

ClassMatic Setup Guide

This tutorial assumes a Windows 7 or above 64-bit computing environment, though the ClassMatic software does work on Linux and Mac, and will also run on a 32-bit architecture. The process is largely the same for other operating systems and configurations, except for where the operating system commands differ. This tutorial includes instructions for setting up a ClassMatic implementation from source code, including using/configuring Git, NodeJS, and NPM.

View basic information about your computer

Windows edition

Windows 8.1
Enterprise
© 2013 Microsoft
Corporation. All
rights reserved.



System

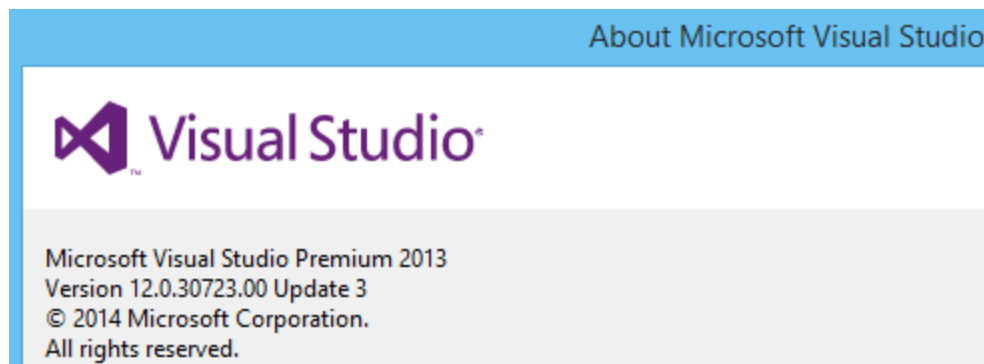
Processor:	Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz 2.39 GHz
Installed memory (RAM):	8.00 GB
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	Pen and Limited Touch Support with 2 Touch Points

This guide is intended to be followed sequentially, and contains relevant pitfall information where appropriate. If the source

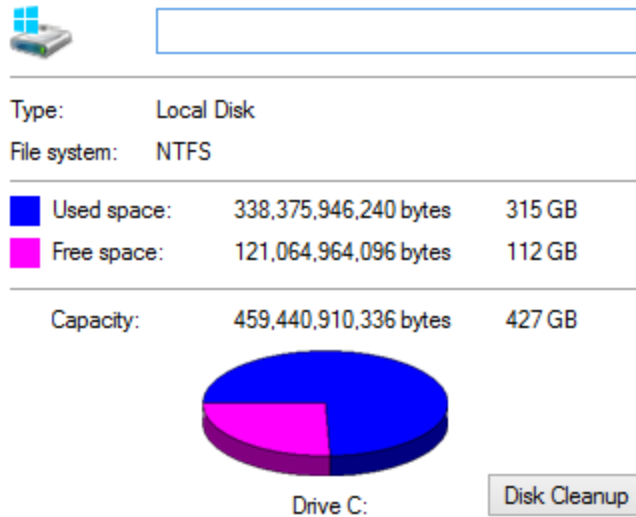
code is unintentionally modified in such a way that breaks the implementation, ClassMatic can be installed fresh by simply deleting the ClassMatic folder and beginning again at the start of this tutorial.

Preliminary Checks

Ensure the computer being used has a C++ compiler installed. If Windows is the platform, Microsoft Visual Studio 2010 or above is required.



Ensure the computer being used has about 2GB of free space available for the lifetime of its use with ClassMatic. This will be enough to install Git, NodeJS, ClassMatic and its dependencies, and house ClassMatic class/file data.



Ensure the computer being used has port 3000 open in its firewall, and that the port is publicly discoverable. The default ClassMatic port can be changed from 3000 to any desired port, as will be covered later in the tutorial. More information about allowing traffic through the Windows Firewall can be found at this URL:

<http://windows.microsoft.com/en-us/windows/open-port-windows-firewall#1TC=windows-7>

Download and Install Git

Git is a file/project source code versioning manager. Git's architecture dictates that project files are saved locally and synced with a web server run by a Git provider of choice (Github is a common one). Regardless of provider, the Git command line tools will work the same given proper syntax, since Git's protocols are standardized. More information can be found here:

https://en.wikipedia.org/wiki/Git_%28software%29



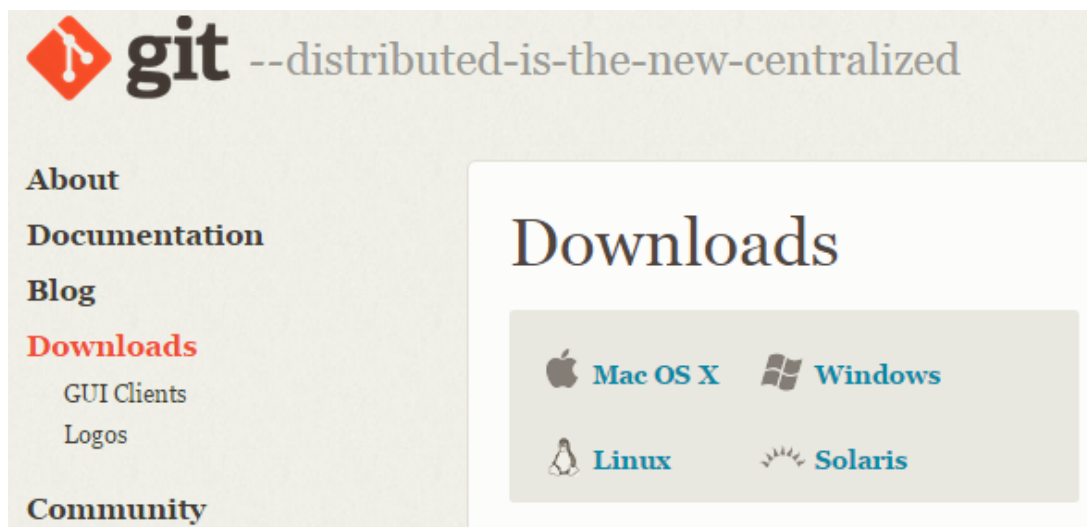
The Git command line tools are needed in order to fetch the existing ClassMatic project and ultimately deploy a copy of it to a server. Alternatively, if a Graphical User Interface (GUI)

is desired, many popular Github clients come bundled with the Git command line tools, so downloading those would work too.



The base command line software can be downloaded at this URL:

<https://git-scm.herokuapp.com/downloads>



The specific version of Git is unimportant, as it will be used only for initial project cloning/download.

Install the Git software accepting all the defaults, or modify them as desired.



Clone the ClassMatic project from the Github repository

For reference, the ClassMatic Github Project URL is:

<https://github.com/cambeyer/ClassMatic.git>

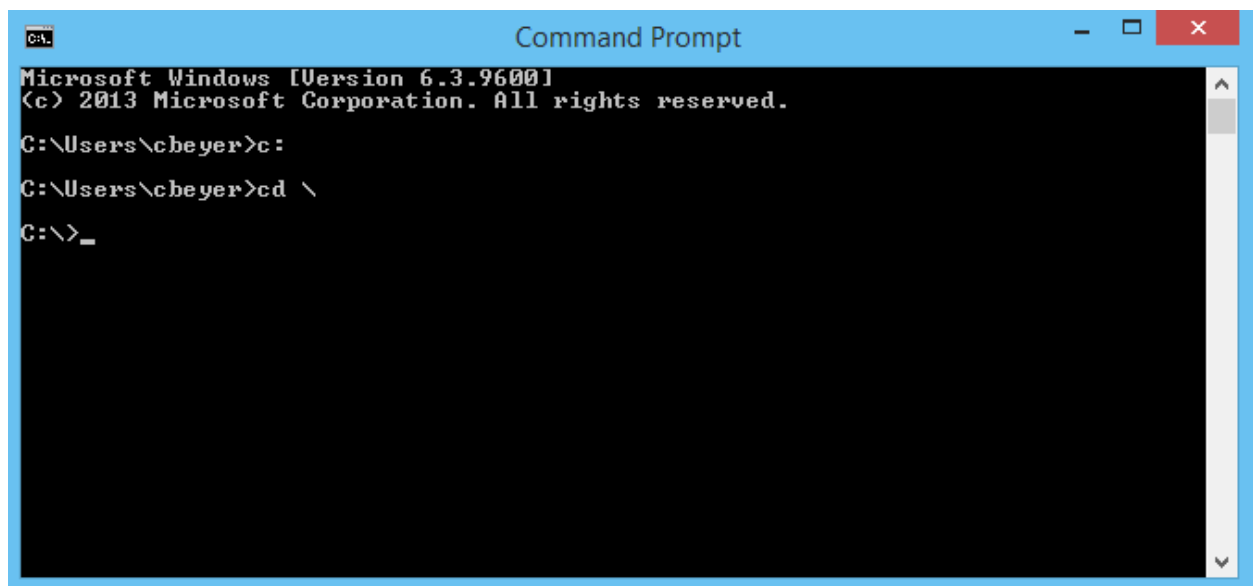
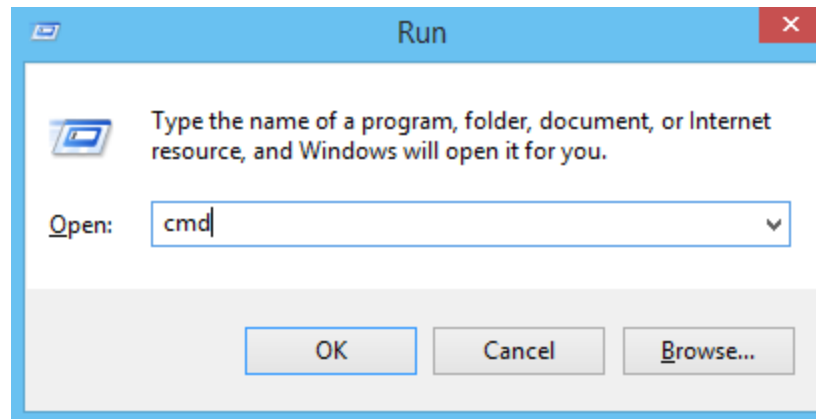
The screenshot shows the GitHub repository page for 'cambeyer / ClassMatic'. At the top, there's a header with the repository name and an 'Unwatch' button. Below this is a section for 'Description' and 'Website' with input fields and 'Save' or 'Cancel' buttons. A progress bar shows 14 commits, 1 branch, 0 releases, and 1 contributor. Below the progress bar is a section for the 'ClassMatic' repository with a dropdown menu for the branch (currently set to 'master') and a '+' button. The main content area shows a list of commits. The most recent commit is by 'cambeyer' an hour ago, with the message 'added comments, tweaked login error condition'. Below this, a table lists files and their commit messages:

File	Commit Message	Time
public	added comments, tweaked login error condition	an hour ago
.gitignore	Initial commit	2 months ago
index.js	added comments, tweaked login error condition	an hour ago
package.json	added loading bar on login, added support for instructor logins to ad...	23 hours ago

The remainder of this tutorial assumes a destination directory of C:\ClassMatic, but can be modified to accommodate any valid destination directory on any platform (Windows, Linux, Mac).

Open the command prompt (press Windows+R and type "cmd", then press Enter) and navigate to the root hard drive directory (run

these commands, pressing Enter after each: "c:", "cd \"). A "C:\>" prompt should be showing.









Verify that Git is registered on the executable PATH in Windows by running "git --version" in the command prompt. If something

like “git version 1.9.5.msysgit.1” is returned, proceed to the next step.

```
C:\>git --version
git version 1.9.5.msysgit.1
```

If a message like “‘git’ is not recognized as an internal or external command, operable program or batch file.” is presented, substitute the full path to the Git executable for the shorthand “git” in the remainder of this tutorial. This executable is typically located at: “C:\Program Files (x86)\Git\bin\git.exe” (the quotation marks are important/necessary, since the Program Files directory contains spaces). For more information on the Windows PATH, visit this URL:

https://en.wikipedia.org/wiki/PATH_variable#DOS.2C_OS.2F2.2C_and_Windows

This PC > Local Disk (C:) > Program Files (x86) > Git > bin			
<input type="checkbox"/> Name	Date modified	Type	Size
 gawk.exe	4/21/2015 3:45 PM	Application	146 KB
 getcp.exe	4/21/2015 3:45 PM	Application	6 KB
<input checked="" type="checkbox"/>  git.exe	4/21/2015 3:45 PM	Application	1,469 KB
 gpg.exe	4/21/2015 3:45 PM	Application	1,208 KB
 gpgkeys_curl.exe	4/21/2015 3:45 PM	Application	39 KB
 gpgkeys_finger.exe	4/21/2015 3:45 PM	Application	95 KB

Run the command: "git clone

<https://github.com/cambeyer/ClassMatic.git>" while still at the "C:\>" in the command prompt. This will create a new folder called "ClassMatic" and copy all of the needed files into the new folder. No authentication is required because the ClassMatic repository has been set up as Public.

```
C:\>git clone https://github.com/cambeyer/ClassMatic.git
Cloning into 'ClassMatic'...
remote: Counting objects: 107, done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 107 (delta 7), reused 0 (delta 0), paceceiving objects:  k-reused
87
Receiving objects: 100% (107/107), 440.56 KiB | 0 bytes/s, done.
Resolving deltas: 100% (55/55), done.
Checking connectivity... done.
```

Verify that the project was cloned successfully by running "cd ClassMatic" and "dir". You should see files called "package.json" and "index.js" listed.

```
C:\>cd ClassMatic

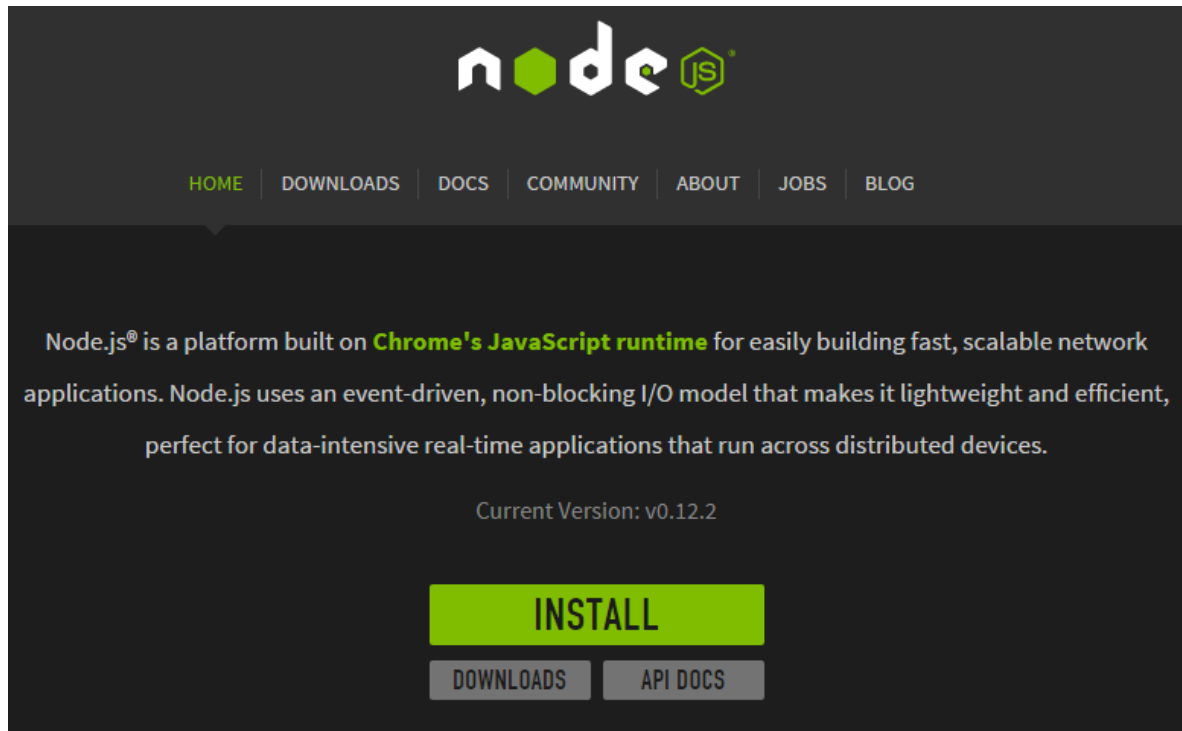
C:\ClassMatic>dir
Volume in drive C has no label.
Volume Serial Number is EC30-F64D

Directory of C:\ClassMatic

05/03/2015  11:07 PM    <DIR>          .
05/03/2015  11:07 PM    <DIR>          ..
05/03/2015  11:07 PM                615 .gitignore
05/03/2015  11:07 PM            31,082 index.js
05/03/2015  11:07 PM            317 package.json
05/03/2015  11:07 PM    <DIR>          public
               3 File(s)              32,014 bytes
               3 Dir(s)  115,054,460,928 bytes free
```

Download and Install NodeJS

NodeJS is the server-side technology being used in ClassMatic. It is based on the Google Chrome JavaScript runtime and provides an asynchronous threaded environment without unnecessary overhead. The implementation model relies on user-authored “modules” which can be specified for any NodeJS project in its “package.json” file. These modules are not stored with the project, since the binary data is easily downloaded using Node’s Package Manager NPM for a newly-cloned project. More information can be found here: <https://nodejs.org/about/> and <https://docs.npmjs.com/>



The main NodeJS software can be downloaded at this URL:

<https://nodejs.org/download/>

Downloads

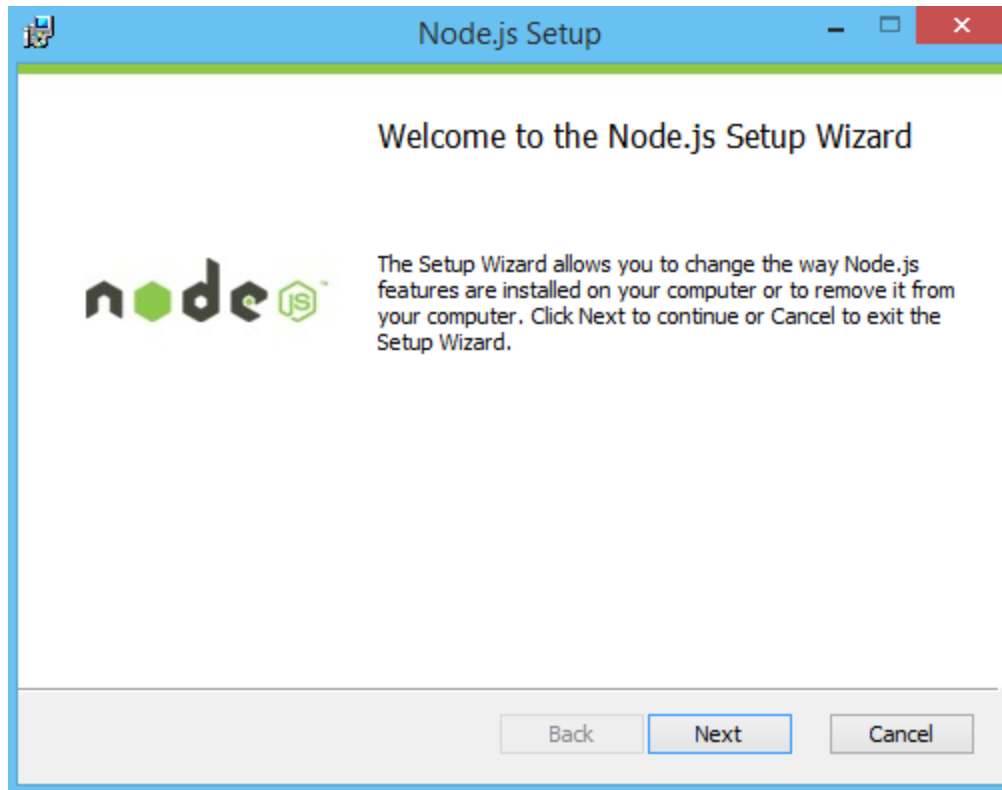
Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Current version: **v0.12.2**



Choose the 64-bit Windows Installer MSI. The current version is v0.12.2 (or above); ClassMatic was developed with version v0.10.36. Ensure the version of NodeJS selected is backward-compatible with version v0.10.36. Most likely, future versions of NodeJS will retain backward-compatibility.

Install the NodeJS platform accepting all the defaults.



Download the ClassMatic Project Dependencies

Open the command prompt and navigate to the ClassMatic project directory (e.g., "C:\ClassMatic"). A quick reminder of the series of commands: Windows+R, "cmd", then "c:", "cd \", "cd ClassMatic". More information on the DOS Command Prompt can be found here:

<http://www.cs.princeton.edu/courses/archive/spr05/cos126/cmd-prompt.html>





Verify that Node is registered on the executable PATH by running "node --version" in the command prompt. If something like "v0.10.36" is returned, proceed to the next step.

```
C:\ClassMatic>node --version
v0.10.36
```

If a message like "'node' is not recognized as an internal or external command, operable program or batch file." is presented, substitute the full path to the Node executable for the shorthand "node" in the remainder of this tutorial. This executable is typically located at: "C:\Program Files\nodejs\node.exe" (the quotation marks are

important/necessary, since the Program Files directory contains spaces).

This PC > Local Disk (C:) > Program Files > nodejs >

<input type="checkbox"/> Name	Date modified	Type	Size
 node_modules	1/26/2015 3:43 PM	File folder	
<input checked="" type="checkbox"/>  node.exe	1/26/2015 7:39 PM	Application	6,444 KB
 node_etw_provider.man	8/7/2014 2:14 AM	MAN File	7 KB
 node_perfctr_provider.man	8/7/2014 2:14 AM	MAN File	5 KB

Run the command: "npm config set msvs_version 2013", replacing "2013" with the appropriate version of Microsoft Visual Studio you have installed. This is necessary because some of the Node modules used with ClassMatic default to looking for version 2010 if no version is specified, and will therefore fail to build correctly. If node was not correctly registered on the executable PATH above, then you may need to replace "npm" with its full path, typically located at: "C:\Program Files\nodejs\npm.cmd". More information on the Visual Studio reconciliation can be found here:

<https://stackoverflow.com/questions/14180012/npm-install-for-some-packages-sqlite3-socket-io-fail-with-error-msb8020-on-wi>

```
C:\ClassMatic>npm config set msvs_version 2013
```

Run the command: "npm install" to download the appropriate versions of all the modules ClassMatic depends on.

```
path@0.11.14 node_modules\path
crypto@0.0.3 node_modules\crypto
fs-extra@0.16.5 node_modules\fs-extra
├── jsonfile@2.0.0
├── graceful-fs@3.0.6
└── rimraf@2.3.3 <glob@4.5.3>
express@4.12.3 node_modules\express
├── merge-descriptors@1.0.0
├── cookie-signature@1.0.6
├── utils-merge@1.0.0
├── fresh@0.2.4
├── methods@1.1.1
├── cookie@0.1.2
├── range-parser@1.0.2
├── escape-html@1.0.1
├── finalhandler@0.3.4
├── content-type@1.0.1
├── parseurl@1.3.0
├── vary@1.0.0
├── serve-static@1.9.2
├── content-disposition@0.5.0
├── path-to-regexp@0.1.3
├── depd@1.0.1
├── on-finished@2.2.1 <ee-first@1.1.0>
├── qs@2.4.1
├── proxy-addr@1.0.7 <forwarded@0.1.0, ipaddr.js@0.1.9>
├── etag@1.5.1 <crc@3.2.1>
├── debug@2.1.3 <ms@0.7.0>
├── send@0.12.2 <destroy@1.0.3, ms@0.7.0, mime@1.3.4>
├── type-is@1.6.1 <media-typer@0.3.0, mime-types@2.0.10>
└── accepts@1.2.5 <negotiator@0.5.1, mime-types@2.0.10>
request@2.55.0 node_modules\request
├── caseless@0.9.0
├── json-stringify-safe@5.0.0
├── forever-agent@0.6.1
├── aws-sign2@0.5.0
├── stringstream@0.0.4
├── tunnel-agent@0.4.0
├── oauth-sign@0.6.0
├── isstream@0.1.2
├── node-uuid@1.4.3
├── combined-stream@0.0.7 <delayed-stream@0.0.5>
├── qs@2.4.1
├── form-data@0.2.0 <async@0.9.0>
├── mime-types@2.0.10 <mime-db@1.8.0>
├── http-signature@0.10.1 <assert-plus@0.1.5, asn1@0.1.11, ctype@0.5.3>
├── bl@0.9.4 <readable-stream@1.0.33>
├── tough-cookie@1.1.0
├── hawk@2.3.1 <cryptiles@2.0.4, sntp@1.0.9, boom@2.7.1, hoek@2.12.0>
├── har-validator@1.7.0 <commander@2.8.1, chalk@1.0.0, is-my-json-valid@2.10.1,
bluebird@2.9.25>
connect-busboy@0.0.2 node_modules\connect-busboy
├── busboy@0.2.9 <readable-stream@1.1.13, dicer@0.2.3>
socket.io@1.3.5 node_modules\socket.io
├── debug@2.1.0 <ms@0.6.2>
├── has-binary-data@0.1.3 <isarray@0.0.1>
├── socket.io-adapter@0.3.1 <object-keys@1.0.1, debug@1.0.2, socket.io-parser@2.
2.2>
├── socket.io-parser@2.2.4 <debug@0.7.4, component-emitter@1.1.2, isarray@0.0.1,
benchmark@1.0.0, json3@3.2.6>
├── engine.io@1.5.1 <base64id@0.1.0, debug@1.0.3, engine.io-parser@1.2.1, ws@0.5
.0>
└── socket.io-client@1.3.5 <to-array@0.1.3, indexof@0.0.1, component-bind@1.0.0,
debug@0.7.4, backo2@1.0.2, object-component@0.0.3, component-emitter@1.1.2, has-
binary@0.1.6, parseuri@0.0.2, engine.io-client@1.5.1>
```

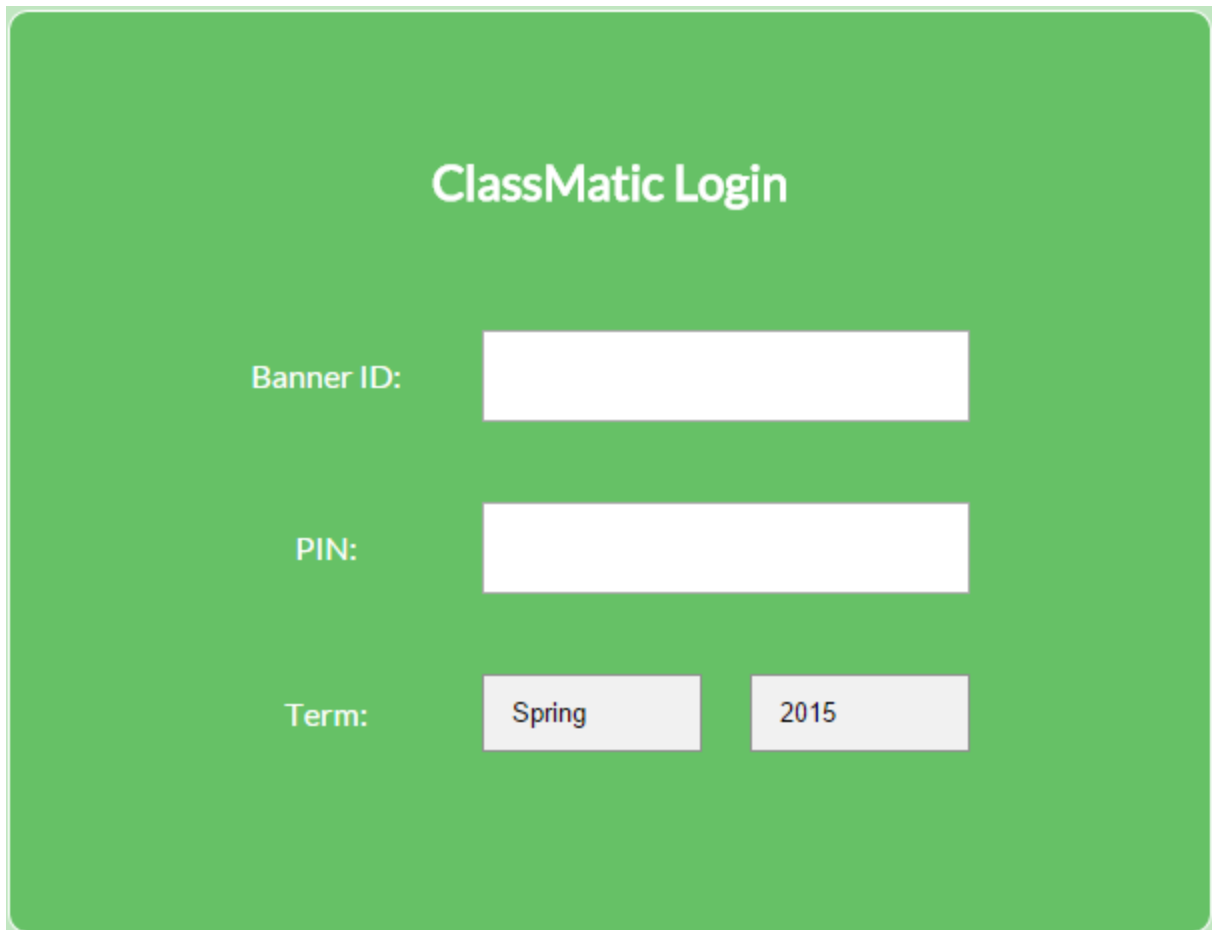
Start and Test the ClassMatic Server

Open a command prompt and navigate to the ClassMatic project folder.

Run the command: "node index" to start the ClassMatic server locally on port 3000.

```
C:\ClassMatic>node index  
listening on *:3000
```

Navigate in a Web Browser to <http://localhost:3000/> and verify that the ClassMatic logon page is presented.

The image shows a login form titled "ClassMatic Login" on a green background. It contains three input fields: "Banner ID:" with a white text box, "PIN:" with a white text box, and "Term:" with two adjacent white buttons labeled "Spring" and "2015".

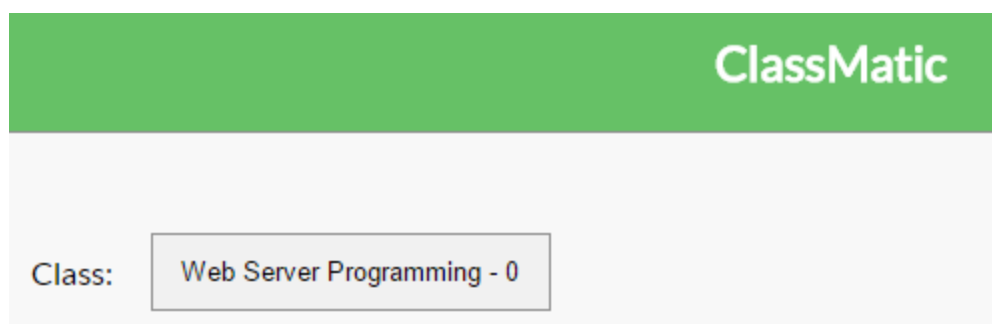
ClassMatic Login

Banner ID:

PIN:

Term:

Login to ClassMatic using your Lawrence Tech BannerWeb ID and PIN. Verify that the classes associated with your username (enrolled or teaching) are present in the dropdown.

The image shows a dropdown menu from the ClassMatic system. It has a green header with the "ClassMatic" logo and a light gray body. The "Class:" label is followed by a dropdown box showing the selected option "Web Server Programming - 0".

ClassMatic

Class: