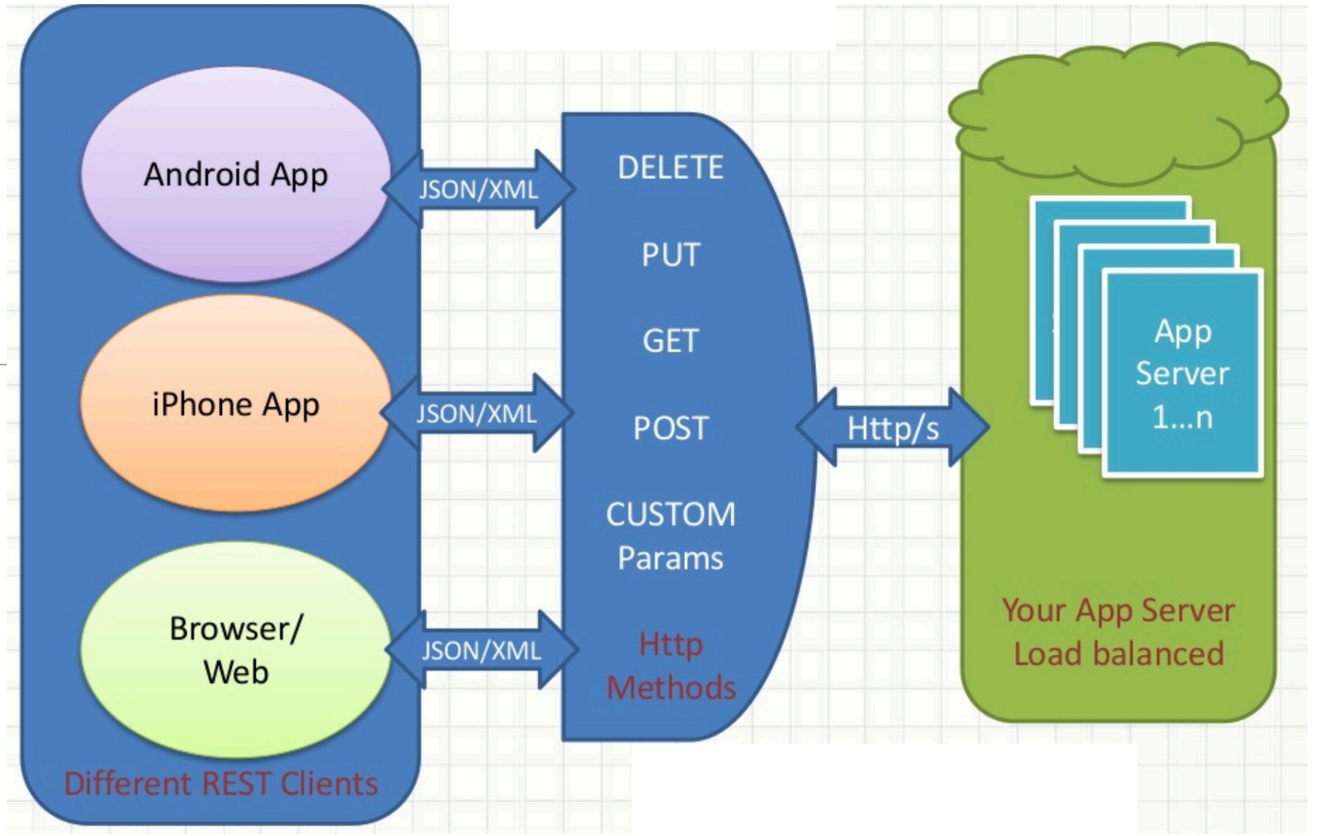
