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

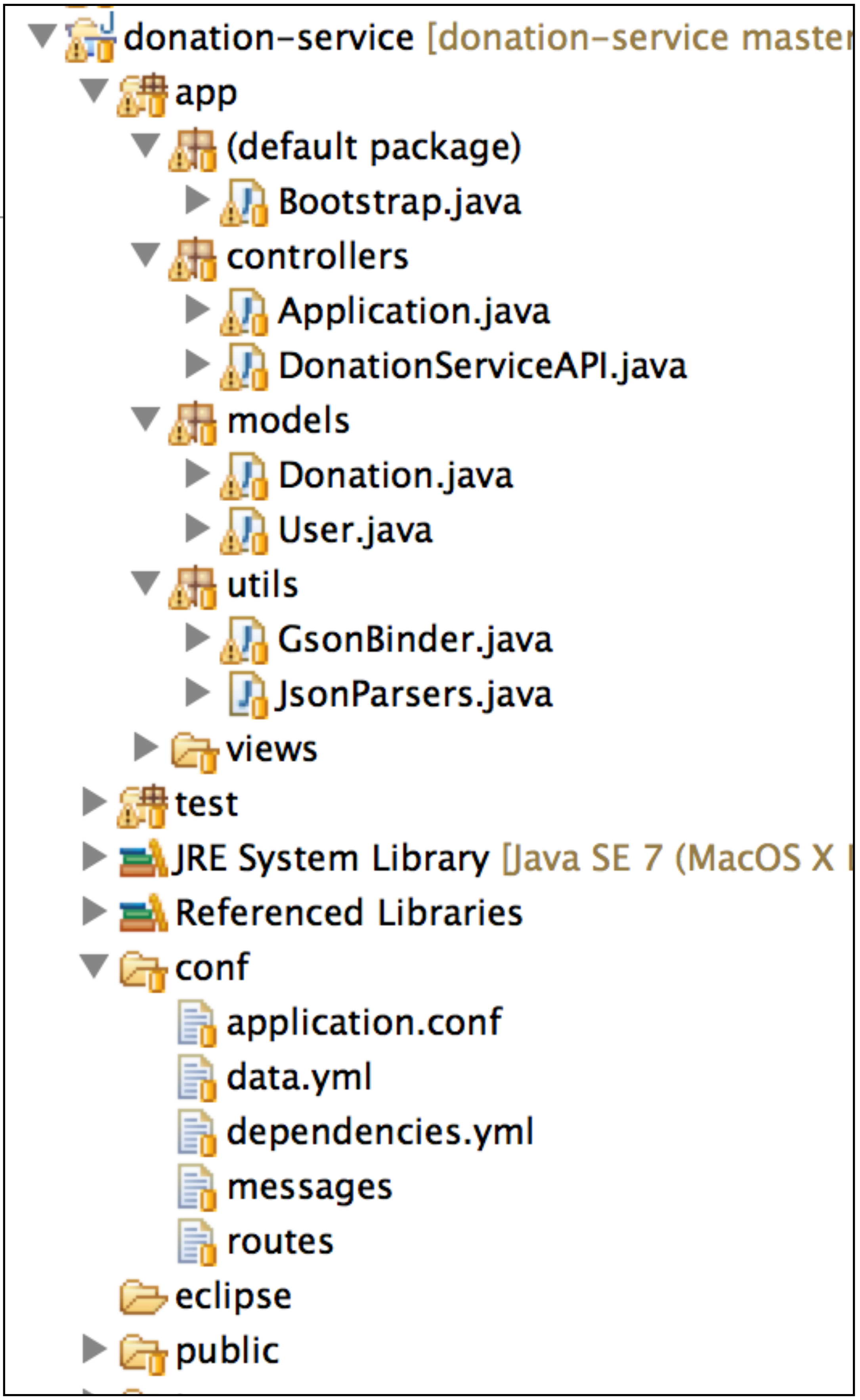
Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit

donation-service-play

donation-service Play App

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



```
▼ donation-service [donation-service maste
  ▼ app
    ▼ (default package)
      ▶ Bootstrap.java
    ▼ controllers
      ▶ Application.java
      ▶ DonationServiceAPI.java
    ▼ models
      ▶ Donation.java
      ▶ User.java
    ▼ utils
      ▶ GsonBinder.java
      ▶ JsonParsers.java
    ▶ views
  ▶ test
  ▶ JRE System Library [Java SE 7 (MacOS X
  ▶ Referenced Libraries
  ▼ conf
      application.conf
      data.yml
      dependencies.yml
      messages
      routes
    eclipse
  ▶ public
```

Models

```java
@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;
  }
}
```

