

# MBaaS in iOS: Backendless - User Services

Mobile Computing - iOS

# Objectives

- Students will be able to:
  - utilize Backendless' User Service APIs to save/update/retrieve/delete objects and manage users
  - regulate access to data based on roles and users

# User Service API

- Backendless' User Service handles:
  - User Registration
  - User Login
  - Password Recovery
  - User Logout
- It defines a **BackendlessUser** class, which can have additional properties associated with it

Note: Since the User Service is courtesy of Backendless, there is no need to demo its many features when presenting your final project. Spend your limited demo time on parts of the app that your team has created.

# User Registration

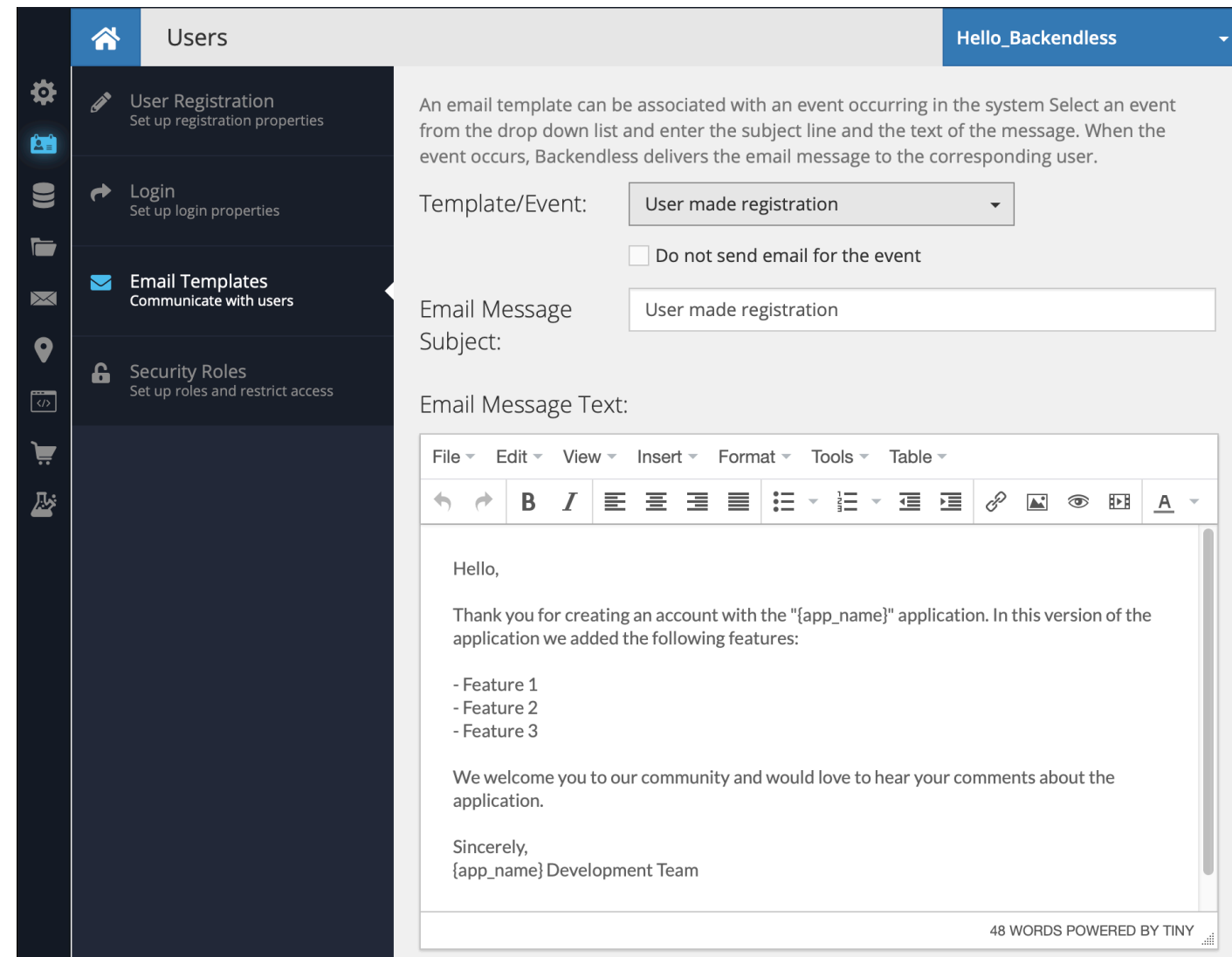
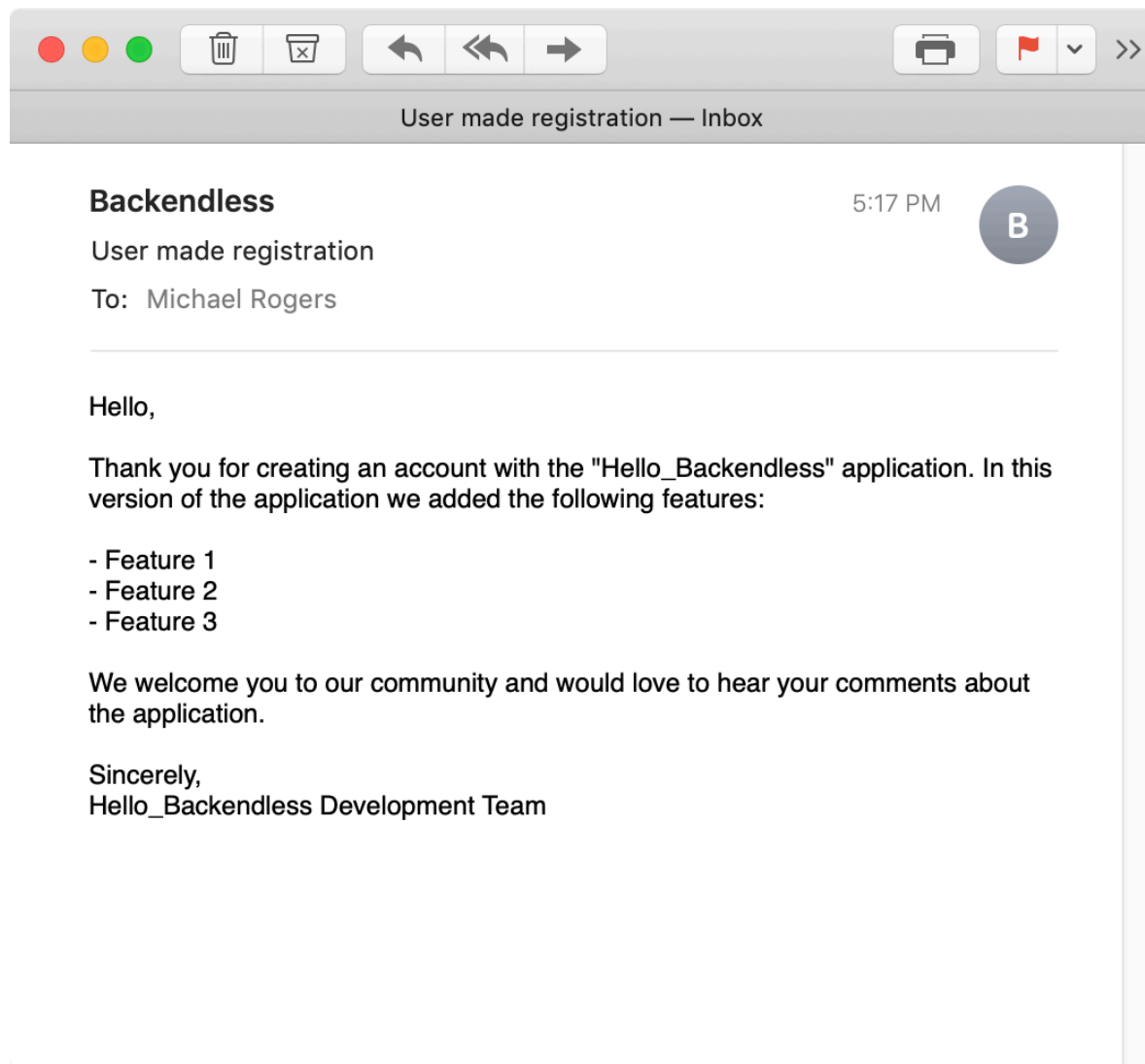
```
let backendless = Backendless.sharedInstance()!

func registerUser() {
    // do not forget to call Backendless.initApp when your app initializes
    let user = BackendlessUser()
    user.setProperty("email", object: "james2.bond@mi6.co.uk")
    user.password = "111"
    backendless.userService.register(user,
        response: {
            (registeredUser : BackendlessUser?) -> Void in
                print("User registered \(registeredUser?.value(forKey: "email"))")
        },
        error: {
            (fault : Fault?) -> Void in
                print("Server reported an error: \(fault?.description)")
            })
}
```

