WordPress Achievements

WordPress Achievements truly is a very simple product to understand and use. Once you understand how it works you will be able to use it without any hassles. For that reason let me give you background info first and then end off with how you can get started using the kit

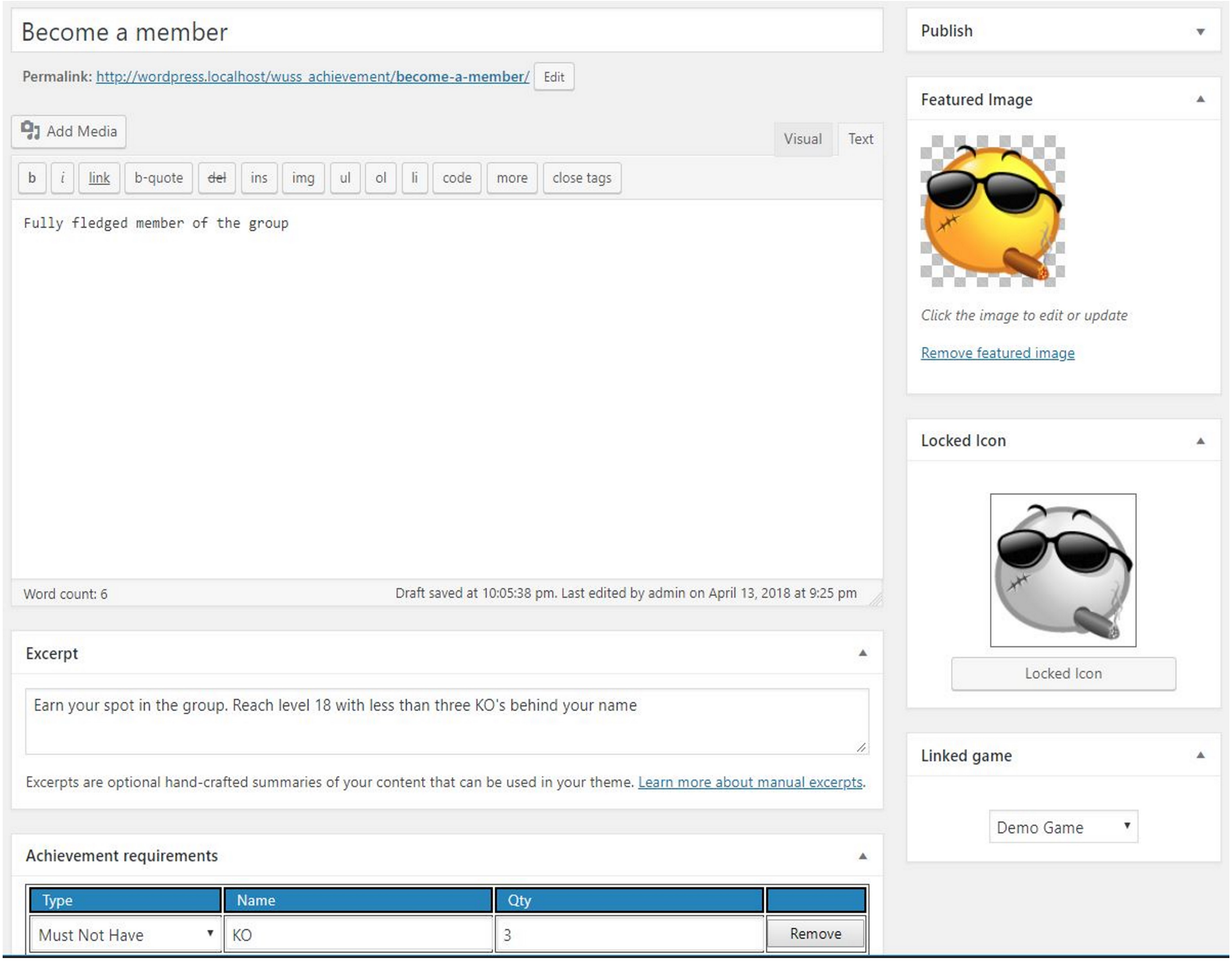
The basic premise is as simple as this

1. You go to your dashboard and create a new achievement, giving it a graphic to display when it has been achieved and a graphic to display when it has not. Next, pick what game this achievement belongs to and you now have a bare-bones achievement. The game identifier and the two graphics are the only things that are really required. Everything else is optional. I would recommend you give it a name also, though, just to make it easier for you to find when you want to modify it.

