Mobile Application Development

Higher Diploma in Science in Computer Science

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

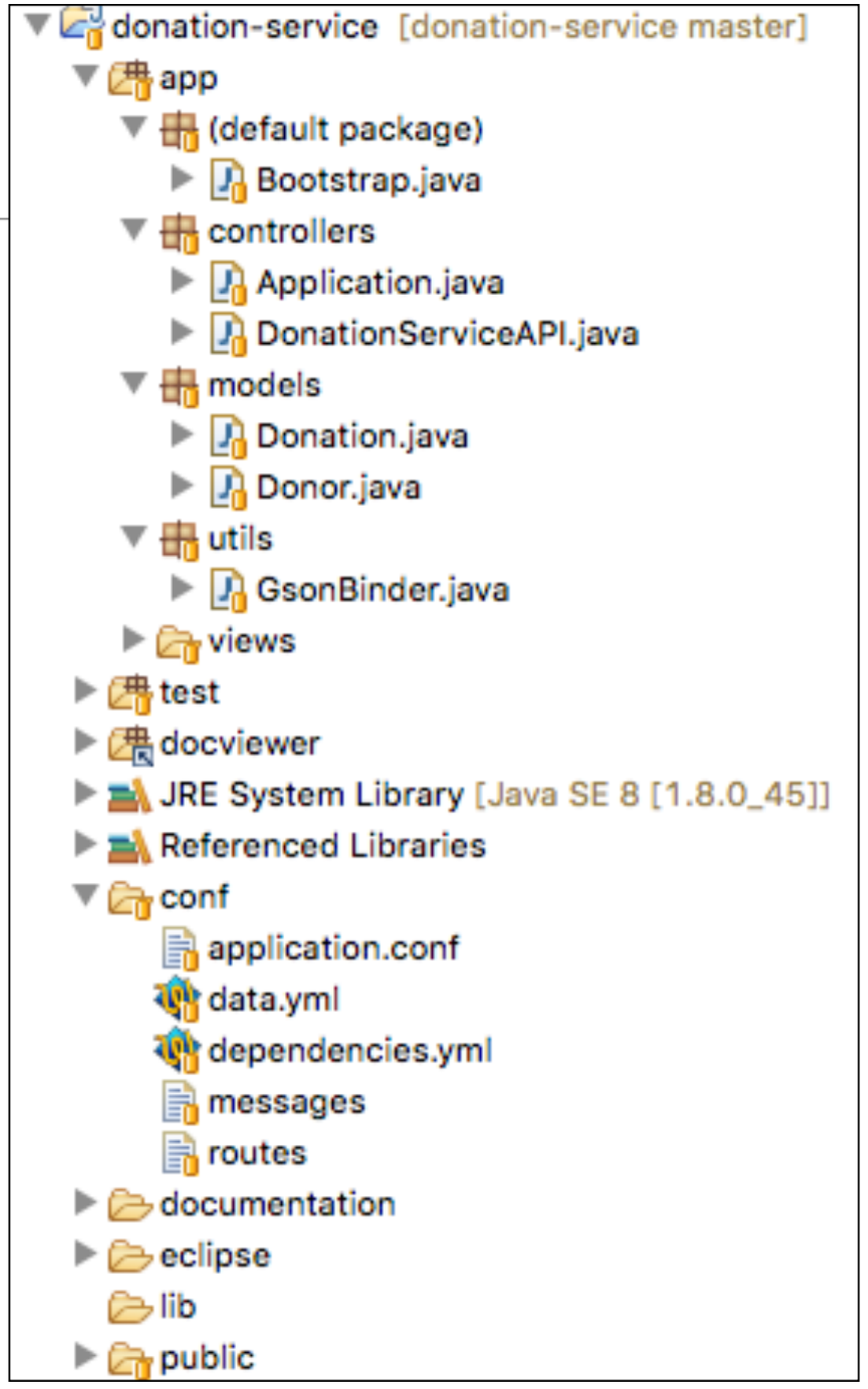
Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit

donation-service

donation-service Play App

- Rebuild simpler version of the donation play app:
  - No User Interface
  - No JPA relationships
- Includes:
  - Converters from Json to User/Donation and back



Models

```
@Entity
public class Donation extends Model
{
  public int    amount;
  public String method;

  public Donation (int amount, String method)
  {
    this.amount = amount;
    this.method = method;
  }

  public String toString()
  {
    return amount + ", " + method;
  }
}
```

```
@Entity
public class Donor extends Model
{
  public String firstName;
  public String lastName;
  public String email;
  public String password;

  public Donor(String firstName,
               String lastName,
               String email, String password)
  {
    this.firstName = firstName;
    this.lastName = lastName;
    this.email = email;
    this.password = password;
  }
}
```

Utils

```
@Global
public class GsonBinder implements TypeBinder<JsonElement>
{
  public Object bind(String name, Annotation[] notes, String value, Class toClass, Type toType) throws Exception
  {
    return new JsonParser().parse(value);
  }
}
```

- Utility class to enable controller actions to translate to/from Json objects.

JSON

http://en.wikipedia.org/wiki/JSON

JSON (/ˈdʒeɪsɒn/ ***jay-sawn***, /ˈdʒeɪsən/ ***jay-sun***), or **JavaScript Object Notation**, is a text-based open standard designed for human-readable data interchange. Derived from the JavaScript scripting language, JSON is a language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, JSON is language-independent, with parsers available for many languages.

The JSON format was originally specified by Douglas Crockford, and is described in RFC 4627. The official Internet media type for JSON is `application/json`. The JSON filename extension is `.json`.

The JSON format is often used for serializing and transmitting structured data over a network connection. It is used primarily to transmit data between a server and web application, serving as an alternative to XML.

```
{
  "name":"mocha",
  "shop":"costa",
  "rating":3.5,
  "price":2.0,
  "favourite":0,
  "id":1
},
{
  "name":"americano",
  "shop":"costa",
  "rating":4.5,
  "price":3.0,
  "favourite":1,
  "id":2
},
{
  "name":"cappuccino lite",
  "shop":"starbucks",
  "rating":1.5,
  "price":4.0,
  "favourite":1,
  "id":3
}
```

google-gson

Gson is a Java library that can be used to convert Java Objects into their JSON representation.

It can also be used to convert a JSON string to an equivalent Java object.

Gson can work with arbitrary Java objects including pre-existing objects that you do not have source-code of.