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

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

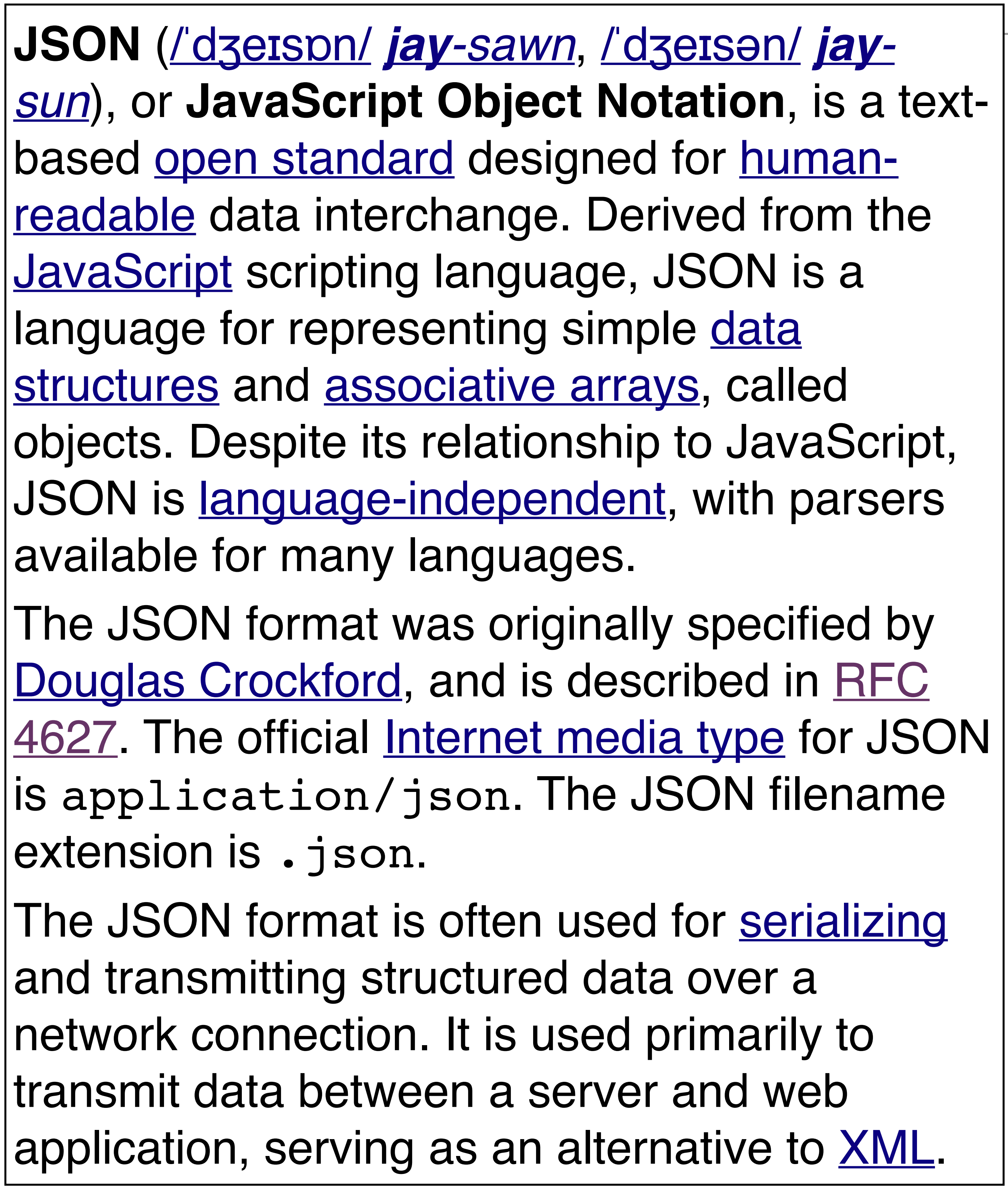
Utils

```
@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 acquire 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
}
```

utils

- Convert Model objects to/from Json
- Include support for list of Model objects

```java
public class JsonParsers
{
  static Gson gson = new Gson();

  public static User json2User(String json)
  {
    return gson.fromJson(json, User.class);
  }

  public static List<User> json2Users(String json)
  {
    Type collectionType = new TypeToken<List<User>>() {}.getType();
    return gson.fromJson(json, collectionType);
  }

  public static String user2Json(Object obj)
  {
    return gson.toJson(obj);
  }

  public static Donation json2Donation(String json)
  {
    return gson.fromJson(json, Donation.class);
  }

  public static String donation2Json(Object obj)
  {
    return gson.toJson(obj);
  }

  public static List<Donation> json2Donations(String json)
  {
    Type collectionType = new TypeToken<List<Donation>>() {}.getType();
    return gson.fromJson(json, collectionType);
  }
}
```

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

```java
public class DonationServiceAPI extends Controller
{
  public static void users()
  {
    List<User> users = User.findAll();
    renderJSON(JsonParsers.user2Json(users));
  }

  public static void user(Long id)
  {
    User user = User.findById(id);
    if (user == null)
    {
      notFound();
    }
    else
    {
      renderJSON(JsonParsers.user2Json(user));
    }
  }

  public static void createUser(JsonElement body)
  {
    User user = JsonParsers.json2User(body.toString());
    user.save();
    renderJSON(JsonParsers.user2Json(user));
  }

  public static void deleteUser(Long id)
  {
    User user = User.findById(id);
    if (user == null)
    {
      notFound();
    }
    else
    {
      user.delete();
      renderText("success");
    }
  }
  //…
 }
```

controllers

```
public class DonationServiceAPI extends Controller
{
  //…

  public static void donations()
  {
    List<Donation> donations = Donation.findAll();
    renderText(JsonParsers.donation2Json(donations));
  }

  public static void donation (Long id)
  {
   Donation donation = Donation.findById(id);
   renderJSON (JsonParsers.donation2Json(donation));
  }

  public static void createDonation(JsonElement body)
  {
    Donation donation = JsonParsers.json2Donation(body.toString());
    Donation newDonation = new Donation (donation.amount, donation.method);
    newDonation.save();
    renderJSON (JsonParsers.donation2Json(newDonation));
  }


  public static void deleteDonation(Long id)
  {
    Donation donation = Donation.findById(id);
    if (donation == null)
    {
      notFound();
    }
    else
    {
      donation.delete();
      ok();
    }
  }
}
```

