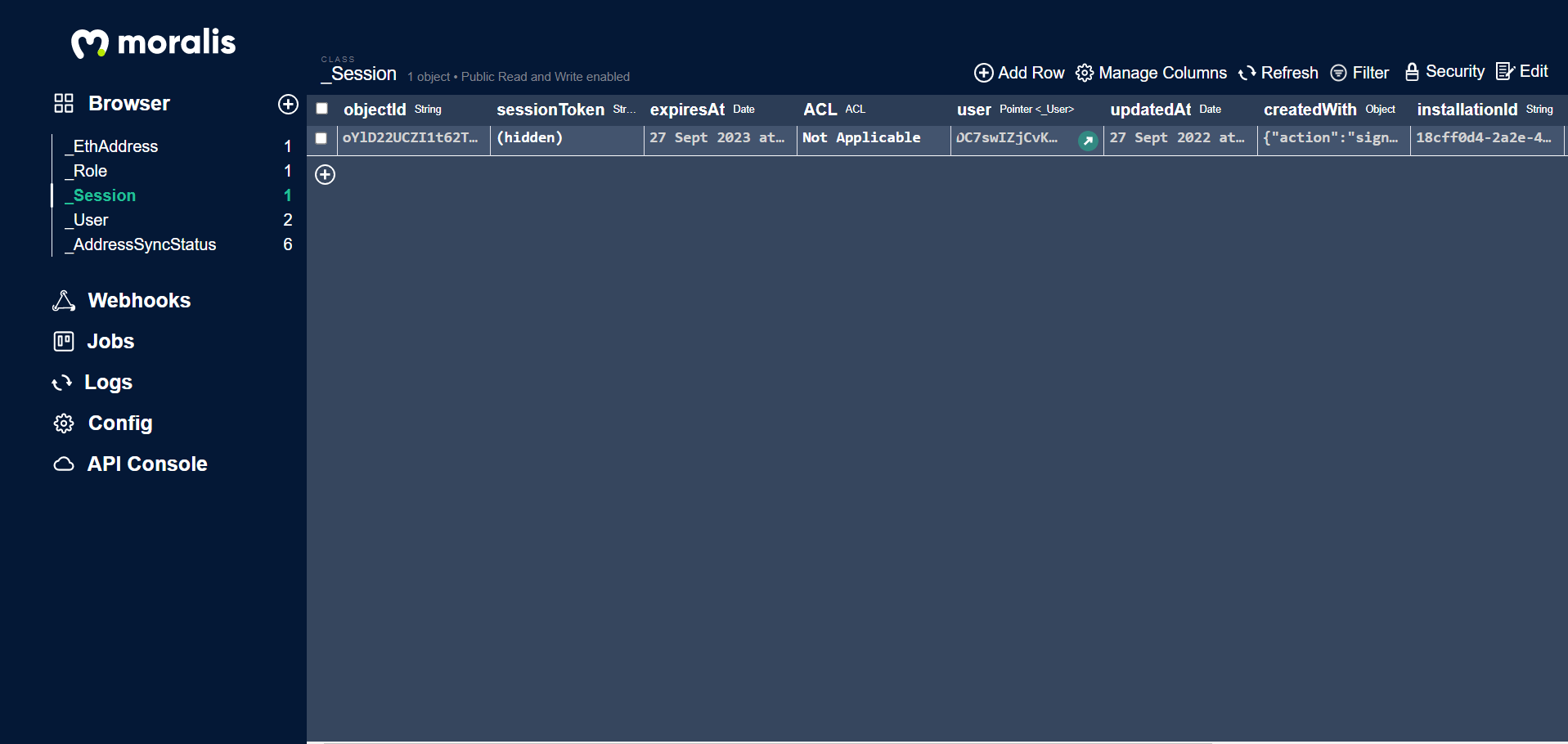
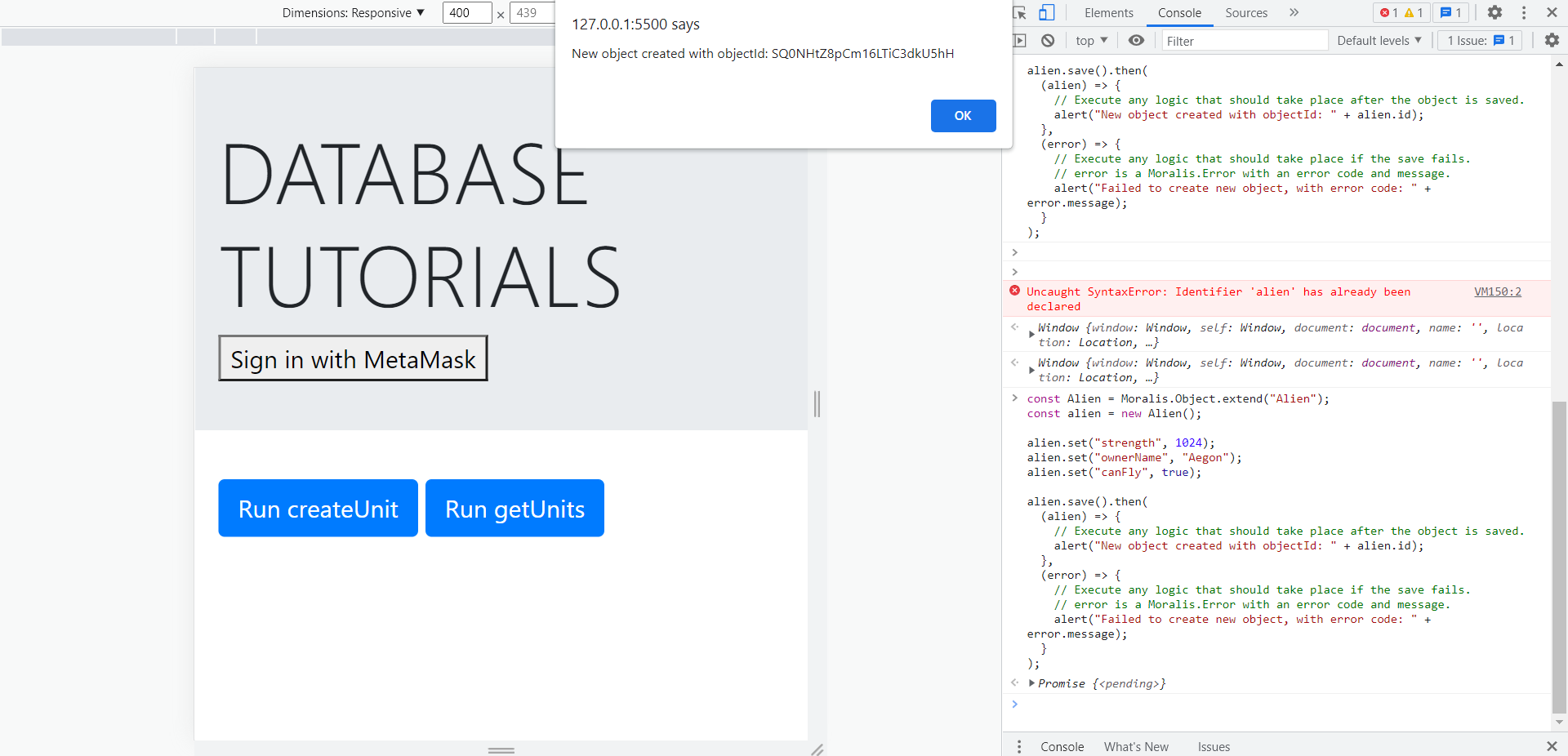
