Pages / Helium Documentation Home / Helium Beginner's Tutorial

# Lesson 21: The Identity Object

Created by Adriaan Klue, last modified by Jacques Marais on Mar 27, 2018

As a System Admin I want to view the names and surnames of System Admins as they have entered it on their user profiles.

- · Lesson Outcomes
- New & Modified App Files
- Definition
- · Populating the Identity Object
- Accessing Identity Attributes from Custom Objects
- · Accessing Identity Attributes from the Identity Object
- · Lesson Source Code

#### **Lesson Outcomes**

After this lesson you should know how to use the Identity built-in object.

## New & Modified App Files

- ./web-app/presenters/user\_managment/SystemAdminUserMgmt.mez
- ./web-app/views/user\_management/SystemAdminUserDetails.vxml

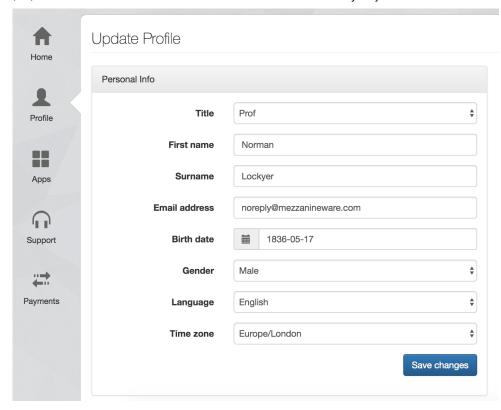
### Definition

The Identity object is an implicit interface that is implemented by every persistence object in your application that has a <code>@Role</code> annotation. This object has the following implicit declaration:

```
1 object Identity {
2    string _firstNames;
3    string _nickName;
4    string _surname;
5    string _locale;
6    string _timeZone;
7 }
```

# Populating the Identity Object

Attributes of the Identity object are automatically populated from the information users update on their profiles. You see this profile screen when you exit the app by clicking on "More Apps" on the right-hand corner mouseover menu. These details apply server wide, and are not app specific.



Outside of above context, the Identity object and all of its attributes are read-only.

# Accessing Identity Attributes from Custom Objects

When displaying a system administrator's details (see Lesson 2 and Lesson 3), we'll give precedence to a user's profile information (which they would have entered themselves) over the values entered when they were invited. Since all persistent custom objects with <code>@Role</code> annotations implement the <code>Identity</code> object we can access the <code>Identity</code> attributes directly from such custom objects.

We can therefore change the SystemAdminUserMgmt unit and SystemAdminUserDetails view as below to check first if the user has entered a name and/or surname on his profile, and if so, display that instead.

```
1
    string systemAdminFirstName;
 2
    string systemAdminLastName;
 3
 4
 5
 6
    void initViewDetails() {
 7
       if (selectedSystemAdmin._firstNames != null) {
 8
           systemAdminFirstName = selectedSystemAdmin._firs
 9
       } else {
10
           systemAdminFirstName = selectedSystemAdmin.first
11
12
       if (selectedSystemAdmin._surname != null) {
13
           systemAdminLastName = selectedSystemAdmin._surna
14
       } else {
15
           systemAdminLastName = selectedSystemAdmin.lastNa
16
```

```
17
     }
     <view label="view_heading.system_admin_user_details" u</pre>
 1
 2
 3
 4
 5
         <info label="info.first_name">
 6
             <binding variable="systemAdminFirstName" />
 7
         </info>
 8
         <info label="info.last_name">
 9
             <binding variable="systemAdminLastName" />
10
         </info>
```

## Accessing Identity Attributes from the Identity Object

As mentioned earlier, the Identity object is read-only. It <u>cannot</u> be directly instantiated by Identity:new(). However, the compiler can implicitly convert any custom persistent object with a **@Role** annotation to an Identity object instance. The following changes to SystemAdminUserMgmt would have achieved the same as the previous approach:

```
1
    string systemAdminFirstName;
 2
    string systemAdminLastName;
 3
 4
 5
 6
    void initViewDetails() {
 7
       Identity identity = selectedSystemAdmin;
 8
       if (selectedSystemAdmin._firstNames != null) {
 9
           systemAdminFirstName = identity._firstNames;
       } else {
10
11
           systemAdminFirstName = identity.firstName;
12
       }
13
       if (selectedSystemAdmin._surname != null) {
14
           systemAdminLastName = identity._surname;
       } else {
15
16
           systemAdminLastName = identity.lastName;
17
       }
18
    }
```

### Lesson Source Code

Lesson 21.zip

No labels