It is possible to set multiple properties at once with a dictionary -- [see the docs](#) for details.

```
func setProperties(_ props: [String : Any]!) -> Void
```

# User Registration Email



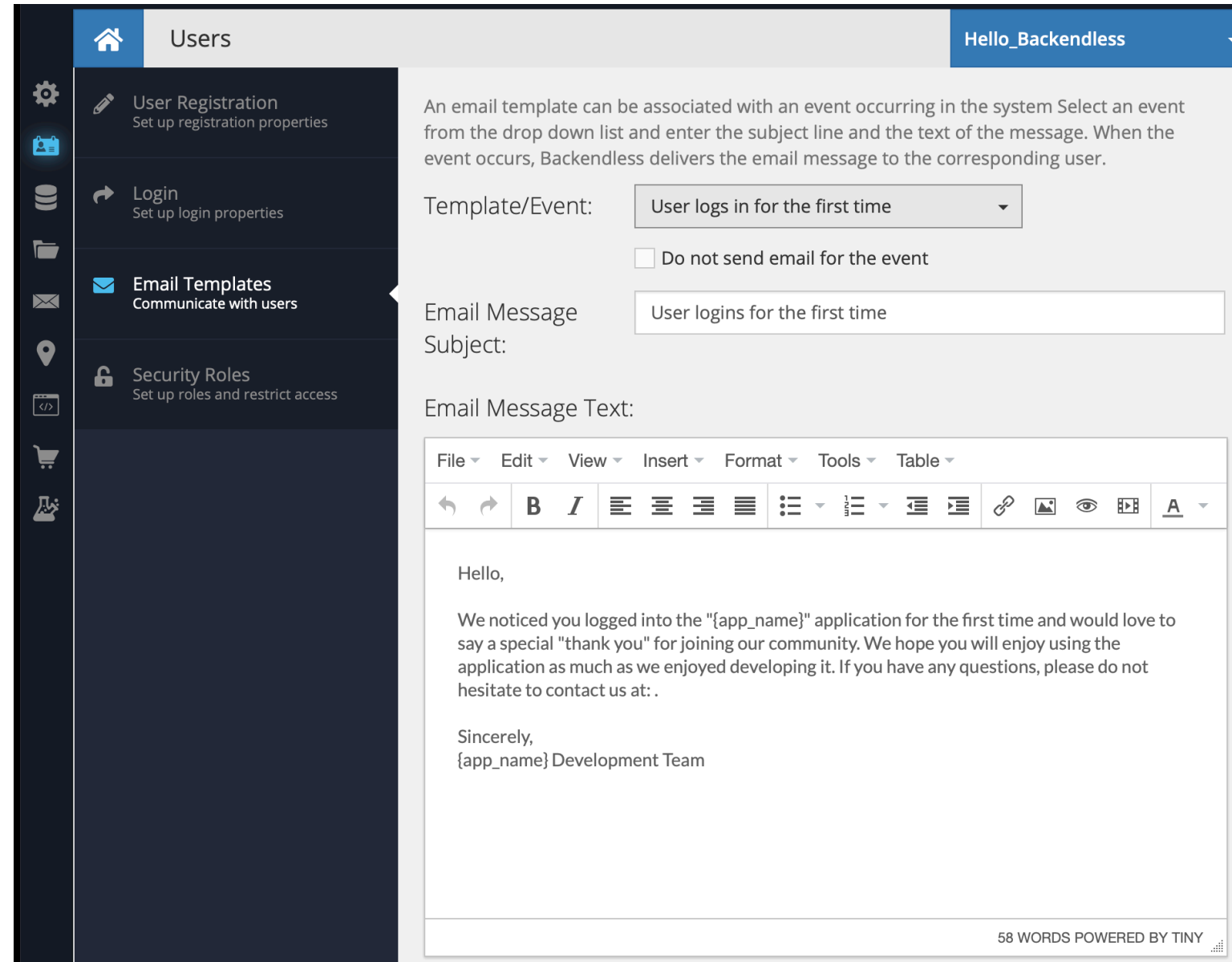
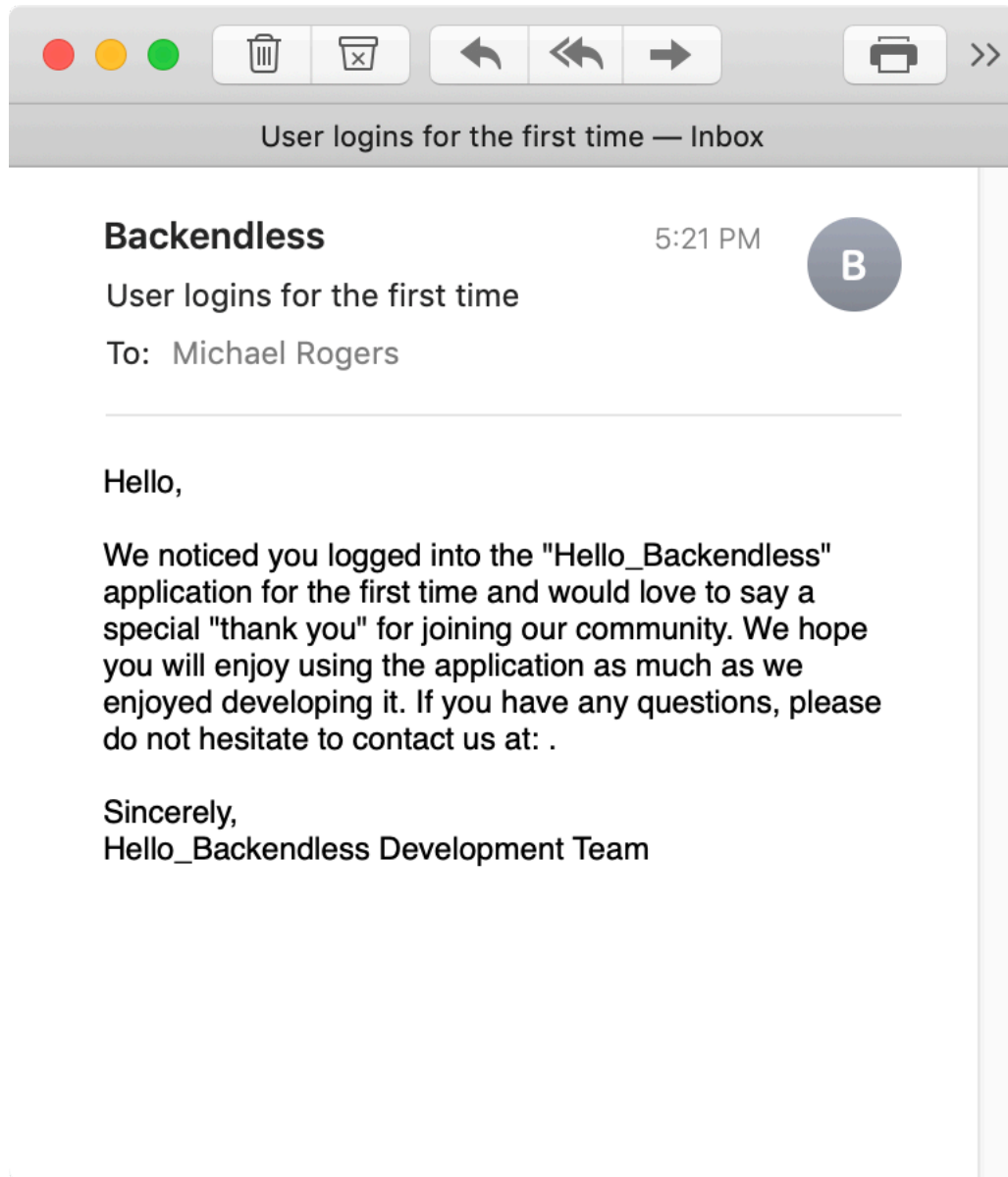
Configure email templates as appropriate

