

En una instalación de alfresco 3.2 que estamos usando desde hace unos meses con unos cuantos centenares de usuarios nos ha surgido un problema a la hora de invitar nuevos usuarios a un espacio.

El formulario de búsqueda de usuarios sólo nos permite buscar por usuario, nombre o apellidos, es decir si buscamos "Jose Serra" buscará un usuario que tenga como login "Jose Serra" o un nombre que tenga "Jose Serra" o un apellido que tenga "Jose Serra". Esto no es demasiado problema en sistemas dónde tenemos pocos usuarios, pero ahora que manejamos centenares de usuarios empieza a ser problemática la búsqueda.

En los logs tenemos la siguiente consulta Lucene:

```
12:16:20,902 User:sapodoble DEBUG [bean.wizard.BaseInviteUsersWizard] Maximum invite users results size: 500
```

```
12:16:20,902 User:sapodoble DEBUG [bean.wizard.BaseInviteUsersWizard] Using query to find users:
```

```
@cm:firstName:"*Jose Serra*" @cm:lastName:"*Jose Serra*" @cm:userName:Jose Serra*
```

La clase que construye la búsqueda es *(org.alfresco.web.bean.wizard) BaseInviteUsersWizard* concretamente la función *pickerCallback()* que

hace uso de

*(org.alfresco.web.ui.common.Utills)*

*Utills.generatePersonSearch(StringBuilder query, String term)* dónde se crea realmente la consulta, por tanto, es esta última la que deberemos modificar:

Clase original de *Utills.class*

```
public static void generatePersonSearch(StringBuilder query, String term)
{
```

```
    query.append("@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:firstName:*");
    query.append(term);
```

```
        query.append("@"  
@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:lastName:");  
        query.append(term);  
        query.append("@"  
@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:userName:");  
        query.append(term);  
        query.append("");  
    }
```

Una solución sencilla pasa por buscar cualquier item de la consulta en cualquier campo, con esto de momento nos soluciona el problema de las búsquedas por nombre y apellidos.

Modificar la clase `org.alfresco.web.ui.common.Utils`

```
public static void generatePersonSearch(StringBuilder query, String term)  
{  
    /*Modificació per poder consultar nom + cognoms*/  
    StringBuffer strb = new StringBuffer ();  
  
    query.append("@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:userN  
ame:");  
    query.append(term);  
  
    StringTokenizer st = new StringTokenizer(term.trim());  
    while(st.hasMoreTokens()){  
        String mtoken = st.nextToken();  
  
        query.append("@"@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:first  
Name:");  
        query.append(mtoken);  
  
        query.append("@"  
@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:lastName:");  
        query.append(mtoken);  
    }  
    query.append("");  
}
```

La solución definitiva que vamos a adoptar es la siguiente:

- Búsquedas con una única cadena: buscará que la cadena forme parte de los atributos: nombre, apellidos, usuario y dirección de correo.
- Búsquedas de 2 cadenas: buscará que la primera cadena forme parte del nombre y la

segunda de los apellidos. A la inversa y que cualquier cadena forme parte del nombre de usuario.

- Búsquedas con 3 o más cadenas: las dos últimas cadenas se consideraran apellidos, y todo lo anterior se considerará nombre para realizar la búsqueda.

Código de la modificación:

```
public static void generatePersonSearch(StringBuilder query, String term)
{
    StringBuffer strb = new StringBuffer ();
    StringTokenizer st = new StringTokenizer(term.trim());

    if (st.countTokens()==1)
    {
        query.append("@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:firstName:");
        query.append(term);
        query.append(" AND ")
        query.append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:lastName:");
        query.append(term);
        query.append(" AND ")
        query.append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:userName:");
        query.append(term);
        query.append(" AND ")
        query.append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:email:");
        query.append(term);
        query.append(" AND ");
    }
    else if(st.countTokens()==2)
    {
        String token1 = new String(st.nextToken());
        String token2 = new String(st.nextToken());

        query.append("(@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:firstName:");
        query.append(token1);
        query.append(" AND ")
        query.append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:lastName:");
        query.append(token2);
        query.append(" AND ");

        query.append(" OR ");
    }
}
```

```
query.append("(").append("@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:lastName:");
    query.append(token1);
    query.append("'" AND
    @").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:firstName:");
    query.append(token2);
    query.append("'"");

    query.append(" OR ");
```

```
query.append("(").append("@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:userName:");
    query.append(token1);
    query.append("'" OR
    @").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:userName:");
    query.append(token2);
    query.append("'"");
} else
{
    query.append("(");
    for (int i=0; i < (st.countTokens()-2) ; i++)
    {
        String mtoken = st.nextToken();
```

```
query.append("@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:firstName:");
    query.append(mtoken);
    query.append("'" ");
    }
    query.append(") AND (");

    String token1 = new String(st.nextToken());
    String token2 = new String(st.nextToken());
```

```
query.append("@").append(NamespaceService.CONTENT_MODEL_PREFIX).append("\:lastName:");
    query.append(token1);
    query.append(" ");
    query.append(token2);
    query.append('');

    query.append(")");
}
}
```