Boostrap

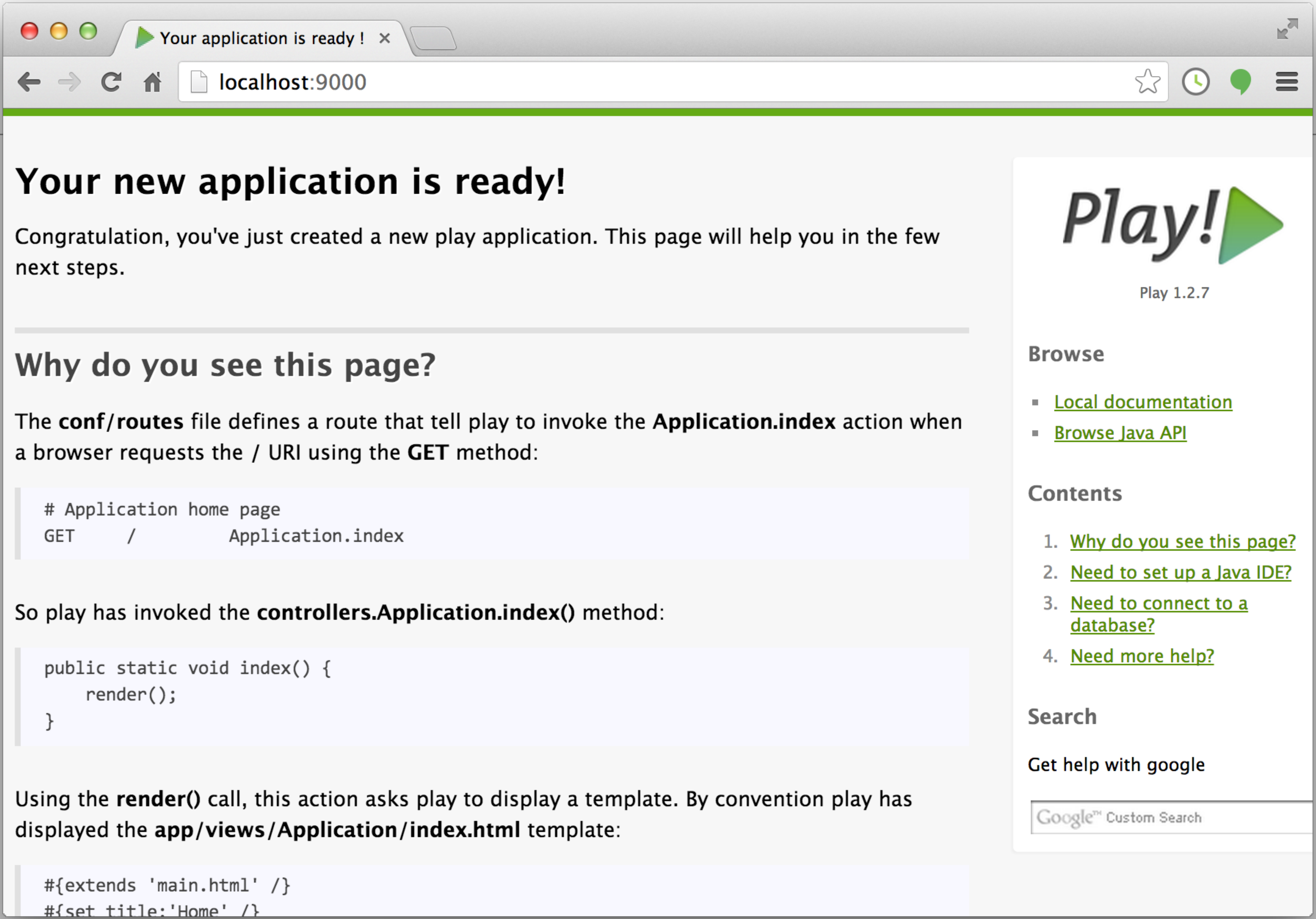
```
@OnApplicationStart
public class Bootstrap extends Job
{
  public void doJob()
  {
    if (User.count() == 0)
    {
     Fixtures.deleteDatabase();
     Fixtures.loadModels("data.yml");
    }
  }
}
```

```
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/users                                  DonationServiceAPI.users
GET     /api/users/{id}                             DonationServiceAPI.user
POST    /api/users                                  DonationServiceAPI.createUser
DELETE  /api/users/{id}                             DonationServiceAPI.deleteUser

GET     /api/donations                              DonationServiceAPI.donations
GET     /api/donations/{id}                         DonationServiceAPI.donation
POST    /api/donations                              DonationServiceAPI.createDonation
DELETE  /api/donations/{id}                         DonationServiceAPI.deleteDonation
```

- Donation Service API
- Client application use these patterns to communicate with service