Goals

- Provide simple toJson() and fromJson() methods to convert Java objects to JSON and vice-versa
- Allow pre-existing unmodifiable objects to be converted to and from JSON
- Extensive support of Java Generics
- Allow custom representations for objects
- Support arbitrarily complex objects (with deep inheritance hierarchies and extensive use of generic types)

gson

Gson User Guide

Authors: Inderjeet Singh, Joel Leitch, Jesse Wilson

Contents

1 Overview
2 Goals for Gson
3 Gson Performance and Scalability
4 Gson Users
5 Using Gson
  5.1 Primitives Examples
  5.2 Object Examples
    5.2.1 Finer Points with Objects
  5.3 Nested Classes (including Inner Classes)
  5.4 Array Examples
  5.5 Collections Examples
    5.5.1 Collections Limitations
  5.6 Serializing and Deserializing Generic Types
  5.7 Serializing and Deserializing Collection with Objects of Arbitrary Types
  5.8 Built-in Serializers and Deserializers
  5.9 Custom Serialization and Deserialization
    5.9.1 Writing a Serializer
    5.9.2 Writing a Deserializer
  5.10 Writing an Instance Creator
    5.10.1 InstanceCreator for a Parameterized Type
  5.11 Compact Vs. Pretty Printing for JSON Output Format
  5.12 Null Object Support
  5.13 Versioning Support
  5.14 Excluding Fields From Serialization and Deserialization
    5.14.1 Java Modifier Exclusion
    5.14.2 Gson's @Expose
    5.14.3 User Defined Exclusion Strategies
  5.15 JSON Field Naming Support
  5.16 Sharing State Across Custom Serializers and Deserializers
  5.17 Streaming
6 Issues in Designing Gson
7 Future Enhancements to Gson

controllers

- Actions do not render any views
- Use ‘renderJSON’ to return json version of model objects
- Other responses include ‘notFound’ and ‘Ok’- which are HTTP response codes

```
public class DonationServiceAPI extends Controller
{
  static Gson gson = new Gson();

  public static void getAllDonors()
  {
    List<Donor> Donors = Donor.findAll();
    renderJSON(gson.toJson(Donors));
  }

  public static void getDonor(Long id)
  {
    Donor donor = Donor.findById(id);
    if (donor == null)
    {
      notFound();
    }
    else
    {
      renderJSON(gson.toJson(donor));
    }
  }

  public static void createDonor(JsonElement body)
  {
    Donor donor = gson.fromJson(body.toString(), Donor.class);
    donor.id = null;
    donor.save();
    renderJSON(gson.toJson(donor));
  }
...
```

controllers

```
public class DonationServiceAPI extends Controller
{
  ...

  public static void deleteDonor(Long id)
  {
    Donor donor = Donor.findById(id);
    if (donor == null)
    {
      notFound("No Doner with ID" + id);
    }
    else
    {
      donor.delete();
      renderJSON(gson.toJson(donor));
    }
  }

  public static void deleteAllDonors()
  {
    Donor.deleteAll();
    renderText("success");
  }
}
```

Boostrap

```
@OnApplicationStart
public class Bootstrap extends Job
{
  public void doJob()
  {
    if (User.count() == 0)
    {
     Fixtures.deleteDatabase();
     Fixtures.loadModels("data.yml");
    }
  }
}
```

```
User(homer):
    usaCitizen: true
    firstName: Homer
    lastName: Simpson
    email: homer@simpson.com
    password: secret

User(marge):
    usaCitizen: true
    firstName: Marge
    lastName: Simpson
    email: marge@simpson.com
    password: secret

User(lisa):
    usaCitizen: true
    firstName: Lisa
    lastName: Simpson
    email: lisa@simpson.com
    password: secret

User(bart):
    usaCitizen: true
    firstName: Bart
    lastName: Simpson
    email: bart@simpson.com
    password: secret

User(maggie):
    usaCitizen: true
    firstName: Maggie
    lastName: Simpson
    email: maggie@simpson.com
    password: secret

Donation(a):
    amount : 210
    method : paypal

Donation(b):
    amount : 20
    method : cash

Donation(c):
    amount : 330
    method : cash
```

Routes

```
GET     /api/donors                         DonationServiceAPI.getAllDonors
GET     /api/donors/{id}                    DonationServiceAPI.getDonor
POST    /api/donors                         DonationServiceAPI.createDonor
DELETE  /api/donors/{id}                    DonationServiceAPI.deleteDonor
DELETE  /api/donors                         DonationServiceAPI.deleteAllDonors


GET     /api/donations                      DonationServiceAPI.getAllDonations
GET     /api/donations/{id}                 DonationServiceAPI.getDonation
POST    /api/donations                      DonationServiceAPI.createDonation
DELETE  /api/donations/{id}                 DonationServiceAPI.deleteDonation
DELETE  /api/donations                      DonationServiceAPI.deleteAllDonations
```

- Donation Service API

- Client application use these patterns to communicate with service

Your application is ready ! ×

localhost:9000

Your new application is ready!

Congratulation, you've just created a new play application. This page will help you in the few next steps.

Why do you see this page?

The **conf/routes** file defines a route that tell play to invoke the **Application.index** action when a browser requests the / URI using the **GET** method:

```
# Application home page
GET     /         Application.index
```

So play has invoked the **controllers.Application.index()** method:

```
public static void index() {
    render();
}
```

Using the **render()** call, this action asks play to display a template. By convention play has displayed the **app/views/Application/index.html** template:

```
#{extends 'main.html' /}
#{set title:'Home' /}
```

Play!

