Create project

File->New->Mule Project

Add quartz:event-generator-job + logger to flow

```
<quartz:inbound-endpoint jobName="eventTimer" repeatInterval="30000">
    <quartz:event-generator-job>
        <quartz:payload>http://espn.go.com</quartz:payload>
    </quartz:event-generator-job>
</quartz:inbound-endpoint>
<logger message="Hello from cron #[message.payload] - #[java.lang.System.currentTimeMillis()]"
     level="INFO" doc:name="Logger"
/>
```

Convert project to mvn

Right click on project

Configure->Convert to Maven

Maven->Update Project

- Every time you change dependencies
  - Update Project

Open phantomjs client in netbeans

File-Open Project...

Add phantomjs dependency

```
<dependencies>
    <dependency>
        <groupId>com.espn.phantomjs.client</groupId>
        <artifactId>phantomjs-client</artifactId>
        <version>local-SNAPSHOT</version>
    </dependency>
</dependencies>
```

Demonstrate js script to get screen shot

```
importClass(Packages.com.espn.phantomjs.PhantomJs);

var phantomjs = new PhantomJs('/usr/local/bin/phantomjs', 60);

var bytes = phantomjs.screenshot('http://espn.go.com');

result = bytes;
```

Add js script to mule print and save bytes to disk

- Drag in file component
- Set Path: /Users/arlethp1/Documents/mule presentation/screenshots
- Set Output Pattern: #[java.lang.System.currentTimeMillis()].screenshot.png

Add phantomjs spring beans

```
<spring:beans>
    <spring:bean
        id="phantomJsWebDriverClientBean"name="phantomJsWebDriverClientBean"
        class="com.espn.phantomjs.client.PhantomJsWebDriverClient"scope="singleton">
        <spring:constructor-arg value="/usr/local/bin/phantomjs"/>
        <spring:constructor-arg value="60"/>
    </spring:bean>
    <spring:bean id="screenshotBean"
        name="screenshotBean"class="com.espn.phantomjs.PhantomJs"scope="singleton">
        <spring:constructor-arg ref="phantomJsWebDriverClientBean"/>
    </spring:bean>
</spring:beans>
```

Call spring bean from script

```
var bean = registry.lookupObject('screenshotBean');
result = bean.screenshot(payload);
```

Call spring bean from set payload

```
#[app.registry['screenshotBean'].screenshot(payload)]
```

Lets talk about coupling...
(no not that kind)

Studio and mvn had to “know” about the dependency to get all of this to work

Not exactly jar hell, but it does make mule very mvn/java centric

Devkit

open connector in netbeans

Compile connector

```
mvn clean install -Ddevkit.studio.package.skip=false
```

Install connector to Studio

Help->Install New Software...

Configure connector in mule

```
<phantomjs:config
    phantomjsBinary="/usr/local/bin/phantomjs"
    timeout="60"
    name="Phantomjs" doc:name="Phantomjs"
/>
```

Add connector to flow

Create flow, and just drag and drop