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 1.2.7

Browse

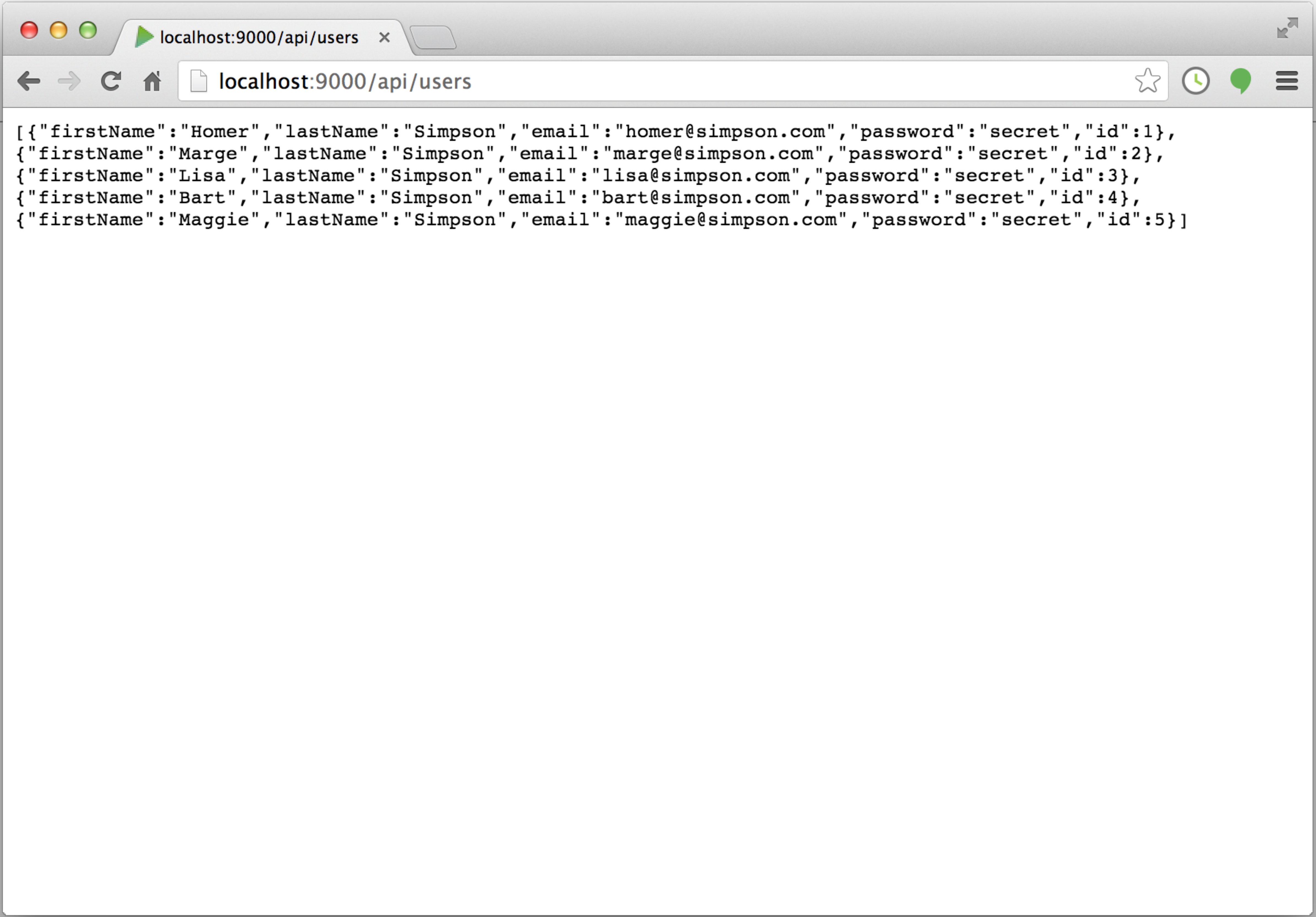
- Local documentation
- Browse Java API

Contents

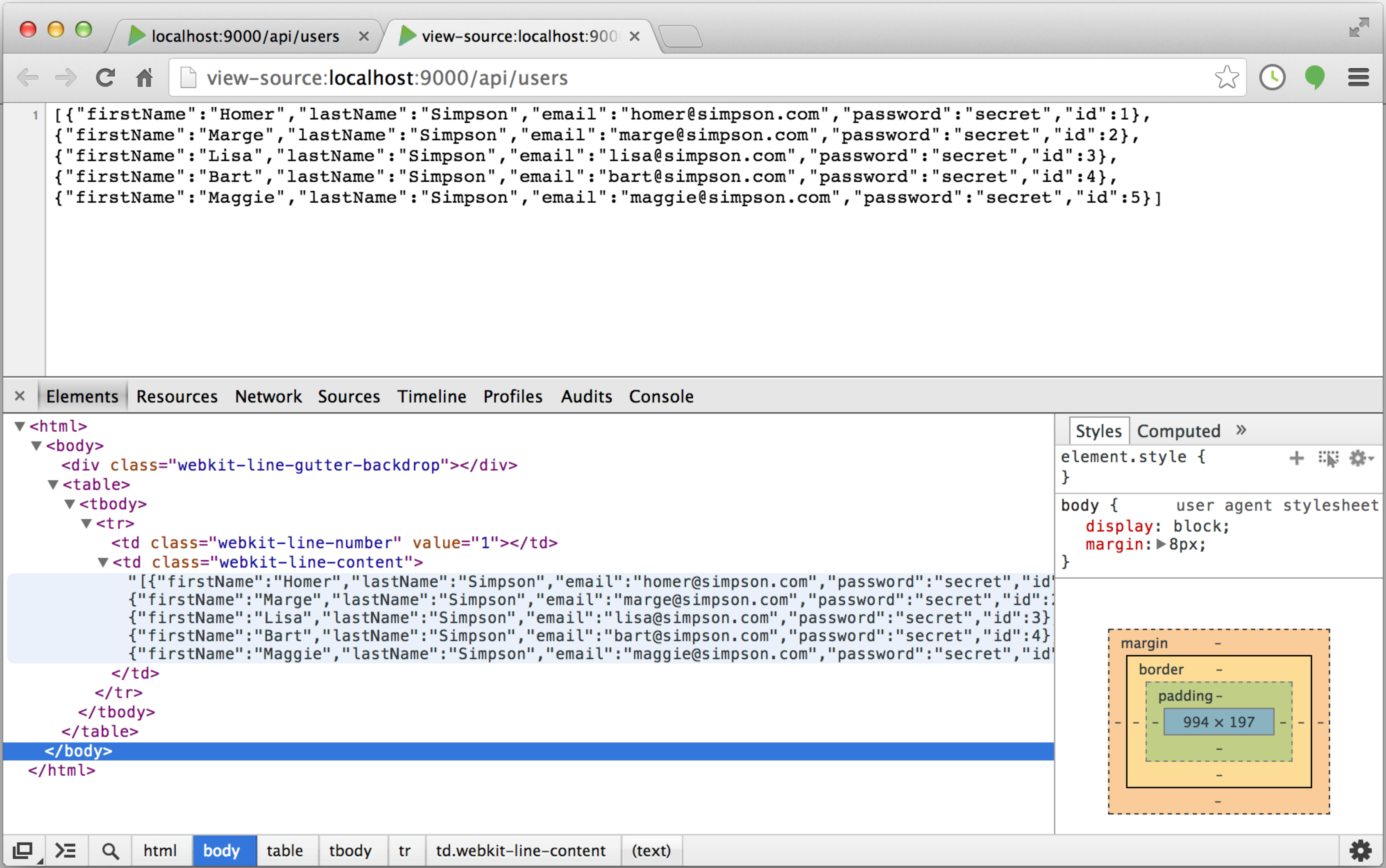
1. Why do you see this page?
2. Need to set up a Java IDE?
3. Need to connect to a database?
4. Need more help?