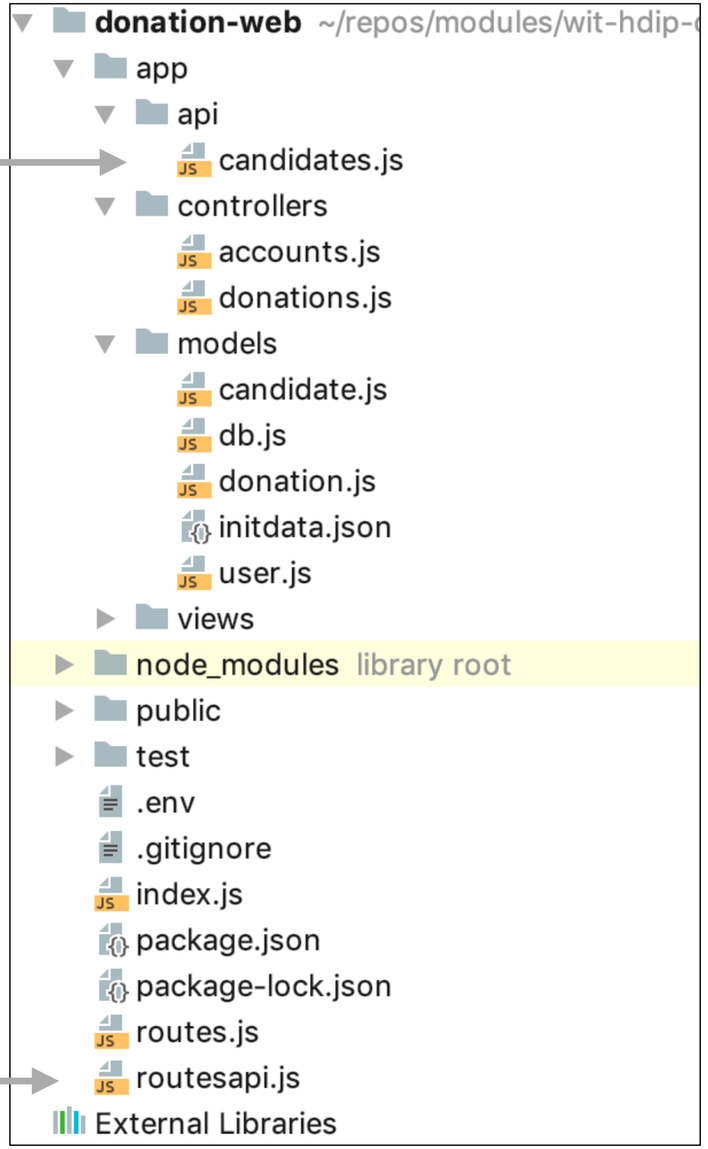
Endpoints