# User Login

```
let backendless = Backendless.sharedInstance()!
```

```
func loginUser () {  
    backendless.userService.login("james.bond@mi6.co.uk", password: "iAmWatchingU",  
                                  response: {  
                                      (loggedUser : BackendlessUser?) -> Void in  
                                          print("User logged in")  
                                  },  
                                  error: {  
                                      (fault : Fault?) -> Void in  
                                          print("Error logging in: \(fault?.description)")  
                                  })  
}
```

# User Login



# Password Recovery

```
let backendless = Backendless.sharedInstance()

backendless.userService.restorePassword("james.bond@mi6.co.uk",
    response: {
        (result : Any) -> Void in
        print("Please check your email inbox to reset your password")
    },
    error: {
        (fault : Fault?) -> Void in
        print("Server reported an error: \(fault?.description)")
    })
```



# Security Roles

- Backendless controls who can access what using Security Roles

| Security Role        | Assigned to user when an API request is made by:                            |
|----------------------|---|
| NotAuthenticatedUser | ...a user who is not logged in.   |
| AuthenticatedUser    | ...a user who is logged in.   |
| SocialUser           | ...a user who is logged in via a social network (Facebook, Twitter, Google) |
| FacebookUser         | ...a user who is logged in via Facebook                                     |
| GooglePlusUser       | ...a user who is logged in via Google                                       |
| TwitterUser          | ...a user who is logged in via Twitter                                      |

# Permissions

- Permissions are arranged in a hierarchy: anything lower in the hierarchy supersedes that higher up.

Global Permissions

- applies to all assets and resources


Asset Container Permissions

- applies to specific data tables, file directories, etc.

Asset Permissions

- applies to specific data records, files within file directories, etc.

# Setting Global Permissions

Users

Manage

Users

Data

Files


Messaging


Geolocation


Business Logic


Marketplace

Code Generation

User Registration  
Set up registration properties

Login  
Set up login properties






















Email Templates  
Communicate with users

Security Roles  
Set up roles and restrict access

Delete SelectedAdd Role

| <input type="checkbox"/> | ROLE NAME            |
|--------------------------|----------------------|
| <input type="checkbox"/> | ASUser               |
| <input type="checkbox"/> | AndroidUser          |
| <input type="checkbox"/> | AuthenticatedUser    |
| <input type="checkbox"/> | DotNetUser           |
| <input type="checkbox"/> | FacebookUser         |
| <input type="checkbox"/> | GooglePlusUser       |
| <input type="checkbox"/> | IOSUser              |
| <input type="checkbox"/> | JSUser               |
| <input type="checkbox"/> | NotAuthenticatedUser |
| <input type="checkbox"/> | RestUser             |
| <input type="checkbox"/> | ServerCodeUser       |
| <input type="checkbox"/> | SocialUser           |
| <input type="checkbox"/> | TwitterUser          |

Permission for the ASUser Role

| DATA  | FILES   | MESSAGING   | GEO  |
|---|---|---|--|
| <br>Create             | <br>Read         | <br>Publish      | <br>Add Geopoint        |
| <br>Update            | <br>Delete      | <br>Subscribe   | <br>Add Categories     |
| <br>Find             | <br>Write      | <br>Send email | <br>Get categories    |
| <br>Remove           | <br>Permission |   | <br>Remove categories |
| <br>Describe         |   |   | <br>Search            |
| <br>Permission       |   |   |  |
| <br>Load relations   |   |   |  |
| <br>Add/Set relation |   |   |  |
| <br>Delete relation  |   |   |  |

# Setting Asset Container\*

## Permissions

The screenshot shows the Backendless web interface. The top navigation bar includes the Backendless logo, buttons for '+ CREATE APP' and 'DOWNLOAD PROJECT TEMPLATE', and user information 'Hi, MP Rogers!'. The left sidebar contains icons for Manage, Users, Data, Files, Messaging, Geolocation, Business Logic, Marketplace, and Code Generation. The main content area is titled 'Data Management' and shows the 'TestTable' selected. The 'PERMISSIONS' tab is active, displaying a table of permissions for various roles.

| ROLE NAME            | CREATE | UPDATE | FIND | REMOVE | DESCRIBE | PERMISSION | ADD/SET RELATION | DELETE RELATION | LOAD RELATIONS |
|----------------------|--------|--------|------|--------|----------|------------|------------------|-----------------|----------------|
| ASUser               | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| AndroidUser          | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| AuthenticatedUser    | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| DotNetUser           | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| FacebookUser         | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| GooglePlusUser       | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| IOSUser              | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| JSUser               | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| NotAuthenticatedUser | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| RestUser             | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| ServerCodeUser       | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| SocialUser           | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |
| TwitterUser          | ✓      | ✓      | ✓    | ✓      | ✓        | ✓          | ✓                | ✓               | ✓              |

\*an asset container is a data table or file folder

# Setting Asset\* Permissions

The screenshot shows the 'Data Management' interface. On the left, a sidebar contains icons for home, settings, app tables, system data, and a location pin. The 'APP TABLES' section is active, showing 'TestTable' with 1 record. The main area has tabs for 'DATA BROWSER', 'SCHEMA', 'PERMISSIONS', 'REST CONSOLE', and 'CONFIGURATION'. The 'DATA BROWSER' tab is selected, displaying a table with columns: 'foo' (STRING | MAX LENGTH: 500), 'objectId' (STRING\_ID), and 'ownerId' (STRING). A search bar and checkboxes for 'SQL Search' and 'Real-time' are at the top. A record with 'objectId' '3D68FC67-A97C-CA35-FF35-B0F11A595500' is highlighted. A lock icon is visible in the 'foo' column of this record. An arrow points from a pink callout box to this lock icon.

click here to change record-specific permissions

Permissions for : TestTable / 3D68FC67-A97C-CA35-FF35-B0F11A595500

USERS PERMISSIONS ROLES PERMISSIONS

| ROLE NAME         | UPDATE ↺ | FIND ↺ | REMOVE ↺ | ADD/SET RELATION ↺ | DELETE RELATION ↺ | LOAD RELATIONS |
|-------------------|----------|--------|----------|--------------------|-------------------|----------------|
| ASUser            | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |
| AndroidUser       | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |
| AuthenticatedUser | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |
| DotNetUser        | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |
| FacebookUser      | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |
| GooglePlusUser    | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |
| IOSUser           | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |
| JSUser            | ✓        | ✓      | ✓        | ✓                  | ✓                 | ✓              |

\*an asset is a data record or file in a file folder

# Resources

- [backendless.com](https://backendless.com)