Search

Get help with google



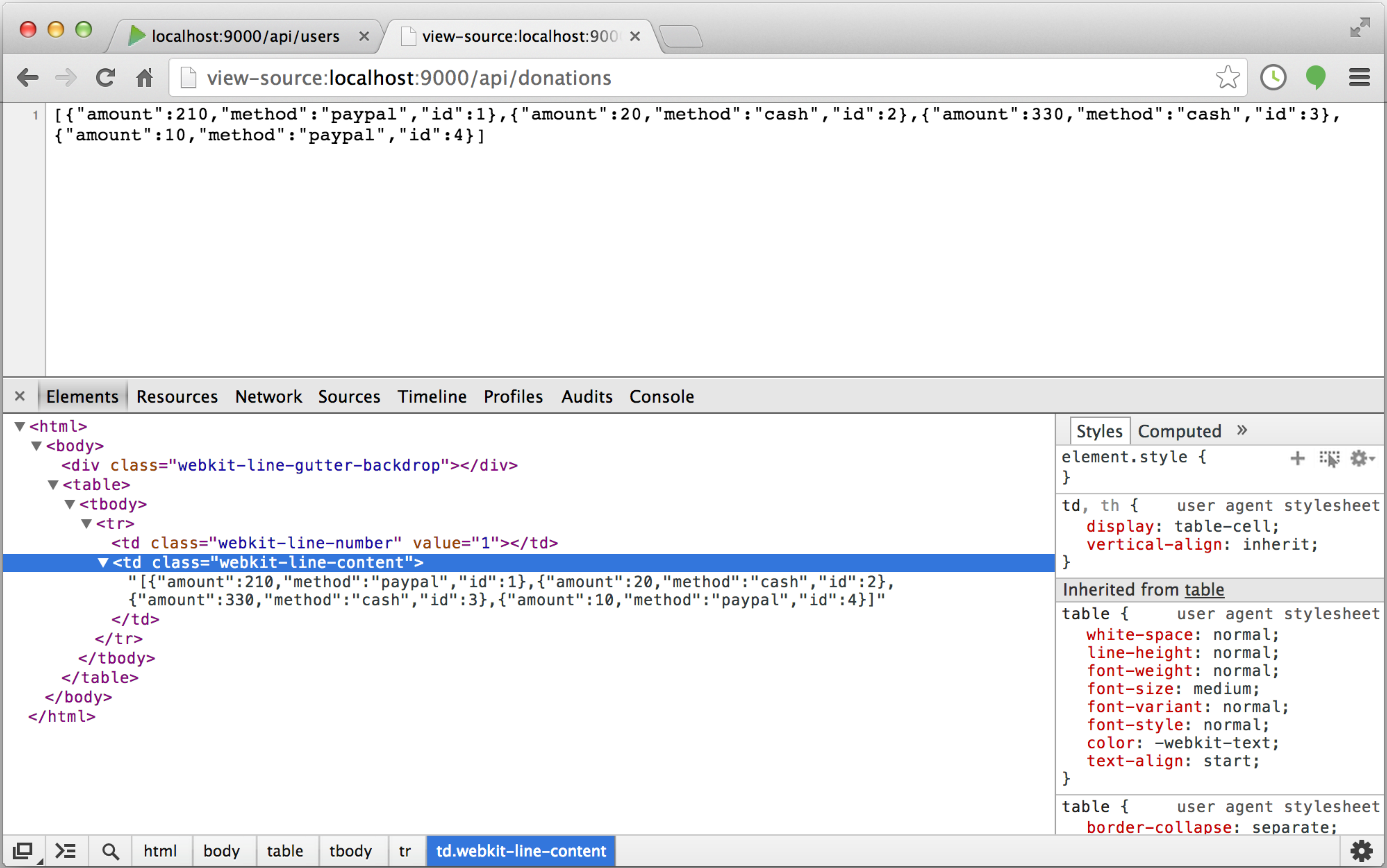
```
[{"firstName":"Homer","lastName":"Simpson","email":"homer@simpson.com","password":"secret","id":1},
{"firstName":"Marge","lastName":"Simpson","email":"marge@simpson.com","password":"secret","id":2},
{"firstName":"Lisa","lastName":"Simpson","email":"lisa@simpson.com","password":"secret","id":3},
{"firstName":"Bart","lastName":"Simpson","email":"bart@simpson.c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



```
[{"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}]
```

```
▼<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'
             {"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>
```

