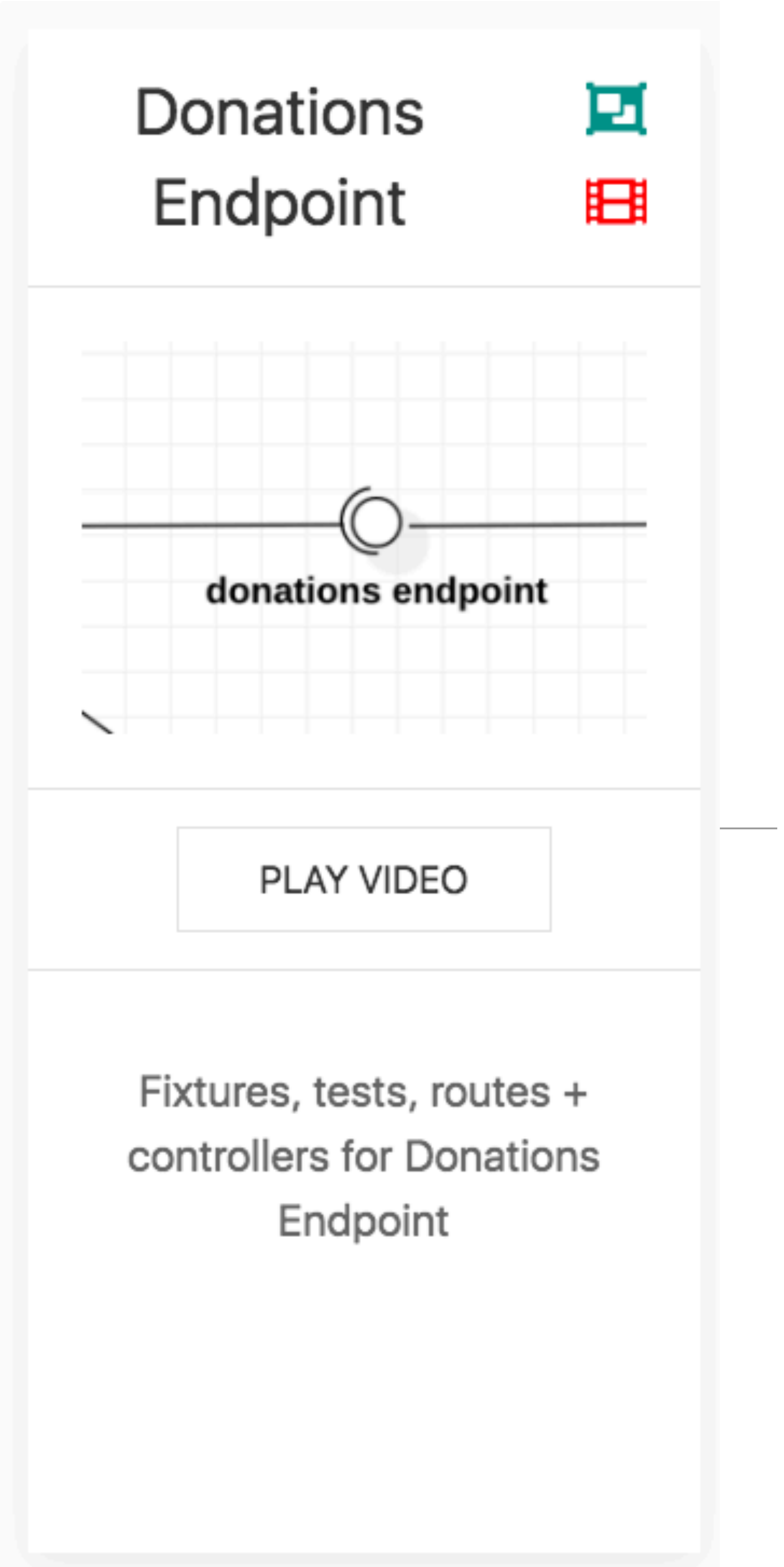
Donations Endpoint



Donations Endpoint

Fixtures, tests, routes + controllers for Donations Endpoint

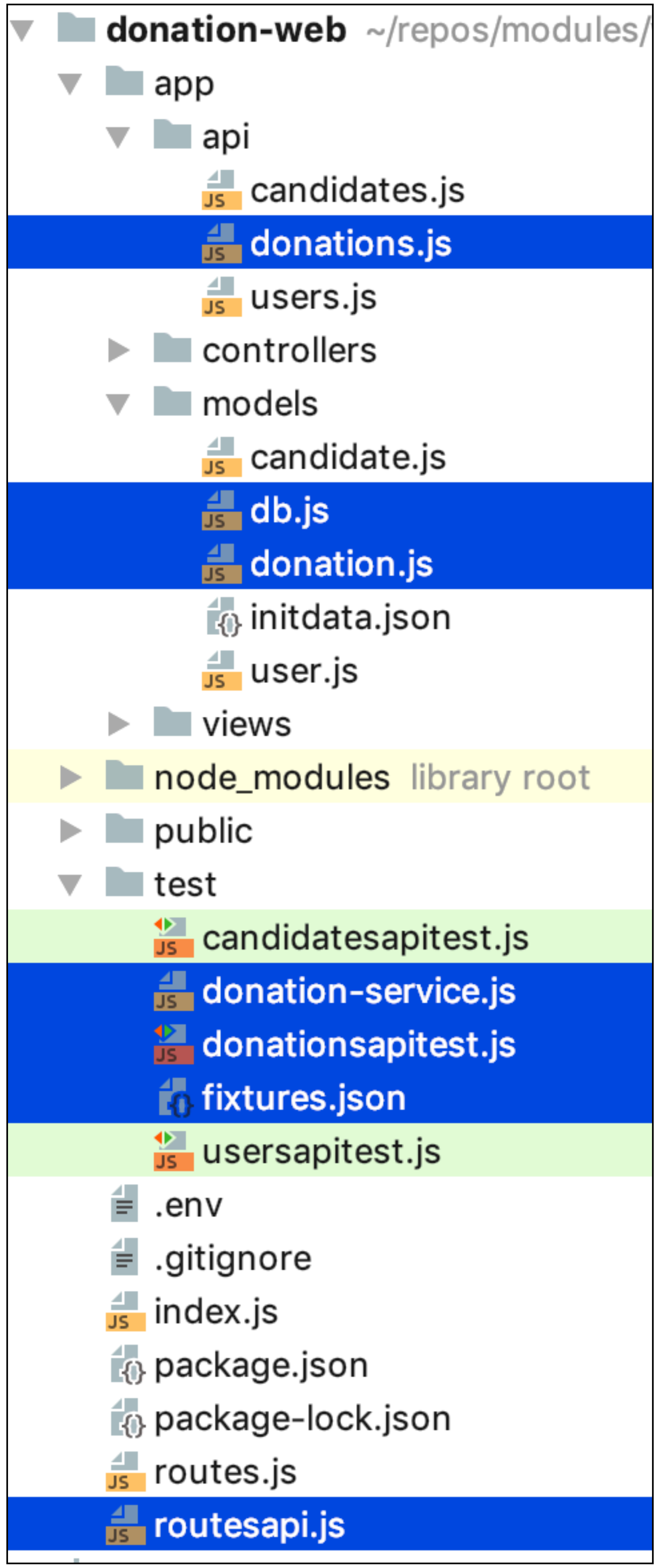
Donations

- Create, Delete and Retrieve donations needs to be supported
- Retrieve Donations options:
  - All donations recorded for all Candidates
  - All Donations for a single candidate
  - All donations made by an singe donor
- Create Donation
  - By a Donor to a Candidate

userstest
fixtures.json
donationstest
DonationService
candidatestest
users endpoint
donations endpoint
candidates endpoint
api
Users
Donations
Candidates
models
UserSchema
DonationSchema
CandidateSchema
db
initdata.json
MongoDb