Android App
iPhone App
Browser/
Web
Different REST Clients
JSON/XML
JSON/XML
JSON/XML
DELETE
PUT
GET
POST
CUSTOM
Params
Http
Methods
Http/s
App
Server
1...n
Your App Server
Load balanced

Project Structure

Create ‘api’ folder to expose endpoints from the application

Create ‘routesapi.js’ to define these endpoints



Get Candidates Endpoint

routeapi.js

```
const CandidatesApi = require('./app/api/candidates');

module.exports = [
  { method: 'GET', path: '/api/candidates', config: Candidates.find },
];
```

candidates.js

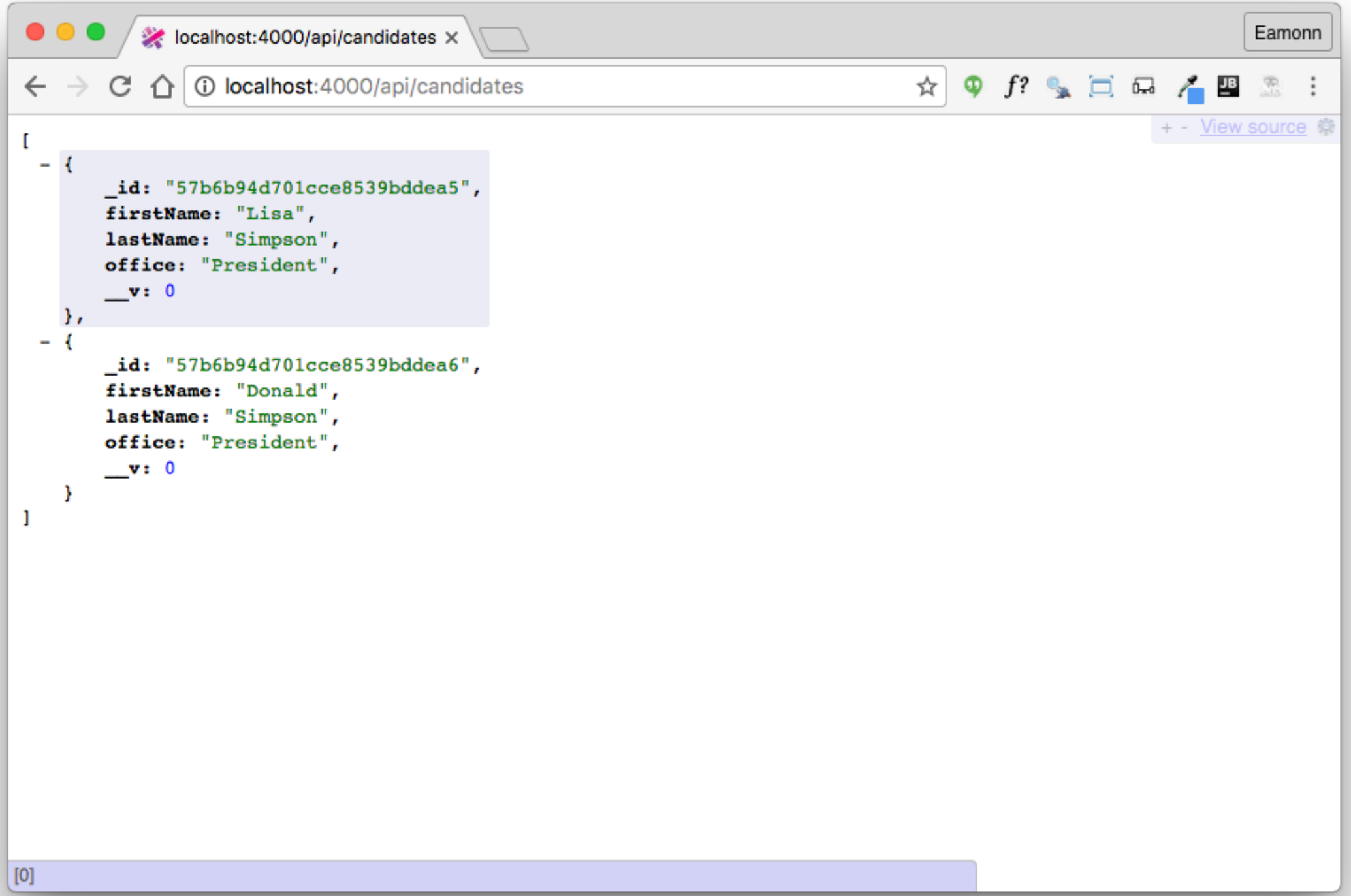
```
const Candidate = require('../models/candidate');

const Candidates = {

  find: {
    auth: false,
    handler: async function(request, h) {
      const candidates = await Candidate.find();
      return candidates;
    }
  },
```

index.js

```
server.route(require('./routes'));
server.route(require('./routesapi'));
await server.start();
```


localhost:4000/api/candidates
Eamonn
localhost:4000/api/candidates
+ - View source
[
- {
"_id": "57b6b94d701cce8539bddea5",
"firstName": "Lisa",
"lastName": "Simpson",
"office": "President",
"__v": 0
},
- {
"_id": "57b6b94d701cce8539bddea6",
"firstName": "Donald",
"lastName": "Simpson",
"office": "President",
"__v": 0
}
]
[0]

https://github.com/hapijs/boom

boom provides a set of utilities for returning HTTP errors. Each utility returns a `Boom` error response object (instance of `Error`) which includes the following properties:

- `isBoom` - if `true`, indicates this is a `Boom` object instance.
- `isServer` - convenience bool indicating status code >= 500.
- `message` - the error message.
- `output` - the formatted response. Can be directly manipulated after object construction to return a custom error response. Allowed root keys:
  - `statusCode` - the HTTP status code (typically 4xx or 5xx).
  - `headers` - an object containing any HTTP headers where each key is a header name and value is the header content.
  - `payload` - the formatted object used as the response payload (stringified). Can be directly manipulated but any changes will be lost if `reformat()` is called. Any content allowed and by default includes the following content:
    - `statusCode` - the HTTP status code, derived from `error.output.statusCode`.
    - `error` - the HTTP status message (e.g. 'Bad Request', 'Internal Server Error') derived from `statusCode`.
    - `message` - the error message derived from `error.message`.
- inherited `Error` properties.

