# Steps to Separate UI code from eCRNow application

1. Create a new directory 'eCRNow-UI'.
2. Copy the contents of frontend folder to eCRNow-UI directory.
3. Delete the frontend folder from eCRNow root directory.
4. Remove the Maven Plugins related to frontend from pom.xml. Below are the 3 plugins need to be removed from pom.xml

1) exec-maven-plugin - npm install

2) exec-maven-plugin - npm run build

3) maven-antrun-plugin - which copies the build files to backend source directory.

1. Then backend Service can be build and run individually without frontend code.

# Changes to be made in UI to run and deploy UI code:

Below are the changes to be done to build the UI code and deploy to Tomcat web server.

### Web.xml

Create a web.xml file under the root directory of frontend code and paste the below code in web.xml file.

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"

        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

        xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee

        http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"

        version="2.4">

  <display-name>eCRNow-UI</display-name>

  <error-page>

    <error-code>404</error-code>

    <location>/index.html</location>

  </error-page>

</web-app>

### Pom.xml

Create a pom.xml under the root directory of frontend code and paste the below code in pom.xml file.

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.drajer</groupId>

    <artifactId>eCRNow-UI</artifactId>

    <version>1.0</version>

    <packaging>war</packaging>

    <properties>

        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

        <npm.output.directory>build</npm.output.directory>

    </properties>

    <build>

        <finalName>${project.artifactId}</finalName>

        <plugins>

            <!-- Standard plugin to generate WAR -->

            <plugin>

                <groupId>org.apache.maven.plugins</groupId>

                <artifactId>maven-war-plugin</artifactId>

                <version>2.1.1</version>

                <configuration>

                    <webResources>

                        <resource>

                            <directory>${npm.output.directory}</directory>

                        </resource>

                    </webResources>

                    <webXml>${basedir}/web.xml</webXml>

                </configuration>

            </plugin>

            <plugin>

                <groupId>org.codehaus.mojo</groupId>

                <artifactId>exec-maven-plugin</artifactId>

                <version>1.3.2</version>

                <executions>

                    <execution>

                        <id>npm run build (compile)</id>

                        <goals>

                            <goal>exec</goal>

                        </goals>

                        <phase>compile</phase>

                        <configuration>

                            <executable>npm</executable>

                            <arguments>

                                <argument>run</argument>

                                <argument>build</argument>

                            </arguments>

                        </configuration>

                    </execution>

                </executions>

                <configuration>

                    <environmentVariables>

                        <CI>false</CI>

                        <NPM\_CONFIG\_PREFIX>${basedir}/npm</NPM\_CONFIG\_PREFIX>

                        <NPM\_CONFIG\_CACHE>${NPM\_CONFIG\_PREFIX}/cache</NPM\_CONFIG\_CACHE>

                        <NPM\_CONFIG\_TMP>${project.build.directory}/npmtmp</NPM\_CONFIG\_TMP>

                    </environmentVariables>

                </configuration>

            </plugin>

        </plugins>

    </build>

    <profiles>

        <profile>

            <id>local</id>

            <activation>

                <activeByDefault>true</activeByDefault>

            </activation>

            <build>

                <plugins>

                    <plugin>

                        <groupId>org.codehaus.mojo</groupId>

                        <artifactId>exec-maven-plugin</artifactId>

                        <configuration>

                            <environmentVariables>

                                <PUBLIC\_URL>http://localhost:8080/${project.artifactId}</PUBLIC\_URL>

                                <REACT\_APP\_ROUTER\_BASE>/${project.artifactId}</REACT\_APP\_ROUTER\_BASE>

                                <REACT\_APP\_ECR\_BASE\_URL>http://localhost:8081</REACT\_APP\_ECR\_BASE\_URL>

                            </environmentVariables>

                        </configuration>

                    </plugin>

                </plugins>

            </build>

        </profile>

        <profile>

            <id>prod</id>

            <build>

                <plugins>

                    <plugin>

                        <groupId>org.codehaus.mojo</groupId>

                        <artifactId>exec-maven-plugin</artifactId>

                        <configuration>

                            <environmentVariables>

                                <PUBLIC\_URL>http://ecr.drajer.com/${project.artifactId}</PUBLIC\_URL>

                                <REACT\_APP\_ROUTER\_BASE>/${project.artifactId}</REACT\_APP\_ROUTER\_BASE>

                                <REACT\_APP\_ECR\_BASE\_URL>http://localhost:8081</REACT\_APP\_ECR\_BASE\_URL>

                            </environmentVariables>

                        </configuration>

                    </plugin>

                </plugins>

            </build>

        </profile>

    </profiles>

</project>

### App.js:

Update the below code App.js file under ‘src’ directory. Add the below code under the render() method in App.js file

<Router basename={process.env.REACT\_APP\_ROUTER\_BASE || ''}>

### ClientDetails.js

Update the below code in ClientDetails.js file. Add the below code under saveClientDetails() method in ClientDetails.js file

fetch(process.env.REACT\_APP\_ECR\_BASE\_URL + "/api/clientDetails", {

### ClientDetailsList.js

Update the below code in ClientDetailsList.js file. Add the below code under getAllClientDetails() method in ClientDetailsList.js file

fetch(process.env.REACT\_APP\_ECR\_BASE\_URL + "/api/clientDetails/", {

### Note:

Change the PUBLIC\_URL, REACT\_APP\_ROUTER\_BASE, REACT\_APP\_ECR\_BASE\_URL values in pom.xml as per your environment.

### Instructions to Build and Deploy onto Tomcat web server:

1. Run the below command to download all the required packages.

npm install

1. Run the below command to build the code and create a package.

mvn clean install (or) mvn package

1. After the build is success, a war file will be generated in the target folder under the root directory of UI.
2. Copy the war file onto tomcat/webapps folder and start the tomcat server.
3. Once the tomcat is up and running you should be able to access the UI from browser.

http://localhost:8080/eCRNow-UI