# LingQ Bulk Lesson Importer Tool

By Green Airplane

**DISCLAIMER**

**I HEREBY DISCLAIM ANY LIABILITY FOR DAMAGES CAUSED BY USING THIS TOOL. THE RESPONSIBILITY FOR USING IT IS YOURS ALONE. I DO NOT CLAIM ANY COPYRIGHT ON THE TOOL. ANYONE IS FREE TO USE IT, MODIFY IT AND REDISTRIBUTE IT.**

## Changelog

**Version 1.1**

* Added retry capability to import function
* Added verification of uploaded lessons (does not verify lesson content, only that it’s present)
* Added script to check files for words that are too long
* Added script to delete all lessons in a given course, so that it can be deleted

## Introduction

This tool automates importing large amounts of text into the language learning system at <https://www.lingq.com>. It can upload entire books in one go. It does this by commandeering a Firefox browser window and making it do the same steps you would do during manual import (Other browsers were tested, with unsatisfactory results). For this purpose it uses the [Selenium Browser Autmation Framework](http://docs.seleniumhq.org/). The tool was implemented as a PowerShell script to make it transparent and easy to modify. The script uses several Selenium DLL files, which can be freely downloaded from the Selenium website. For the sake of convenience I have also included them with the tool. I do not claim any ownership of these DLLs.

The tool assumes you already have your text prepared - split into small plain-text files, properly named and numbered. This task depends on your specific circumstances (such as the format of your e-book), therefore making a generic tool that handles this is too difficult. (I may release something that will help a bit in the future)

## Improvements in the latest version

While I was using the tool to import large books, I discovered that sometimes one or more lessons were not imported. This is usually caused by the LingQ site responding too slowly. Because LingQ presents lessons for study in the order in which they were imported, if a lesson is not uploaded, it will mess up the whole course. To work around this issue, I’ve added the ability to retry importing a lesson given number of times if it fails. Because the failures are mostly caused by slow response from the LingQ site, the script waits for a gradually increasing amount of time before each retry.

The script isn’t always capable of detecting an import failure. I’ve therefore added a verification step after the import. It will detect all lessons that are present in the input folder, but not on LingQ. This step only checks lesson titles, not their contents. Checking the contents would take too long, and I haven’t had a lesson import with incomplete contents yet. It seems that if a course is imported, it’s imported whole.

While still not perfect, these changes make the tool fairly robust and reliable. And if some lessons do fail to import, the script will tell you exactly which ones they are, so you can add them manually.

I’ve also discovered that you can’t delete a course if it has lessons in it – which can mean deleting hundreds of lessons manually. So I made a script that does that for you.

Lastly, I’ve expanded the instructions on how to use the script, as well as how to prepare your lessons to avoid import failures. Please read these instructions carefully.

## Prerequisites

* **Windows 7 or later.** (May work on Vista or XP, this has not been tested.)
* **Mozilla Firefox browser**
* **.NET Framework 4.0 or higher**. The .NET framework is usually shipped with your operating system, and Windows Update keeps it up to date. If this is not the case, you can download it here <https://msdn.microsoft.com/en-us/vstudio/aa496123>
* **PowerShell 3.0 or higher.** Depending on your operating system, it's possible you already have PowerShell. If this is not the case, you can install it by following the instructions here <https://technet.microsoft.com/en-us/library/hh847837.aspx>

## Parts of the solution

* **LingQBulkUploader.ps1** - script for importing lessons to LingQ
* **FindLongWords.ps1** - script for detecting words longer than 60 characters
* **ClearCourse.ps1** - script for removing all lessons from a course, so it can be deleted
* The DLLs:

*ThoughtWorks.Selenium.Core.dll*

*Selenium.WebDriverBackedSelenium.dll*

*Selenium.WebDriver.dll*

*WebDriver.dll*

*WebDriver.Support.dll*

**Note: I included these DLL files for convenience only. They are from the standard Selenium distributable package. You can download them yourself from here** [**http://selenium-release.storage.googleapis.com/2.45/selenium-dotnet-2.45.0.zip**](http://selenium-release.storage.googleapis.com/2.45/selenium-dotnet-2.45.0.zip) **Copy the DLLs from the net40 folder to the folder where the script file is located.**

## Preparation

**1. LingQ Course**

Make sure you have a course ready on the LingQ site. I recommend you create a new course for the import. Don't use a course where you already have a lot of lessons. If you make a mistake, or the import fails halfway through for whatever reason, you'll have to manually delete all the unwanted lessons, which is a pain (or use my ClearCourse script, which will delete your old lessons as well).

Once your course is created, navigate to it in your browser. You should have a url that looks like this:

https://www.lingq.com/learn/**ja**/courses/**199553**/

You will need two values from the url:

* Language code - in my case "ja"
* Course id - in my case "199553"

**2. Files to import**

Place all the files you want to import in a single folder. Use only plain text files (.txt). Do not put anything else in the folder (the script will try to import any file present in the folder). The files will be imported in alphabetical order, taking into account numbers in file names (natural ordering is used, i.e. "part 1 of 10" will be first and "part 10 of 10" last). The lesson title will be the file name without extension (without the ".txt").