   Should you want to, you can also add:
   - text to display when the achievement has been unlocked
   - text to explain to the player how they can unlock the achievement
   - any number of requirements to test if an achievement can be unlocked

2. Inside your game you decide for yourself when someone deserves an achievement to be unlocked and if you feel that is the case then all you need to do is call *WUAchieve.UnlockAchievement* and pass it the achievement's id. The achievement is now unlocked.

That's it. That easy.
Like I said. Very simple to understand and use.



Become a member

Permalink: http://wordpress.localhost/wuss_achievement/become-a-member/ Edit

```
Fully fledged member of the group
```

Word count: 6

Draft saved at 10:05:38 pm. Last edited by admin on April 13, 2018 at 9:25 pm

Excerpt

Earn your spot in the group. Reach level 18 with less than three KO's behind your name

Excerpts are optional hand-crafted summaries of your content that can be used in your theme. Learn more about manual excerpts.

Achievement requirements

Type	Name	Qty	
Must Not Have ▾	KO	3	Remove

Featured Image

Click the image to edit or update

Remove featured image

Locked Icon

Linked game

Demo Game ▾

There HAS to be more to it than that, surely?

No, not really. You create the achievement, you get the ID, you decide when to award it and when you are ready you just call that function and on the back end everything is hunky dory. That now only leaves the front end… i.e. how do you show the achievements in Unity?

That, however, depends on each individual game. I won't tie you down to a single way of how to display the awards. To that end I provide you with a host of functions so that you can decide for yourself how you want to use and display achievements in your game.

Some games show only a graphic and then a tool tip on mouse over. Others show a full screen background with pages filled with scrolling lists of achievements with texts and progress bars and all manner of things. How you decide to use/display achievements is up to you. What this asset endeavors to do is to enable you to do whatever you decide upon doing... and to that end it provides a whole host of functions for you to use…

Functions? What functions?

All these functions reside in the static WUAchieve class so they can be called from anywhere at any time giving you the freedom to work on your design, not worrying about running into limitations and finding workarounds for this system.
The functions at your disposal are:

These functions operate on your own achievements:

- FetchEverything - Returns all info you generated for all achievements including whether you unlocked it already
- FetchUnlocked - Returns all info on unlocked achievements

- FetchLocked - Returns all info on locked achievements
- FetchUnlockedIDs - Returns a comma delimited string with the ID's of unlocked achievements
- FetchLockedIDs - Returns a comma delimited string with the ID's of locked achievements
- FetchAchievement - Returns all info of a single achievement
- UnlockAchievement - Unlocks a specific achievement by ID
- LockAchievement - Locks a specific achievement by ID
- ToggleAchievement - Locks or unlocks the achievement, whichever it's not. Only really useful in the dashboard
- IsAchievementUnlocked - Returns true if the achievement was unlocked

The following derivatives works on someone else's data and is to be used with authoritative games

They work identical to the above counterparts except they require the other user's id as an additional argument

- FetchUserUnlockedIDs
- FetchUserLockedIDs
- UnlockUserAchievement
- LockUserAchievement
- ToggleUserAchievement
- IsUserAchievementUnlocked

With all those functions this is starting to look complicated

It really isn't. I include a working prefab that demonstrates all the advanced stuff you could possibly want to do and comment the code heavily. Once you see it in action, you'll understand why I can so confidently say that this system is super easy to use. It boils down to this:

1. Request a list of all achievements
2. I return all info on all achievements and for each I indicate whether or not the player has already unlocked it
3. Now you just display that info in any way you choose

Inside this prefab I make provision for awards that you can unlock manually but I also make provision for awarding the achievements on your behalf.

Automatic awarding of achievements

Requirements allow you to specify specific requirements directly on your website (and thus you can change the requirements at any time even after the game was published) if you so wanted to. You can either keep track of a user's actions and inventory and progress and whatever else it is you want to keep track of using some custom system you already have or want to create and then decide in your own code "Now the person is due this achievement"... sure... or you can just specify the achievements on the website in advance and in the game you just store their progress. Every time you store something just check "Does this latest progress qualify the player for any new achievement?". A single function to test all achievements based on this predefined method of tracking achievements.

