- What is JNDI?
  - Java Naming and Directory Interface
    - Its a Catalogue/Inventory/List/Archive... Directory mapping Names to Resources (Bindings)
    - Not designed to replace existing systems
      - e.g. Corba Naming, DNS, LDAP, NIS, etc.
    - More like a Common Interfaces to various existing Directories
    - Has Two Goals -
      1. Bind Name to Resource
      2. Lookup Resource by Name
    - You ask for an InitialContext to Query JNDI, arguments dictate the Context -

      ```
      new javax.naming.InitialContext()
      ```

      - Contexts exist for -
        - LDAP access – InitialLdapContext
        - FileSystem access – RefFSContext
        - Java EE EJB – e.g. OpenEJB
        - Inside Tomcat WebApps, default JNDI Context is Tomcats Env.
        - You can create your own... Implement javax.naming.InitialContextFactory

- JNDI in Tomcat

env-entry and **Environment**

- Allows you to set variables, accessible from the JNDI environment
- Aimed at deployment time (context.xml/web.xml) / server config time (server.xml)
- For per-application - Should set in web.xml

```
<env-entry>
        <env-entry-name>testName</env-entry-name>
        <env-entry-type>java.lang.String</env-entry-type>
        <env-entry-value>Adam Retter</env-entry-value>
</env-entry>
```

- Or per-application... context.xml/server.xml Context (if manual deployment controlled by Sysadmins?)

```
<Environment name="testName" value="Adam Retter" type="java.lang.String" override="false"/>
```

- For Global variables – Should set in server.xml

```
<GlobalNamingResources ...>
    ...
    <Environment name="testName" value="Adam Retter" type="java.lang.String" override="false"/>
    ...
</GlobalNamingResources>
```

- How to get our Tomcat JNDI Environment Variable?

```
// Obtain our environment naming context
Context initCtx = new InitialContext();
Context envCtx = (Context) initCtx.lookup("java:comp/env");

//get our environment entry
String name = (String) envCtx.lookup("testName");
```

- JNDI in Tomcat

resource-ref and **Resource**

- Allows you to bind a Resource to a Name in the JNDI Environment
- Resources are looked up in
  - web.xml
  - context.xml
  - server.xml Context
- Aimed at deployment time (context.xml/web.xml) / server config time (server.xml)
- For per-application - Should set in web.xml

```
<resource-ref>
        <res-ref-name>mail/Session</res-ref-name>
        <res-type>javax.mail.Session</res-type>
        <res-auth>Container</res-auth>
</resource-ref>
```

- Or per-application... context.xml (if manual deployment controlled by Sysadmins?)

```
<Resource name="mail/Session" type="javax.mail.Session" auth="Container" mail.debug="true"
mail.smtp.auth="true" mail.smtp.host="my.mail.server.com" mail.user="my_mail_username"
password="my_mail_password" />
```

http://www.adamretter.co.uk
adam.retter@googlemail.com

- JNDI in Tomcat

resource-ref and **Resource** and **ResourceLink**

- Global Resources can be defined in server.xml

```
<GlobalNamingResources ...>
   ...
   <Resource name="mail/Session" type="javax.mail.Session" auth="Container" mail.debug="true"
      mail.smtp.auth="true" mail.smtp.host="my.mail.server.com" mail.user="my_mail_username"
      password="my_mail_password" />
      ...
</GlobalNamingResources>
```

- Global Resources are imported via your Web App's context.xml -

```
<ResourceLink name="mail/Session" global="mail/Session" type="javax.mail.Session"/>
```

- If your Resources are not defined in web.xml, it is considered good practice to add a resource-ref to web.xml for them regardless. This ensures your app is self-documenting about its JNDI dependencies.

- How to get our Tomcat JNDI Resource?

```
// Obtain our environment naming context
Context initCtx = new InitialContext();
Context envCtx = (Context) initCtx.lookup("java:comp/env");

//get our environment Resource
Session mailSession =  (Session)envCtx.lookup("mail/Session");
```

- NB. Exactly the same approach as getting a Tomcat JNDI Environment Entry!
- NB. Your jar files need to go into $CATALINA_HOME/lib if this is not defined in web.xml
- NB If you place jar's in $CATALINA_HOME/lib for JNDI,
  DO NOT also place them in webapp/WEB-INF/lib...

- What is JNDI?

