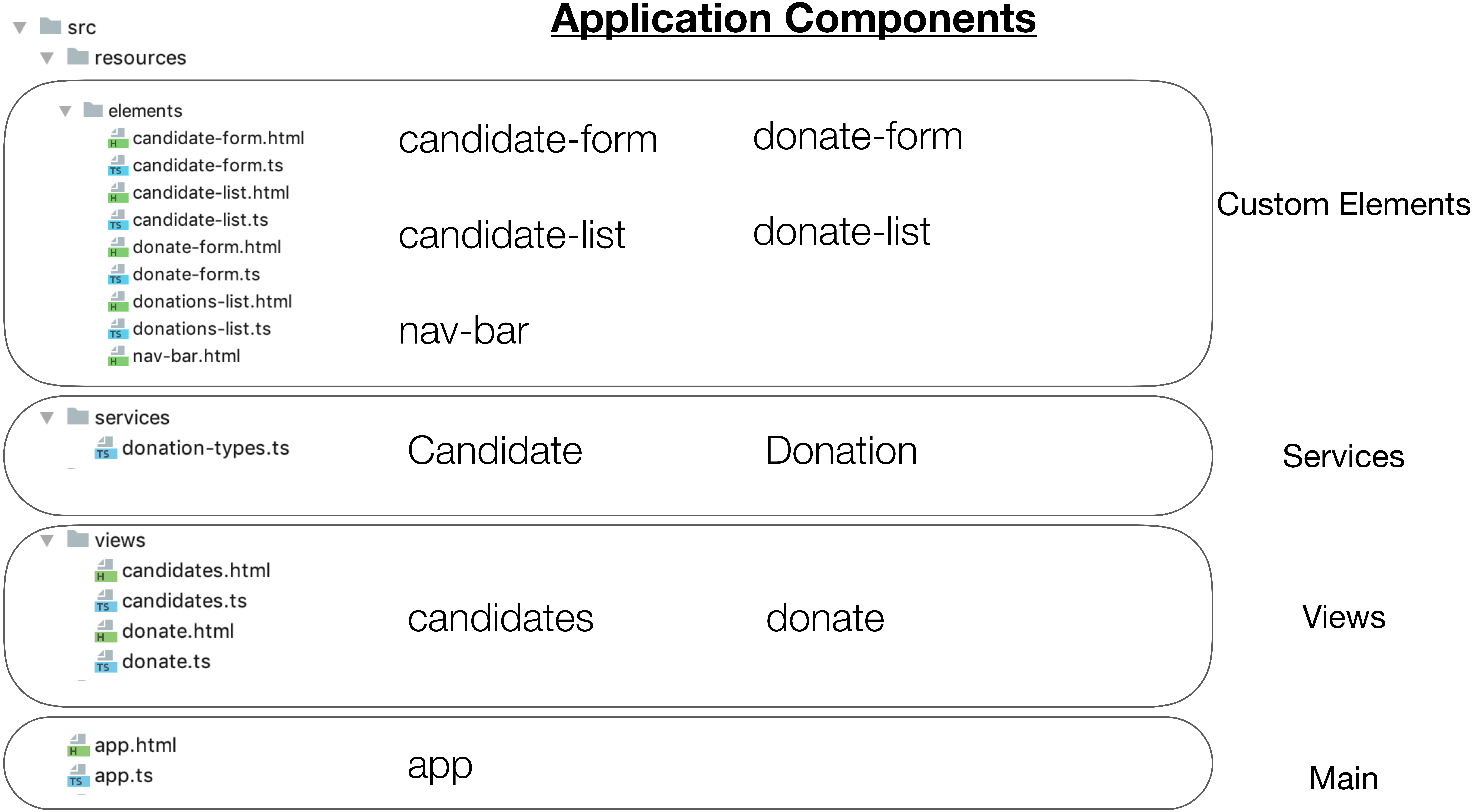
Donation Router Component



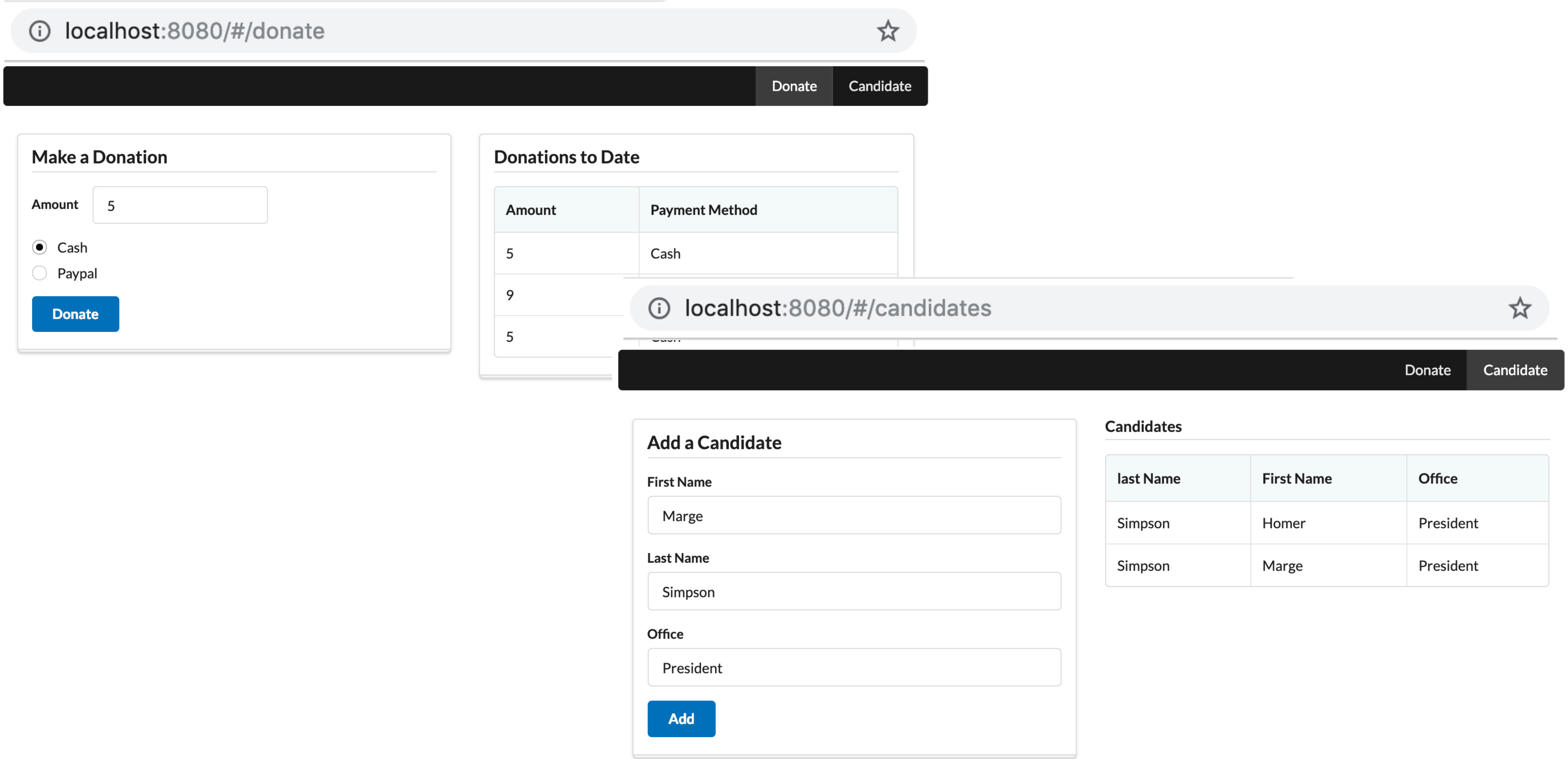
Donation Router Component

Donation components - including the a Router

Application Components



```
▼ src
  ▼ resources
    ▼ elements
        candidate-form.html
        candidate-form.ts
        candidate-list.html
        candidate-list.ts
        donate-form.html
        donate-form.ts
        donations-list.html
        donations-list.ts
        nav-bar.html
  ▼ services
      donation-types.ts
  ▼ views
      candidates.html
      candidates.ts
      donate.html
      donate.ts
    app.html
    app.ts
```



