

Login Module

This login system works by sending the relevant information to php scripts on your server, which then send it to the MySQL database.

Features

- Registration
- Email verification and activation
- Password recovery and reset
- Remember functionality
- Easy to use API
- Example interface included

Base Setup

MySQL Databases allow you to store lots of information in an easy to access manner. The databases themselves are not easily read by humans. To use a database, you'll need to create it. Only MySQL Users that have privileges to access a database can read from or write to that database.

You should follow the instructions from your hosting provider on how to create a database, because it can vary. After you created the database and have setuped a user granded with privileges, you can start to import the login.sql script that is included in the php folder. Most hosting provider have a tool called phpMyAdmin. This tool allows you to easily import a tables to your database. In the top menu there is a button "Import", clicking on it will give you all information that are required to import the login.sql script.

The next step is to upload the php scripts and the crossdomain.xml file. Please note that the crossdomain.xml file needs to be in the root folder. Before you can run the login system, you should fill out the config.php file with the created database information.

It is mostly done. The last step is to setup the LoginSettings asset that is located in the Resources folder. The most important setup is here, your own server adress with a link to the folder where your php files are located. I recommend to type this link into your browser to verify that you do not have a spelling mistake and then just copy and paste it into the server adress field.

API

`LoginSystem.LoginAccount(string username, string password, UnityAction<bool> callback)`

Description

Try to login with an existing account. The response callback should contain a bool parameter that will tell you whether the login was successful or failed.

`LoginSystem.RecoverPassword(string email, UnityAction<bool> callback)`

Description

Try to recover a password. The response callback should contain a bool parameter that will tell you whether the password recovery was successful. A new generated password will be send to that email, because the password is stored in one way hash algorithm.

`LoginSystem.ResetPassword(string username, string password, UnityAction<bool> callback)`

Description

Resets the old password with a new password. This is mostly used when the player is logged in.

`LoginSystem.CreateAccount(string username, string password, string email, UnityAction<bool> callback)`

Description

Create a new account. The response callback should contain a bool parameter that will tell you wheather the new account was created or if the username already exists in the database.

`LoginSystem.ValidateEmail(string email)`

Description

A simple regex helper method, to check if the email is in correct format.

Contact

Name: Konstantin Janson

Email: konstantin.janson@freenet.de

Support Forum: <http://zerano-unity3d.com/Forums/forum-10.html>