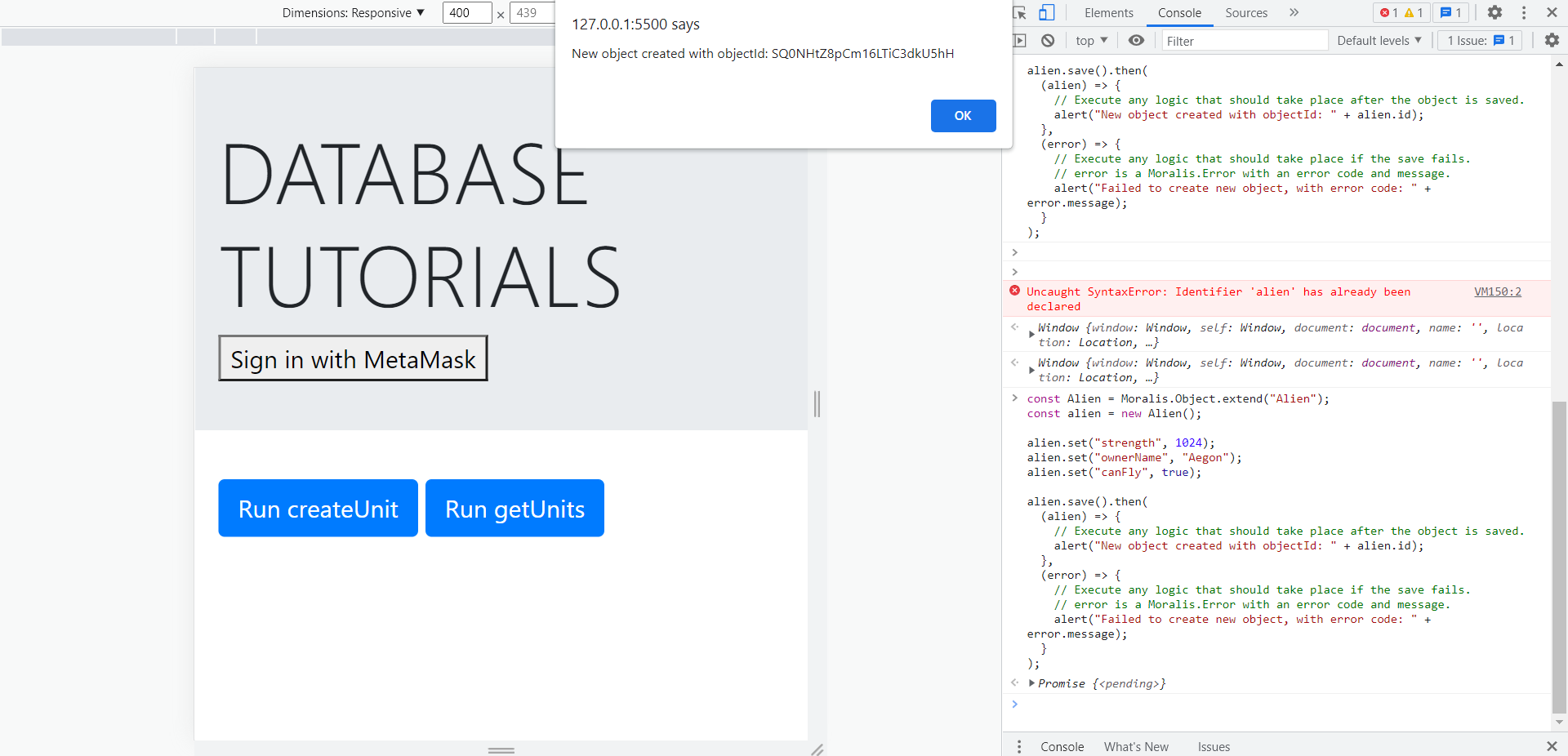
601