localhost:8080/#/donate

Make a Donation

Amount: 5

● Cash
○ Paypal

Donations to Date

Amount	Payment Method
5	Cash
9	
5	Cash

localhost:8080/#/candidates

Add a Candidate

First Name: Marge

Last Name: Simpson

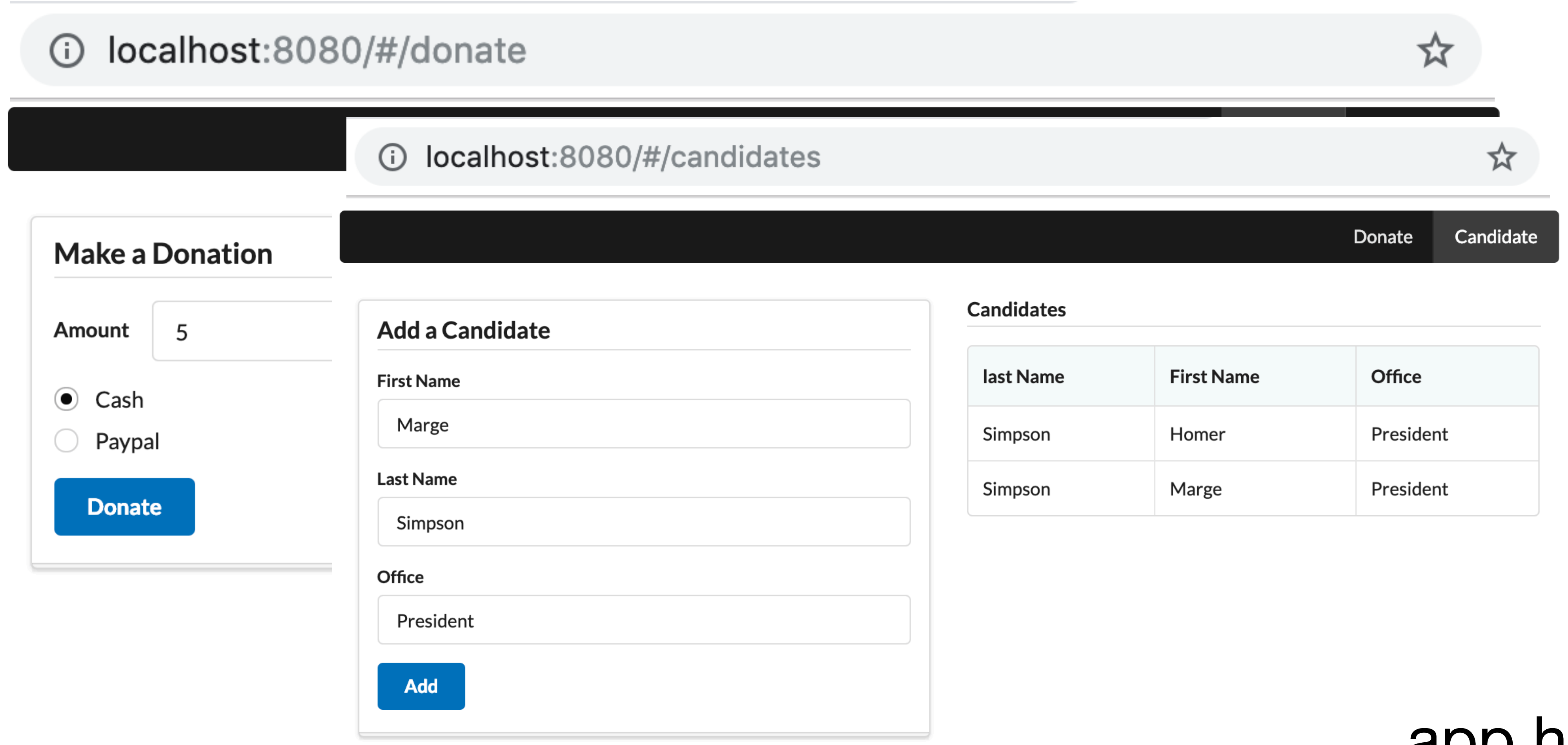
Office: President

Candidates

last Name	First Name	Office
Simpson	Homer	President
Simpson	Marge	President

Main

app.html
app.ts



localhost:8080/#/donate

Make a Donation

Amount 5

● Cash
○ Paypal

Donate

localhost:8080/#/candidates

Add a Candidate

First Name
Marge

Last Name
Simpson

Office
President

Add

Candidates

last Name	First Name	Office
Simpson	Homer	President
Simpson	Marge	President

app.ts

```
import { RouterConfiguration, Router } from 'aurelia-router';
import { PLATFORM } from 'aurelia-pal';

export class App {
  router: Router;

  configureRouter(config: RouterConfiguration, router: Router) {
    config.map([
      {
        route: ['', 'donate'],
        name: 'Donate',
        moduleId: PLATFORM.moduleName('views/donate'),
        nav: true,
        title: 'Donate'
      },
      {
        route: 'candidates',
        name: 'candidates',
        moduleId: PLATFORM.moduleName('views/candidates'),
        nav: true,
        title: 'Candidate'
      }
    ]);
    this.router = router;
  }
}
```

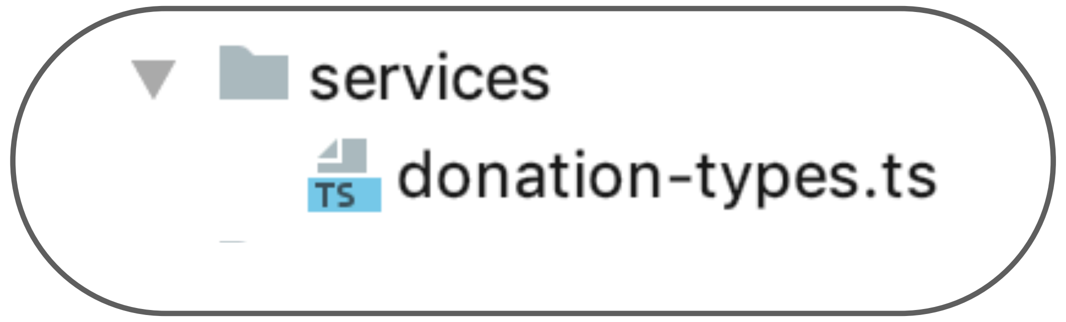
app.html

```
<template>
  <require from="resources/elements/nav-bar.html"></require>
  <div class="ui container">
    <section class="ui raised segment">
      <h3 class="ui headder"> Donation </h3>
    </section>
    <nav-bar router.bind="router"></nav-bar>
    <div class="ui basic segment">
      <router-view></router-view>
    </div>
  </div>
</template>
```