Play 1.2.7

Browse

- Local documentation
- Browse Java API

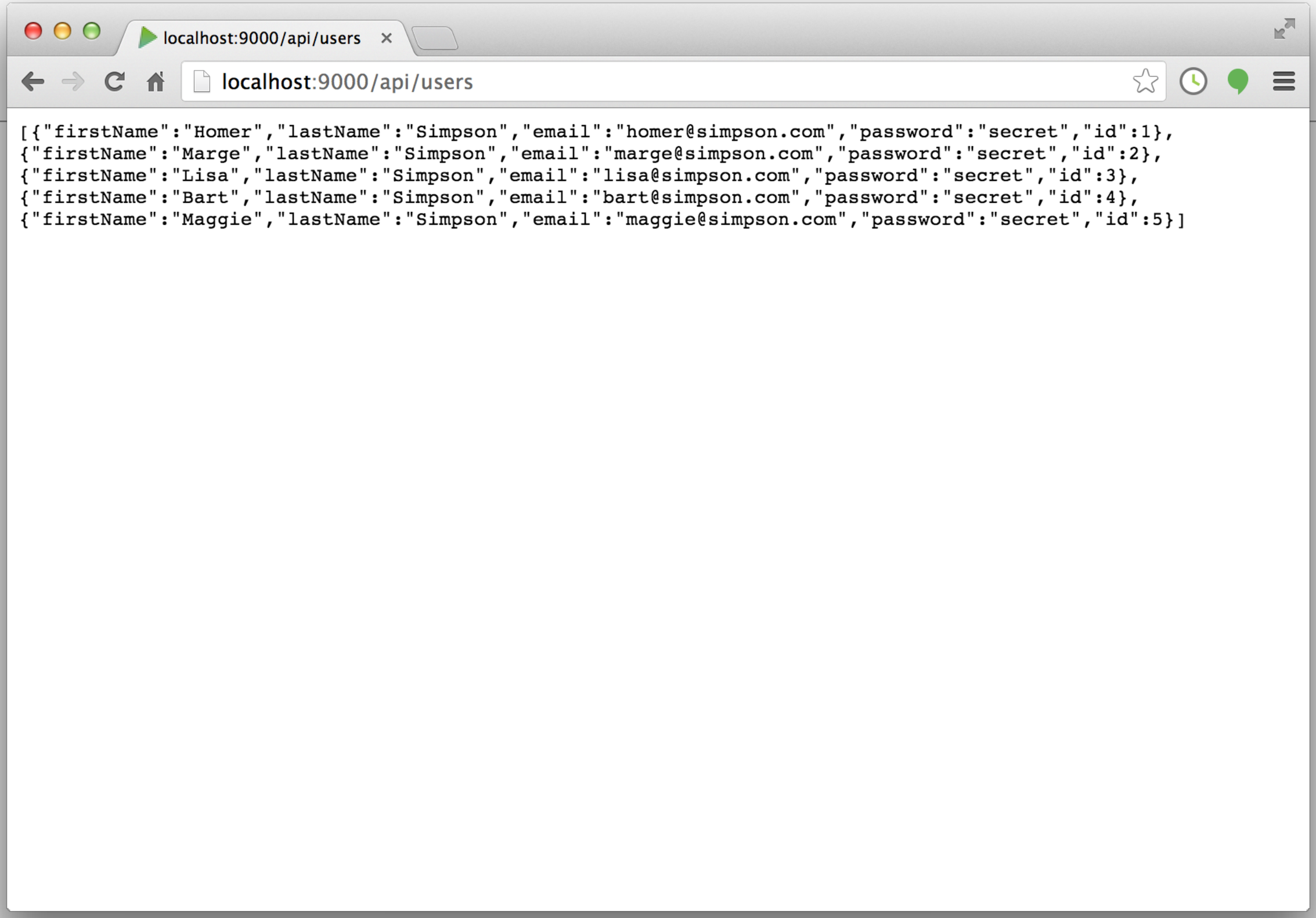
Contents

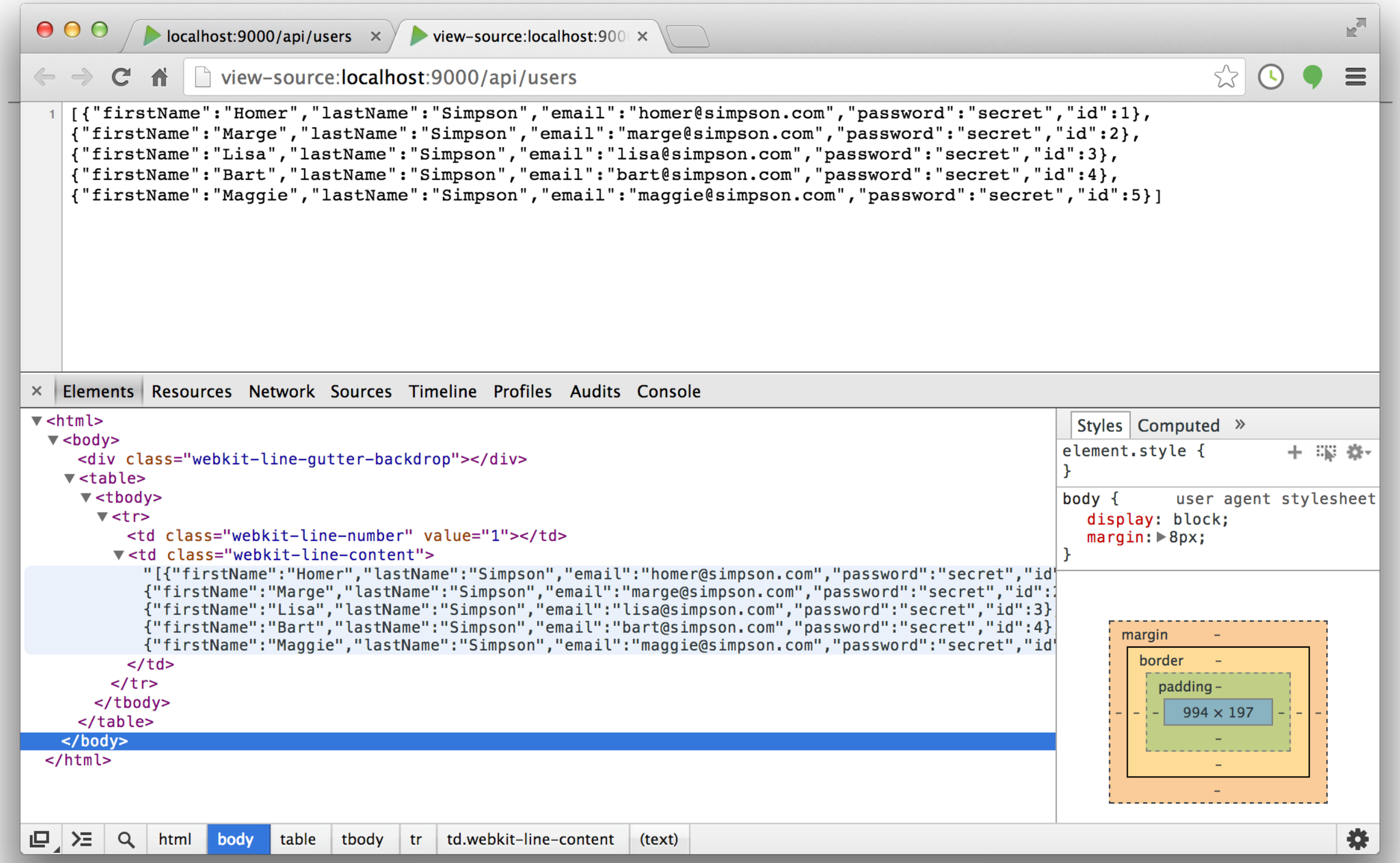
1. Why do you see this page?
2. Need to set up a Java IDE?
3. Need to connect to a database?
4. Need more help?