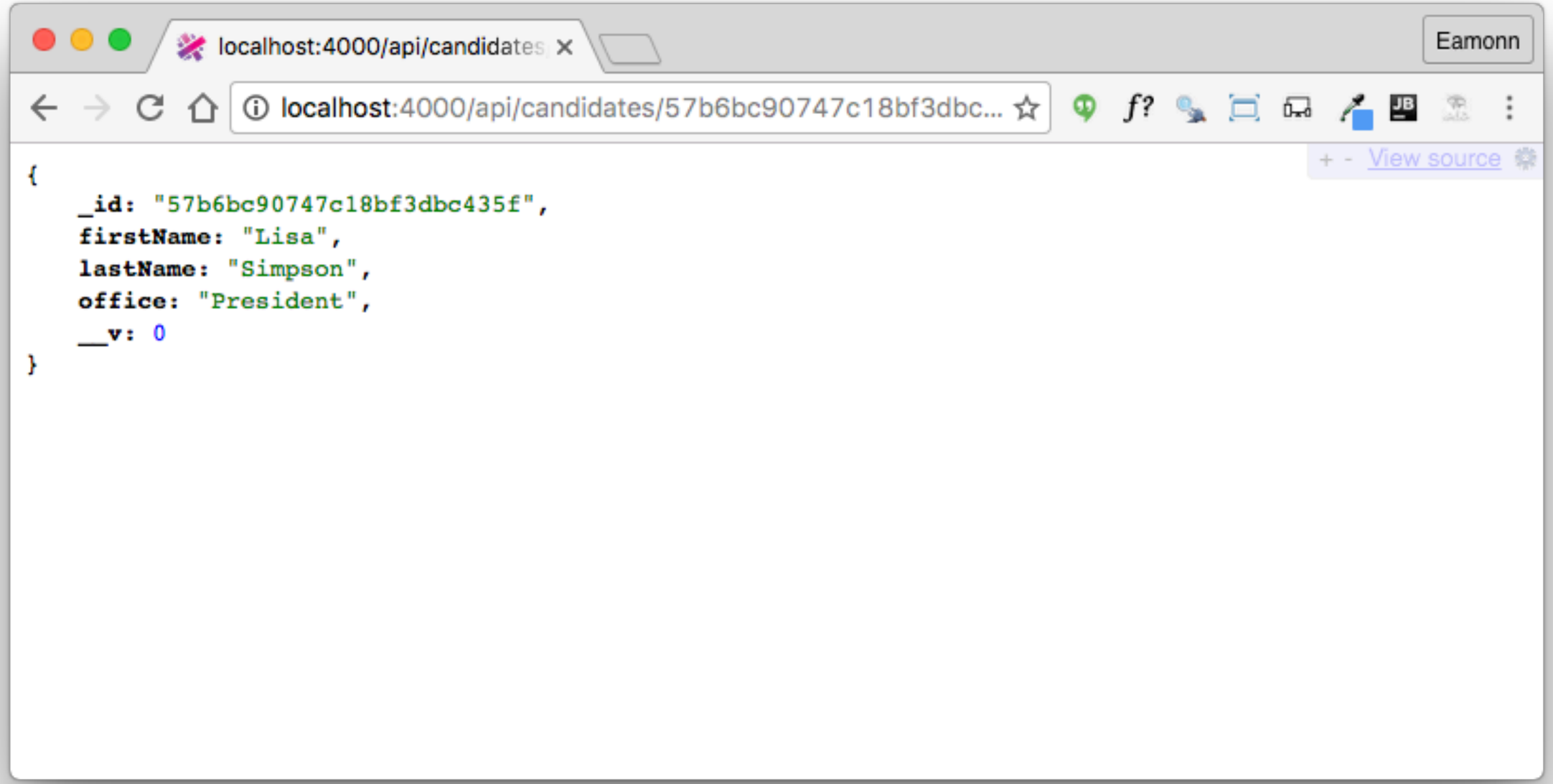
Get Candidate (singular) Endpoint

```
{ method: 'GET', path: '/api/candidates/{id}', config: Candidates.findOne },
```

