# Installation guideline for Booktera

## System requirements

**.NET based components**

* Application Core, ASP.NET MVC web server
  + Windows 8.1
  + Visual Studio 2013
  + MS SQL Server 2012
  + .NET Framework 4.5
* Windows Phone client
  + Windows Phone SDK 8.0
    - Windows Phone 8 Emulator (or WP8 device)
    - The requirements for the Emulator:
      * Windows 8 Pro edition or greater
      * Requires a processor that supports Second Level Address Translation (SLAT)
  + TODO ?
* Windows 8.1 client
  + *(nothing extra)*

**Java based components**

* + Java SE Development Kit 1.8
* Play web server
  + Play Framework 2.3.4
  + Apache Maven 3.2.3
* Android client
  + IntelliJ IDEA 13
  + TODO ?

## Setup application’s core

There are 2 important things before start:

* Always run Visual Studio with administrator rights to avoid misfunctionality caused by access restrictions (the correct functioning is not guaranteed without this; also WcfHost cannot be hosted without admin rights)
* There have to be a Default Database Instance with "." alias, using Windows authentication, and Booktera will need privileges to access it

Through the following steps you will install the core application. This means you will have the full backend (db with data, wcf layer ready to host [see in the next chapter]), and you can run the ASP.NET MVC 4 based web application, and the Windows 8.1 client.

1. **Copy** to your disk the BookTera folder
   1. Do not alter the folder structure under it!
   2. The path to it must not contain special characters or white spaces
2. **Start** Visual Studio with administrator privileges, and open the project. The project file’s location: \BookTera\Solution\theSolution\BookTera.sln
   1. Register for Windows Phone Developer license (it’s free) and whatever VS ask you to do for the first time
   2. If appears, ignore the warning that the WEB project needs SQL Server Express.
3. **Build** the solution (Ctrl+Shift+B)
   1. At the first build, a NuGet Restore process will be executed. This way NuGet downloads the referenced packages, and copies the dll-s of those to a local directory (\Solution\theSolution\packages) , so the projects can reach them.
4. Browse to project **Tools/InitSolution**, then right click on it 🡪 Set as startup project; then press Ctrl+F5 to run the project without debugging
   1. If you choose to start any project with debugging, first check this: General troubleshooting / [1]
   2. Note: We can follow in the Console window the flow of the process. The most time-consuming task is here to download a ~5MB archive file. This will take time depends upon your internet bandwidth.
5. Browse to project **DataBase/DB**, then publish it with ebookDB.publish.xml (double click/right click, publish; then click the Publish button)
   1. This will create our (empty) database, with name “BookTera”
6. Browse to project **Tools/TestData**, then right click on it 🡪 Set as startup project; then press Ctrl+F5 to run the project without debugging
   1. If you choose to start any project with debugging, first check this: General troubleshooting / [1]
   2. By running the TestData project, we fill the empty database with data. Most of it is mock data for books and transactions, but there is valid, important data as well, like categories, type enumerations, etc.
   3. Note: We can follow in the Console window the flow of the process. It will take about 0.5 – 2 minutes. There will be exception messages listed with green and red during generation mock transaction data, it’s expected, don’t worry about it.

#### Troubleshooting

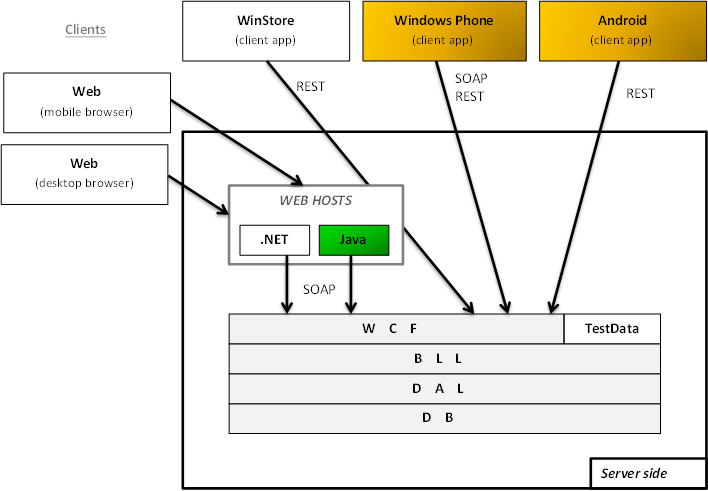
* If something went wrong, you can purge the db via step 4, then refill it via step 5.
* Troubleshooting for step InitSolution.
  + It (re)creates junctions pointing to content directories. If any of these would fail, first check you have started VS with admin privileges. If so, you can do this step manually.  
    You can create the junctions yourself using the “mklink /J” command prompt command. The junctions to create:  
    BookTera\Java\WebPlay\public\images < > BookTera\Solution\WEB\Content\Images  
    BookTera\Java\WebPlay\public\javascripts < > BookTera\Solution\WEB\Scripts\   
    BookTera\Java\WebPlay\public\stylesheets\shared< > BookTera\Solution\WEB\Content\themes
  + It also downloads resources for the TestData project. If it fails, you can manually download them from here:  
    <https://drive.google.com/uc?export=download&id=0B2aHj_zBJI1KRGFQbnhOdDJ0WW8>  
    This is an archive file. Decompress it’s content directly to here:  
    \Solution\Tools\TestData\Resources\

## Initialize the ground for the applications

The application BookTera, as mentioned before, consists of 5 runnable submodules/runnables:

* ASP.NET MVC 4 based web application
* Windows 8.1 client
* Play Framework 2.3.4 based web application
* Windows Phone 8 client
* Android 4.4 client

As you can see below on the project’s architecture image, these modules all depend on the WCF module (layer).



Therefore, **whatever runnable you want to use, first you have to start the WCF module**, which will start an ASP.NET MVC 4 based web host for the wcf services (via IIS Express). If you try to start any of the runnables before wcf process is hosted, you will get some socket layer exceptions, and the applications can’t run at all.

This (starting the WcfHost) you can achieve by browsing to project WCF/WcfHost, then right click 🡪 Set as StartUp project; then press Ctrl+F5 to run without debugging. (You may run it in debug mode, but it’s not necessary). This will also open a new tab in your default browser and browse to the root of the wcf project; and you also can see in your system tray that IIS Express started as well.

To start the ASP.NET MVC 4 web application, simply run the WEB project (right click, set as startup project, ctrl+F5)

#### Troubleshooting

There can be some (magic) errors by starting the WcfHost project.

* Sometimes a service can’t be activated (they are activated the first time requesting them). Don’t be afraid about this, the project is hosted via IIS Express… All you have to do is rebuild (and with it republish) the WcfHost project
* At a time, there were instability problems with Newtonsoft.Json package. It has sometimes caused some exceptions with this message:  
  A(z) „Newtonsoft.Json” fájl vagy szerelvény, illetve annak egyik függősége nem tölthető be. A megtalált szerelvény jegyzékdefiníciója nem egyezik a szerelvény hivatkozásával. (A kivétel HRESULT-értéke: 0x80131040)  
  Probably the error was caused because there were different versions installed to different projects. Luckily, after unifying the version, this problem has not appeared. But in case the source of the problem was still something other, if you see the above error message, try the followings:

1. Reload the requested page
2. Rebuild the project owns the error
3. Reinstall the NuGet package.   
   In Visual Studio: Tools/Nuget Package Manager/ Package Manager Console  
   Update-Package Newtonsoft.Json -ProjectName {...} -Reinstall

## Mock users and passwords

The application has many mock users, you can sing in with any of them.  
All of them have the password: "**asdqwe123**".  
An example username is: "**ZomiDudu**"

## Setup and run subapplications

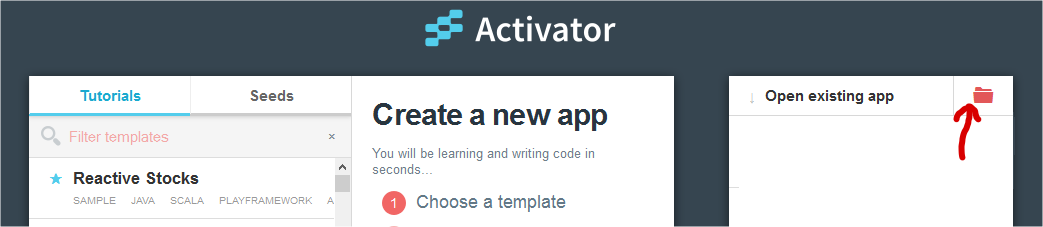
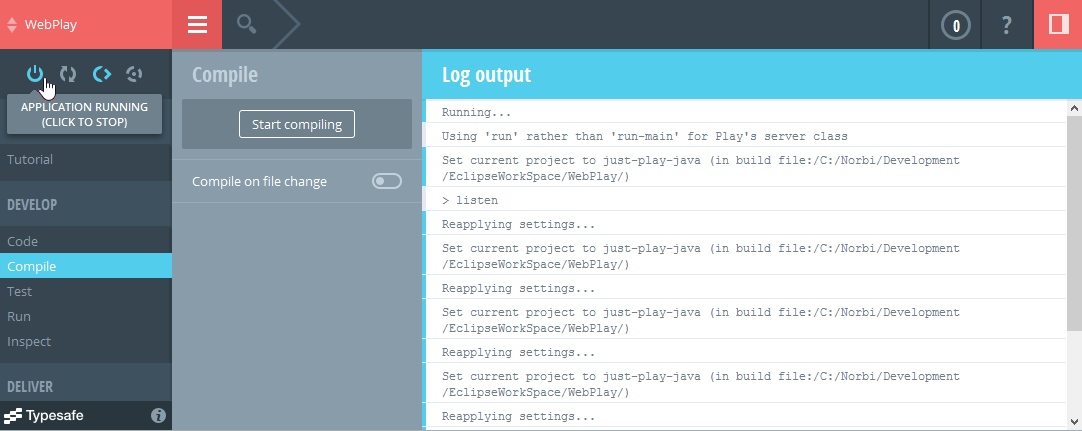
### Play

To try out the WebPlay project, first you have to install the Play Framework 2.3.4. If you also want to use an IDE for the project, you can find an "internal" guideline next to this document: "*Play/How to import WebPlay project into IntelliJ.docx*". This was made for own use, so it’s not too verbose. After all, you do not need any IDE to try out this project.

You also need to install Apache Maven 3.2.3. Before launching the WebPlay application, you have to install the ServiceClientProxy module to your local maven repository. To achieve this, follow these steps:

1. Browse to the directory: BookTera\Java\ServiceClientProxy\
2. Start a command prompt (cmd)
3. Type in: mvn clean install

To start the WebPlay project, follow these steps:

1. Start a command prompt (cmd)
2. Type in: activator ui  
   (Note: if the play framework’s root was not getting in the PATH environment variable during installation, you have to put it there now; or if you would don’t want to do this, you have to navigate into the framework’s root directory now to run this command)
3. On the appearing web site, click the icon next to “Open existing app”, and browse to {project root’s parent folder}/Java/WebPlay  
   
4. Click on the power icon to compile and run the application. Note that it could take a long time to resolve all the projects (sbt and maven) dependencies, and to compile all the sources. So please be with patience.  
   
5. After the project is compiled, deployed and run (the circling wait icons disappear, the power button lights with cyan, and the log also says so), you can browse to the web application at <http://localhost:9000>

Note: if you go to the WebPlay project’s directory, and type “activator ui”, you can continue with step 4.

#### Troubleshooting

* If you have problems with the character encoding, set the JVM’s default encoding to UTF8 via this: <http://stackoverflow.com/questions/361975/setting-the-default-java-character-encoding/623036#623036>. It should not be happen, the Play project’s default char encoding is set to UTF8
* Don’t do a clean-and-then-make (or recompile) via any IDE. Play hacks the bytecode in the background. If you clean the project, first you will only be able to recompile it via Play.
* Don’t worry, if the IDE mark for you some errors, but the project compiles and runs. The framework’s IDE support is not perfect yet.

### Windows Phone

First of all, you have to download and install the **Windows Phone SDK 8.0**.

Then, you have to complete the steps 1-4 here at the „*Quick solution with IIS Express*” section here:  
<http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj684580%28v=vs.105%29.aspx>  
(I suggest rather the saved version: WP8/**How to connect to a local web service from WP8.mht**)

* **Step 1**: Create a firewall exception to allow HTTP requests through the firewall on the port that IIS Express is using. These ports are **50135** (*WcfHost*) and **50308** (*WEB*)
* **Step 4**
  + Before starting this step, be sure you have at least once deployed (started) the *WcfHost* and the *WEB* projects. Without this, the application.config file will not contain the necessery sections to modify
  + You have to see something like this (\* the underlined rows are to be inserted by you):

<sites>

<site name=**"WEB"** id=**"2"**>

<application path=**"/"** applicationPool=**"Clr4IntegratedAppPool"**>

<virtualDirectory path=**"/"** physicalPath=**"C:\Norbi\Development\BookTera\Solution\WEB"** />

</application>

<bindings>

<binding protocol=**"http"** bindingInformation=**"\*:50308:localhost"** />

<binding protocol=**"http"** bindingInformation=**"\*:50308:192.168.1.102"** />

</bindings>

</site>

<site name=**"WcfHost"** id=**"3"**>

<application path=**"/"** applicationPool=**"Clr4IntegratedAppPool"**>

<virtualDirectory path=**"/"** physicalPath=**"C:\Norbi\Development\BookTera\Solution\WCF\WcfHost"** />

</application>

<bindings>

<binding protocol=**"http"** bindingInformation=**"\*:50135:localhost"** />

<binding protocol=**"http"** bindingInformation=**"\*:50135:192.168.1.102"** />

</bindings>

</site>

</sites>

* + Of course, you have to write your own local IP address to the application.config file, the above is just my example to make things easier.

After you are done with these first steps, you have to run the **SetupLocalIp**program, which you can find under the Solution folder: Solution/WP8/SetupLocalIp. This app will guide you and tell everything it have done.

Note: Every time your local IP address changes, you have to re-run this *SetupLocalIp* tool; and then restart the Visual Studio.

#### Running on device

First, you have to **developer unlock** your device. Follow this guidelines to achieve it:  
<http://msdn.microsoft.com/library/windows/apps/ff769508(v=vs.105).aspx>  
(I suggest rather the saved version: WP8/How to register your phone for development for Windows Phone 8.mht)

If you get here, you are able now to deploy (run) the application to the device. Congratulations ☺

Note, that if you also want to use the application, the phone’s wifi have to be on; and the phone have to have a common LAN access with the computer.

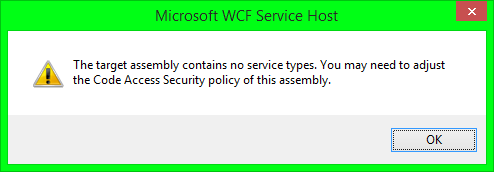
#### Troubleshooting

* Hint: I experienced a strange debug error once. It was on a Nokia Lumia 520. This phone can get crazy sometimes. So, the debugging have not worked for me: it started successfully, then after ~10-15s, it exited without prompts or logs. But if I started the app without debugging, nothing wrong happened. The solution was just to restart the phone…

### Android

TODO

# General troubleshooting

1. If you debug any project in this solution, you might see this alert window:  
     
   Do not take much attention to this, it’s a framework bug; it will cause no problem for this application. Except one thing: if you switch to another application after starting the Debug process, then this alert popups in the background (you will only hear a windows sound); you won’t be able to seek this windows by switching back to VS. First you will have to go to the desktop (win + D, or click in the bottom-right corner), and only then to VS; other way this windows won’t appear; and until you don’t click OK for it, the whole process will pause.