Search

Get help with google

Google™ Custom Search


localhost:9000/api/users
localhost:9000/api/users
[{"firstName":"Homer","lastName":"Simpson","email":"homer@simpson.com","password":"secret","id":1},
{"firstName":"Marge","lastName":"Simpson","email":"marge@simpson.com","password":"secret","id":2},
{"firstName":"Lisa","lastName":"Simpson","email":"lisa@simpson.com","password":"secret","id":3},
{"firstName":"Bart","lastName":"Simpson","email":"bart@simpson.com","password":"secret","id":4},
{"firstName":"Maggie","lastName":"Simpson","email":"maggie@simpson.com","password":"secret","id":5}]


localhost:9000/api/users
view-source:localhost:900
view-source:localhost:9000/api/users
1
[{"firstName":"Homer","lastName":"Simpson","email":"homer@simpson.com","password":"secret","id":1},
{"firstName":"Marge","lastName":"Simpson","email":"marge@simpson.com","password":"secret","id":2},
{"firstName":"Lisa","lastName":"Simpson","email":"lisa@simpson.com","password":"secret","id":3},
{"firstName":"Bart","lastName":"Simpson","email":"bart@simpson.com","password":"secret","id":4},
{"firstName":"Maggie","lastName":"Simpson","email":"maggie@simpson.com","password":"secret","id":5}]
Elements Resources Network Sources Timeline Profiles Audits Console
<html>
<body>

<table>
<tbody>
<tr>
<td class="webkit-line-number" value="1"></td>
<td class="webkit-line-content">
"[{"firstName":"Homer","lastName":"Simpson","email":"homer@simpson.com","password":"secret","id
{"firstName":"Marge","lastName":"Simpson","email":"marge@simpson.com","password":"secret","id":
{"firstName":"Lisa","lastName":"Simpson","email":"lisa@simpson.com","password":"secret","id":3}
{"firstName":"Bart","lastName":"Simpson","email":"bart@simpson.com","password":"secret","id":4}
{"firstName":"Maggie","lastName":"Simpson","email":"maggie@simpson.com","password":"secret","id
</td>
</tr>
</tbody>
</table>
</body>
</html>
Styles Computed
element.style {
}
body { user agent stylesheet
display: block;
margin: 8px;
}
margin
border
padding
994 × 197
html body table tbody tr td.webkit-line-content (text)

localhost:9000/api/users | view-source:localhost:900

view-source:localhost:9000/api/donations

```
1 [{"amount":210,"method":"paypal","id":1},{"amount":20,"method":"cash","id":2},{"amount":330,"method":"cash","id":3},
{"amount":10,"method":"paypal","id":4}]
```

Elements Resources Network Sources Timeline Profiles Audits Console

```
▼<html>
  ▼<body>
    <div class="webkit-line-gutter-backdrop"></div>
    ▼<table>
      ▼<tbody>
        ▼<tr>
          <td class="webkit-line-number" value="1"></td>
          ▼<td class="webkit-line-content">
            "[{"amount":210,"method":"paypal","id":1},{"amount":20,"method":"cash","id":2},
            {"amount":330,"method":"cash","id":3},{"amount":10,"method":"paypal","id":4}]"
          </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Styles Computed »

```
element.style {
}
td, th {    user agent stylesheet
  display: table-cell;
  vertical-align: inherit;
}
```

Inherited from table

```
table {     user agent stylesheet
  white-space: normal;
  line-height: normal;
  font-weight: normal;
  font-size: medium;
  font-variant: normal;
  font-style: normal;
  color: -webkit-text;
  text-align: start;
}
table {     user agent stylesheet
  border-collapse: separate;
```

html body table tbody tr td.webkit-line-content

localhost:9000/api/users

view-source:localhost:900

view-source:localhost:9000/api/users/1

```
1 {"firstName":"Homer","lastName":"Simpson","email":"homer@simpson.com","password":"secret","id":1}
```

Elements Resources Network Sources Timeline Profiles Audits Console

```
▼<html>
  ▼<body>
    <div class="webkit-line-gutter-backdrop"></div>
    ▼<table>
      ▼<tbody>
        ▼<tr>
          <td class="webkit-line-number" value="1"></td>
          ▼<td class="webkit-line-content">
            "{"firstName":"Homer","lastName":"Simpson","email":"homer@simpson.com","password":"secret","id"
          </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Styles Computed »

```
element.style {
}
```

```
td, th {    user agent stylesheet
  display: table-cell;
  vertical-align: inherit;
}
```

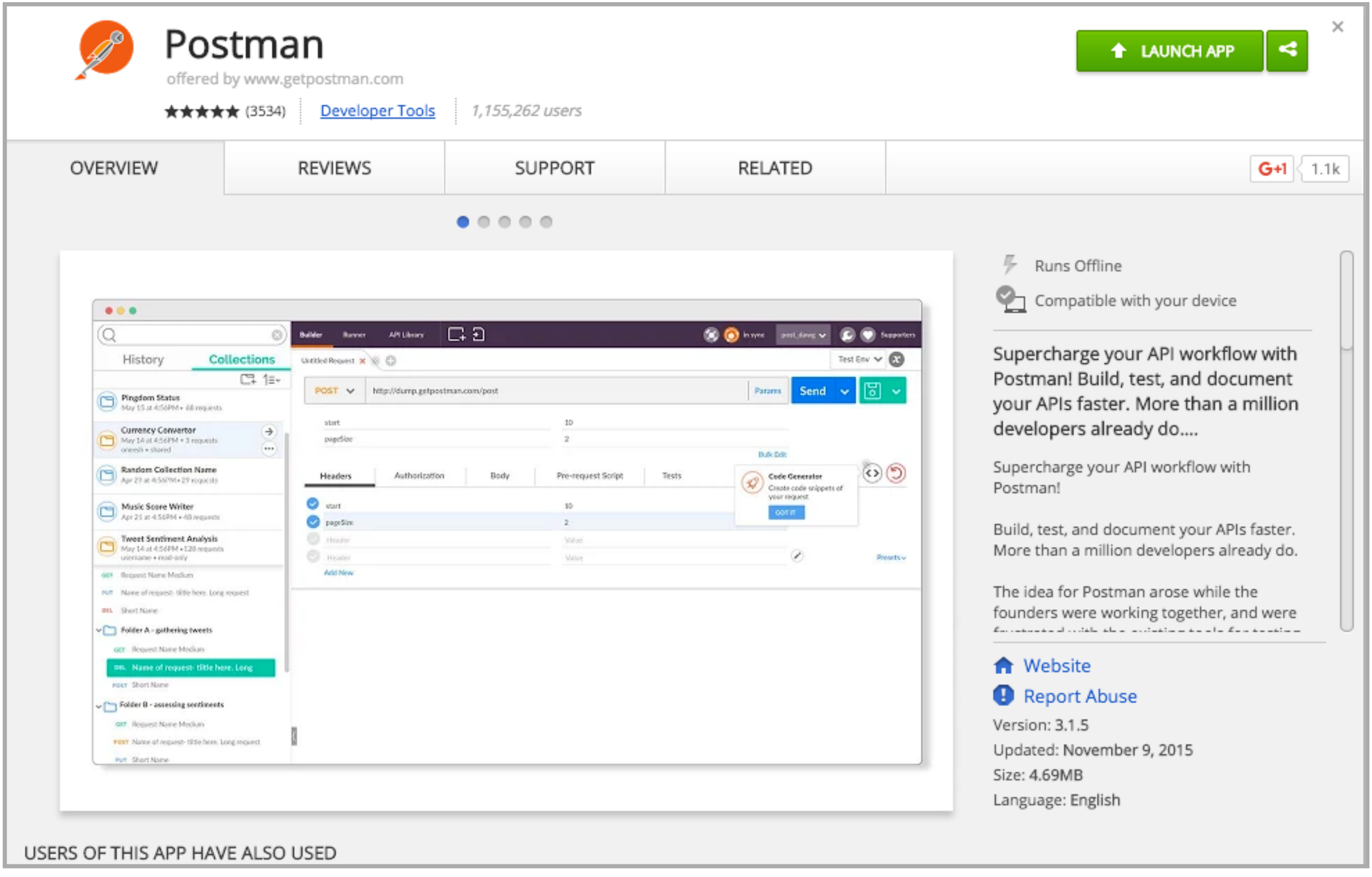
Inherited from table

```
table {     user agent stylesheet
  white-space: normal;
  line-height: normal;
  font-weight: normal;
  font-size: medium;
  font-variant: normal;
  font-style: normal;
  color: -webkit-text;
  text-align: start;
}
```