Donations

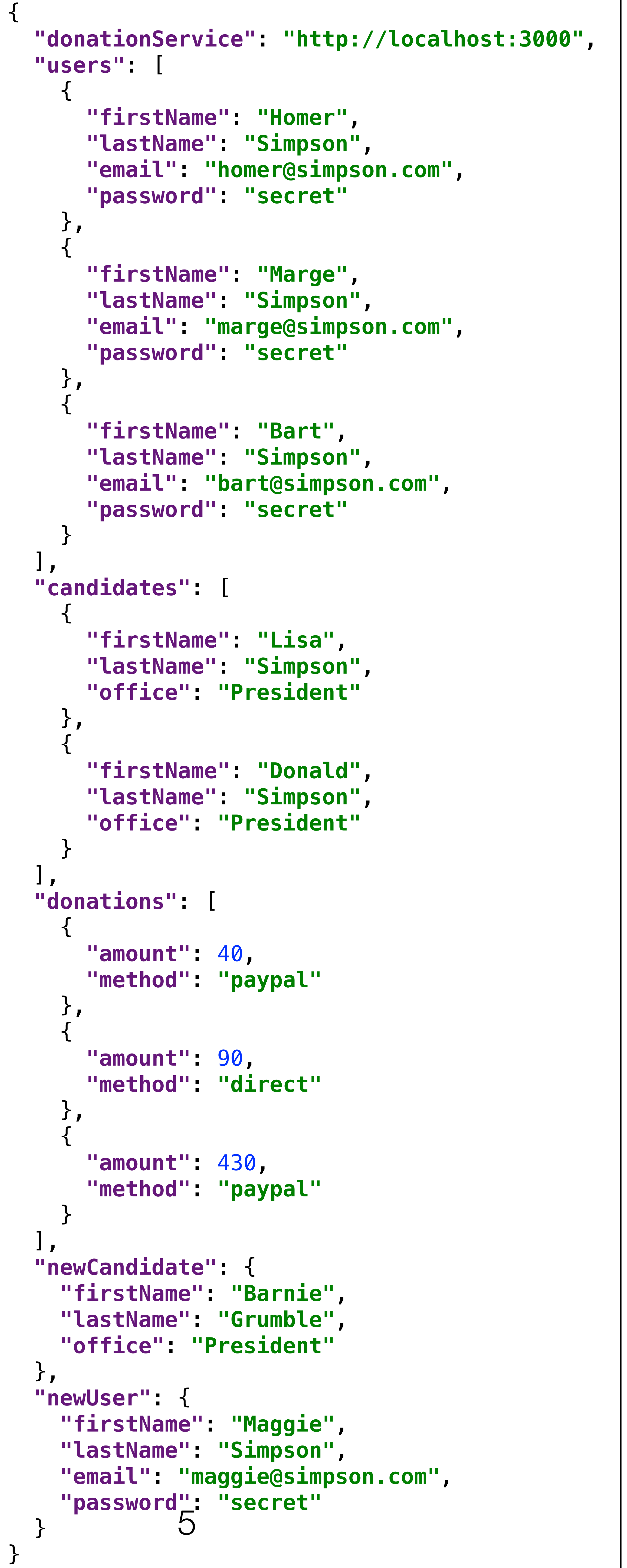
- Fixtures
- Routes
- Controller
- DonationService
- Unit Tests



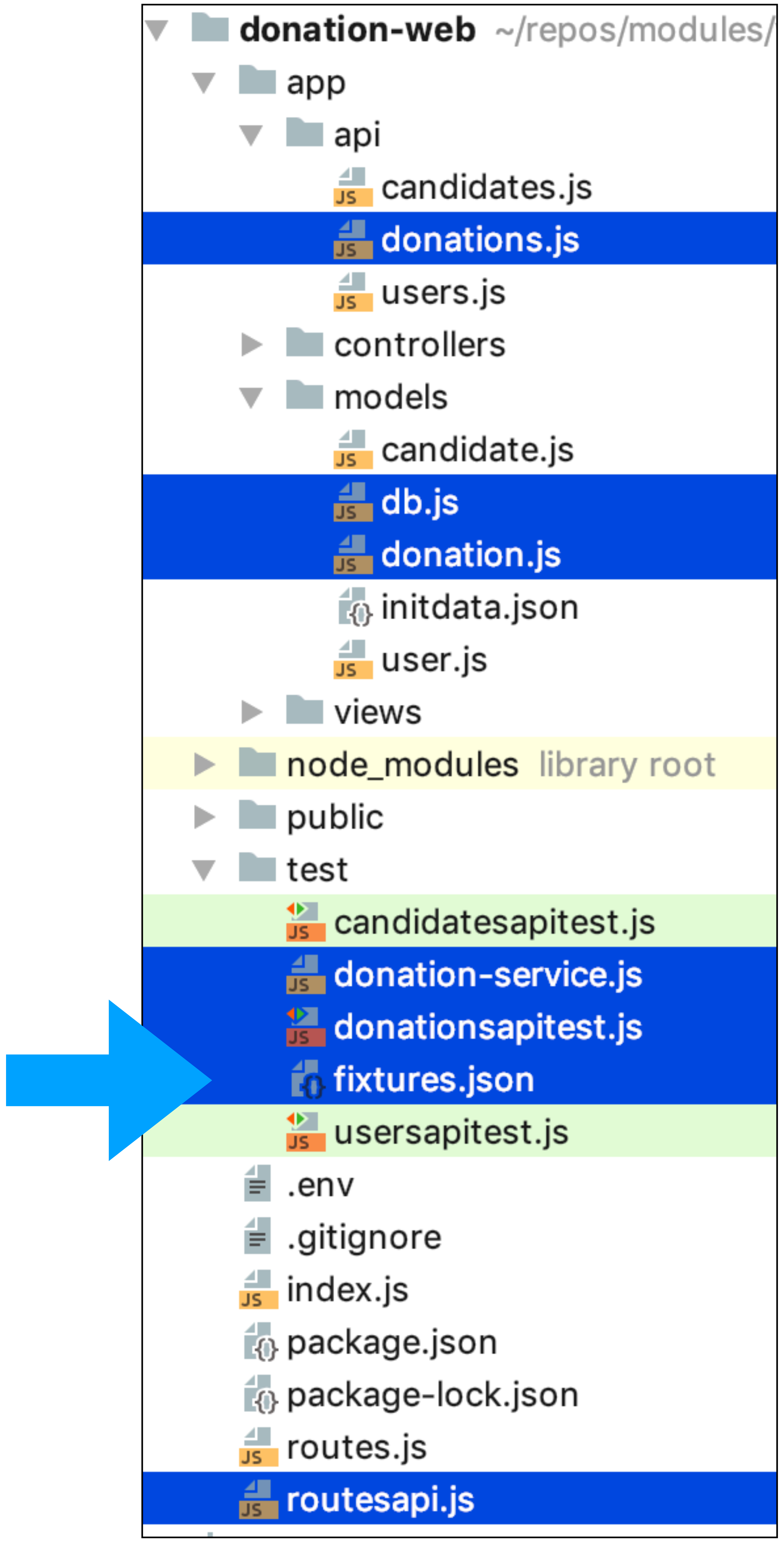
```
▼ donation-web ~/repos/modules/
  ▼ app
    ▼ api
        candidates.js
        donations.js
        users.js
    ▶ controllers
    ▼ models
        candidate.js
        db.js
        donation.js
        initdata.json
        user.js
    ▶ views
  ▶ node_modules library root
  ▶ public
  ▼ test
      candidatesapitest.js
      donation-service.js
      donationsapitest.js
      fixtures.json
      usersapitest.js
    .env
    .gitignore
    index.js
    package.json
    package-lock.json
    routes.js
    routesapi.js
```

Fixtures

- Fixtures to enhance reusability & readability of unit tests



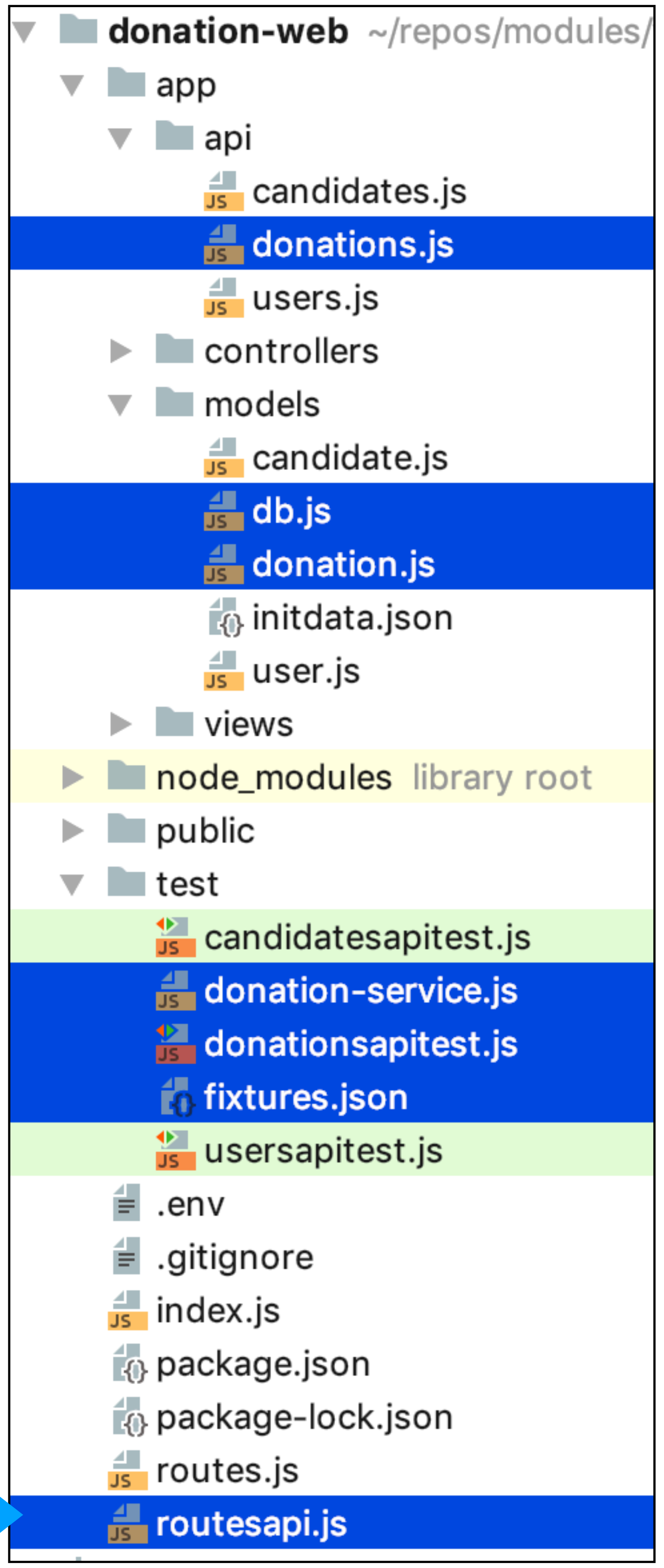
```
{
  "donationService": "http://localhost:3000",
  "users": [
    {
      "firstName": "Homer",
      "lastName": "Simpson",
      "email": "homer@simpson.com",
      "password": "secret"
    },
    {
      "firstName": "Marge",
      "lastName": "Simpson",
      "email": "marge@simpson.com",
      "password": "secret"
    },
    {
      "firstName": "Bart",
      "lastName": "Simpson",
      "email": "bart@simpson.com",
      "password": "secret"
    }
  ],
  "candidates": [
    {
      "firstName": "Lisa",
      "lastName": "Simpson",
      "office": "President"
    },
    {
      "firstName": "Donald",
      "lastName": "Simpson",
      "office": "President"
    }
  ],
  "donations": [
    {
      "amount": 40,
      "method": "paypal"
    },
    {
      "amount": 90,
      "method": "direct"
    },
    {
      "amount": 430,
      "method": "paypal"
    }
  ],
  "newCandidate": {
    "firstName": "Barnie",
    "lastName": "Grumble",
    "office": "President"
  },
  "newUser": {
    "firstName": "Maggie",
    "lastName": "Simpson",
    "email": "maggie@simpson.com",
    "password": "secret"
  }
}
```



Routes

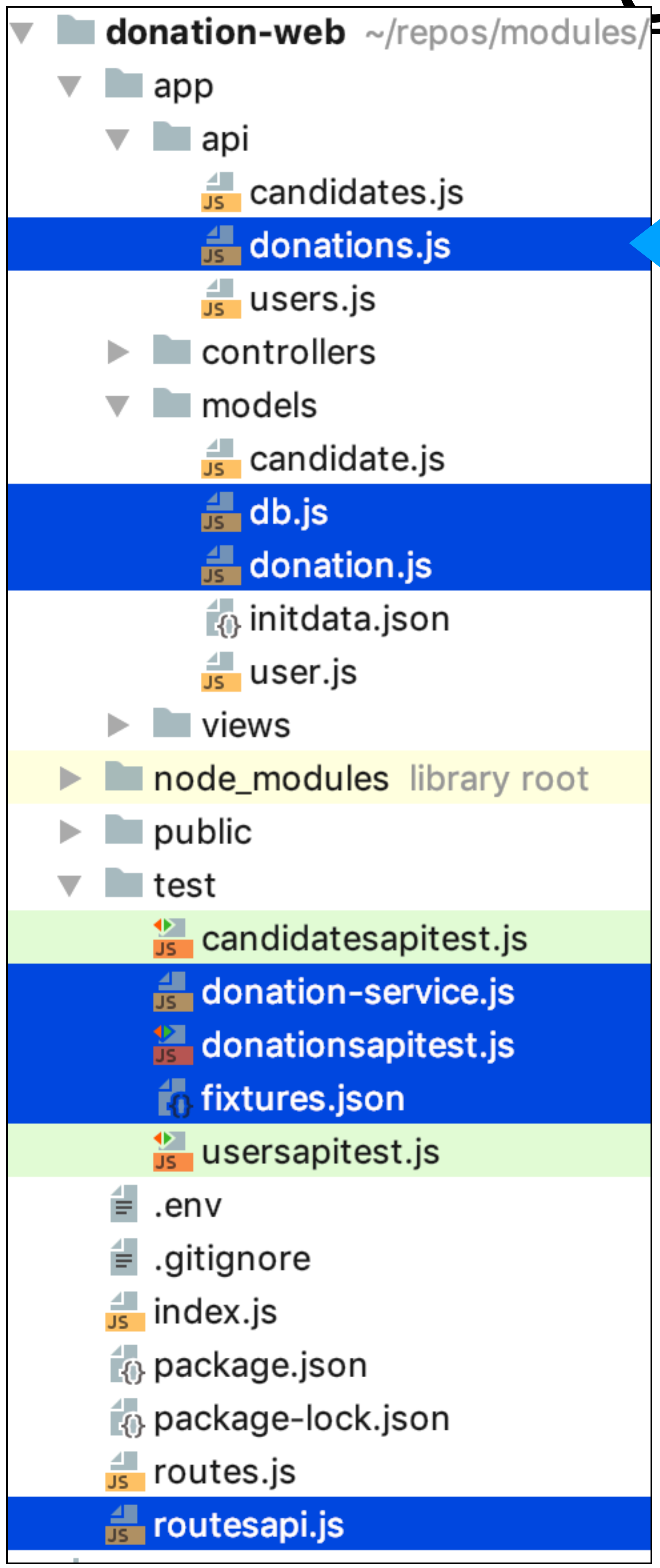
```
{ method: 'GET', path: '/api/donations', config: Donations.findAll },
{ method: 'GET', path: '/api/candidates/{id}/donations', config: Donations.findByCandidate },
{ method: 'POST', path: '/api/candidates/{id}/donations', config: Donations.makeDonation },
{ method: 'DELETE', path: '/api/donations', config: Donations.deleteAll }
```

- All donations recorded for all Candidates
- All Donations for a single candidate
- Make a donation to a candidate
- Delete all donations



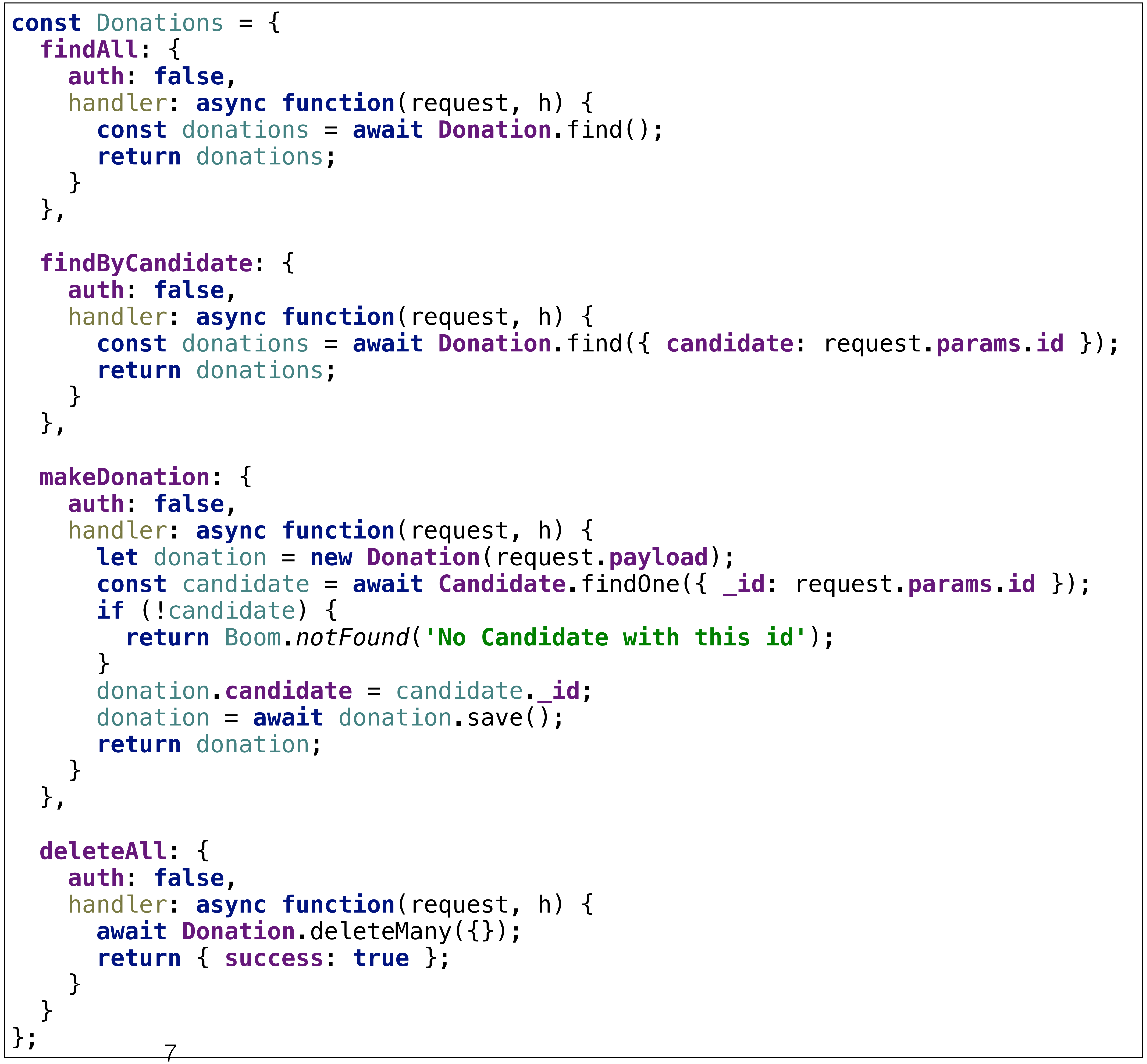
```
▼ donation-web ~/repos/modules/
  ▼ app
    ▼ api
        candidates.js
        donations.js
        users.js
    ▶ controllers
    ▼ models
        candidate.js
        db.js
        donation.js
        initdata.json
        user.js
    ▶ views
  ▶ node_modules library root
  ▶ public
  ▼ test
      candidatesapitest.js
      donation-service.js
      donationsapitest.js
      fixtures.json
      usersapitest.js
    .env
    .gitignore
    index.js
    package.json
    package-lock.json
    routes.js
    routesapi.js
```

Controller



Donations Endpoint

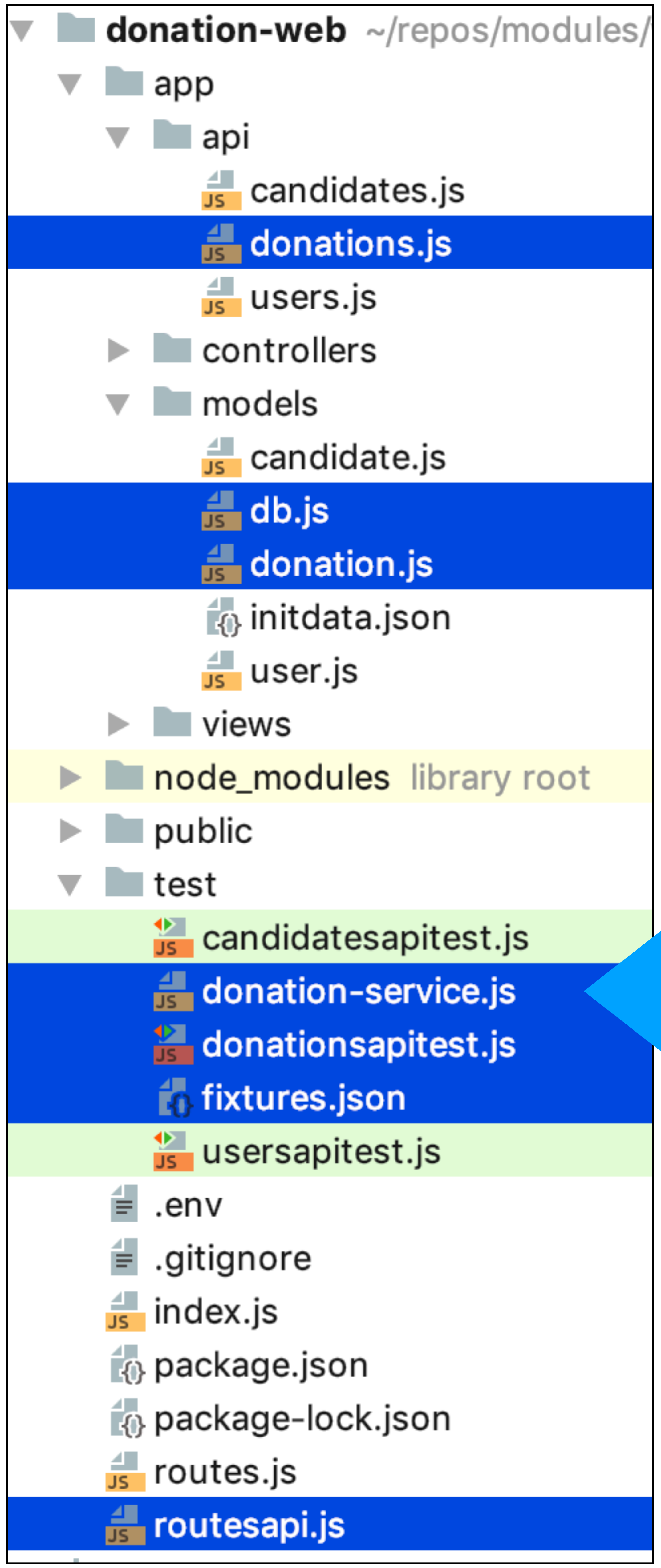
- findAll
- findByCandidate
- makeDonation
- deleteAll



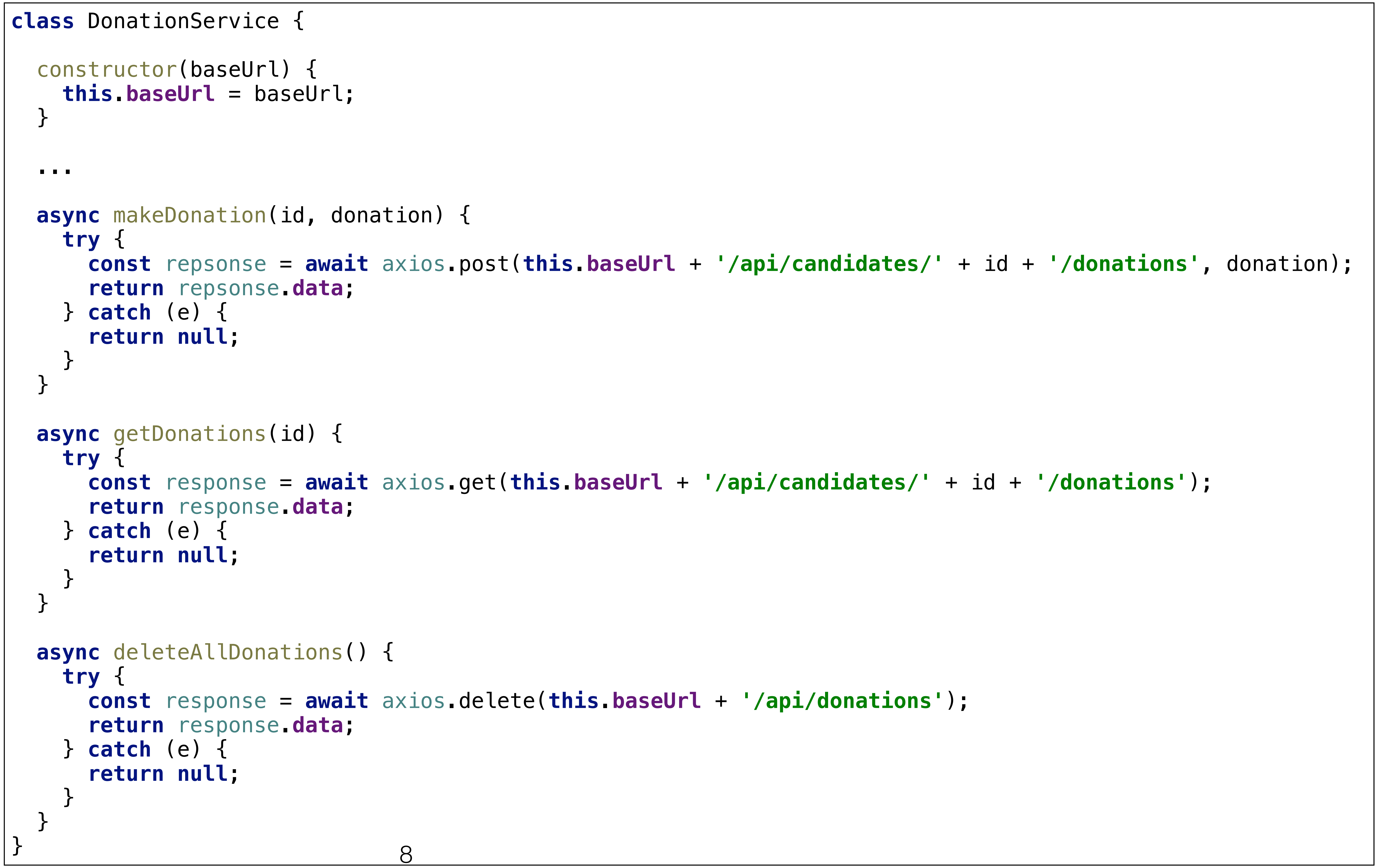
```
const Donations = {
  findAll: {
    auth: false,
    handler: async function(request, h) {
      const donations = await Donation.find();
      return donations;
    }
  },

  findByCandidate: {
    auth: false,
    handler: async function(request, h) {
      const donations = await Donation.find({ candidate: request.params.id });
      return donations;
    }
  },

  makeDonation: {
    auth: false,
    handler: async function(request, h) {
      let donation = new Donation(request.payload);
      const candidate = await Candidate.findOne({ _id: request.params.id });
      if (!candidate) {
        return Boom.notFound('No Candidate with this id');
      }
      donation.candidate = candidate._id;
      donation = await donation.save();
      return donation;
    }
  },

  deleteAll: {
    auth: false,
    handler: async function(request, h) {
      await Donation.deleteMany({});
      return { success: true };
    }
  }
};
```

DonationService

- Class to encapsulate client side access to the donation service
- Simplifies unit tests



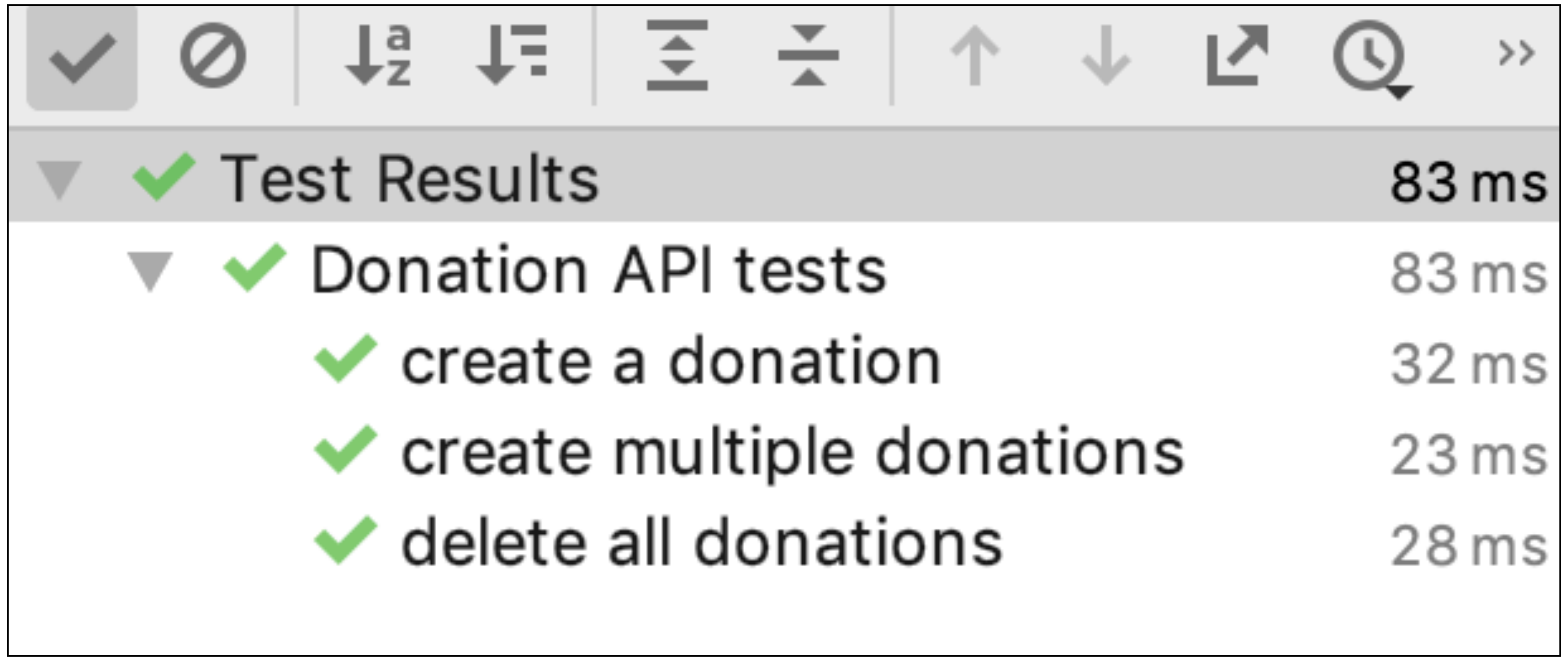
```
▼ donation-web ~/repos/modules/
  ▼ app
    ▼ api
        candidates.js
        donations.js
        users.js
    ▶ controllers
    ▼ models
        candidate.js
        db.js
        donation.js
        initdata.json
        user.js
    ▶ views
  ▶ node_modules library root
  ▶ public
  ▼ test
      candidatesapitest.js
      donation-service.js
      donationsapitest.js
      fixtures.json
      usersapitest.js
    .env
    .gitignore
    index.js
    package.json
    package-lock.json
    routes.js
    routesapi.js
```



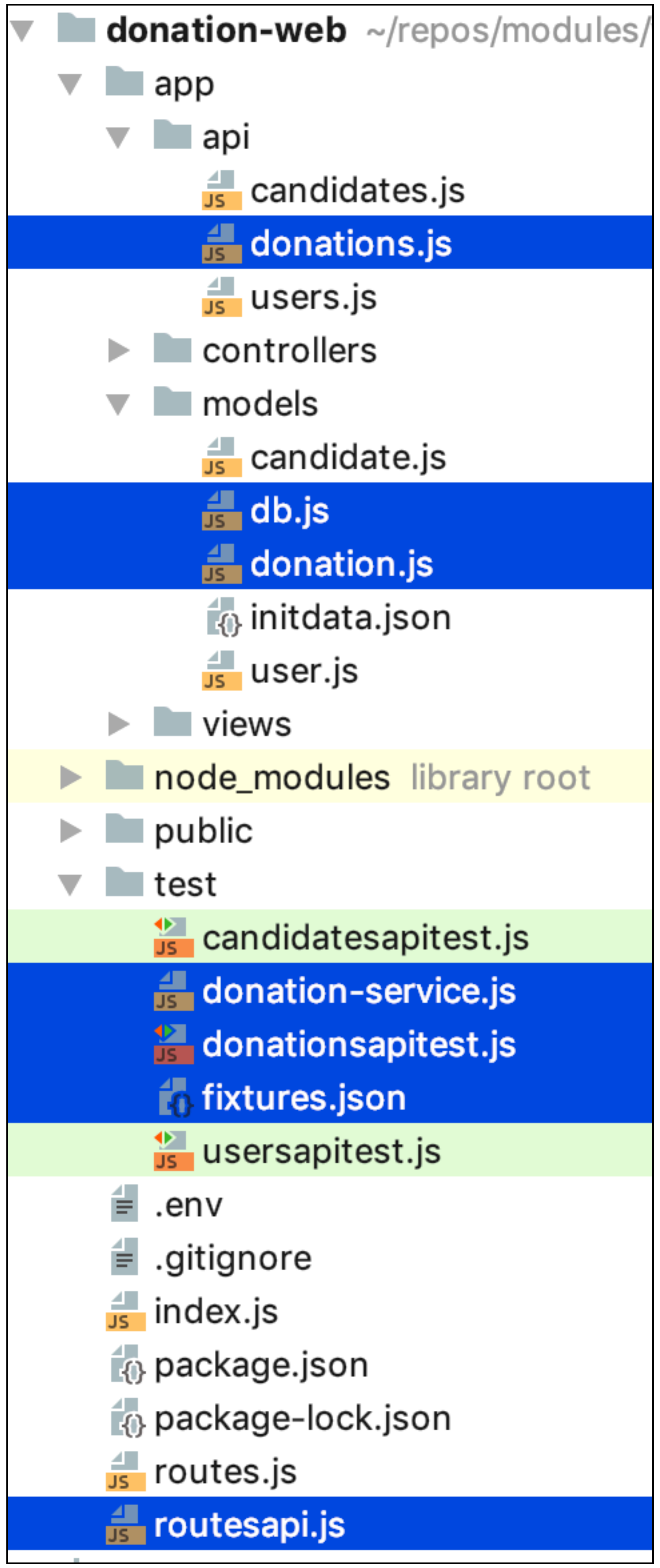
```
class DonationService {

  constructor(baseUrl) {
    this.baseUrl = baseUrl;
  }

  ...

  async makeDonation(id, donation) {
    try {
      const repsonse = await axios.post(this.baseUrl + '/api/candidates/' + id + '/donations', donation);
      return repsonse.data;
    } catch (e) {
      return null;
    }
  }

  async getDonations(id) {
    try {
      const response = await axios.get(this.baseUrl + '/api/candidates/' + id + '/donations');
      return response.data;
    } catch (e) {
      return null;
    }
  }

  async deleteAllDonations() {
    try {
      const response = await axios.delete(this.baseUrl + '/api/donations');
      return response.data;
    } catch (e) {
      return null;
    }
  }
}
```

Unit Tests

- Unit tests to verify the users endpoint
- Exercise all aspects of the endpoint.



```
▼ ✓ Test Results                      83 ms
  ▼ ✓ Donation API tests              83 ms
      ✓ create a donation             32 ms
      ✓ create multiple donations     23 ms
      ✓ delete all donations          28 ms
```



```
▼ donation-web ~/repos/modules/
  ▼ app
    ▼ api
        candidates.js
        donations.js
        users.js
    ▶ controllers
    ▼ models
        candidate.js
        db.js
        donation.js
        initdata.json
        user.js
    ▶ views
  ▶ node_modules library root
  ▶ public
  ▼ test
      candidatesapitest.js
      donation-service.js
      donationsapitest.js
      fixtures.json
      usersapitest.js
    .env
    .gitignore
    index.js
    package.json
    package-lock.json
    routes.js
    routesapi.js
```

Unit Tests

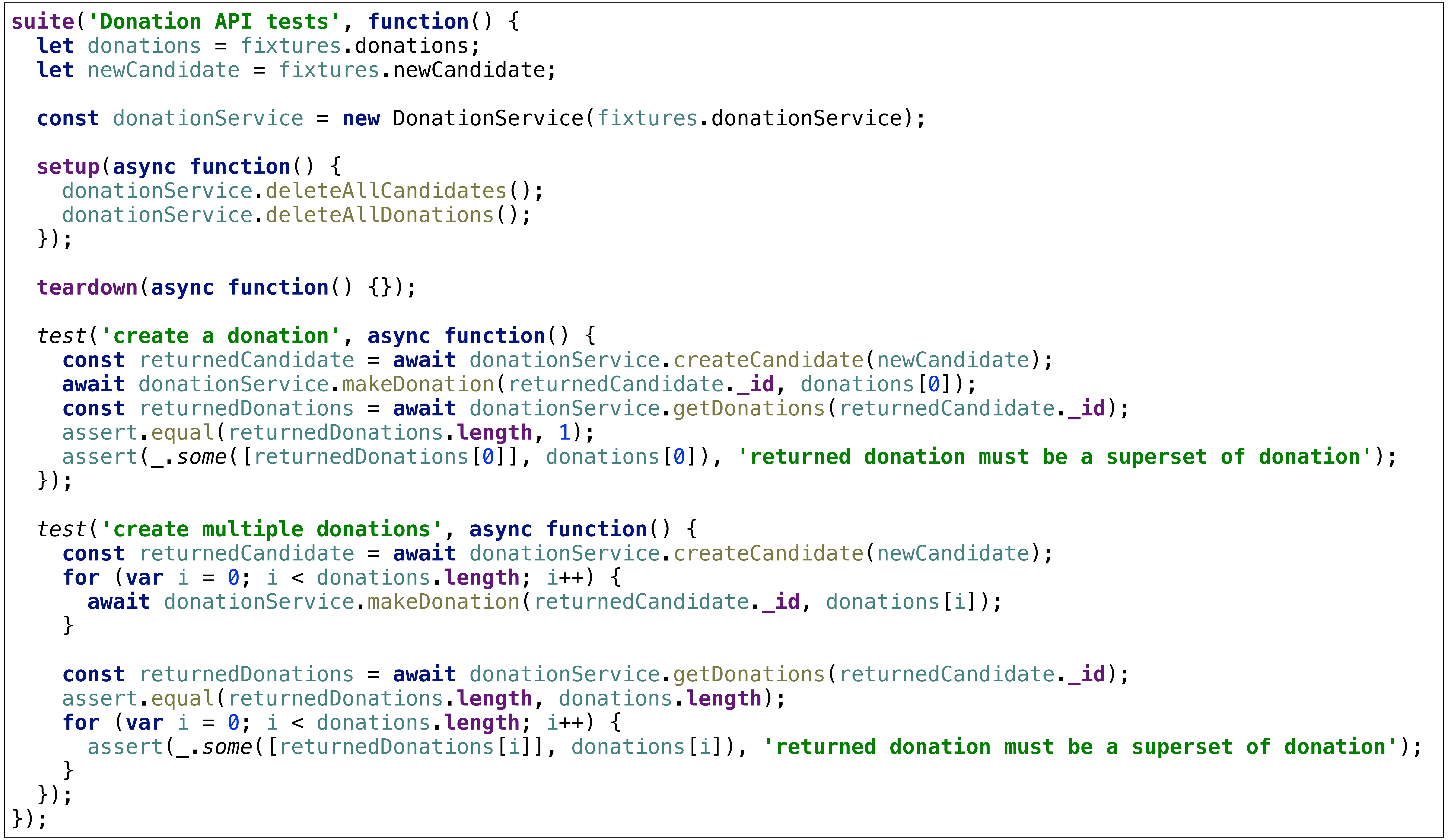
- Create donationService interface (with service url loaded from fixture)

- Remove all donations & candidates

- Tests:

  - Create a Donation

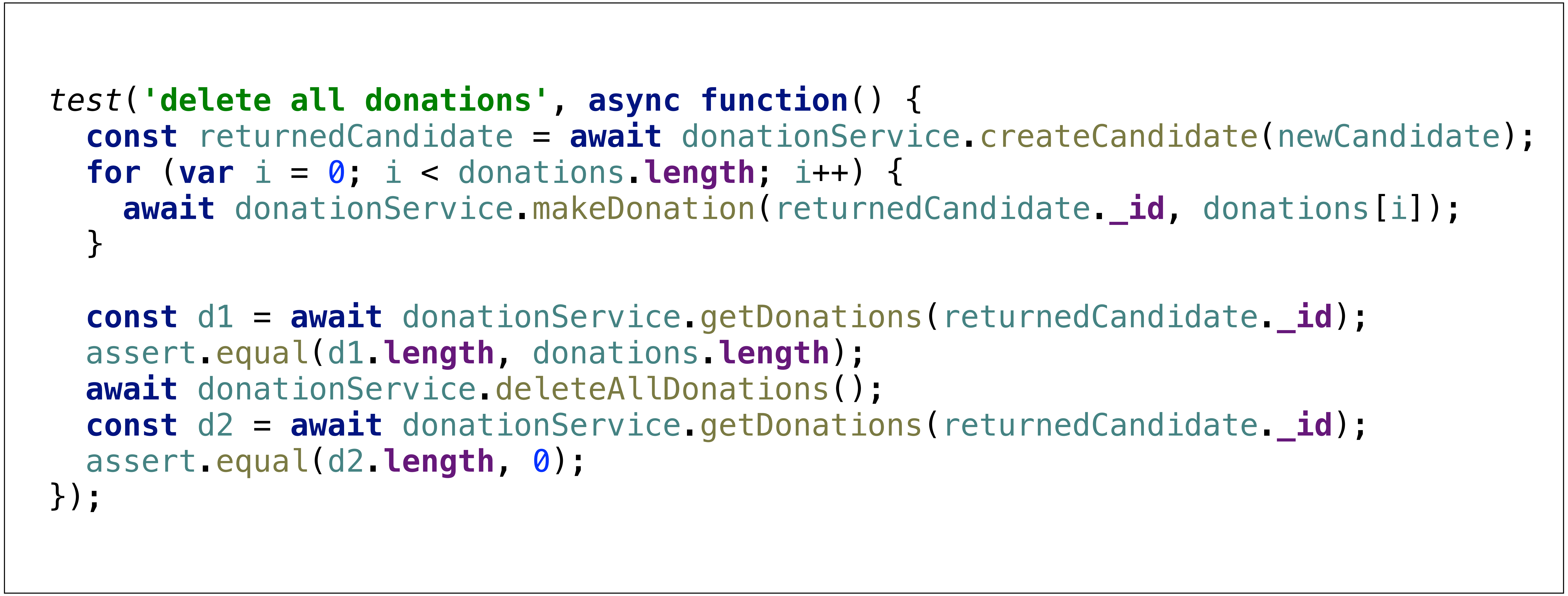
  - Create Multiple Donations



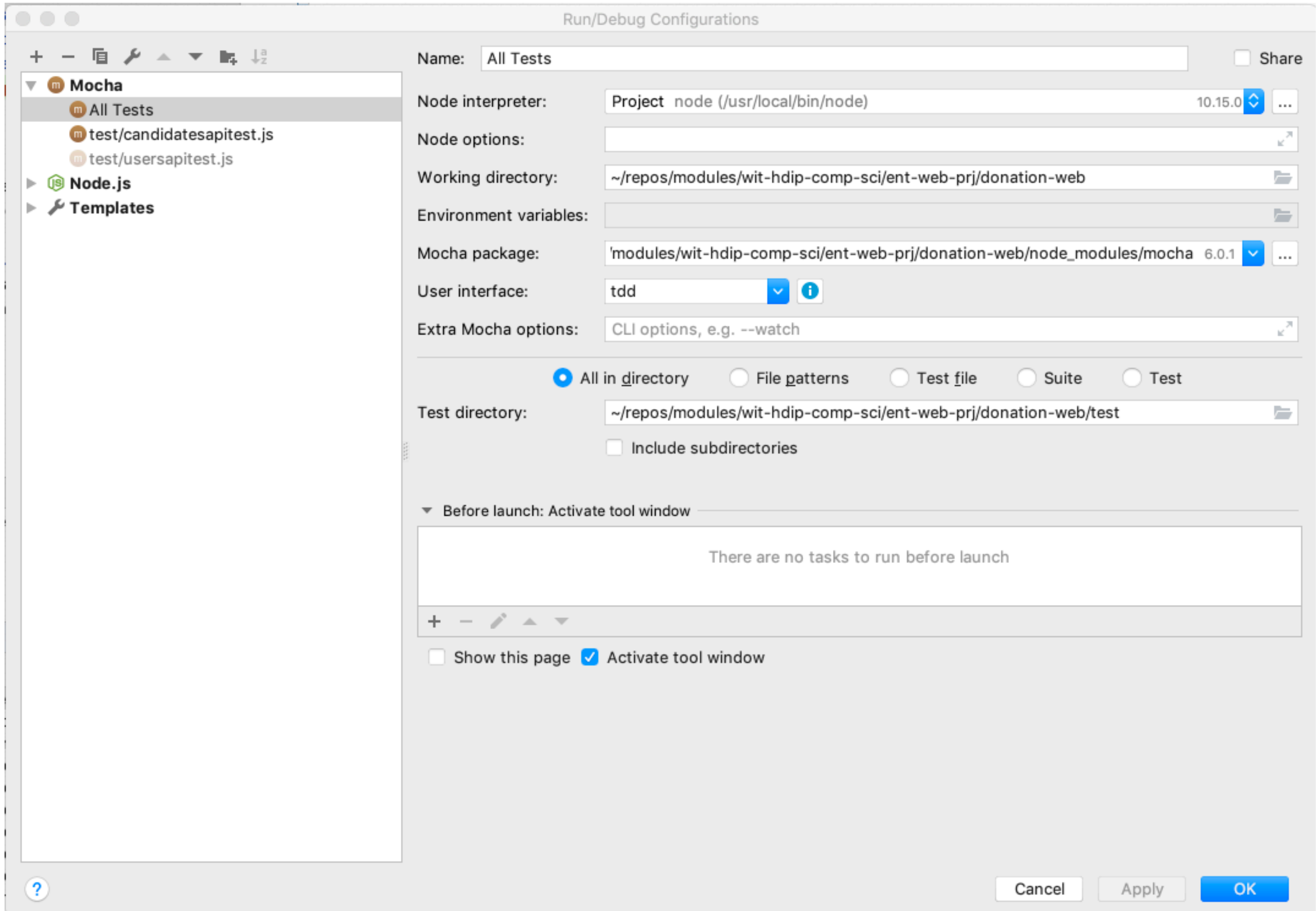
```
suite('Donation API tests', function() {
  let donations = fixtures.donations;
  let newCandidate = fixtures.newCandidate;

  const donationService = new DonationService(fixtures.donationService);

  setup(async function() {
    donationService.deleteAllCandidates();
    donationService.deleteAllDonations();
  });

  teardown(async function() {});

  test('create a donation', async function() {
    const returnedCandidate = await donationService.createCandidate(newCandidate);
    await donationService.makeDonation(returnedCandidate._id, donations[0]);
    const returnedDonations = await donationService.getDonations(returnedCandidate._id);
    assert.equal(returnedDonations.length, 1);
    assert(_.some([returnedDonations[0]], donations[0]), 'returned donation must be a superset of donation');
  });

  test('create multiple donations', async function() {
    const returnedCandidate = await donationService.createCandidate(newCandidate);
    for (var i = 0; i < donations.length; i++) {
      await donationService.makeDonation(returnedCandidate._id, donations[i]);
    }

    const returnedDonations = await donationService.getDonations(returnedCandidate._id);
    assert.equal(returnedDonations.length, donations.length);
    for (var i = 0; i < donations.length; i++) {
      assert(_.some([returnedDonations[i]], donations[i]), 'returned donation must be a superset of donation');
    }
  });
});
```

Unit Tests

- Delete Donations



```
test('delete all donations', async function() {
  const returnedCandidate = await donationService.createCandidate(newCandidate);
  for (var i = 0; i < donations.length; i++) {
    await donationService.makeDonation(returnedCandidate._id, donations[i]);
  }

  const d1 = await donationService.getDonations(returnedCandidate._id);
  assert.equal(d1.length, donations.length);
  await donationService.deleteAllDonations();
  const d2 = await donationService.getDonations(returnedCandidate._id);
  assert.equal(d2.length, 0);
});
```



Run/Debug Configurations

Name: All Tests □ Share

Node interpreter: Project node (/usr/local/bin/node) 10.15.0

Node options:

Working directory: ~/repos/modules/wit-hdip-comp-sci/ent-web-prj/donation-web

Environment variables:

Mocha package: 'modules/wit-hdip-comp-sci/ent-web-prj/donation-web/node_modules/mocha 6.0.1