In the demo prefab I first fetch all achievements and display them on the background image. I check to see if it is unlocked or not and display the relevant graphic and text for each achievement. Next I check to see if any of those achievements that are still locked has any server side defined requirements and, if so, I add them to the list of achievements I track.

Finally, I created some buttons above the prefab to allow you to simulate what would happen in game and allow you to toggle awards on or off manually. I also enable you to simulate game progress by adding or removing game keys. Each time you add or remove a game key I check to see if an award is now obtained and if so I unlock it on the server and stop tracking it locally.

Basically, I only track locked achievements that have server side requirements and nothing else. Unlocked achievements are just displayed and locked achievements without requirements you can unlock whenever you so please. Of course, you can unlock keys with requirements at any time also, even if the requirements are not met. It is entirely at your discretion. In this case, if the same achievement is unlocked more than once, nothing strange will happen on consequent unlocks.

It will essentially say "The unlocked achievement is still unlocked. Nothing to see here. Move along"

What are those game keys you mentioned?

Game keys are how I track game progress in the demo prefab.
They are just simple strings and can be anything you want them to be.
Examples include: gold, level, enemy_kills, headshots, reputation... quite literally anything you want.

While the game is playing, whenever something of any importance happens, you simply add a key to signal that thing has happened and that way you keep track of how many times it has happened thus far.
The prefab has a simple function called UpdateKeys which takes the name of a key and a positive or negative value to adjust the quantity of the key's by and that is how simple game keys are to work with.

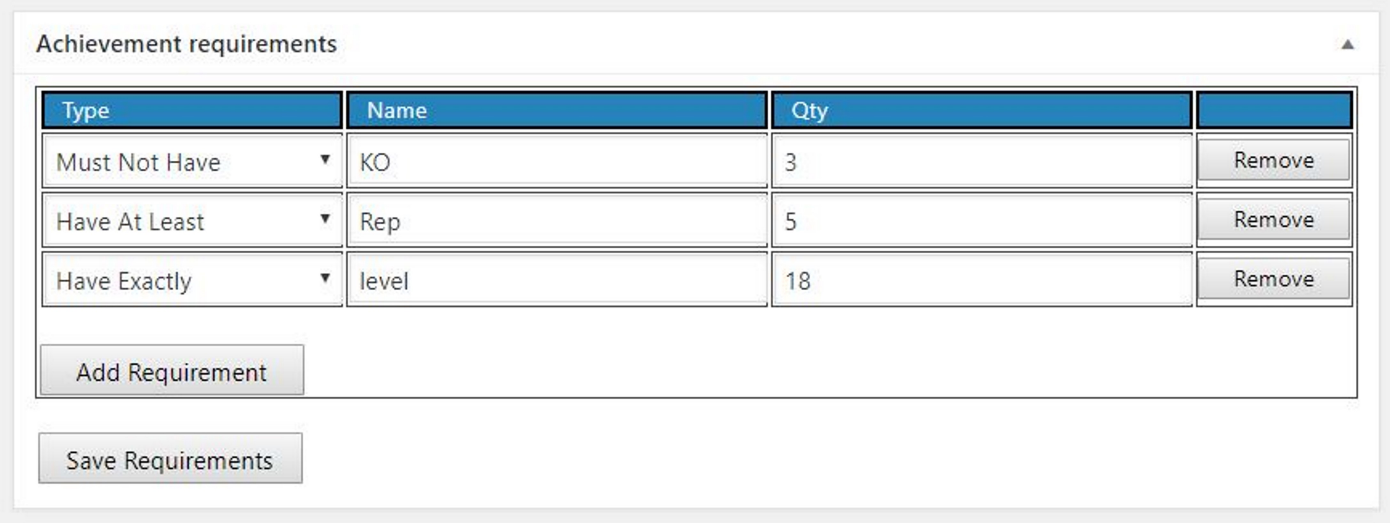
For example, say your character does a head shot... You would just call:

```
UpdateKeys("HeadShots", 1);
```

...and with that single line of code you now have one more head shot key than you had before. Each time you add a key I automatically save the keys for you so these keys persist between game sessions.

So how do they work, exactly?

Again, this is so simple it is almost criminal. In your dashboard when you define your achievement you can add as many requirements as you like. Each requirement consists out of 3 fields.



Achievement requirements

Type	Name	Qty	
Must Not Have ▾	KO	3	Remove
Have At Least ▾	Rep	5	Remove
Have Exactly ▾	level	18	Remove

The first is a drop down box that you just choose a value from to determine if this achievement:

1. Requires AT LEAST...
2. Must have LESS THAN...
3. Or must HAVE EXACTLY...

...the amount of keys you specify.

The second field is a text field where you specify the name of the key (case sensitive). The final field is a number representing how many keys would qualify this requirement as being met. All requirements must be met before an achievement is unlocked.

Example:

Let's say you create an achievement called "Master Sniper" and you set three requirements.
After setting the three values for each of your three requirements they will essentially read like this :

IF you have AT LEAST 10 HeadShots keys

AND you have LESS THAN 1 Missed keys
AND you have EXACTLY 3 Lives keys
THEN receive the Master Sniper achievement.

You can make this as simple as "Killed at least 100 Orcs" or you can make an achievement have 500 requirements. How easy or difficult you make it is entirely up to you.

While playing the game you just call UpdateKeys whenever something happens and I will keep checking all the various keys against all the various requirements of all the locked achievements and award the achievements when appropriate. You just sit back and relax. Did I mention this kit is easy to use? :)

Quick recap...

You can have two types of awards:

1. You don't specify any requirements and then in your game you just call WUAchieve.UnlockAchievement whenever you determine it is the right time to do so
2. You can specify any number of requirements you want and then keep checking the requirements inside a loop, awarding the achievement from a central function in your project

In both cases you will need a way to track what the player has and has not done so you might already have another system in place and would prefer to use that method instead. With method two, above, the prefab takes care of storing and retrieving your progress for you and truly automates the entire process.

As an aside, the prefab also has one other feature you might find useful. If you specify a graphic to use as an achievement and a graphic with the same name is found in the Resources folder inside your project then the prefab will use the local graphic instead of fetching the graphic from your website. Only if it is not available locally will I go fetch it online.

I just ran the demo and nothing happened. How do I get started?

Now that you understand the ins and outs of this system I can explain how to get started and everything will just make sense. So let's get started…