```
element.style {
}
body {         user agent stylesheet
   display: block;
   margin: ▶8px;
}
```



localhost:9000/api/users

view-source:localhost:9000/api/donations

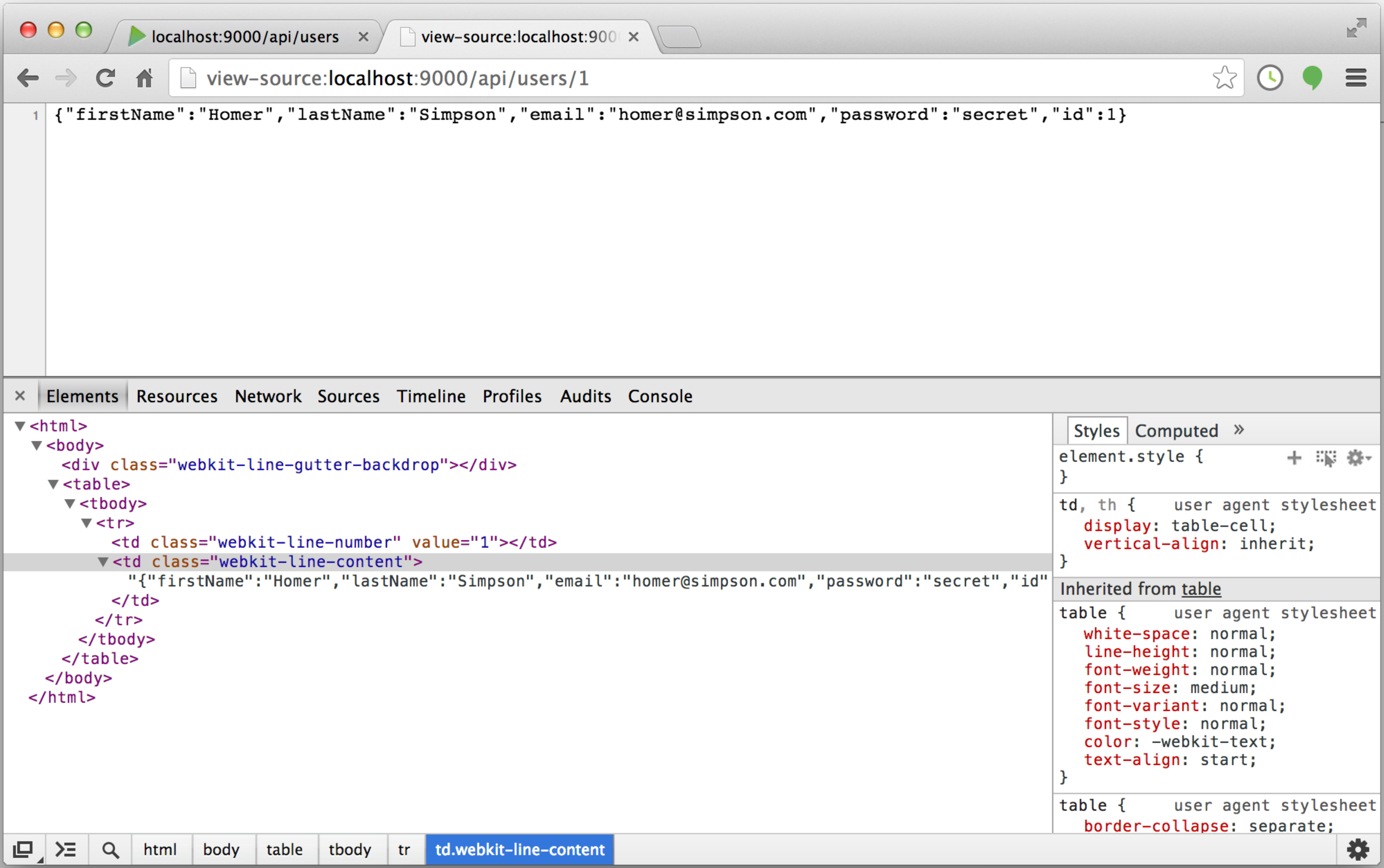
```
[{"amount":210,"method":"paypal","id":1},{"amount":20,"method":"cash","id":2},{"amount":330,"method":"cash","id":3},
{"amount":10,"method":"paypal","id":4}]
```