Example of input folder contents:

第1章　最悪の誕生日　The Worst Birthday - part 1 of 4.txt

第1章　最悪の誕生日　The Worst Birthday - part 2 of 4.txt

第1章　最悪の誕生日　The Worst Birthday - part 3 of 4.txt

第1章　最悪の誕生日　The Worst Birthday - part 4 of 4.txt

第2章　ドビーの警告　Dobby's Warning - part 1 of 7.txt

第2章　ドビーの警告　Dobby's Warning - part 2 of 7.txt

第2章　ドビーの警告　Dobby's Warning - part 3 of 7.txt

第2章　ドビーの警告　Dobby's Warning - part 4 of 7.txt

第2章　ドビーの警告　Dobby's Warning - part 5 of 7.txt

第2章　ドビーの警告　Dobby's Warning - part 6 of 7.txt

第2章　ドビーの警告　Dobby's Warning - part 7 of 7.txt

...

第18章　ドビーのごほうび　Dobby's Reward - part 8 of 8.txt

I recommend importing the whole book in one go. If you do it in parts, sometimes LingQ likes to mess up the lesson order a bit.

**Maximum word length**

The LingQ system does not allow individual words that are longer than 60 characters. If you try to import a lesson with such a word, it will fail.

|  |
| --- |
| You may think there is no way your book contains such words, but I’ve found strings like *“»A­A­A­A­A­A­A­A­A­A­A­A­A­A­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­R­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H­H!”* in mine. |

You can use the script **FindLongWords.ps1** to check your files for these words. The script has one parameter: path to either a file, or a folder containing your files.

Example usage:

FindLongWords.ps1 –path “N:\DE\HP books\HP5.txt”

or

FindLongWords.ps1 –path “N:\DE\HP books\HP5 chapters”

**For instructions on how to run PowerShell scripts see the next section – “Running the import script”**

This script will only work properly on plain text files. For each input file the script will print out the file name followed by any words longer than 60 characters present in the file. If such words are present, you need to open the files in a text editor, find the words using the search function, and edit them. I recommend splitting them into smaller words.

**NOTE: This doesn’t apply to certain Asian languages such as Japanese and Chinese, that don’t use spaces between words.** LingQ handles these languages differently – it tries to identify the individial words, and then inserts spaces between them upon import. Do not run the script on Japanese or Chinese texts, as it won’t work properly.

**Maximum lesson title length**

The hard limit on the length of a lesson title is 80 characters. The input field in the browser will not allow you to type in any more, and if you try to import such lesson, the title will be truncated. This will not cause the lesson import to fail, but the final verificaiton will not find it, and it will be listed as missing.

The import script creates lesson titles from file names. To avoid any issues, visually check your input folder and look for file names that look like they could be longer than 80 characters. Rename the files to something shorter, while keeping their alphabetical order in the folder intact.

|  |
| --- |
| I learned this the hard way when the lesson titled  *“ 第６章　９と３／４番線からの旅 The Journey from Platform Nine and Three Quarters - part 9 of 12“*  (80 characrters) would import properly, but the next one  *“第６章　９と３／４番線からの旅 The Journey from Platform Nine and Three Quarters - part 10 of 12“*  (81 characters) would not. In any case, most of your lesson names should be much shorter, so the long ones should be easy to spot. |

## Running the import script

The script has 6 parameters. The first 5 are necessary for the script to work. The last one is optional, if you don’t specify it, the default value will be used.

|  |  |
| --- | --- |
| username | Your LingQ username. |
| password | Your LingQ password. |
| courseId | ID number of the course you want to import into. See previous section for details. |
| languageCode | Two-letter code of your course's language, such us ja, de, es... |
| inputFolder | Folder where your files are |
| retries | Number of times lesson import should be retried if it fails. Default value is 10 |

1. If you haven't done so already, download the tool. Go to <https://github.com/TheGreenAirplane/LingQBulkImporter> and click **Download ZIP** in the lower right part of the page. Unzip the downloaded archive.
2. **Open PowerShell console**. Click the Start button, type in **PoweShell**. In the search result under **Programs** click **Windows PowerShell**. (If PowerShell doesn't come up in the search, it means you don't have it installed. Refer to section "Prerequisites")

**NOTE**: If you’ve never run PowerShell scripts before, it’s likely that your PowerShell **Execution Policy** is set to “**Restricted**”. If you try to run the script with restricted execution policy, you’ll get an error that looks like this:

**LingQBulkUpload.ps1 cannot be loaded because running scripts is disabled**

**on this system.**

You can also verify this by typing *Get-ExecutionPolicy* in the PowerShell console window.

To change execution policy, you must run the PowerShell console with administrator privileges. Right-click on the PowerShell icon and choose “Run as Administrator”. If a dialog window appears, asking “Do you want to allow the following program to make changes to your computer?” click “Yes”. In the PowerShell console window type:

*Set-ExecutionPolicy Unrestricted*

PowerShell will ask you if you really want to change the execution policy. Press “Y” and then Enter. You can now run the script (proceed to step 3). After the import is finished, you can set the execution policy back to “Restricted” by typing

*Set-ExecutionPolicy Restricted*

Or you can leave it unrestricted.

1. **Navigate to the folder where the script file is located (where you unzipped the downloaded file).** For example, if the script is in C:\downloads\LingQBulkImporter-master, type   
   **cd C:\downloads\LingQBulkImporter-master** in the PowerShell console window.
2. **Run the script with necessary parameters.**Example usage:  
   *.\LingQBulkUpload.ps1 -username myusername -password mypassword -courseId 199553 -languageCode ja -inputFolder "C:\My books\HP2\Files to import"*

or:

*.\LingQBulkUpload.ps1 -username myusername -password mypassword -courseId 199553 -languageCode ja -inputFolder "C:\My books\HP2\Files to import" –retries 5*

1. A Firefox window will launch. It will try to log into LingQ with your username and password, and then import the lessons one by one. Observe this process for a while. Make sure it has logged in correctly and the first one or two lessons were imported. Do not close or interfere with the Firefox window until all the files are imported.

**TIP:** You can stop a PowerShell script at any time by bringing up the console window in which it’s running, and pressing Ctrl+C. If you want to continue the import later, prepare an input folder with only the remaining lessons and run the script again.

## Deleting a course

The ClearCourse.ps1 script can be used to delete all lessons in a given course, so it can then be deleted. The script has 4 parameters. All are necessary for the script to work.

|  |  |
| --- | --- |
| username | Your LingQ username. |
| password | Your LingQ password. |
| courseId | ID number of the course you want to import into. See previous section for details. |
| languageCode | Two-letter code of your course's language, such us ja, de, es... |

Example usage:

*.\ClearCourse.ps1 -username myusername -password mypassword -courseId 199553 -languageCode ja*

This script works on the same principle as the import script, using Selenium to control a browser window. Once started, the script will login to LingQ and immediately start to delete lessons from the course. So it goes without saying, **make sure that this is what you want to do and you have the correct course ID, before you run the script.** After the script has finished, you can manually delete the course. If for some reason the script hasn’t deleted all the lessons, you can run it again as many times as you need until all the lessons are gone.

**Once again, be EXTRA CAREFUL when running this script. If you make a mistake, you can easily lose lessons you didn’t want to delete.**

## Future improvements

This tool still has one potential problem: If some lessons from the middle of the bulk fail to import, and are imported afterward, it will mess up the lesson order. Fixing this order manually is a painful and tedious process, especially if you have hundreds of lessons in a course. The next version of the script should be able to automatically reorder the lessons to mimic the file order in your input folder.

A step after that would be to integrate everything into one solution that will do the following:

1. Examine the existing course
2. Delete all lessons that are not supposed to be there (i.e. they are not in the input folder)
3. Upload only the lessons that are missing, but check them for word length and title length first
4. If necessary, reorder the lessons
5. Repeat any of these steps as necessary until
   1. The course mimics the input folder perfectly
   2. The user chooses to terminate the script

The reason I chose this approach is simple: as the input folder is in each user’s full control, you can easily arrange it as you see fit. The script will then take the folder and put it on LingQ, “as is”. Once again, I make no promises as to when, (or if at all) I’ll get to it.

## Speed issues

In its present form the script is fairly slow. Depending on factors such as lesson size, and mainly the speed and stability of your internet connection, it takes on average about 30 seconds to import a single lesson. This means that if you have a course with 300 lessons, it can take up to 2.5 hours to upload. Deleting lessons is faster, but it still takes about 20 minutes to delete 300 lessons.

It will still save you a lot of time, because it’s much faster than doing it manually, and you do other stuff while it’s working. You can even keep using your computer, provided you don’t mess with the PowerShell window or the browser window that’s doing the work. But in my opinion the tool is ridiculously, and what’s worse, needlessly slow.

Unfortunately, I’m limited by what options the LingQ website provides. LingQ has a developer API, which is intended to support custom tools. But the functionality that deals with imports is either broken or not yet implemented. I brought this to the attention of LingQ staff, however, fixing it doesn’t seem to be a priority. If you’d like to help, you can petition them to make it work. If they see there’s demand, they might get around to it sooner rather than later, and then I can make a much better tool.

## Support & Feedback

If you have problems running the script, you can contact me via the LingQ forum, or at **green.airplane@gmail.com**. Make sure you put "LINGQ TOOL" in the email subject. I may try to help, if my schedule permits. However, bear in mind that **I do not make any commitment or promise to provide technical support for this tool. I made this tool for my own purposes, and then released it out of good will. All support (if any) I may choose to provide, will be on a good will basis as well.**

You may also send me feedback. **I am especially eager to hear if you had trouble following these instructions, or if something needs to be clearer or more detailed.** I am a software developer, I sometimes have trouble judging how much information people who are not skilled with computers really need.