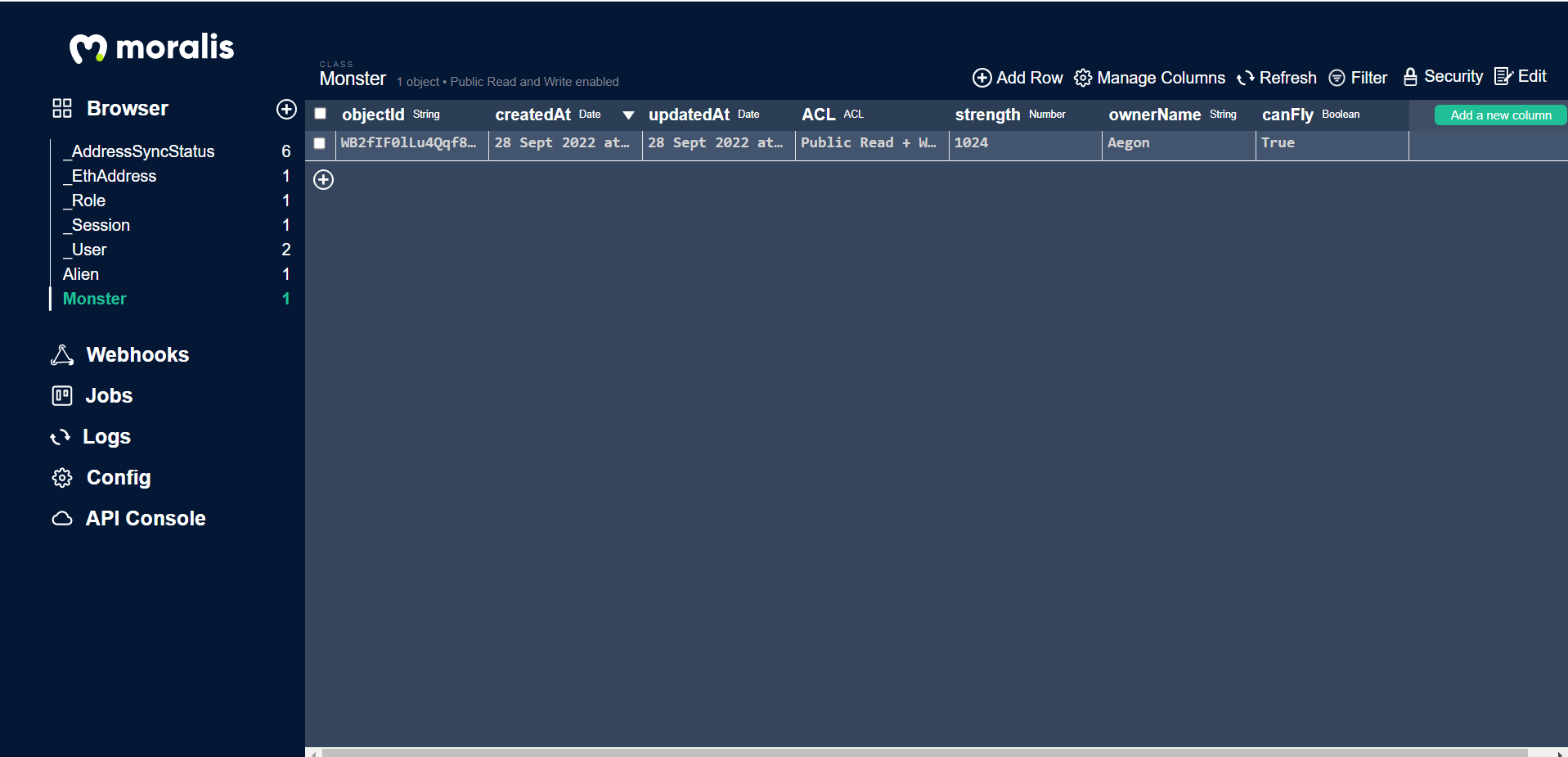
Initial database:



