PathFinder deployment

A simple guide to deploy PathFinder to a server.

Things you need to have in the server

1. Java 7.
2. Java application server — i.e: Apache Tomcat, WildFly (formerly JBoss AS), GlassFish.
3. At least one directory with RW access for the application server. This will be used by the application to create the database, and to store uploaded files. You need to have the path to this directory before **preparing the deployment file**. I would recommend to have two directories; one for the database and another for the files. For example: `/pathfinder/data` and `/pathfinder/file`.
4. A space about 50 MB. Application size is estimated to be less than 23 MB. There is no known limit to how large the database and the files can grow. However I would recommend to have at least 20 MB for both database and files. So, altogether is about 50 MB.

Preparing the deployment file

The deployment file can be prepared by executing a gradle command.

However, you have to configure something first.

Configure the path to the database and file

1. Open `app.properties` file located in `src/main/resources/`.
2. First, change the value for *key* **data.path** from `data/data.txt` to whatever path that you have created.
3. Then, change the value for *key* **file.path** from `file/` to whatever path that has been created.
4. Below is a sample of properties file that you might have ended up with:

```
data.path=/pathfinder/data/data.txt
file.path=/pathfinder/file/
max.content=5000
socket.timeout=10000
connect.timeout=10000
cache.timeout=600000
```

Generate the WAR file

The WAR file is the actual application that you want to deploy in the server. It contains all of the application modules including the back-end engine and the front-end UI. And it's in a form of an archive. That's why it's called a WAR file or **web application archive**.

To generate the WAR file simply run this gradle command:

```
gradle war
```

If success, then the WAR file would have been generated in `build/libs` in the same directory that you have executed the gradle command.

Deploying to the server

Deploying the WAR file to the server depends on the type of the application server that your server is using. By right, it could be trivial or not too hard to deploy the application to any application servers. It could be as simple as copy the WAR file to a certain directory. Then you might have to restart the server, or in many times not at all. It depends on the application server. Some might have hot deploy capability to detect changes in its deployment target, to trigger automatic deployment.