

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>



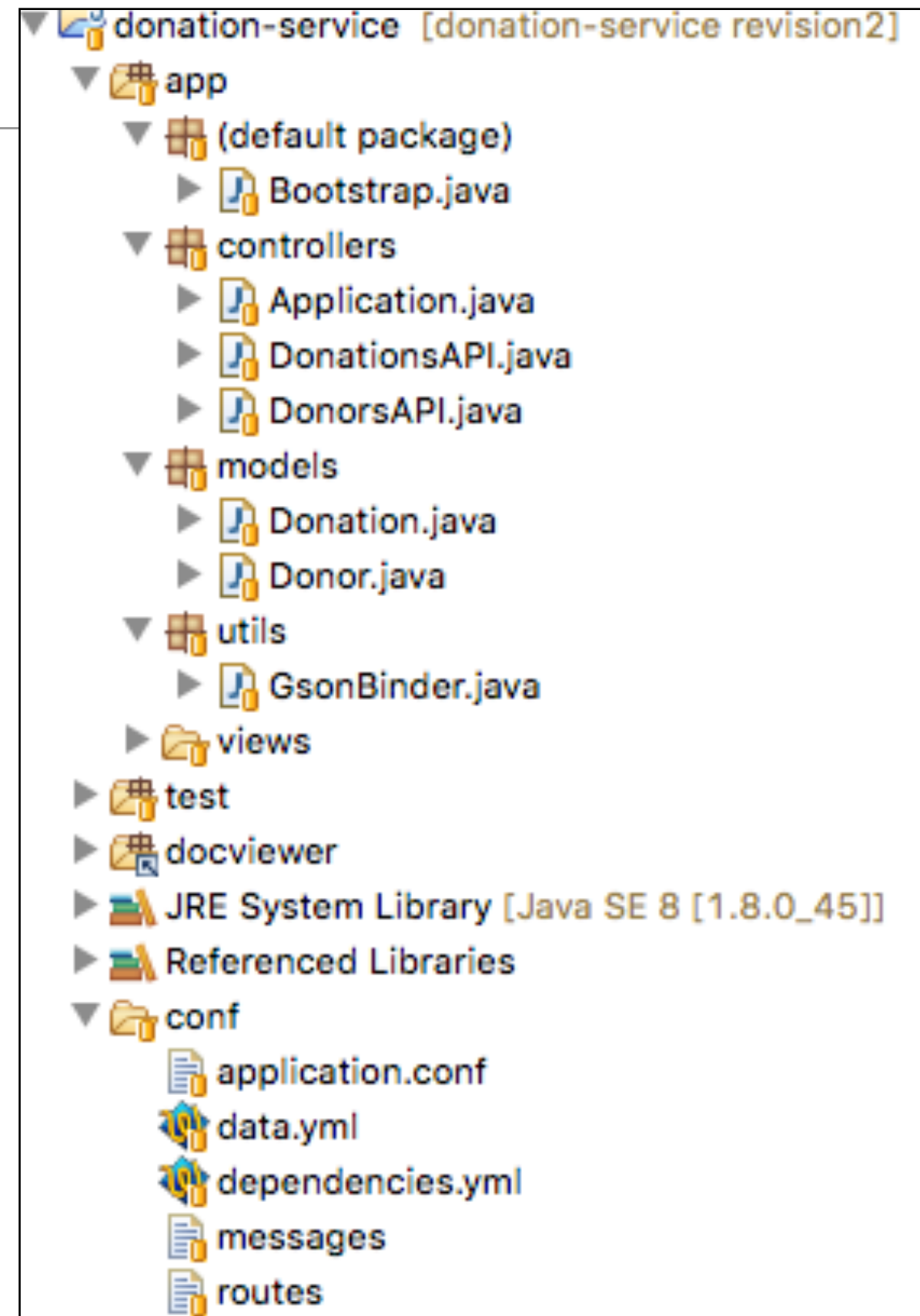
Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



donation-service

donation-service Play App

- Rebuild simpler version of the donation play app:
- No User Interface
- One-to-Many Donor-Donation relationship
- Includes:
 - Converters from Json to Donor/Donation and back



Models

```
@Entity
public class Donor extends Model
{
    public String firstName;
    public String lastName;
    public String email;
    public String password;
    @OneToMany(cascade = CascadeType.ALL)
    public List<Donation>
        donations = new ArrayList<Donation>();

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