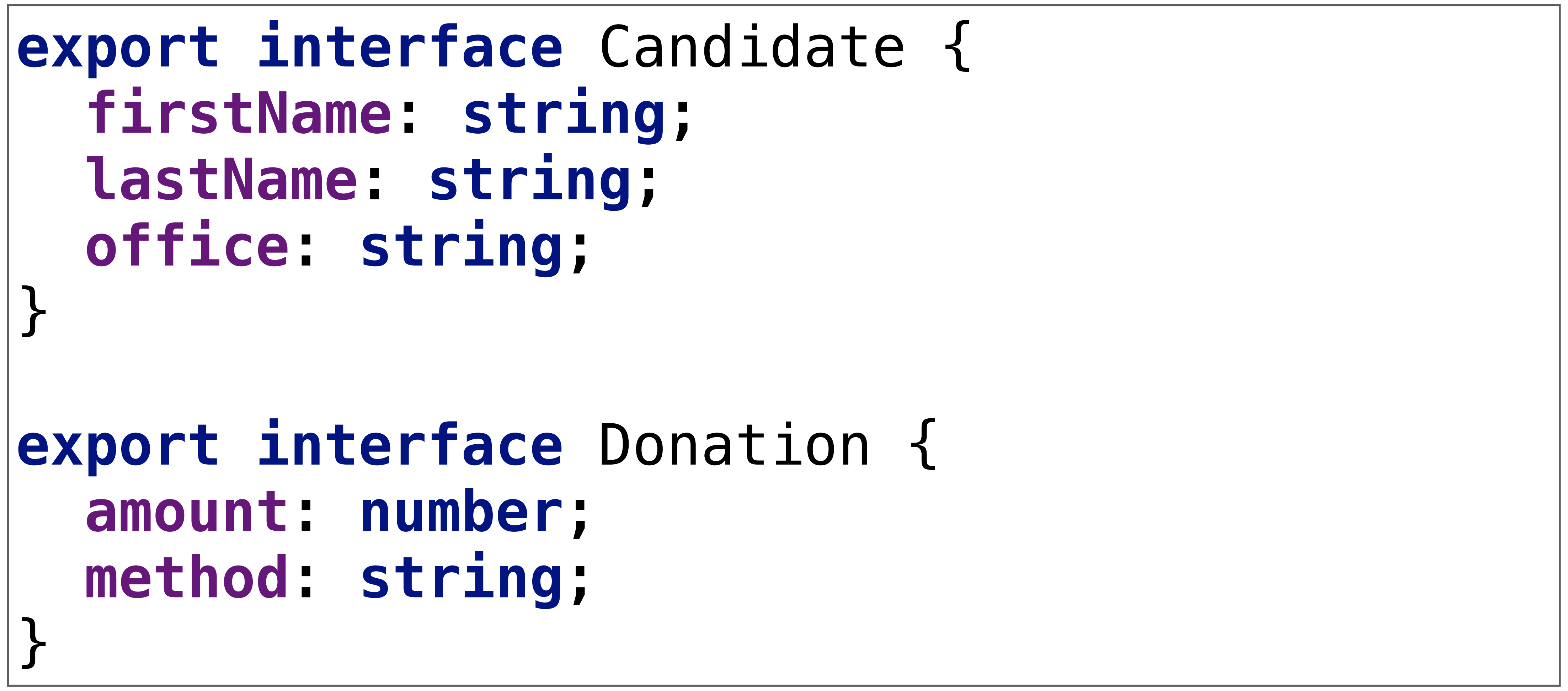
nav-bar.html

```
<template bindable="router">
  <nav class="ui inverted menu">
    <div class="right menu">
      <div repeat.for="row of router.navigation">
        <a class="${row.isActive ? 'active' : ''} item"  href.bind="row.href">${row.title}</a>
      </div>
    </div>
  </nav>
</template>
```

Services

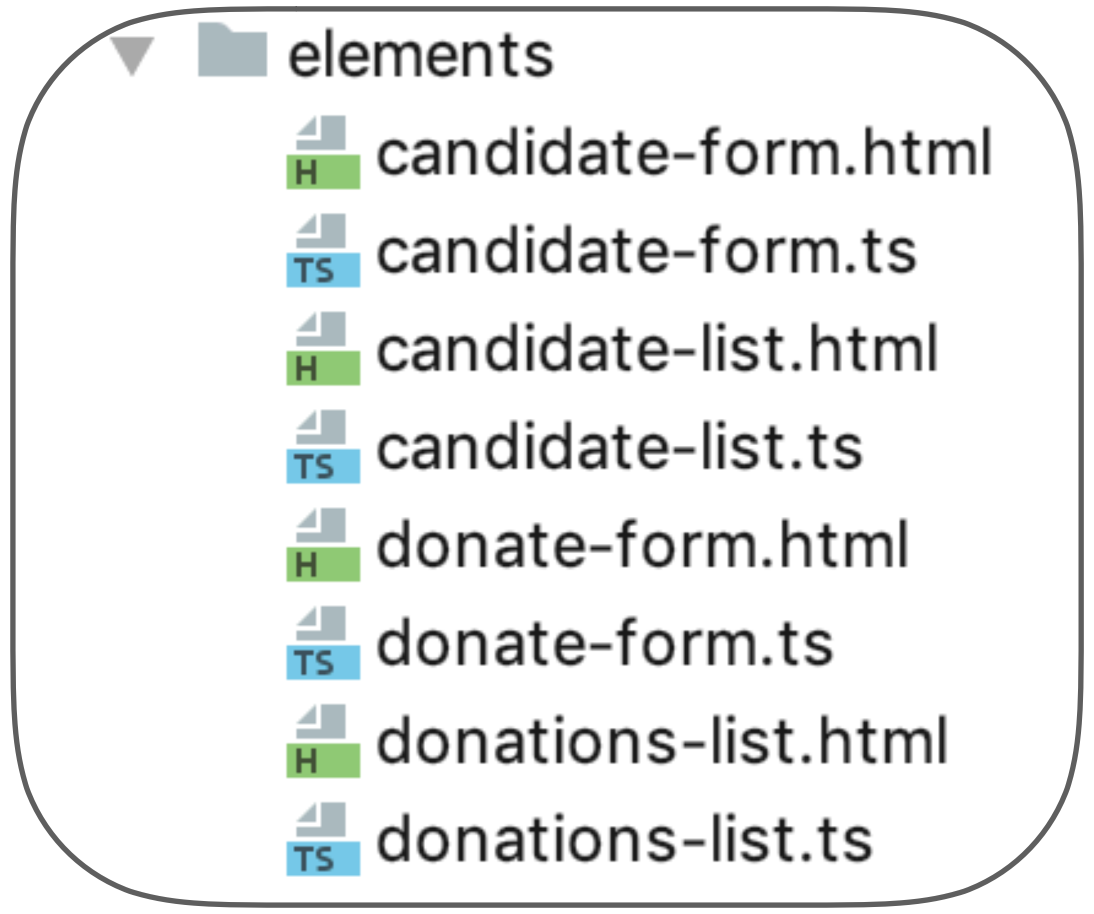


```
▼ services
    donation-types.ts
```



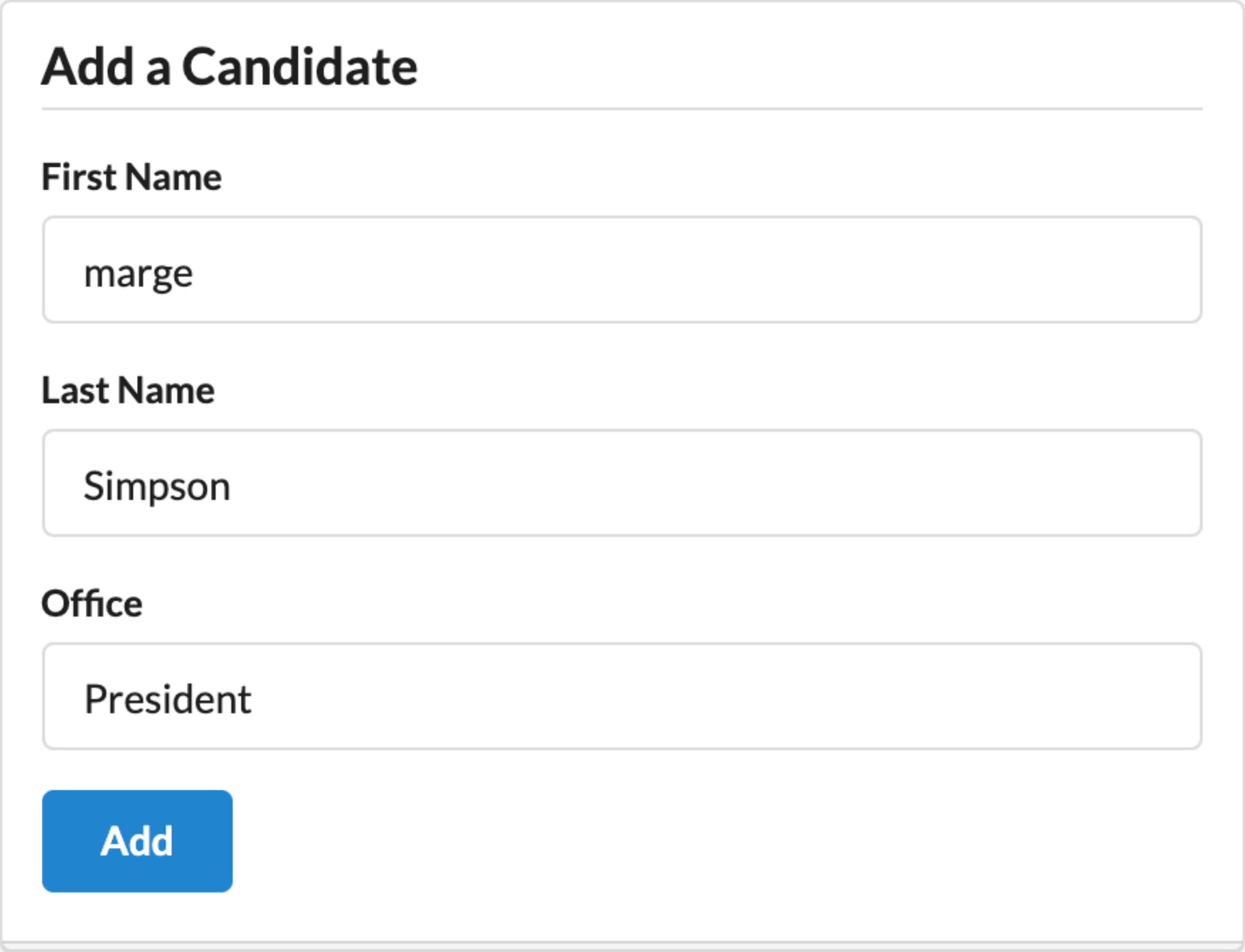
```
export interface Candidate {
  firstName: string;
  lastName: string;
  office: string;
}

export interface Donation {
  amount: number;
  method: string;
}
```