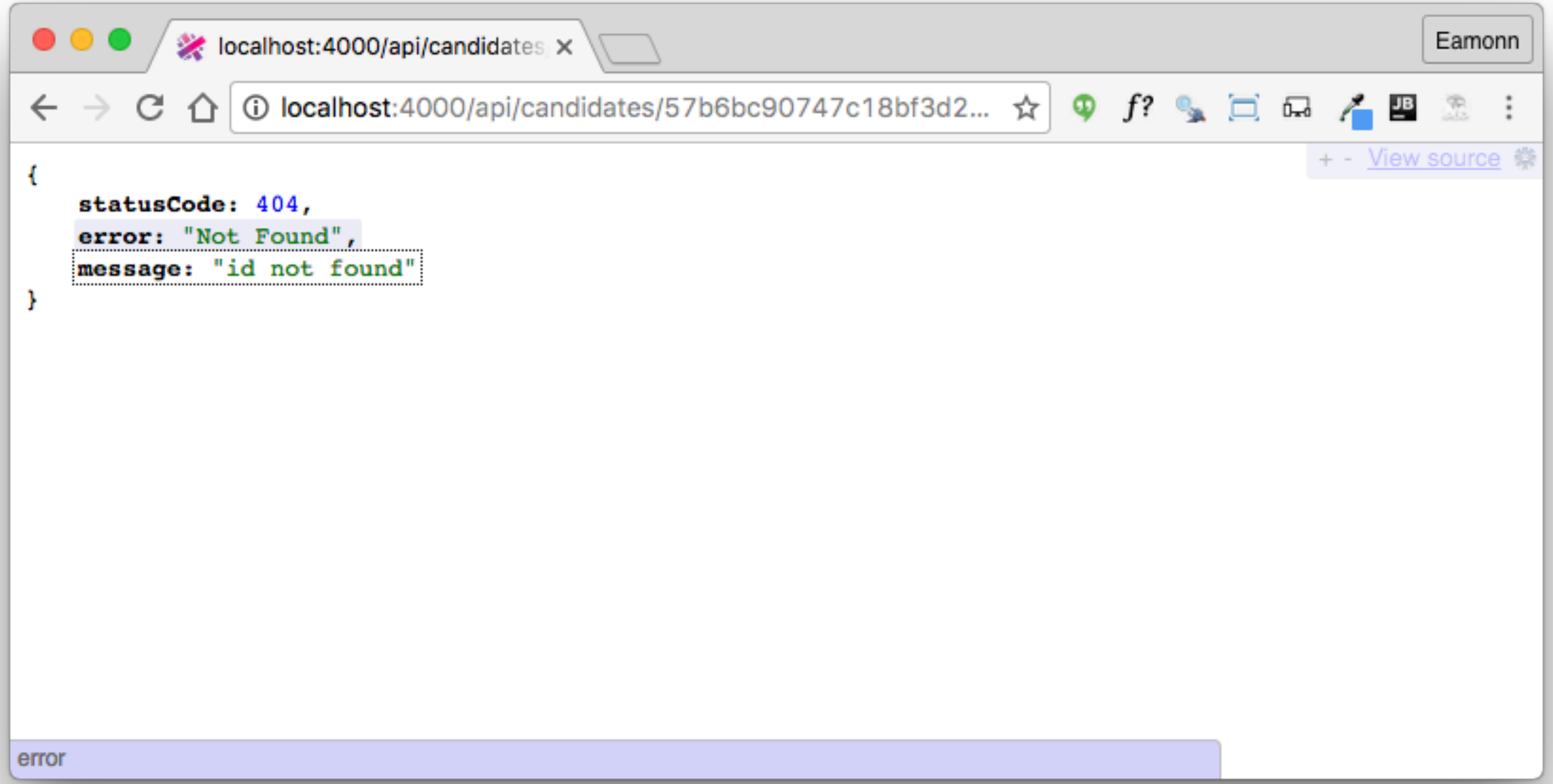
- Object ID passed as part of path

```
findOne: {
  auth: false,
  handler: async function(request, h) {
    try {
      const candidate = await Candidate.findOne({ _id: request.params.id });
      if (!candidate) {
        return Boom.notFound('No Candidate with this id');
      }
      return candidate;
    } catch (err) {
      return Boom.notFound('No Candidate with this id');
    }
  }
},
```