```
@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;
    }
}
```

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.

controllers - DonorsAPI

- 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 DonorsAPI 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));
    }

    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");
    }
}
```

controllers- DonationAPI

```
public static void createDonation(Long id, JsonElement body)
{
    Donor donor = Donor.findById(id);
    Donation donation = gson.fromJson(body.toString(), Donation.class);
    donation.id = null;
    donor.donations.add(donation);
    donor.save();
    renderJSON (gson.toJson(donation));
}

public static void deleteDonation(Long id, Long donationId)
{
    Donation donation = Donation.findById(donationId);
    if (donation == null)
    {
        notFound();
    }
    else
    {
        donation.delete();
        renderJSON (gson.toJson(donation));
    }
}
}
```

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

    public static void getAllDonations()
    {
        List<Donation> donations = Donation.findAll();
        renderJSON(gson.toJson(donations));
    }

    public static void deleteAllDonations()
    {
        Donation.deleteAll();
        renderText("success");
    }

    public static void getDonations (Long id)
    {
        Donor donor = Donor.findById(id);
        renderJSON (gson.toJson(donor.donations));
    }

    public static void getDonation (Long id, Long donationId)
    {
        Donation donation = Donation.findById(donationId);
        renderJSON (gson.toJson(donation));
    }
}
```

Bootstrap

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

```
Donation(a):
  amount : 210
  method : paypal
```

```
Donation(b):
  amount : 20
  method : cash
```

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

```
Donation(d):
  amount : 10
  method : paypal
```

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

```
Donor(marge):
  usaCitizen: true
  firstName: Marge
  lastName: Simpson
  email: marge@simpson.com
  password: secret
  donations:
    - a
    - b
    - c
```

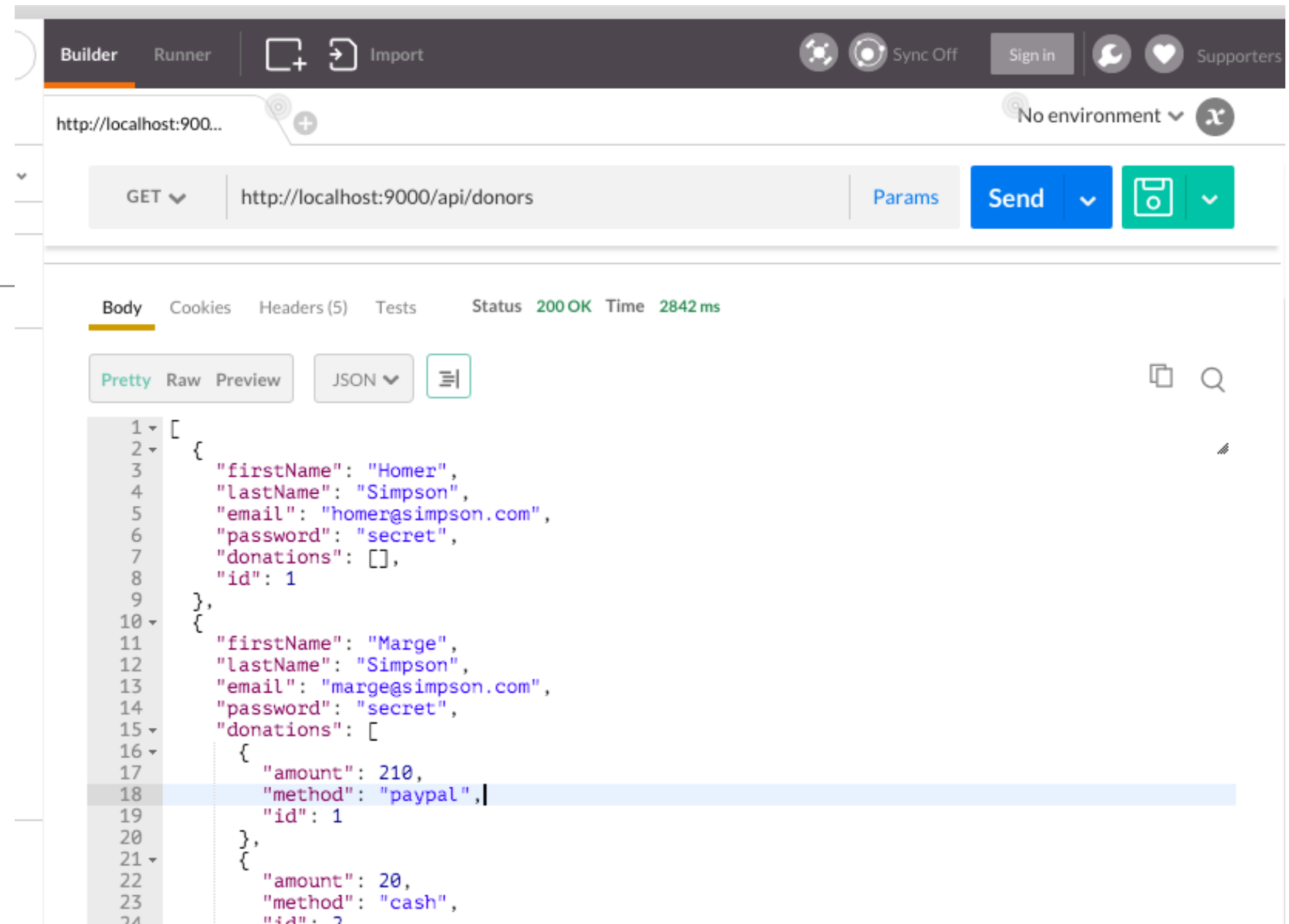
```
Donor(lisa):
  usaCitizen: true
  firstName: Lisa
  lastName: Simpson
  email: lisa@simpson.com
  password: secret
  donations:
    - d
```


Routes

GET	/api/donors	DonorsAPI.getAllDonors
GET	/api/donors/{id}	DonorsAPI.getDonor
POST	/api/donors	DonorsAPI.createDonor
DELETE	/api/donors/{id}	DonorsAPI.deleteDonor
DELETE	/api/donors	DonorsAPI.deleteAllDonors
GET	/api/donations	DonationsAPI.getAllDonations
DELETE	/api/donations	DonationsAPI.deleteAllDonations
GET	/api/donors/{id}/donations	DonationsAPI.getDonations
GET	/api/donors/{id}/donations/{donationId}	DonationsAPI.getDonation
POST	/api/donors/{id}/donations	DonationsAPI.createDonation
DELETE	/api/donors/{id}/donations/{donationId}	DonationsAPI.deleteDonation

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

Postman



- `http://localhost:9000/api/donors`