Custom Elements



- elements
  - candidate-form.html
  - candidate-form.ts
  - candidate-list.html
  - candidate-list.ts
  - donate-form.html
  - donate-form.ts
  - donations-list.html
  - donations-list.ts

candidate-form



Add a Candidate

First Name: marge

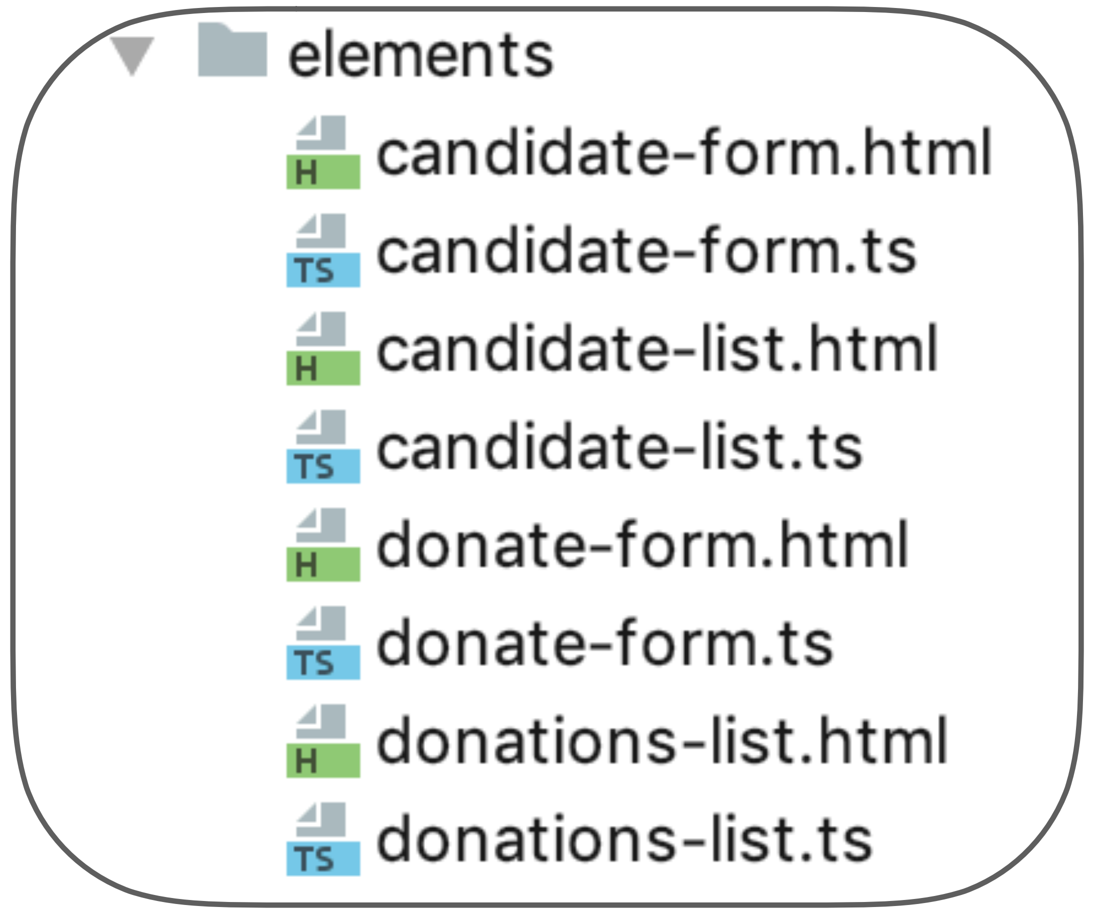
Last Name: Simpson

Office: President

```
import { bindable } from 'aurelia-framework';
import { Candidate } from '../../services/donation-types';

export class CandidateForm {
  firstName: string;
  lastName: string;
  office: string;
  @bindable
  candidates: Candidate[];

  addCandidate() {
    const candidate = {
      firstName: this.firstName,
      lastName: this.lastName,
      office: this.office
    };
    this.candidates.push(candidate);
    console.log(candidate);
  }
}
```

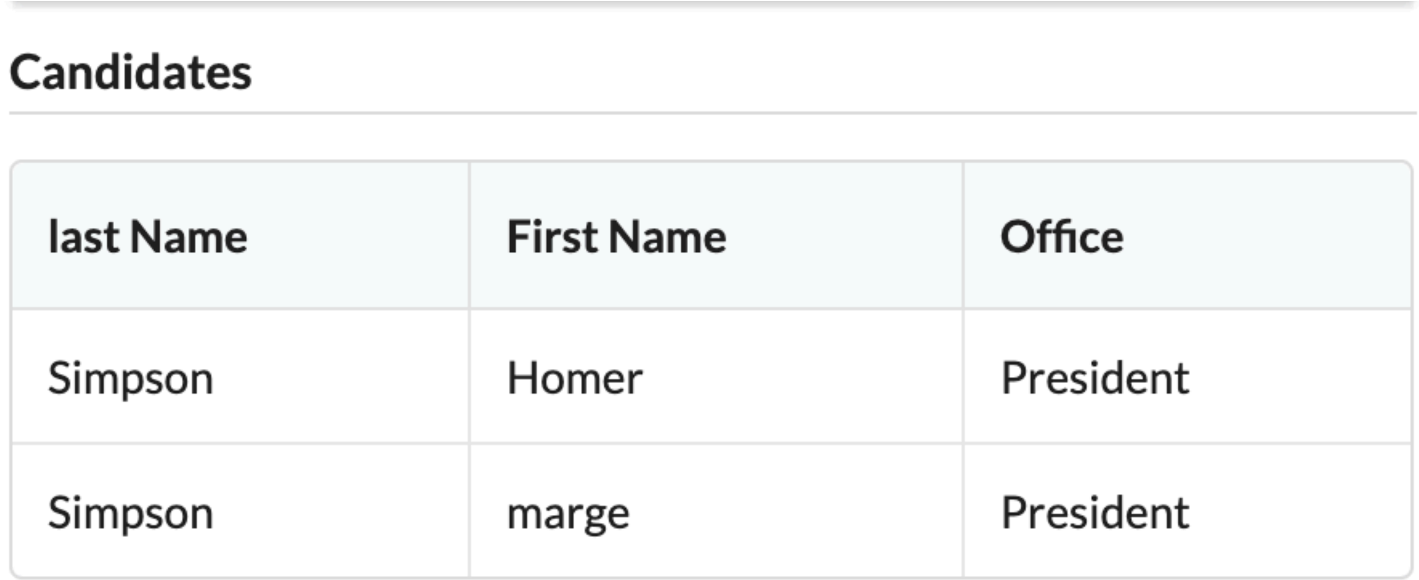
```
<template>
  <form submit.trigger="addCandidate()" class="ui form stacked segment">
    <h3 class="ui dividing header"> Add a Candidate </h3>
    <div class="field">
      <label>First Name </label> <input value.bind="firstName">
    </div>
    <div class="field">
      <label>Last Name </label> <input value.bind="lastName">
    </div>
    <div class="field">
      <label>Office </label> <input value.bind="office">
    </div>
    <button class="ui blue submit button">Add</button>
  </form>
</template>
```