- Java Naming and Directory Interface
  - Its a Catalogue/Inventory/List/Archive... Directory mapping Names to Resources (Bindings)
  - Not designed to replace existing systems
    - e.g. Corba Naming, DNS, LDAP, NIS, etc.
  - More like a Common Interfaces to various existing Directories
  - Has Two Goals -
    1. Bind Name to Resource
    2. Lookup Resource by Name
  - You ask for an InitialContext to Query JNDI, arguments dictate the Context -

    ```
    new javax.naming.InitialContext()
    ```

    - Contexts exist for -
      - LDAP access – InitialLdapContext
      - FileSystem access – RefFSContext
      - Java EE EJB – e.g. OpenEJB
    - Inside Tomcat WebApps, default JNDI Context is Tomcats Env.
    - You can create your own... Implement javax.naming.InitialContextFactory

Adam Retter
Verhoef Trainer and Independent Consultant
http://www.adamretter.co.uk
adam.retter@googlemail.com

- JNDI in Tomcat

env-entry and **Environment**

- Allows you to set variables, accessible from the JNDI environment
- Aimed at deployment time (context.xml/web.xml) / server config time (server.xml)
- For per-application - Should set in web.xml

```
<env-entry>
        <env-entry-name>testName</env-entry-name>
        <env-entry-type>java.lang.String</env-entry-type>
        <env-entry-value>Adam Retter</env-entry-value>
</env-entry>
```

- Or per-application... context.xml/server.xml Context (if manual deployment controlled by Sysadmins?)

```
<Environment name="testName" value="Adam Retter" type="java.lang.String" override="false"/>
```

- For Global variables – Should set in server.xml

```
<GlobalNamingResources ...>
    ...
    <Environment name="testName" value="Adam Retter" type="java.lang.String" override="false"/>
    ...
</GlobalNamingResources>
```

override="false" – determines whether this can be overridden by web.xml or higher level context.xml

- How to get our Tomcat JNDI Environment Variable?

```
// Obtain our environment naming context
Context initCtx = new InitialContext();
Context envCtx = (Context) initCtx.lookup("java:comp/env");

//get our environment entry
String name = (String) envCtx.lookup("testName");
```

- JNDI in Tomcat

resource-ref and **Resource**

- Allows you to bind a Resource to a Name in the JNDI Environment
- Resources are looked up in
  - web.xml
  - context.xml
  - server.xml Context
- Aimed at deployment time (context.xml/web.xml) / server config time (server.xml)
- For per-application - Should set in web.xml

```
<resource-ref>
        <res-ref-name>mail/Session</res-ref-name>
        <res-type>javax.mail.Session</res-type>
        <res-auth>Container</res-auth>
</resource-ref>
```

- Or per-application... context.xml (if manual deployment controlled by Sysadmins?)

```
<Resource name="mail/Session" type="javax.mail.Session" auth="Container" mail.debug="true"
mail.smtp.auth="true" mail.smtp.host="my.mail.server.com" mail.user="my_mail_username"
password="my_mail_password" />
```

- JNDI in Tomcat

resource-ref and **Resource** and **ResourceLink**

- Global Resources can be defined in server.xml

```
<GlobalNamingResources ...>
  ...
  <Resource name="mail/Session" type="javax.mail.Session" auth="Container" mail.debug="true"
    mail.smtp.auth="true" mail.smtp.host="my.mail.server.com" mail.user="my_mail_username"
    password="my_mail_password" />
    ...
</GlobalNamingResources>
```

- Global Resources are imported via your Web App's context.xml -

```
<ResourceLink name="mail/Session" global="mail/Session" type="javax.mail.Session"/>
```

- If your Resources are not defined in web.xml, it is considered good practice to add a resource-ref to web.xml for them regardless. This ensures your app is self-documenting about its JNDI dependencies.

- How to get our Tomcat JNDI Resource?

```
// Obtain our environment naming context
Context initCtx = new InitialContext();
Context envCtx = (Context) initCtx.lookup("java:comp/env");

//get our environment Resource
Session mailSession =  (Session)envCtx.lookup("mail/Session");
```

- NB. Exactly the same approach as getting a Tomcat JNDI Environment Entry!
- NB. Your jar files need to go into $CATALINA_HOME/lib if this is not defined in web.xml
- NB If you place jar's in $CATALINA_HOME/lib for JNDI,
  DO NOT also place them in webapp/WEB-INF/lib...