User interface: tdd

Extra Mocha options: CLI options, e.g. --watch

● All in directory ○ File patterns ○ Test file ○ Suite ○ Test

Test directory: ~/repos/modules/wit-hdip-comp-sci/ent-web-prj/donation-web/test

□ Include subdirectories

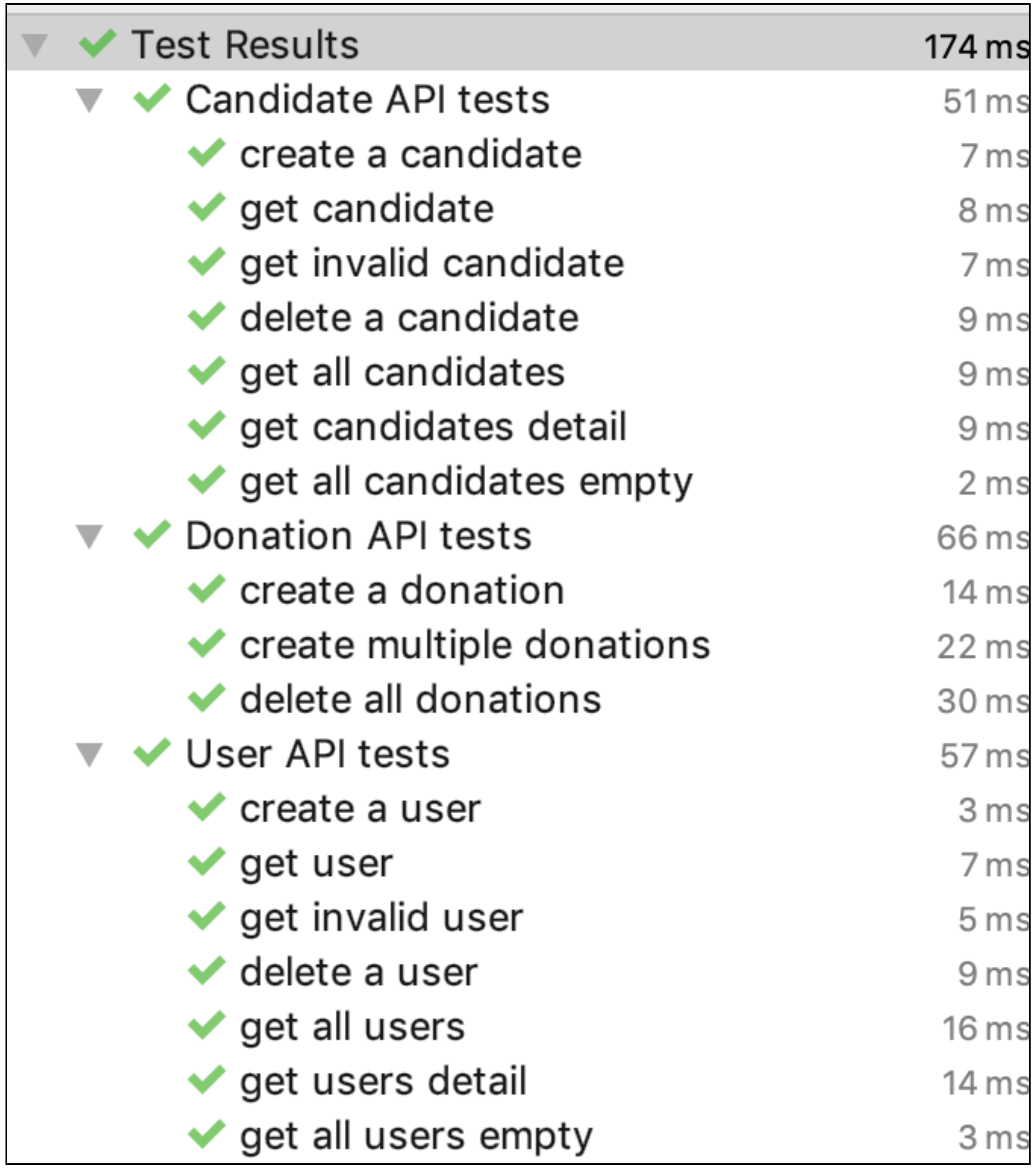
Before launch: Activate tool window

There are no tasks to run before launch

□ Show this page ☑ Activate tool window

- Comprehensive unit tests

- Run full suite



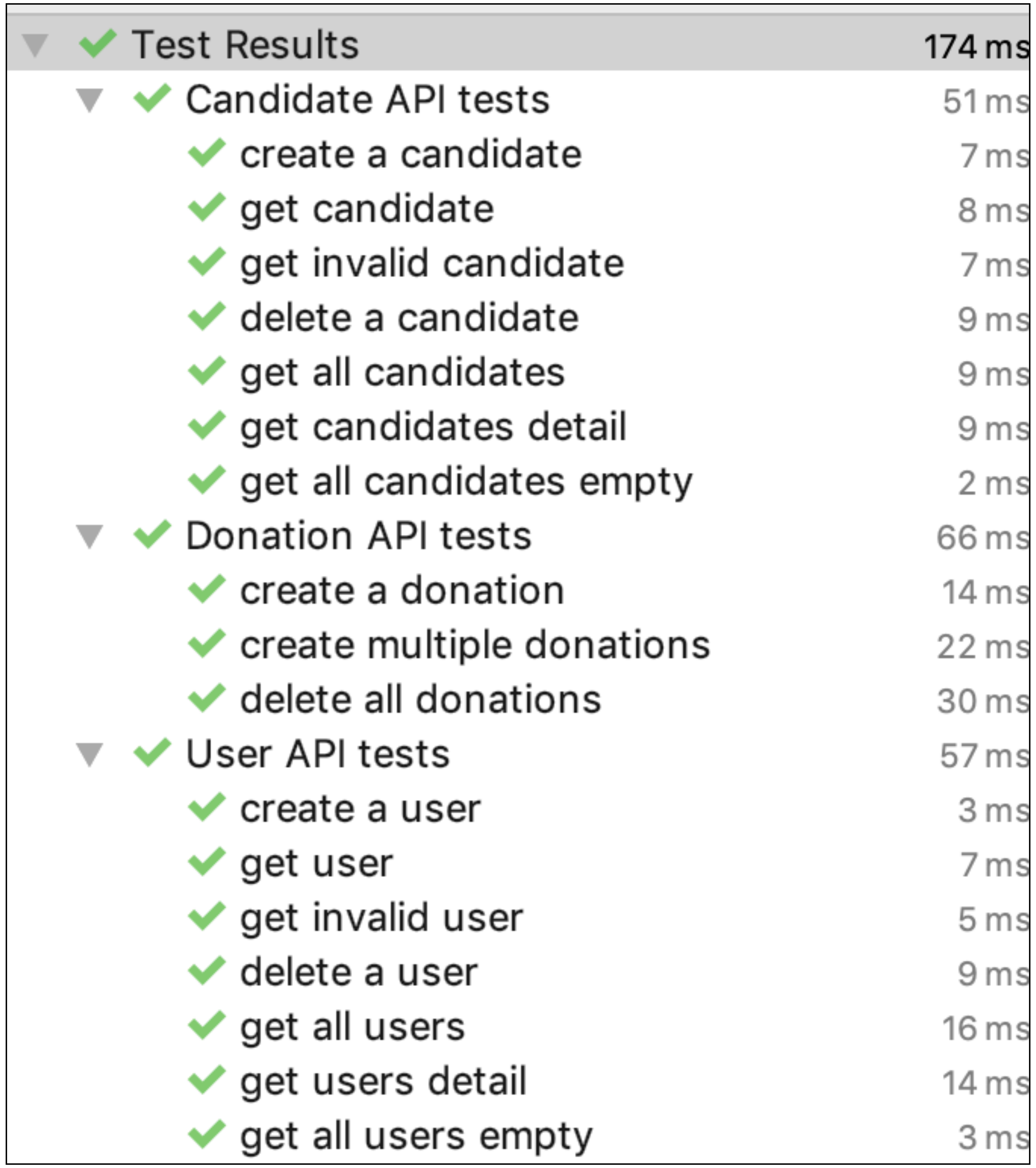
```
▼ ✓ Test Results                              174 ms
   ▼ ✓ Candidate API tests                     51 ms
        ✓ create a candidate                    7 ms
        ✓ get candidate                         8 ms
        ✓ get invalid candidate                 7 ms
        ✓ delete a candidate                    9 ms
        ✓ get all candidates                    9 ms
        ✓ get candidates detail                 9 ms
        ✓ get all candidates empty              2 ms
   ▼ ✓ Donation API tests                      66 ms
        ✓ create a donation                    14 ms
        ✓ create multiple donations            22 ms
        ✓ delete all donations                 30 ms
   ▼ ✓ User API tests                          57 ms
        ✓ create a user                         3 ms
        ✓ get user                              7 ms
        ✓ get invalid user                      5 ms
        ✓ delete a user                         9 ms
        ✓ get all users                        16 ms
        ✓ get users detail                     14 ms
        ✓ get all users empty                   3 ms
```

Regression Tests

- Powerful capability to comprehensively test the API

- Further development/ enhancement of the API not has large range of regression tests

- These will keep the API development stable as new endpoints are introduced



```
▼ ✓ Test Results                          174 ms
    ▼ ✓ Candidate API tests                51 ms
          ✓ create a candidate              7 ms
          ✓ get candidate                   8 ms
          ✓ get invalid candidate           7 ms
          ✓ delete a candidate              9 ms
          ✓ get all candidates              9 ms
          ✓ get candidates detail           9 ms
          ✓ get all candidates empty        2 ms
    ▼ ✓ Donation API tests                 66 ms
          ✓ create a donation              14 ms
          ✓ create multiple donations      22 ms
          ✓ delete all donations           30 ms
    ▼ ✓ User API tests                     57 ms
          ✓ create a user                   3 ms
          ✓ get user                        7 ms
          ✓ get invalid user                5 ms
          ✓ delete a user                   9 ms
          ✓ get all users                  16 ms
          ✓ get users detail               14 ms
          ✓ get all users empty             3 ms
```

userstest
fixtures.json
donationstest
DonationService
candidatestest
users endpoint
donations endpoint
candidates endpoint
api
Users
Donations
Candidates
models
UserSchema
DonationSchema
CandidateSchema
db
initdata.json
MongoDb