Created objects and saving:

* Monster
* Alien
* 
* 

Database which now has monster as well as alien



Console Code

*const Alien = Moralis.Object.extend("Alien");*

*const alien = new Alien();*

*alien.set("strength", 1024);*

*alien.set("ownerName", "Aegon");*

*alien.set("canFly", true);*

*alien.save().then(*

*(alien) => {*

*// Execute any logic that should take place after the object is saved.*

*alert("New object created with objectId: " + alien.id);*

*},*

*(error) => {*

*// Execute any logic that should take place if the save fails.*

*// error is a Moralis.Error with an error code and message.*

*alert("Failed to create new object, with error code: " + error.message);*

*}*

*);*

Added a new entry in the table Alien with name Tyrion

Code:

*const Alien = Moralis.Object.extend("Alien");*

*const alien = new Alien();*

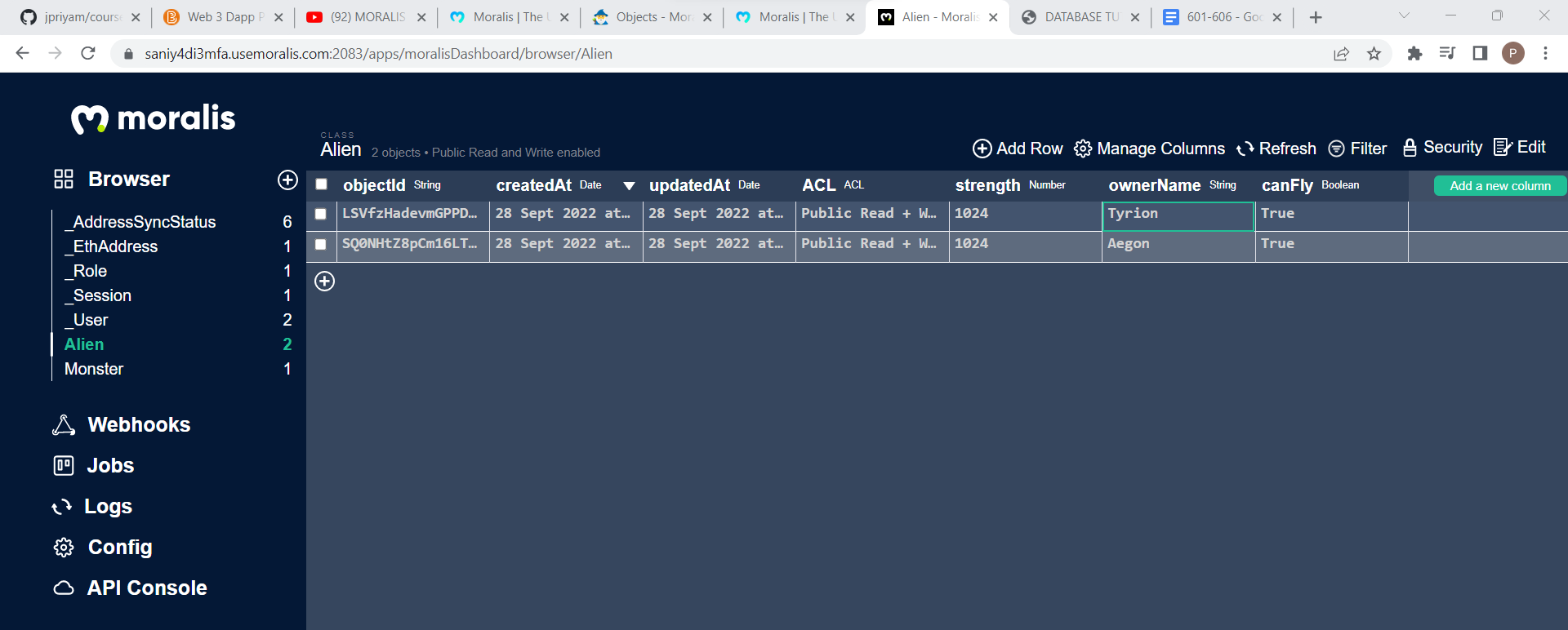
*alien.set("strength", 1024);*

*alien.set("ownerName", "Tyrion");*

*alien.set("canFly", true);*

*alien.save();*

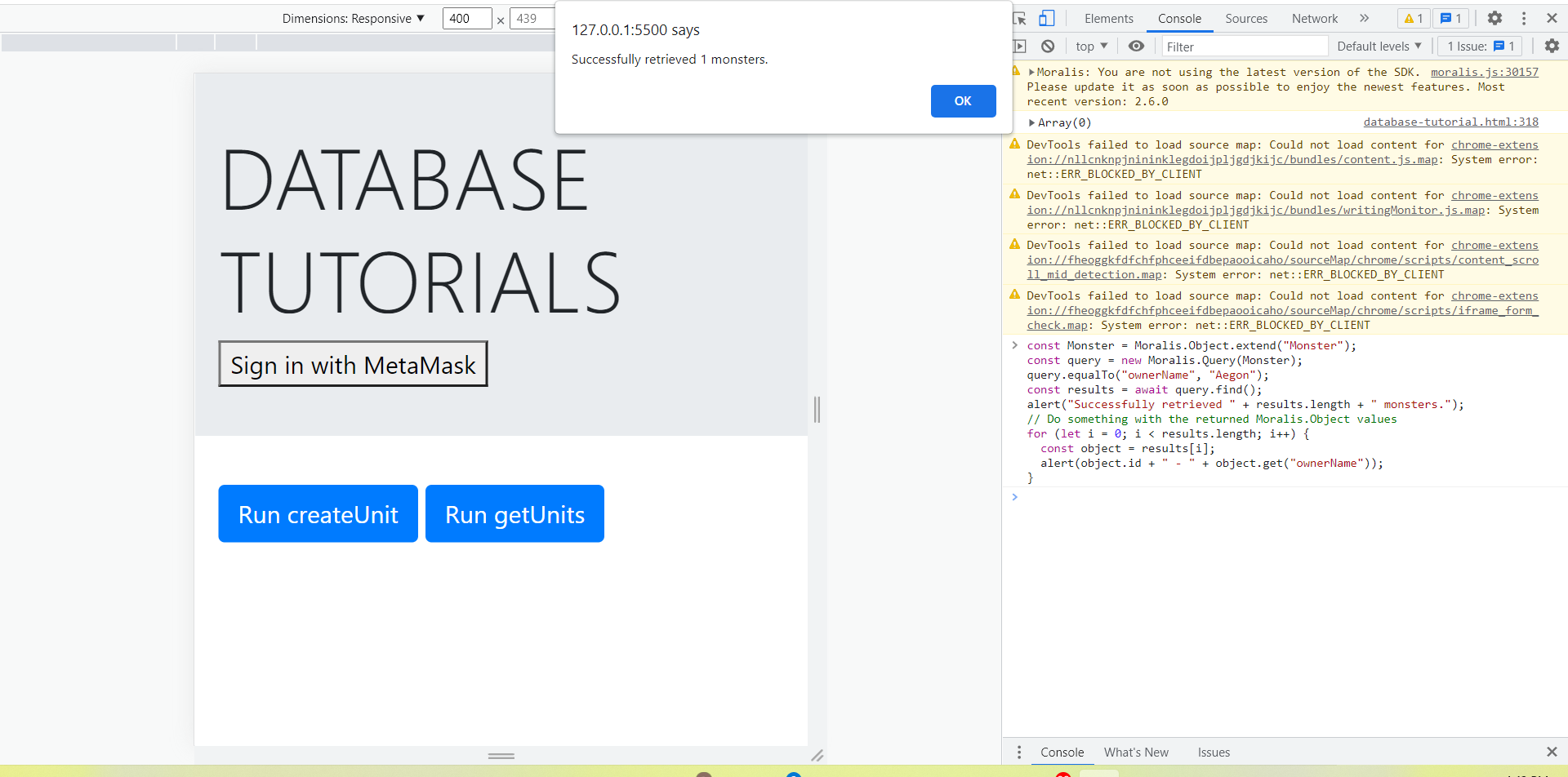
Output:



602

Querying data in order to retrieve information:

Console code to retrieve monster object:



Code

*const Alien = Moralis.Object.extend("Alien");*

*const query = new Moralis.Query(Alien);*

*query.equalTo("ownerName", "Tyrion");*

*const results = await query.find();*

*alert("Successfully retrieved " + results.length + " aliens.");*

*// Do something with the returned Moralis.Object values*

*for (let i = 0; i < results.length; i++) {*

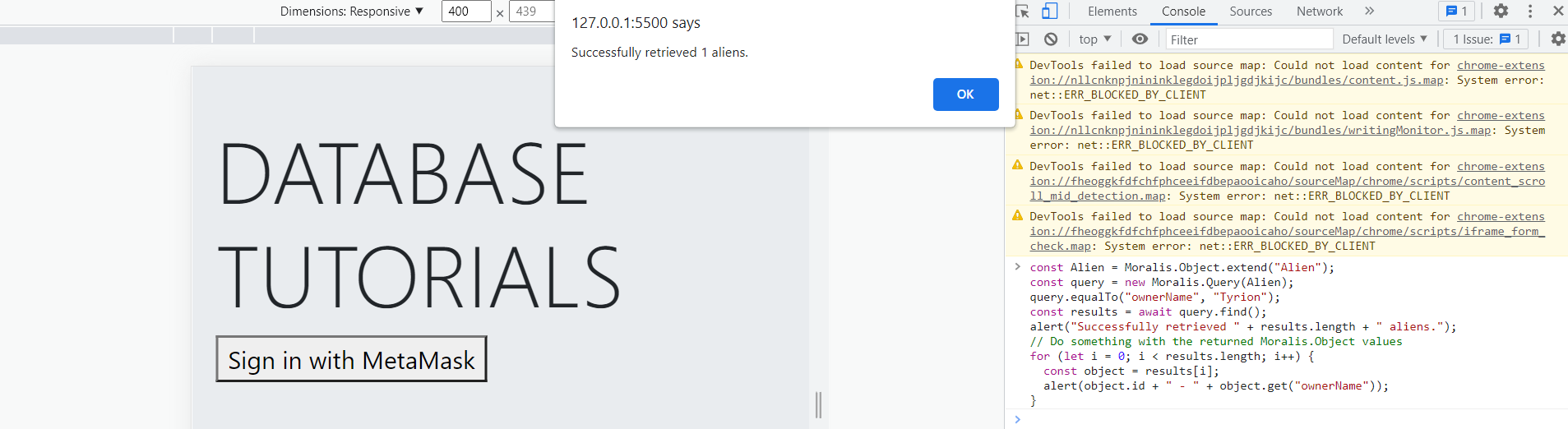
*const object = results[i];*

*alert(object.id + " - " + object.get("ownerName"));*

*}*

Since there is one entry, it retrieved one monster.

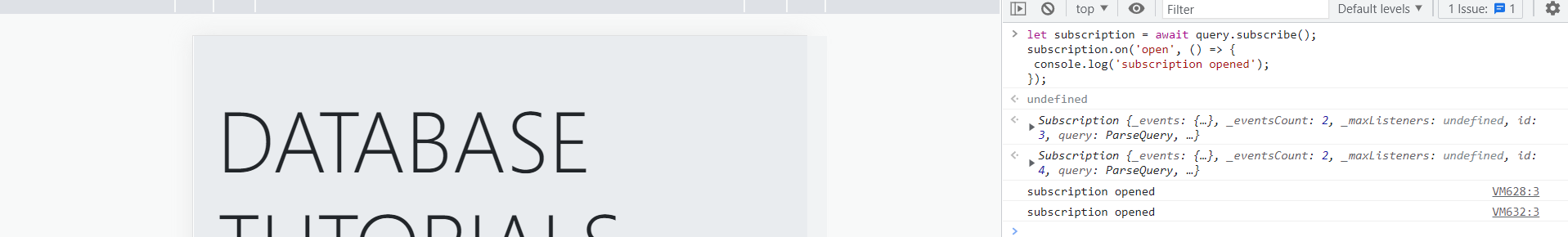
Now, we try to retrieve the Alien named Tyrion