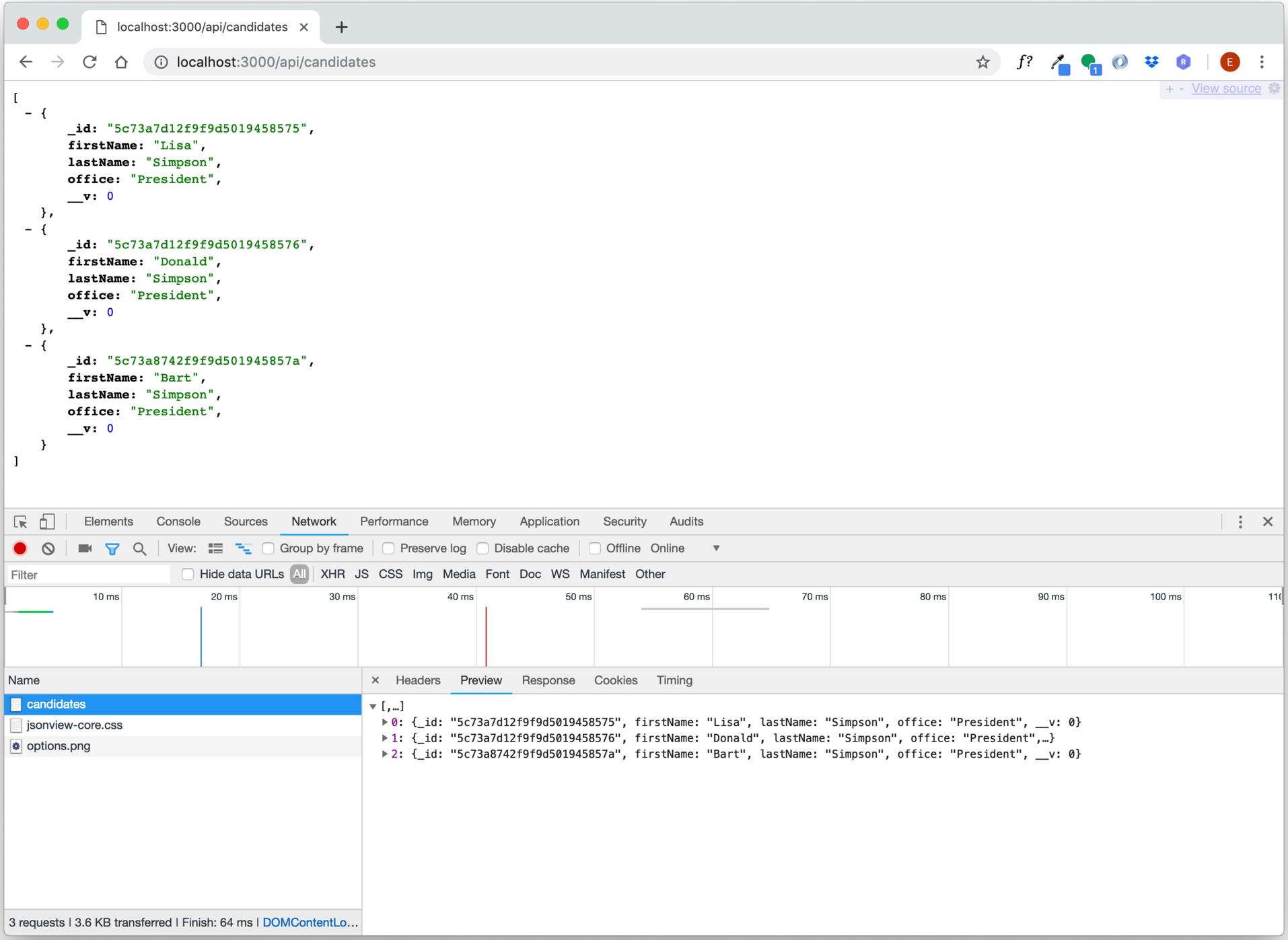
Get Candidate (singular) Endpoint



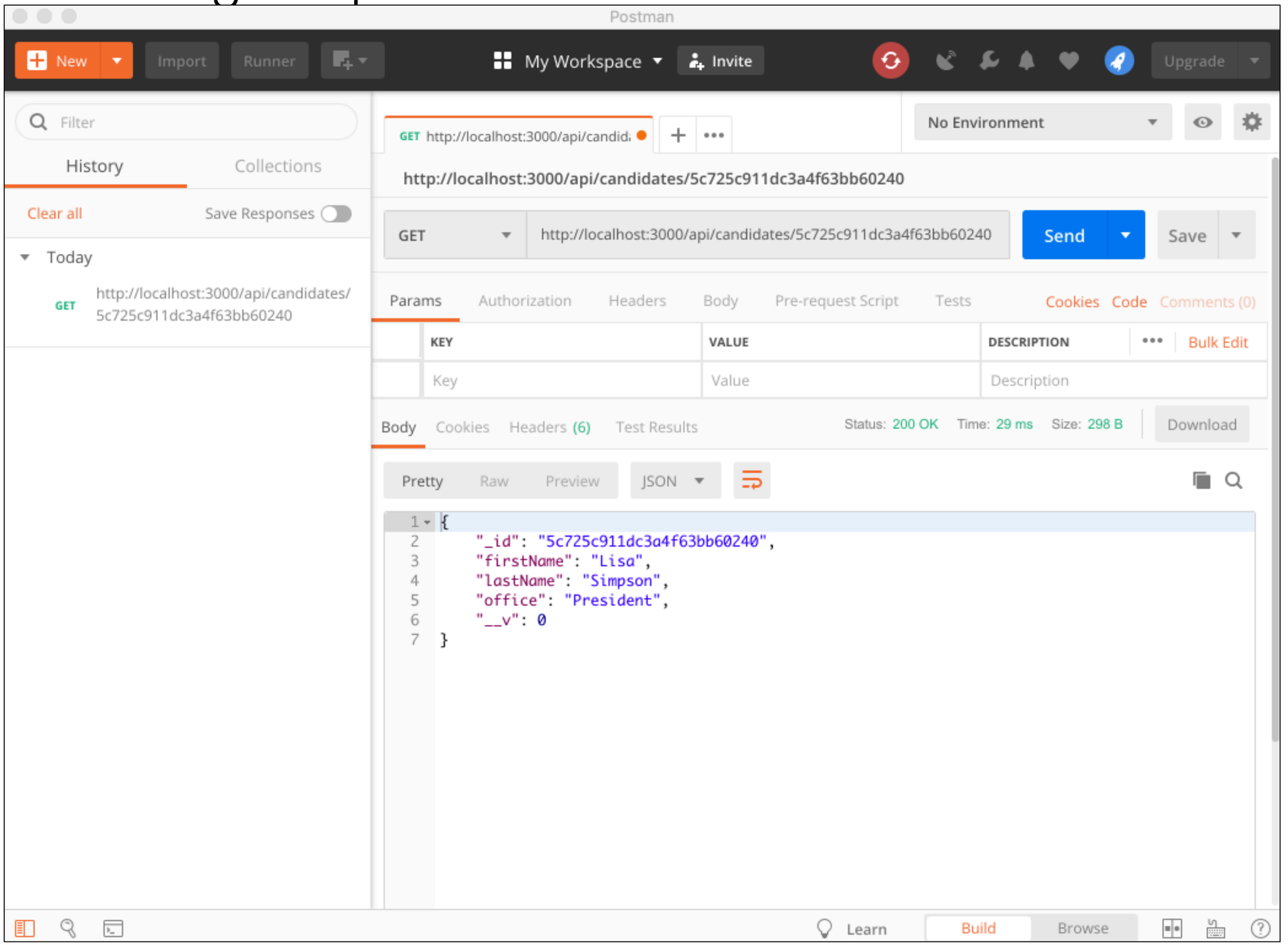
Boom Error Messages



Exercising Endpoints: Chrome



Exercising Endpoints: Postman



More Candidate Routes

```
const Candidates = require('./app/api/candidates');

module.exports = [
  { method: 'GET', path: '/api/candidates', config: Candidates.find },
  { method: 'GET', path: '/api/candidates/{id}', config: Candidates.findOne },
  { method: 'POST', path: '/api/candidates', config: Candidates.create },
  { method: 'DELETE', path: '/api/candidates/{id}', config: Candidates.deleteOne },
  { method: 'DELETE', path: '/api/candidates', config: Candidates.deleteAll },
];
```

- Get all Candidates
- Get One Candidate
- Create One Candidate
- Delete One Candidate
- Delete All Candidates