Custom Elements



- elements
  - candidate-form.html
  - candidate-form.ts
  - candidate-list.html
  - candidate-list.ts
  - donate-form.html
  - donate-form.ts
  - donations-list.html
  - donations-list.ts

candidate-list



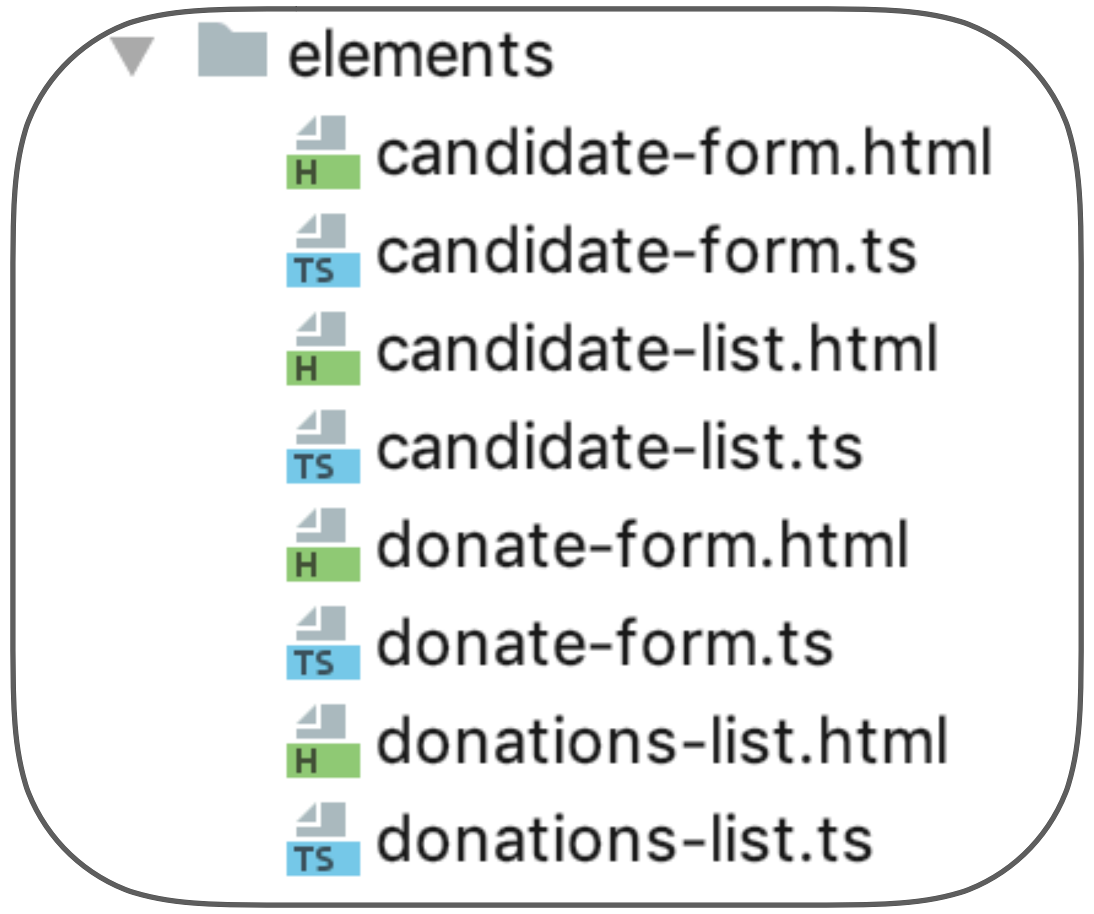
Candidates

last Name	First Name	Office
Simpson	Homer	President
Simpson	marge	President

```
import { bindable } from 'aurelia-framework';
import { Candidate } from '../../services/donation-types';

export class CandidateList {
  @bindable
  candidates: Candidate[];
}
```

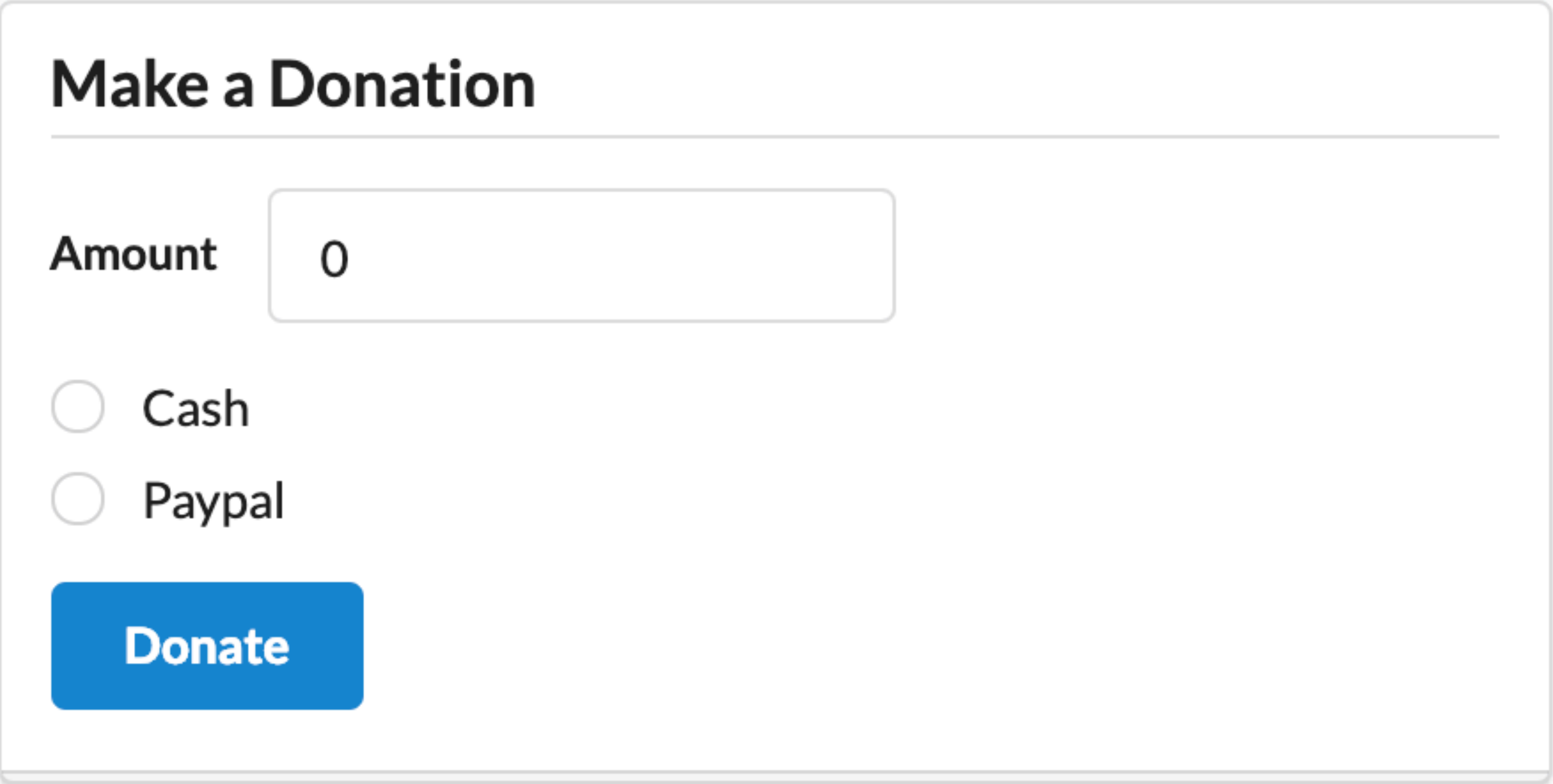
```
<template>
  <h4 class="ui dividing header"> Candidates </h4>
  <table class="ui celled table segment">
    <thead>
      <tr>
        <th>last Name</th>
        <th>First Name</th>
        <th>Office</th>
      </tr>
    </thead>
    <tbody>
      <tr repeat.for="candidate of candidates">
        <td>
          ${candidate.lastName}
        <td>
          ${candidate.firstName}
        </td>
        <td>
          ${candidate.office}
        </td>
      </tr>
    </tbody>
  </table>
</template>
```