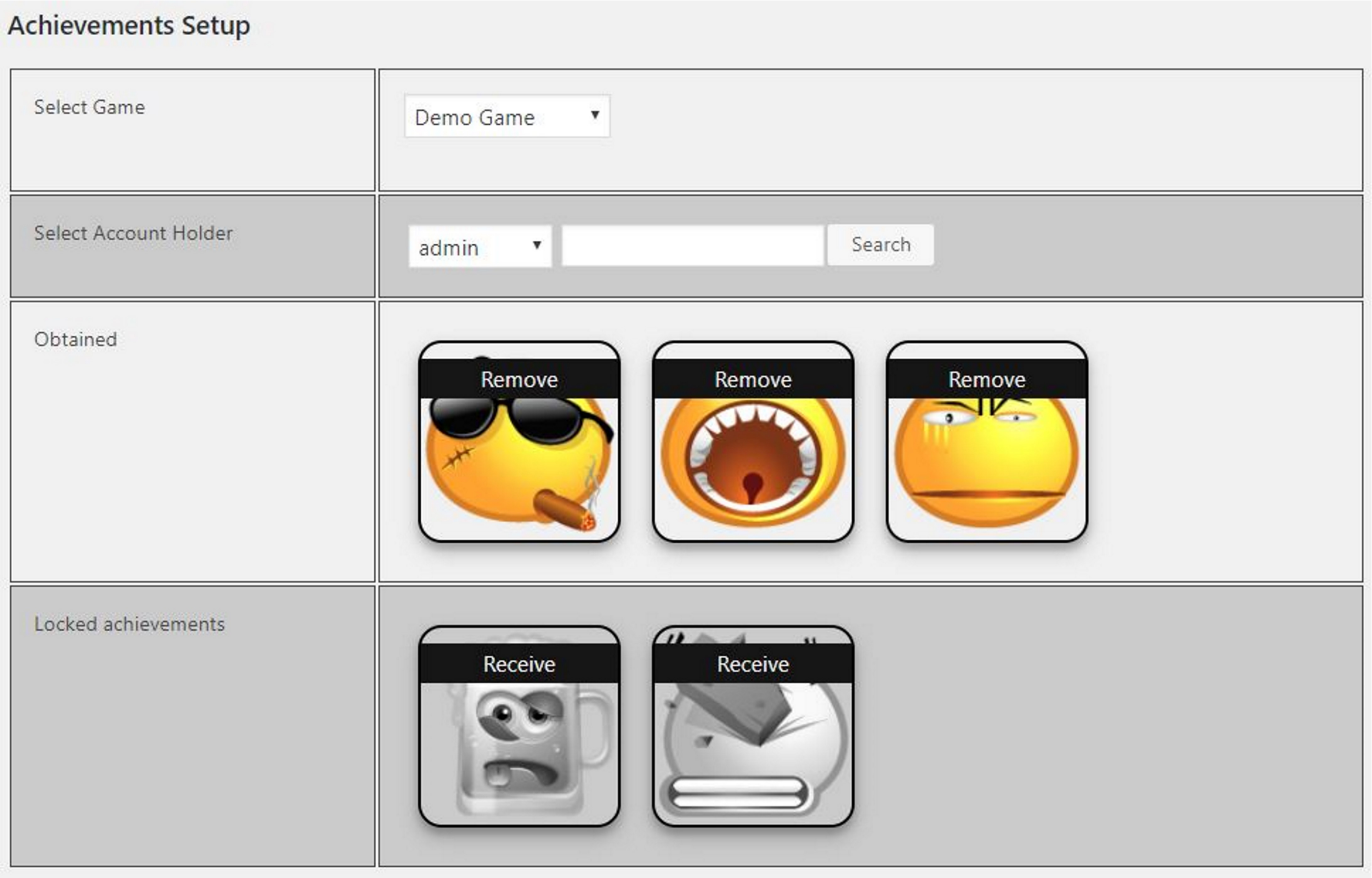
> **NOTE:** *I am going to assume that you already followed the steps for installing the* WordPress Login kit *and creating your first game. After all, this kit is a plugin for* WordPress Login *and needs the latest updated version of it up and running before you can use the Achievements plugin.*

Step by step…

1. Go to your WordPress dashboard and install the wuss_achievements plugin that is found inside:
   Assets/myBadStudios/WUSS/Wordpress/Plugins/
2. Go to the WUSS menu and click on the All Achievements sub menu. On the page that opens, click "Add New"
3. On the page that opens enter a name for your achievement as the post's title
   The post content box is where you type in the text that is shown after the achievement is unlocked
   The Excerpt panel is where you enter the unlock requirements in plain text
   The Featured Image box is where you define the graphic to show after the achievement was unlocked
   The Locked Icon box is where you define the locked image, as the name implies
   Finally there is the Requirements tab. If you want to add any just click on the Add Requirement button to get started
   NOTE: *Be sure to click on Save Requirements before you update the page (!!!)*
   NOTE: *There are some sample graphics in the project you can use to get started with making achievements*
4. Now just publish or update the post and you are done here.
5. When you return to the WUSS main menu and then click on the Badges tab you will see all the

achievements listed per person. Simply select a user via the drop down box to see which achievements they have unlocked. Obviously a this point everything will still be locked but on this page you can manually assign or remove any achievement from any person simply by clicking on the black button floating over the achievement.

6. Now go to Unity and open the Achievements demo scene. Click on the canvas and make sure to configure the *WPServer* component to point to your website and for this particular demo scene enter your login username and password in the *DemoScript* component.



Achievements Setup

Select Game	Demo Game ▾
Select Account Holder	admin ▾
Obtained	
Locked achievements	

That is all there is to it. You can now run the demo and start playing around with adding keys.

NOTE: *Be sure to look at the demo prefab's source code for incredibly detailed notes on everything I do and you will be ready to use this inside your own games in no time at all!*