```
30   },
31   ],
32   "id": 2
33 },
34 {
35   "firstName": "Lisa",
36   "lastName": "Simpson",
37   "email": "lisa@simpson.com",
38   "password": "secret",
39   "donations": [
40     {
41       "amount": 10,
42       "method": "paypal",
43       "id": 4
44     }
45   ],
46   "id": 3
47 },
```

```
[
  {
    "firstName": "Homer",
    "lastName": "Simpson",
    "email": "homer@simpson.com",
    "password": "secret",
    "donations": [],
    "id": 1
  },
  {
    "firstName": "Marge",
    "lastName": "Simpson",
    "email": "marge@simpson.com",
    "password": "secret",
    "donations": [
      {
        "amount": 210,
        "method": "paypal",
        "id": 1
      },
      {
        "amount": 20,
        "method": "cash",
        "id": 2
      },
      {
        "amount": 330,
        "method": "cash",
        "id": 3
      }
    ]
  },
  {
    "id": 2
  },
]
```

```
{
  "firstName": "Lisa",
  "lastName": "Simpson",
  "email": "lisa@simpson.com",
  "password": "secret",
  "donations": [
    {
      "amount": 10,
      "method": "paypal",
      "id": 4
    }
  ],
  "id": 3
},
]
```

```
Donation(a):
  amount : 210
  method : paypal

Donation(b):
  amount : 20
  method : cash

Donation(c):
  amount : 330
  method : cash

Donation(d):
  amount : 10
  method : paypal

Donor(homer):
  firstName: Homer
  lastName: Simpson
  email: homer@simpson.com
  password: secret

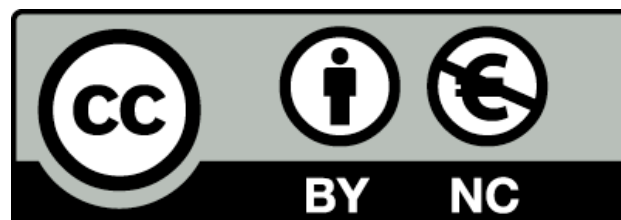
Donor(marge):
  firstName: Marge
  lastName: Simpson
  email: marge@simpson.com
  password: secret
  donations:
    - a
    - b
    - c

Donor(lisa):
  firstName: Lisa
  lastName: Simpson
  email: lisa@simpson.com
  password: secret
  donations:
    - d
```

createDonation example

POST	/api/donors/{id}/donations	DonationsAPI.createDonation
------	----------------------------	-----------------------------

```
public static void createDonation(Long id, JsonElement body)
{
    Donor donor = Donor.findById(id);
    Donation donation = gson.fromJson(body.toString(), Donation.class);
    donation.id = null;
    donor.donations.add(donation);
    donor.save();
    renderJSON (gson.toJson(donation));
}
```



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/>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