create a Candidate

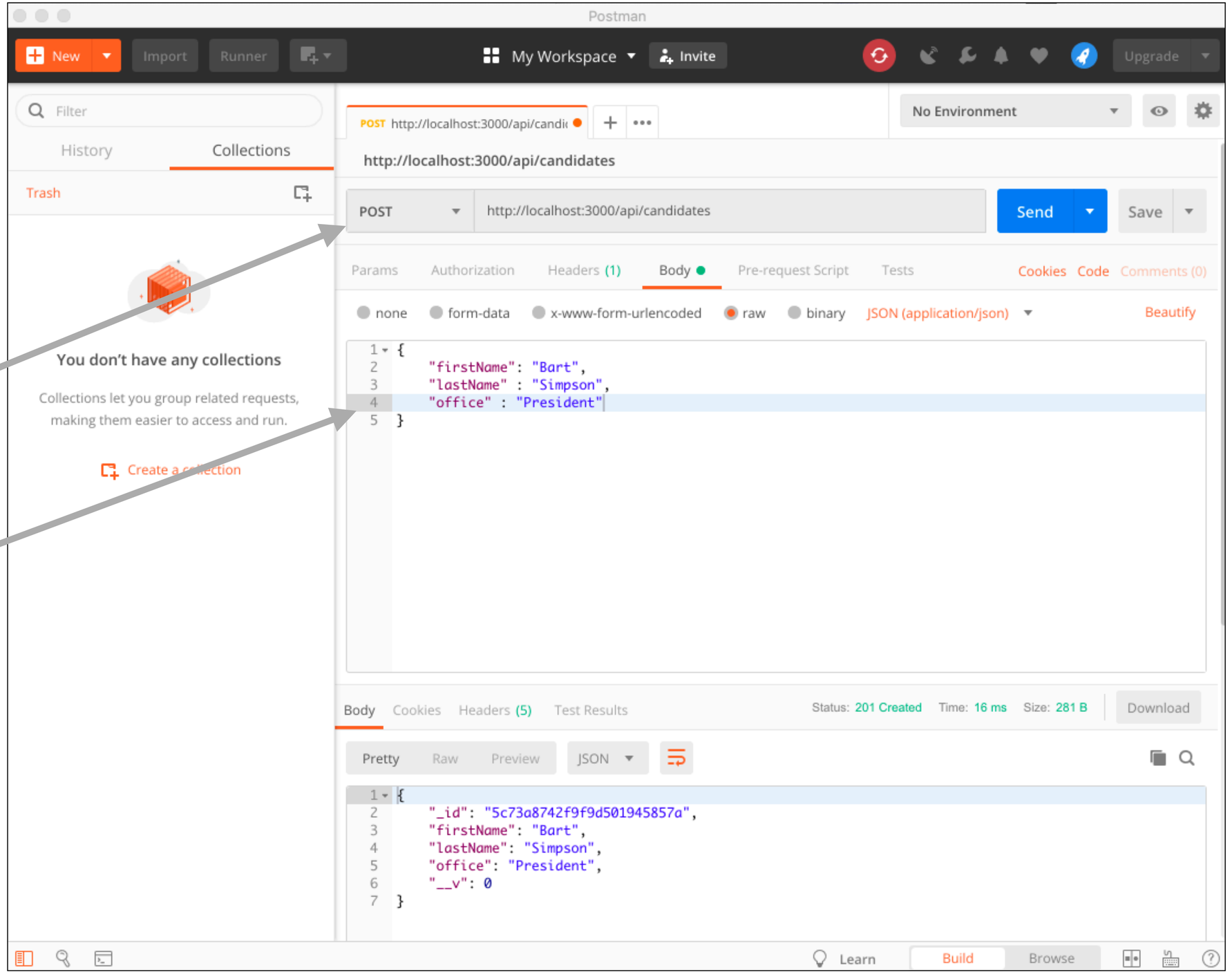
- POST instead of GET

```
{ method: 'POST', path: '/api/candidates', config: Candidates.create },
```

```
create: {
  auth: false,
  handler: async function(request, h) {
    const newCandidate = new Candidate(request.payload);
    const candidate = await newCandidate.save();
    if (candidate) {
      return h.response(candidate).code(201);
    }
    return Boom.badImplementation('error creating candidate');
  }
},
```

  - Create a new Candidate, and return it

- Browser cannot be used to conveniently send a ‘POST’ request.
- Use Postman instead.
- POST
- Json Body (candidate)



Delete a single candidate

```
{ method: 'DELETE', path: '/api/candidates/{id}', config: Candidates.deleteOne },
```

```
deleteOne: {
  auth: false,
  handler: async function(request, h) {
    const candidate = await Candidate.remove({ _id: request.params.id });
    if (candidate) {
      return { success: true };
    }
    return Boom.notFound('id not found');
  }
}
```

Delete all candidates

```
{ method: 'DELETE', path: '/api/candidates', config: Candidates.deleteAll },
```

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