603

Live queries basically involve event handling

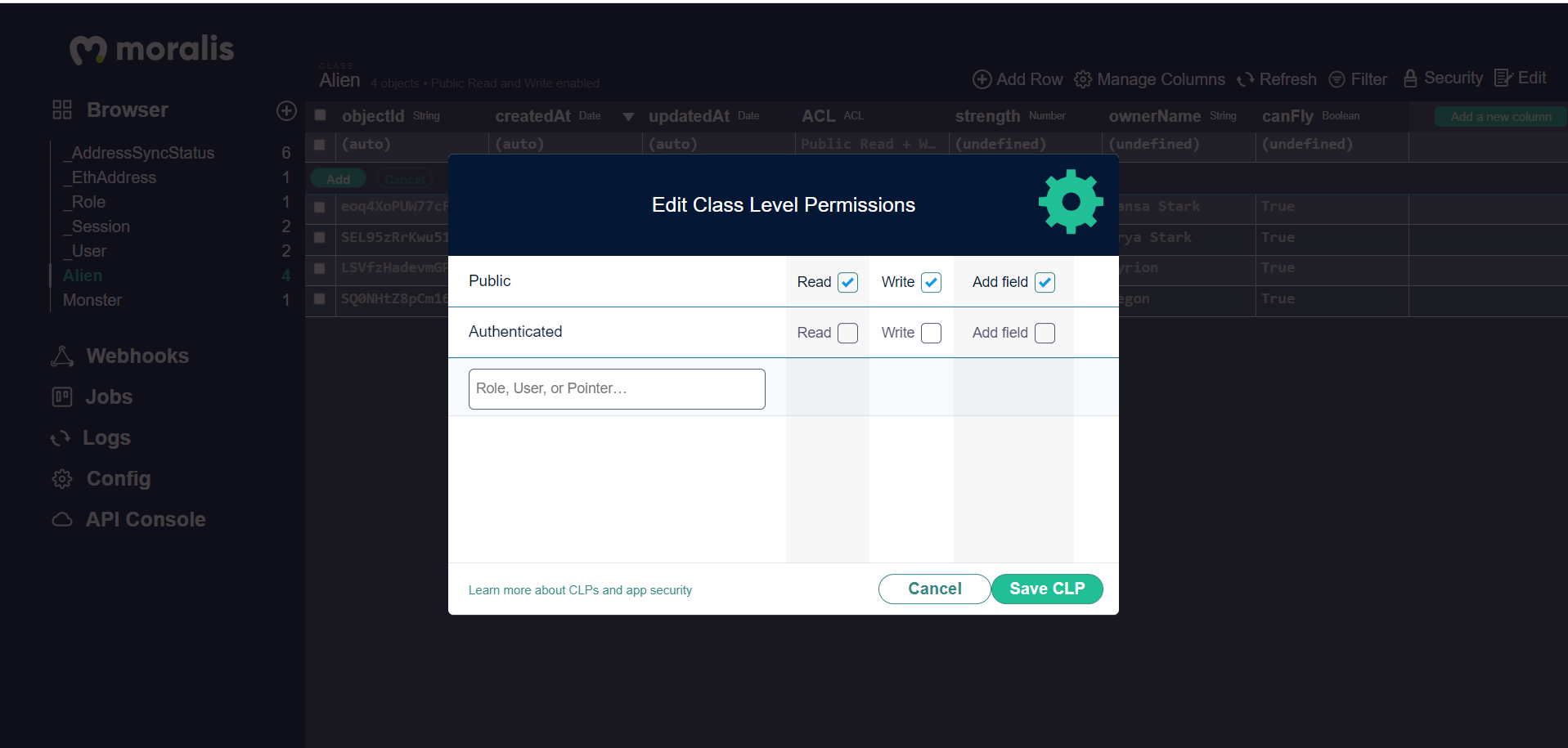
Open event



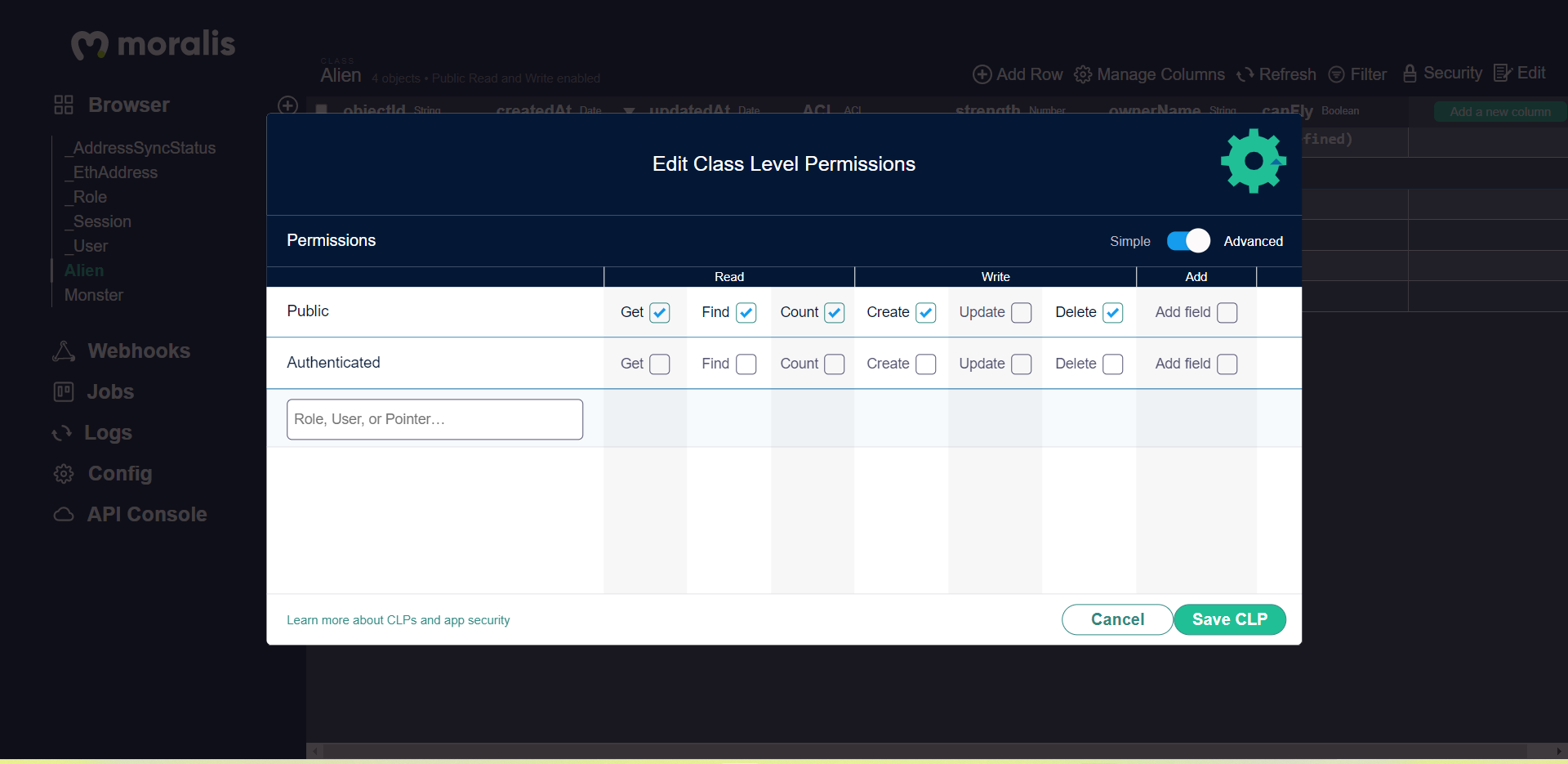
604

Security

Class level permissions



Advance level settings in terms of security



605

// creating another role

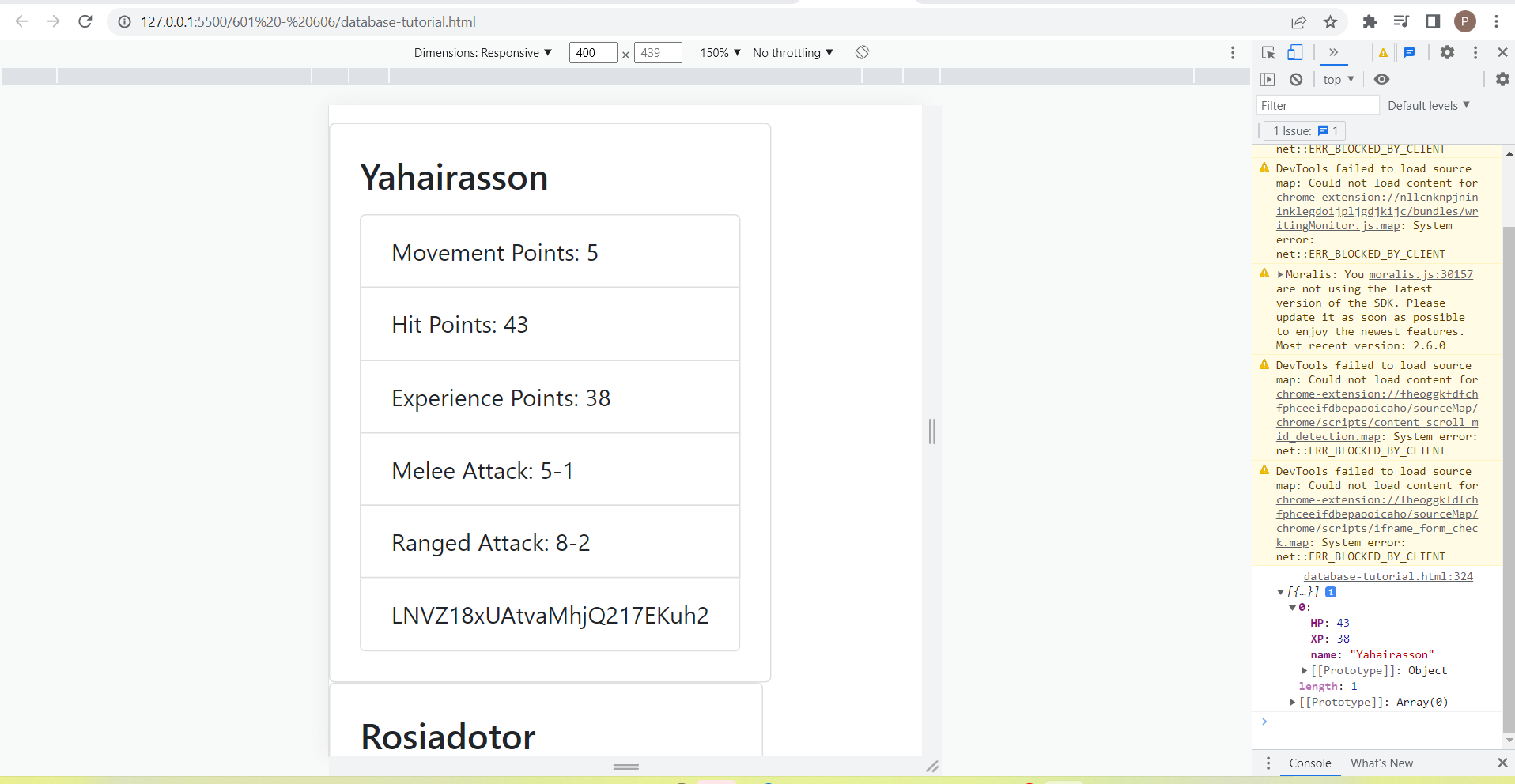
const role = new Moralis.Role("test Role", roleACL);

role.getUsers().add(usersToAddToRole);

role.getRoles().add(rolesToAddToRole);

role.save();

Query invoked when app runs



606



* Raw json objects are stored and can be edited.
* The objectId, createdAt, updatedAt fields cannot be edited (these are set automatically).
* The value “(empty)” denotes that the field has not been set for that particular object (this is different than null).
* You can remove a field’s value by hitting your "Delete" key while the value is selected.