```
▼<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>
```

```
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;
```

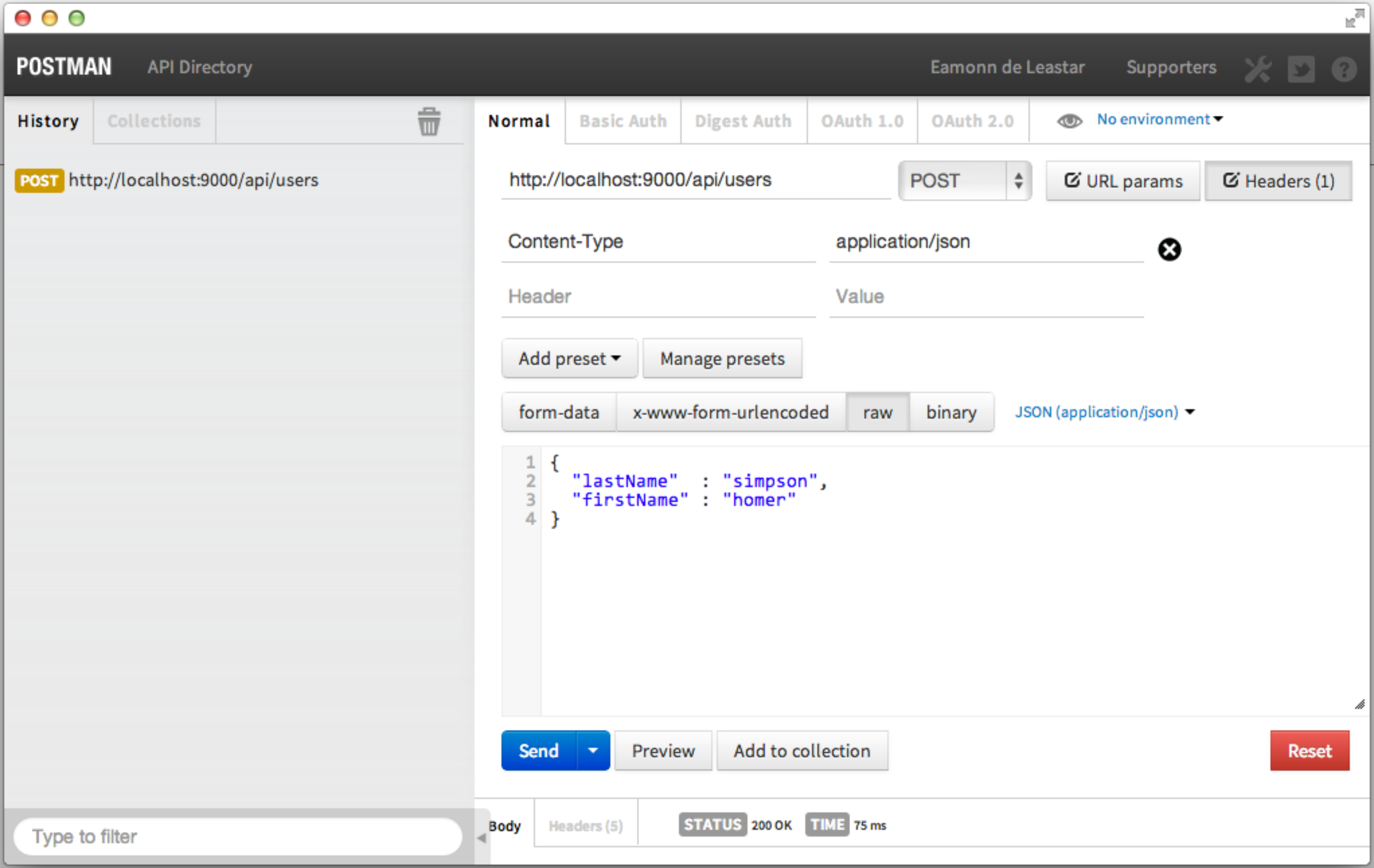


localhost:9000/api/users
view-source:localhost:900
view-source:localhost:9000/api/users/1

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

```
▼<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>
```

```
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;
```



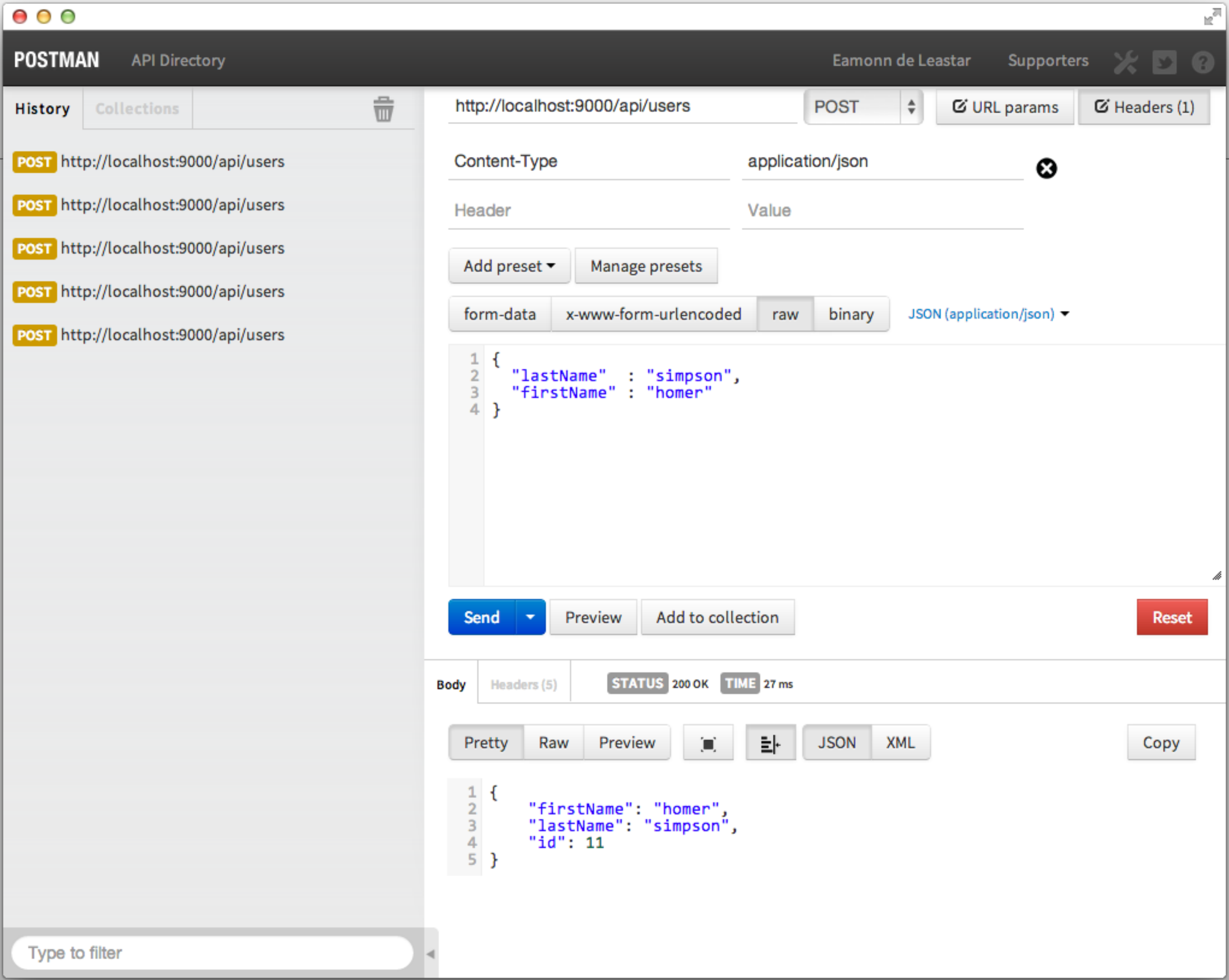
POST http://localhost:9000/api/users

http://localhost:9000/api/users POST

Content-Type	application/json
Header	Value

```
{
  "lastName"  : "simpson",
  "firstName" : "homer"
}
```

STATUS 200 OK TIME 75 ms



POST http://localhost:9000/api/users
POST http://localhost:9000/api/users
POST http://localhost:9000/api/users
POST http://localhost:9000/api/users
POST http://localhost:9000/api/users

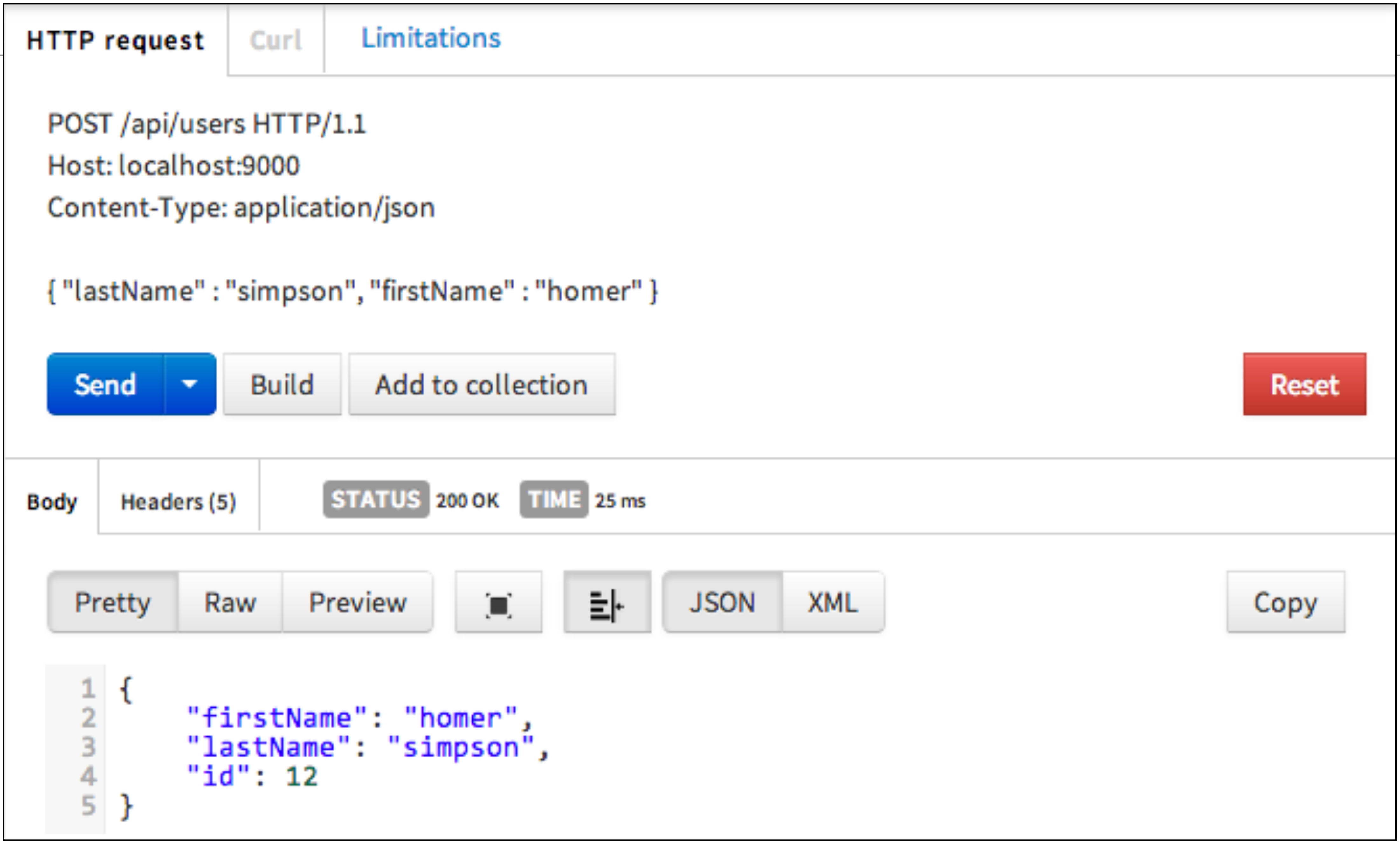
http://localhost:9000/api/users POST

Header	Value
Content-Type	application/json

```
{
  "lastName"  : "simpson",
  "firstName" : "homer"
}
```

STATUS 200 OK TIME 27 ms

```
{
    "firstName": "homer",
    "lastName": "simpson",
    "id": 11
}
```



```
POST /api/users HTTP/1.1
Host: localhost:9000
Content-Type: application/json

{ "lastName" : "simpson", "firstName" : "homer" }
```

STATUS 200 OK TIME 25 ms

```
{
    "firstName": "homer",
    "lastName": "simpson",
    "id": 12
}
```

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit