Provisioning production (Windows) servers

The following software needs installing on all servers:

Application Request Routing 3.0 (with URL rewrite 2.0)

This may well be installed already. You can check because it adds an ARR icon to the features view of the root node in IIS Manager. Otherwise it can be installed using the Web Platform Installer.

Open the icon mentioned above and choose proxy configuration from the right hand pane.

1. There is a checkbox to enable reverse proxying that needs to be checked.
2. Uncheck the checkbox related to rewriting hosts in the response headers

Node.js

Install node. The application is currently being tested in Circle CI with Node version 0.10.31. When you install node, check that npm is installed with it. Check that you can run both node and npm from a Windows command prompt (you may need to adjust the path settings).

IISNode

There is an x64 MSI link and install instructions here: https://github.com/tjanczuk/iisnode.

Note: for url rewriting provider development info see http://www.iis.net/learn/extensions/url-rewrite-module/developing-a-custom-rewrite-provider-for-url-rewrite-module

Setup of production servers

Little web (mobile, node.js) and Big web (PC, ASP.NET MVC4) run side by side on ports 82 and 83. Requests are reverse proxied to them by a despatcher site running on port 80. This uses Application Request Routing to proxy requests to the correct site based on the rules listed below.

Move Big web sites (grocery and login) from port 80 to 83

These are PowerShell commands to add the Grocery big-web and Login big-web with just the main domains modified to listen on port 83 instead of 80:

```
appcmd add site /name:"Grocery"  /physicalPath:$groceryPath
/bindings:"http/*:83:Nakup.itesco.cz,http/*:83:Zabezpeceni.itesco.cz,http/*:83:ezak
upy.tesco.pl,http/*:83:s.tesco.pl,http/*:83:potravinydomov.itesco.sk,http/*:83:s.it
esco.sk,http/*:83:shoponline.tescolotus.com,http/*:83:s.tescolotus.com,http/*:83:es
hop.tesco.com.my,http/*:83:s.tesco.com.my,http/*:83:bevasarlas.tesco.hu,http/*:83:s.
tesco.hu,http/*:83:js.ce-tescoassets.com,http/*:83:js.ap-tescoassets.com,http/*:83:
assets.ce-tescoassets.com,http/*:83:assets.ap-tescoassets.com,http/*:83:pi.ce-tesco
assets.com,http/*:83:pi.ap-tescoassets.com,http/*:83:elegou.cn.tesco.com,http/*:83:
pi.cn-tescoassets.com,http/*:83:js.cn-tescoassets.com,http/*:83:assets.cn-tescoasse
ts.com,http/*:83:kapimda.kipa.com.tr" /applicationDefaults.applicationPool:"Grocery"
```

```
appcmd add site /name:"Login"  /physicalPath:$loginPath
/bindings:"http/*:83:Prihlaseni.itesco.cz,http/*:83:r.tesco.pl,http/*:83:r.itesco.s
k,http/*:83:r.tescolotus.com,http/*:83:r.tesco.com.my,http/*:83:r.tesco.hu,http/*:8
3:r.cn.tesco.com,http/*:83:r.kipa.com.tr"
/applicationDefaults.applicationPool:"Grocery"
```

Hosts file

It is important that the Dispatcher calls the correct upstream virtual hosts when reverse proxying requests. These virtual hosts are on the same physical machine as the instance of the Dispatcher that is making the upstream calls. So the hosts file `c:\windows\system32\drivers\etc\hosts` must contain `localhost` aliases for all the relevant domains e.g.

```
127.0.0.1 ezakupy.tesco.pl r.tesco.pl s.tesco.pl
```

Additional `hosts` file entries (and IIS bindings) need to be added for the new APIs (Login and Grocery). e.g.:

```
127.0.0.1 grocery-api login-api
```

With IIS bindings on the Grocery and Login sites respectively:

`http://grocery-api:83` and `http://login-api:83`

Add Dispatcher site with url rewriting rules

This site is empty apart from the web.config.

In PPE and production SSL off-loading has already happened by the time the dispatcher receives the request and so it only needs bindings for `*:80`.

In DEV and STG the dispatcher now off-loads SSL negotiation and so has bindings for all hosts on port 80 and all hosts on port 443 (with self-sign SSL certificate).

The URL rewriting rules in the web config, in order, are:

1. user agent inspection for any remaining UsableNet traffic
2. querystring ux=mobile to port 82
3. querystring ui=* to port 83
4. cookie with ux=mobile to port 82
5. cookie with ui=* to port 83
6. device family == mobile to port 82
7. default to port 83

Nolio will install the custom rewrite provider in the GAC so that the 5th rule above can run.

Note: when creating the 2 sites (ighs-web and dispatcher), create appropriate service accounts and set the relevant application pool to use these credentials.

Deployment of Dispatcher to production servers

1. Code for Dispatcher is in TFS (`$/InternationalIntegration/Main/Tesco/Com/Web/UI/DeviceFamilyRewriteProvider`). An MSI will be generated as part of gated check-in.
2. Build definition for Dispatcher Manifest generation: `$InternationalDeployment/Builds/Component_Manifest_MobileDispatcher`
3. Run relevant Nolio workflow (Dev machines are UK DEV66CE and UK DEV66AP)

Deployment of Mobile Web to production servers

Jenkins is at http://ukdbt66build02v:8080/, and is set up to do the following:

1. fetch master branch from github
2. install local dependencies:`set NODE_ENV=development`
   `npm install`
3. Build assets `set NODE_ENV=production`
   `gulp`
   The gulp build step does the following:
   1. Less -> css
   2. use browserify to package client side script
   3. use envify to suggest redundant code blocks for removal by uglify
   4. use uglify to minify JS
   5. insert content hashes into file names
   6. massage css to reflect new filenames in urls
   7. minify css
4. remove node_modules directory and check them out again: `git checkout -f node_modules` This is important in order to get rid of dev dependencies.
5. `npm rebuild` to build any binaries for windows.
6. zip it all up

Then:

1. Create a manifest by kicking off a build of `$InternationalDeployment/Builds/Component_Manifest_MobileWebsite`

Then:

1. Run the Nolio deployment, specifying the build number from the step above.

Big Web URL Rewrite Outbound Rule:

Note that in order for redirects to work correctly in Big Web (e.g. those generated by a `RedirectToAction` call) we need a response header rewrite on the `Location` header that removes the port from the URL:

This is the regex:

```
^(https?)://([^/:]+)(:\d+)?(.*)
```

And this is the replacement value:

```
{R:1}://{R:2}{R:4}
```