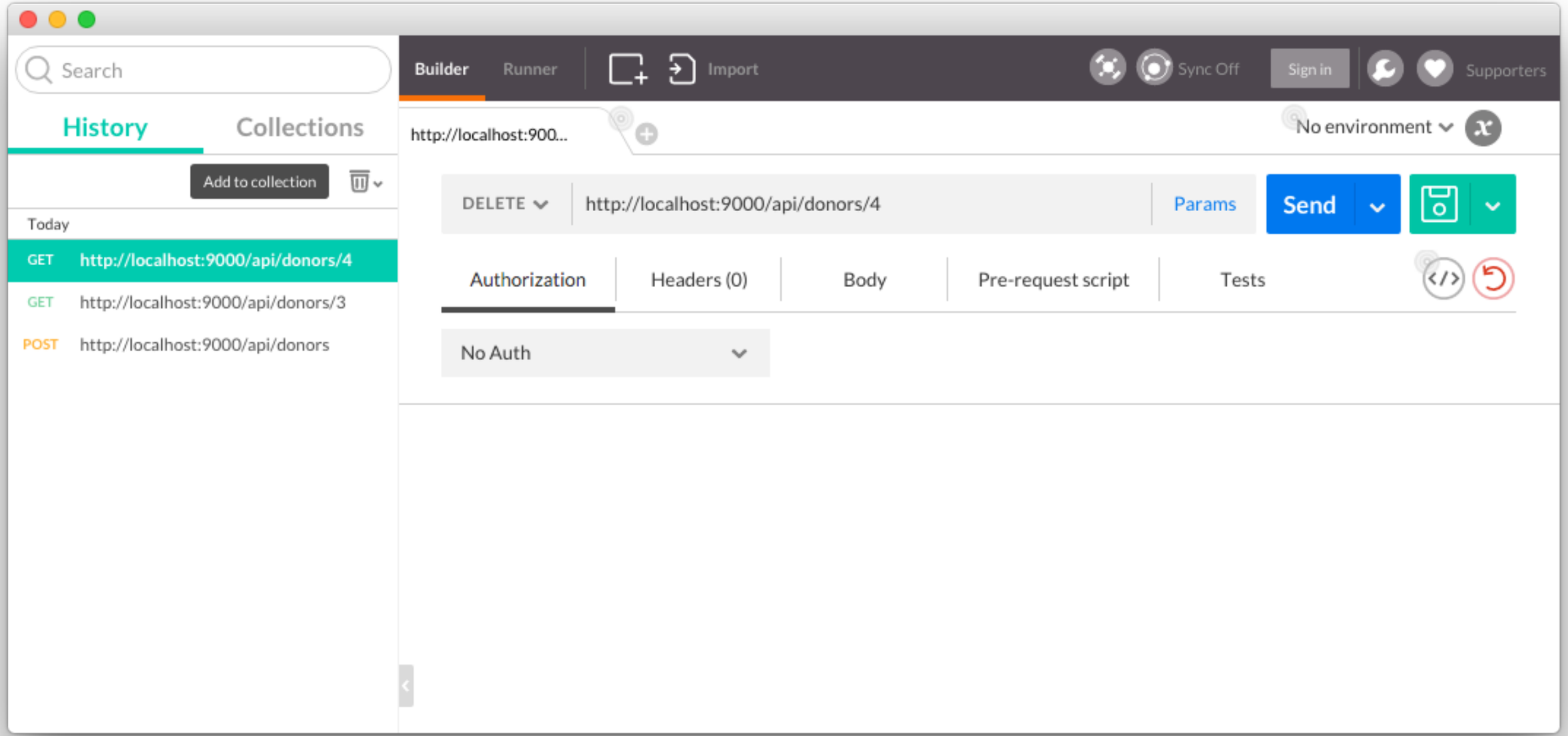
```
table {     user agent stylesheet
  border-collapse: separate;
```

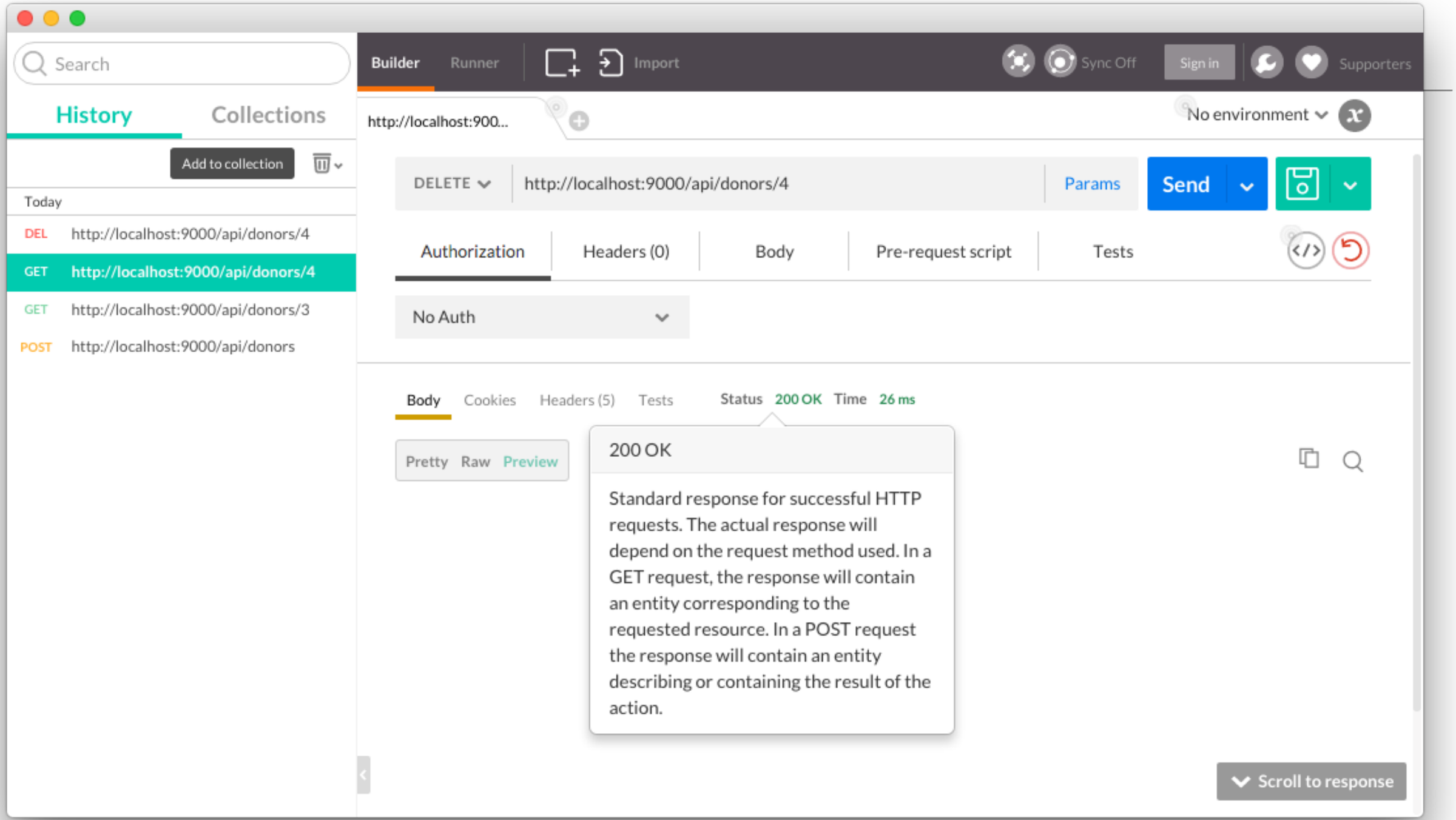
html body table tbody tr td.webkit-line-content

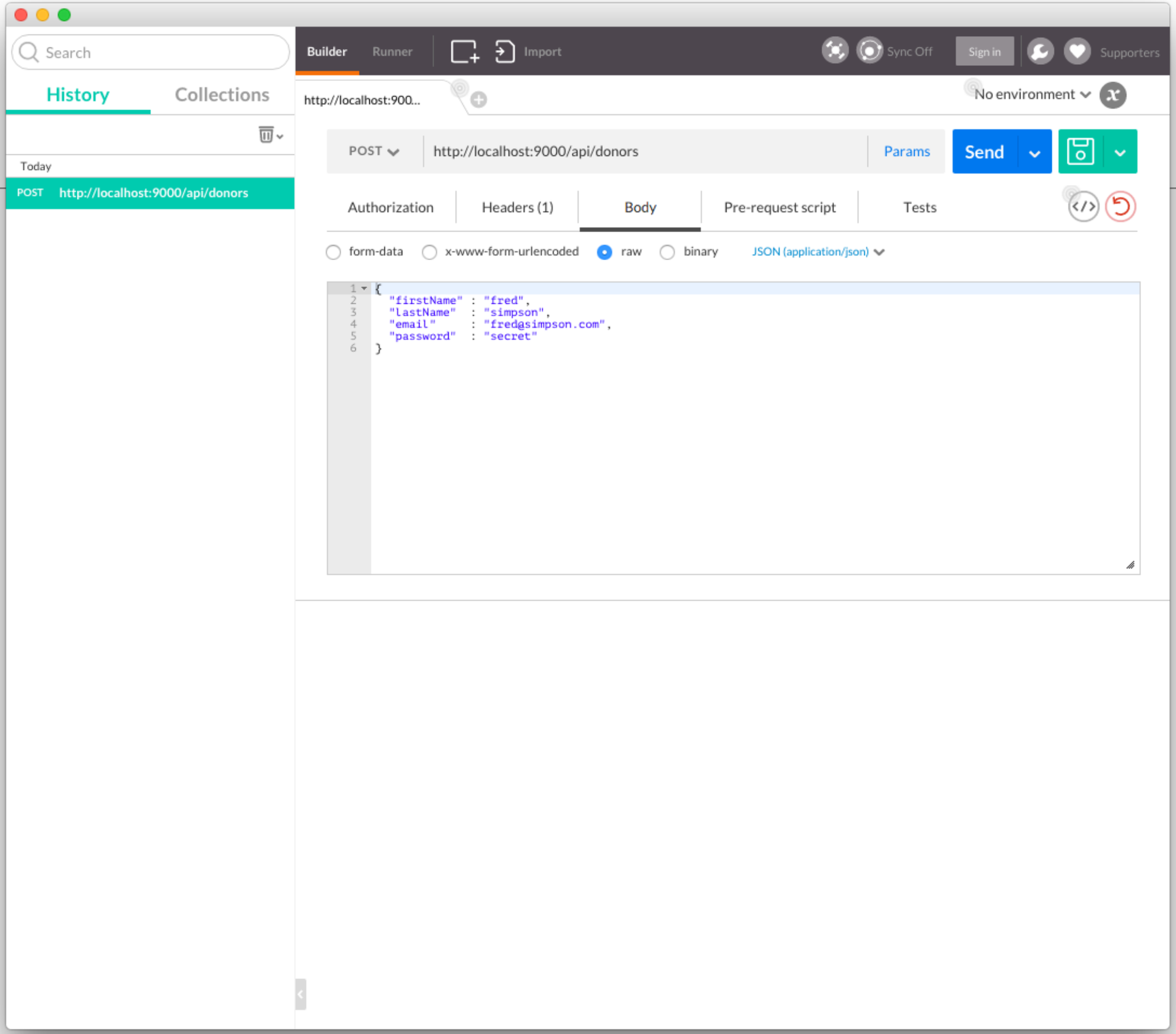
Postman


Postman
offered by www.getpostman.com
(3534)
Developer Tools
1,155,262 users
LAUNCH APP
OVERVIEW
REVIEWS
SUPPORT
RELATED
G+1
1.1k
Runs Offline
Compatible with your device
Supercharge your API workflow with Postman! Build, test, and document your APIs faster. More than a million developers already do....
Supercharge your API workflow with Postman!
Build, test, and document your APIs faster. More than a million developers already do.
The idea for Postman arose while the founders were working together, and were
Website
Report Abuse
Version: 3.1.5
Updated: November 9, 2015
Size: 4.69MB
Language: English
USERS OF THIS APP HAVE ALSO USED

Portman


Search
History
Collections
Add to collection
Today
GET http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/3
POST http://localhost:9000/api/donors
Builder
Runner
Import
Sync Off
Sign in
Supporters
http://localhost:900...
No environment
DELETE
http://localhost:9000/api/donors/4
Params
Send
Authorization
Headers (0)
Body
Pre-request script
Tests
No Auth


Search
History
Collections
Add to collection
Today
DEL http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/3
POST http://localhost:9000/api/donors
Builder
Runner
Import
Sync Off
Sign in
Supporters
http://localhost:900...
No environment
DELETE
http://localhost:9000/api/donors/4
Params
Send
Authorization
Headers (0)
Body
Pre-request script
Tests
No Auth
Body
Cookies
Headers (5)
Tests
Status 200 OK Time 26 ms
Pretty
Raw
Preview
200 OK
Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.
Scroll to response


Search
History
Collections
Today
POST http://localhost:9000/api/donors
Builder
Runner
Import
Sync Off
Sign in
Supporters
http://localhost:900...
No environment
POST
http://localhost:9000/api/donors
Params
Send
Authorization
Headers (1)
Body
Pre-request script
Tests
form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)
{
"firstName" : "fred",
"lastName" : "simpson",
"email" : "fred@simpson.com",
"password" : "secret"
}

Search

History | Collections

Today

POST http://localhost:9000/api/donors

POST http://localhost:9000/api/donors

Builder | Runner | Import | Sync Off | Sign in | Supporters

http://localhost:900...

No environment

POST | http://localhost:9000/api/donors | Params | Send

Authorization | Headers (1) | Body | Pre-request script | Tests

form-data | x-www-form-urlencoded | raw | binary | JSON (application/json)

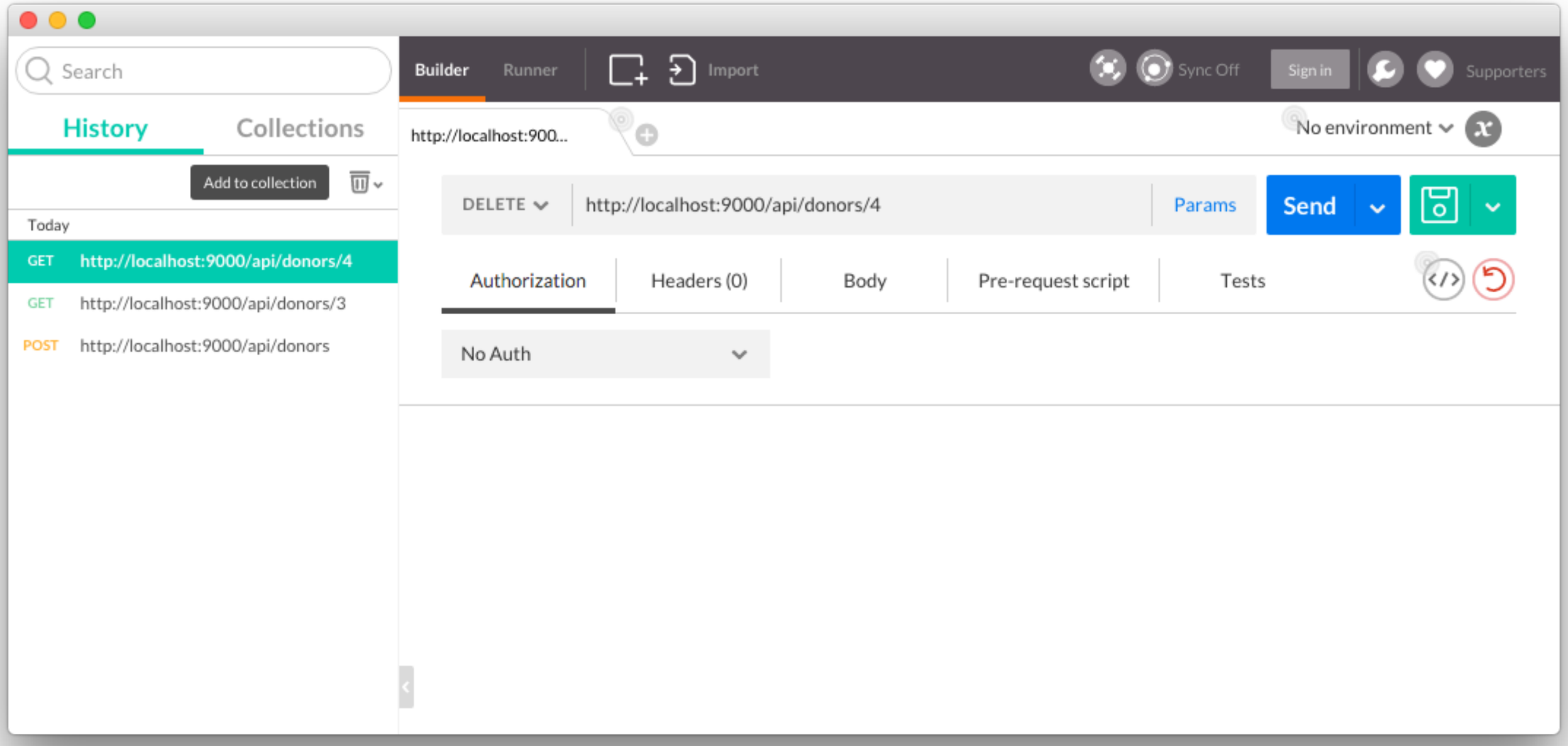
```
{
  "firstName" : "fred",
  "lastName"  : "simpson",
  "email"     : "fred@simpson.com",
  "password"  : "secret"
}
```

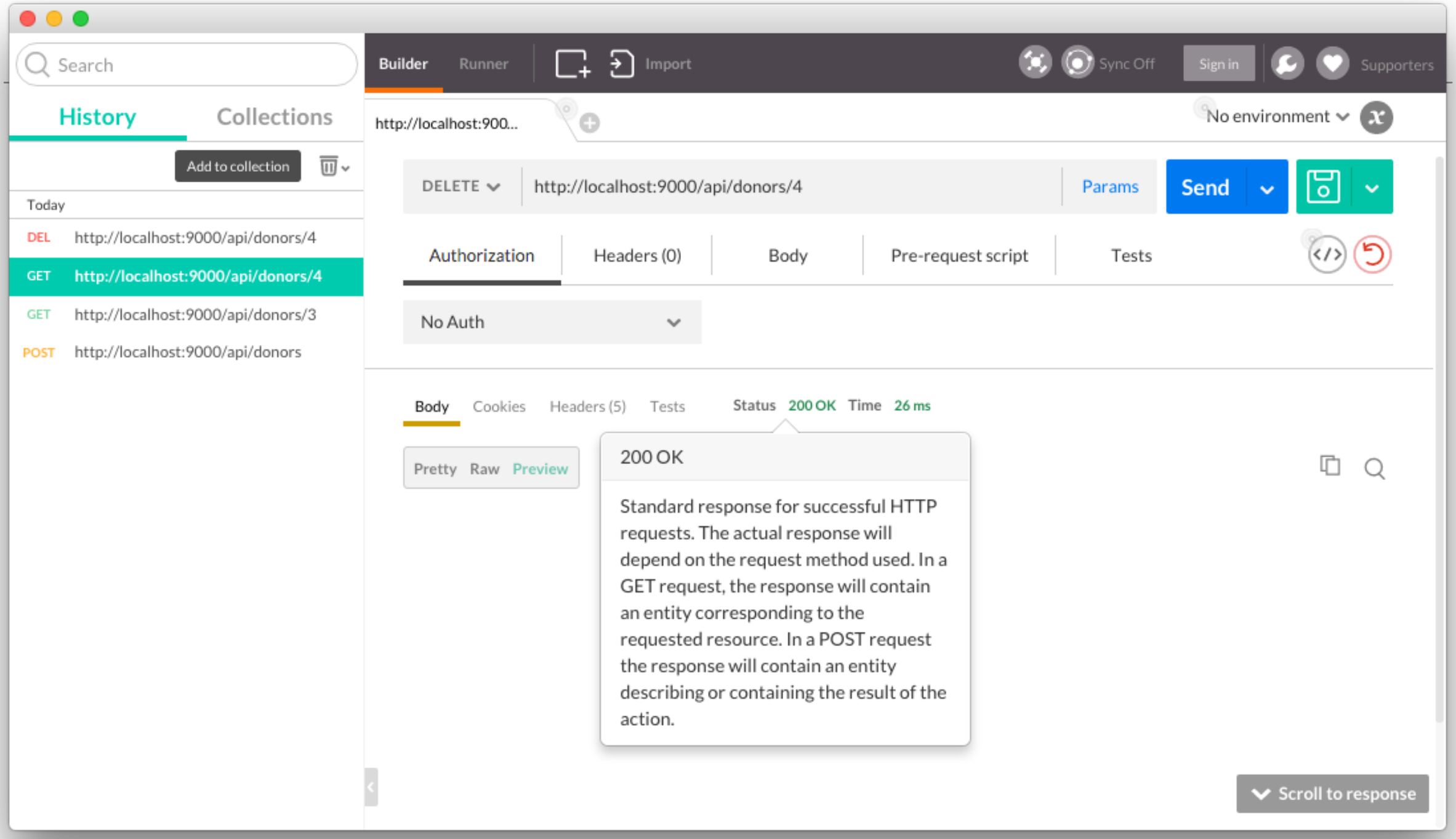
Body | Cookies | Headers (5) | Tests | Status 200 OK | Time 12 ms

Pretty | Raw | Preview

{"firstName":"fred","lastName":"simpson","email":"fred@simpson.com","password":"secret","id":8}

Scroll to response


Search
History
Collections
Add to collection
Today
GET http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/3
POST http://localhost:9000/api/donors
Builder
Runner
Import
Sync Off
Sign in
Supporters
http://localhost:900...
No environment
DELETE
http://localhost:9000/api/donors/4
Params
Send
Authorization
Headers (0)
Body
Pre-request script
Tests
No Auth


Search
History
Collections
Add to collection
Today
DEL http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/3
POST http://localhost:9000/api/donors
Builder
Runner
Import
Sync Off
Sign in
Supporters
http://localhost:900...
No environment
DELETE
http://localhost:9000/api/donors/4
Params
Send
Authorization
Headers (0)
Body
Pre-request script
Tests
No Auth
Body
Cookies
Headers (5)
Tests
Status 200 OK Time 26 ms
Pretty Raw Preview
200 OK
Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request the response will contain an entity describing or containing the result of the action.
Scroll to response

Search

History | Collections

Add to collection

Today

DEL http://localhost:9000/api/donors/4
DEL http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/4
GET http://localhost:9000/api/donors/3
POST http://localhost:9000/api/donors

Builder | Runner | Import | Sync Off | Sign in | Supporters

http://localhost:900...

No environment

DELETE | http://localhost:9000/api/donors/4 | Params | Send

Authorization | Headers (0) | Body | Pre-request script | Tests

form-data | x-www-form-urlencoded | raw | binary

Key | Value | Text

Body | Cookies | Headers (5) | Tests | Status 404 Not Found | Time 155 ms

Pretty | Raw | Preview

Not found

No Doner with ID4

These routes have been tried, in this order :

```
1.  GET     /@projectdocs/files/{name}                          ProjectDocumentation.file
2.  GET     /@projectdocs/images/{name}                         ProjectDocumentation.image
3.  GET     /@projectdocs/{id}                                  ProjectDocumentation.page
4.  GET     /@projectdocs/?                                     ProjectDocumentation.index
5.  GET     /@documentation/{docLang}/cheatsheet/{category}     PlayDocumentation.cheatShe
6.  GET     /@documentation/cheatsheet/{category}               PlayDocumentation.cheatShe
7.  GET     /@documentation/modules/{module}/files/{name}       PlayDocumentation.file
8.  GET     /@documentation/modules/{module}/images/{name}      PlayDocumentation.image
```

Scroll to response

Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see http://creativecommons.org/licenses/by-nc/3.0/