Custom Elements



elements
- candidate-form.html
- candidate-form.ts
- candidate-list.html
- candidate-list.ts
- donate-form.html
- donate-form.ts
- donations-list.html
- donations-list.ts

donations-form



Make a Donation

Amount 0

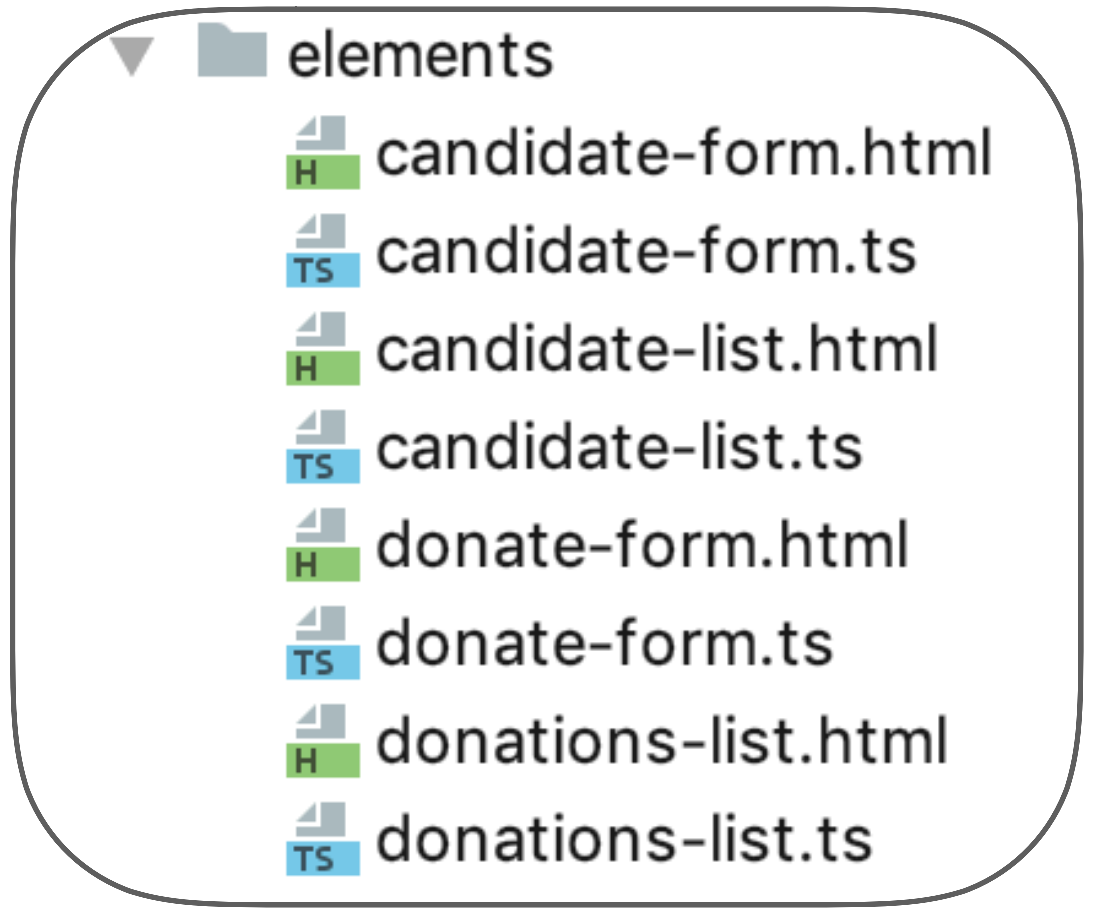
○ Cash

○ Paypal

```
import { bindable } from 'aurelia-framework';
import {Donation} from "../../services/donation-types";

export class DonateForm {
  amount = '0';
  @bindable
  donations: Donation[] = [];
  @bindable
  paymentMethods: string[];

  selectedMethod = '';

  makeDonation() {
    const donation = {
      amount: parseInt(this.amount),
      method : this.selectedMethod
    };
    this.donations.push(donation);
    console.log(this.donations);
  }
}
```

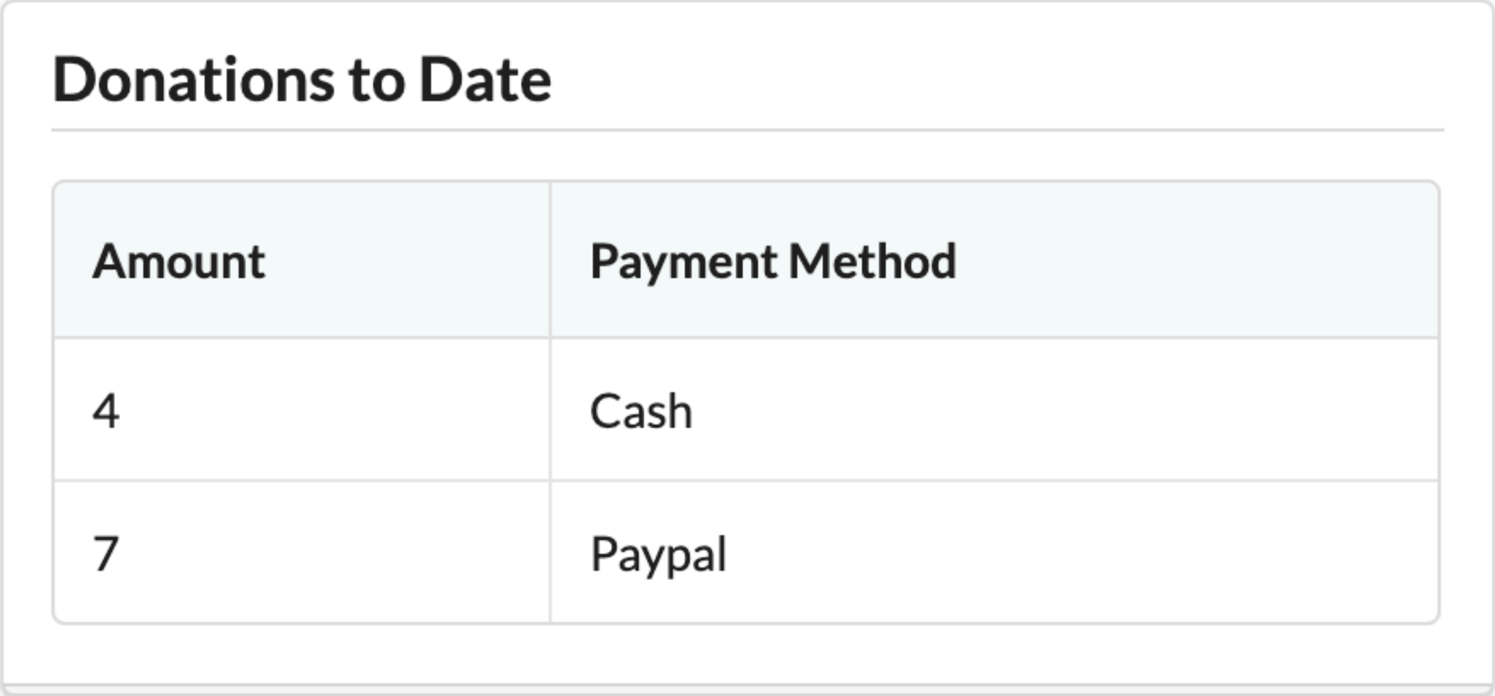
```
<template>
  <form submit.trigger="makeDonation()" class="ui form stacked segment">
    <h3 class='ui dividing header'> Make a Donation </h3>
    <div class="grouped inline fields">
      <div class="field">
        <label>Amount</label> <input type="number" value.bind="amount">
      </div>
    </div>
    <div class="grouped inline fields">
      <div class="field" repeat.for="method of paymentMethods">
        <div class="ui radio checkbox">
          <input type="radio" model.bind="method" checked.bind="selectedMethod">
          <label>${method}</label>
        </div>
      </div>
    </div>
    <button class="ui blue submit button">Donate</button>
  </form>
</template>
```

Custom Elements



- elements
  - candidate-form.html
  - candidate-form.ts
  - candidate-list.html
  - candidate-list.ts
  - donate-form.html
  - donate-form.ts
  - donations-list.html
  - donations-list.ts

donations-list



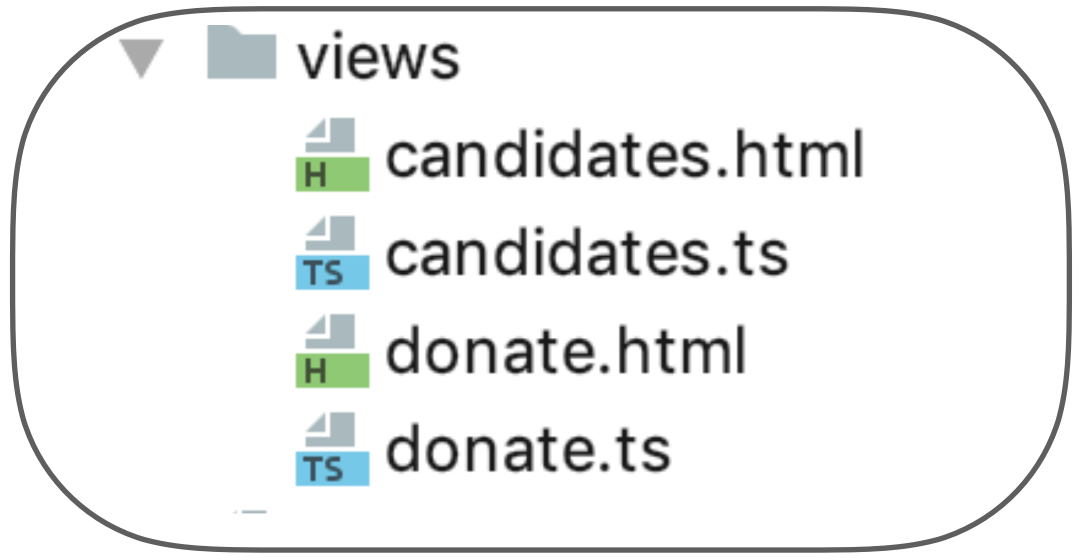
Donations to Date

Amount	Payment Method
4	Cash
7	Paypal

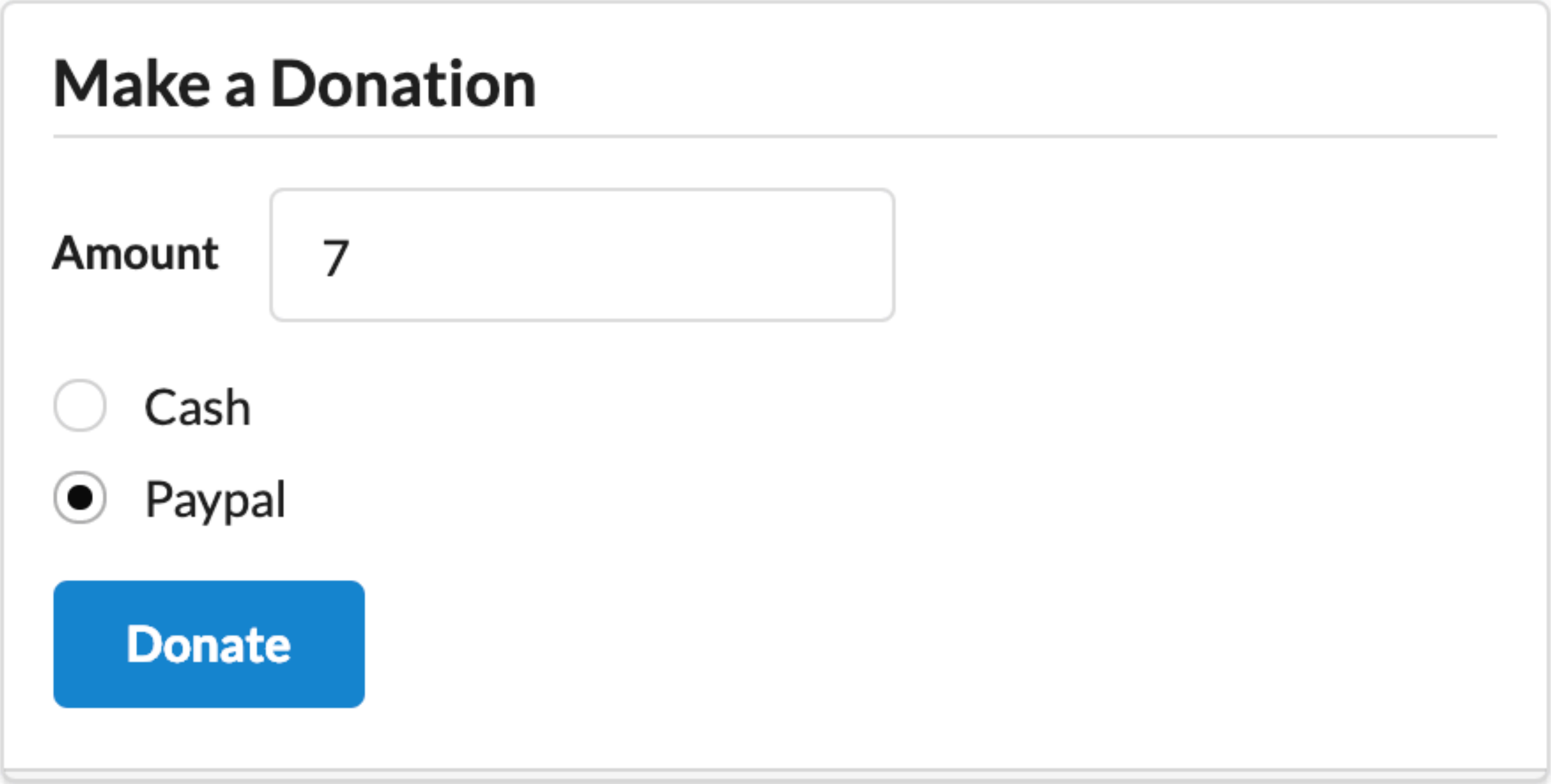
```
import { bindable } from 'aurelia-framework';

export class DonationsList {
  @bindable
  donations : number[];
}
```

```
<template>
  <div class="ui stacked segment">
    <h3 class="ui dividing header"> Donations to Date </h3>
    <table class="ui celled table segment">
      <thead>
        <tr>
          <th>Amount</th>
          <th>Payment Method</th>
        </tr>
      </thead>
      <tbody>
        <tr repeat.for="donation of donations">
          <td> ${donation.amount}</td>
          <td> ${donation.method}</td>
        </tr>
      </tbody>
    </table>
  </div>
</template>
```

Views



- views
  - candidates.html
  - candidates.ts
  - donate.html
  - donate.ts

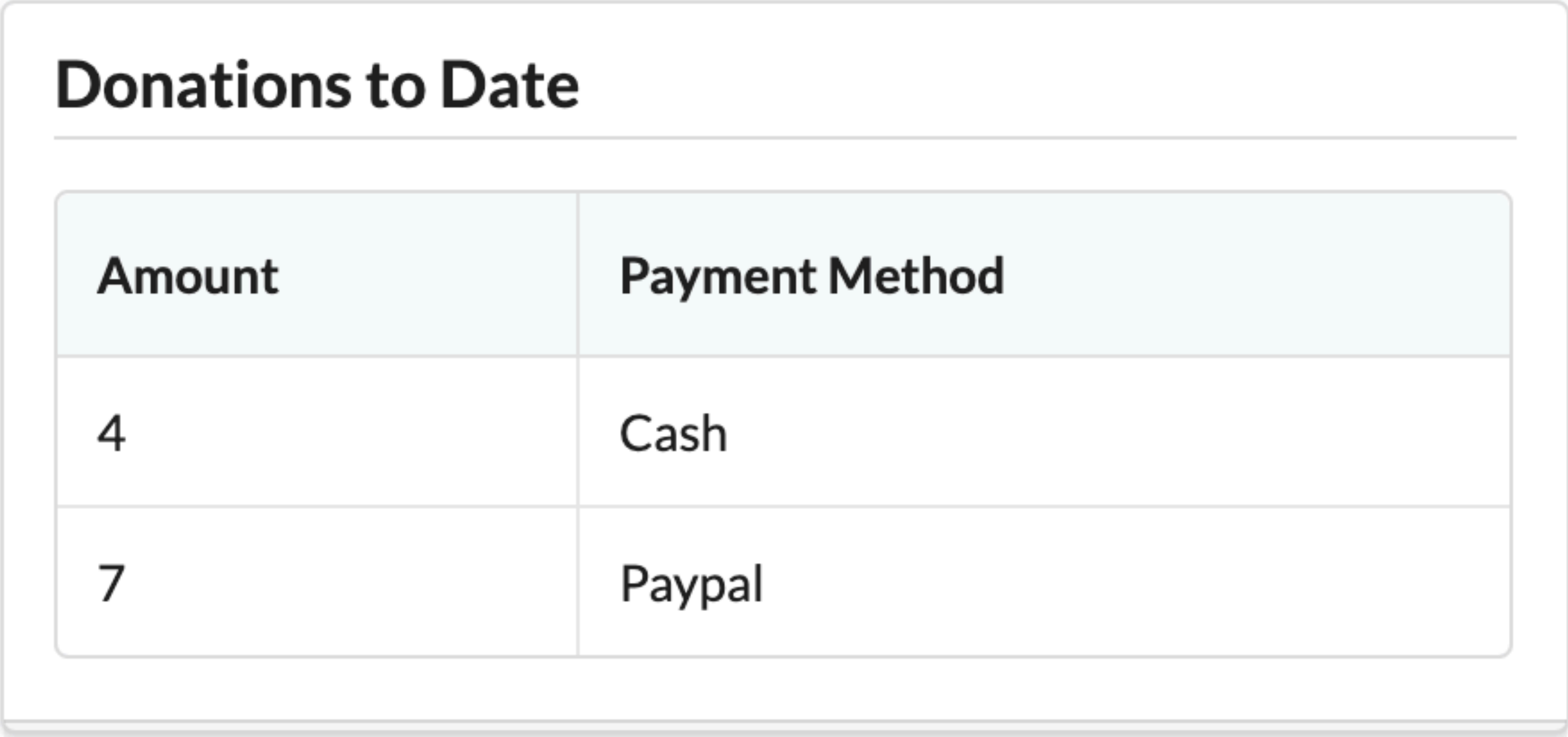


Make a Donation

Amount: 7

○ Cash

● Paypal



Donations to Date

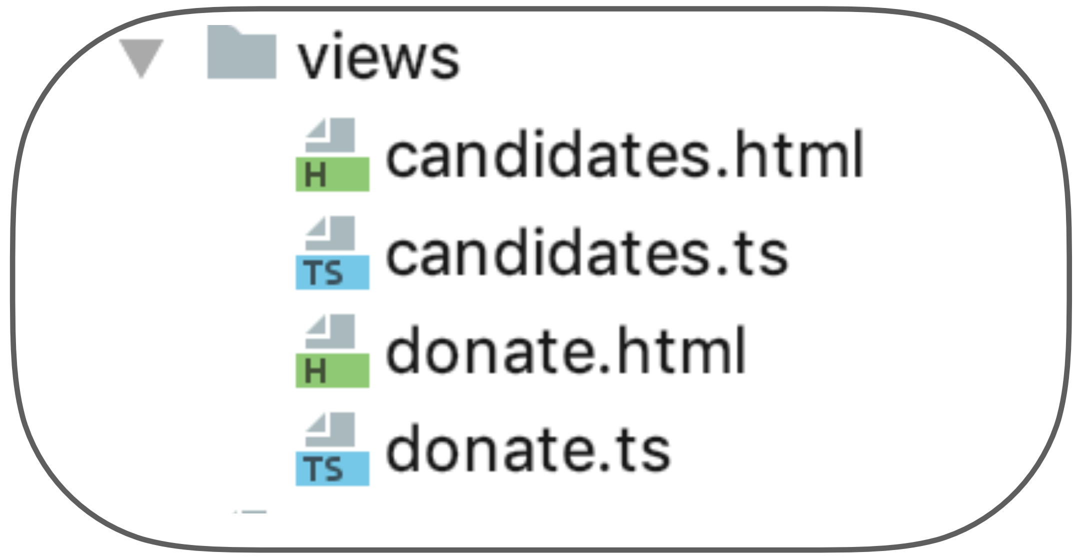
Amount	Payment Method
4	Cash
7	Paypal

donate

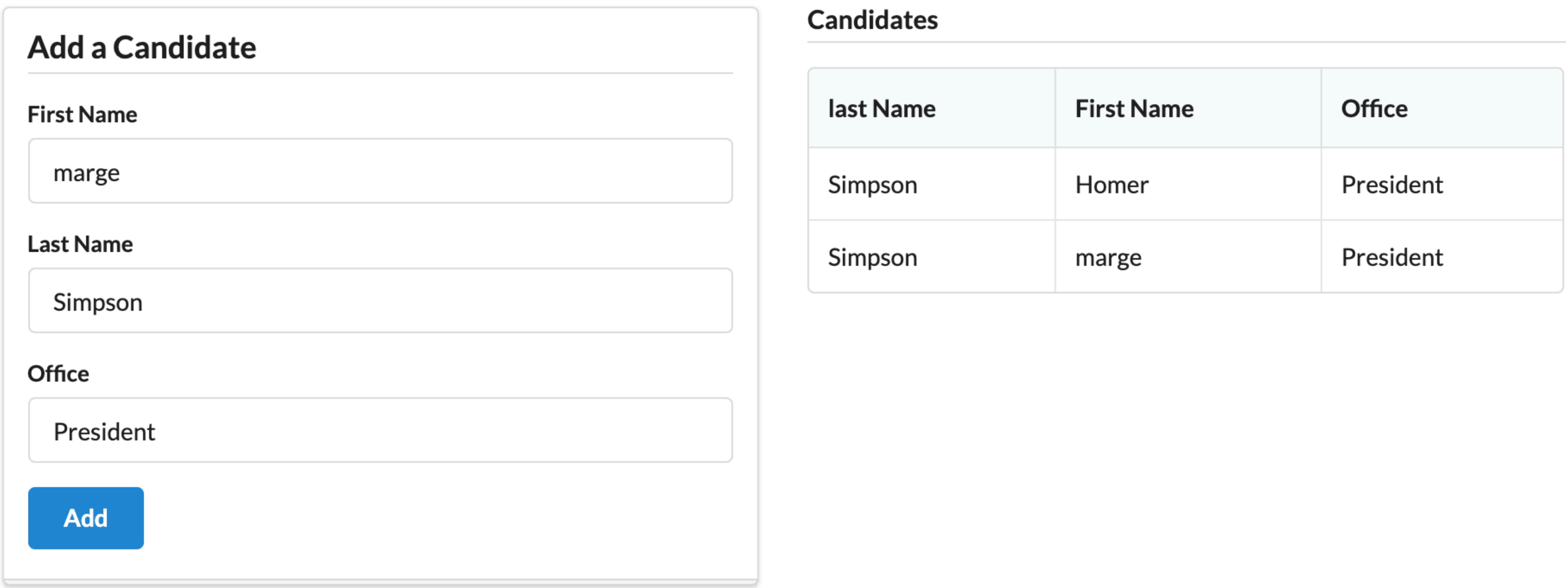
```
import {Donation} from "../services/donation-types";

export class Donate {
  donations: Donation[] = [];
  paymentMethods = ['Cash', 'Paypal'];
}
```

```
<template>
  <require from="../resources/elements/donations-list"></require>
  <require from="../resources/elements/donate-form"></require>

  <div class="ui stackable two column grid">
    <div class="column">
      <donate-form donations.bind="donations" payment-methods.bind="paymentMethods"></donate-form>
    </div>
    <div class="column">
      <donations-list donations.bind="donations"></donations-list>
    </div>
  </div>
</template>
```

Views



- views
  - candidates.html
  - candidates.ts
  - donate.html
  - donate.ts



Add a Candidate

First Name: marge

Last Name: Simpson

Office: President

Candidates

last Name	First Name	Office
Simpson	Homer	President
Simpson	marge	President

candidates

```
import { Candidate } from '../services/donation-types';

export class Candidates {
  candidates: Candidate[] = [];
}
```

```
<template>
  <require from="../resources/elements/candidate-form"></require>
  <require from="../resources/elements/candidate-list"></require>

  <div class="ui stackable two column grid">
    <div class="column">
      <candidate-form candidates.bind="candidates"></candidate-form>
    </div>
    <div class="column">
      <candidate-list candidates.bind="candidates"></candidate-list>
    </div>
  </div>
</template>
```