Formation ESUP Portail
Cours 4

http://www.esup-portail.org

Plan

- Prérequis
- Outils
- Normes Esup portail
- Canal uPortal
- Publication d'un canal
- Web services

Prérequis

- Java
- Ant
- XML
- XSL/XSLT

Outils

- http://www.esup-portail.org/consortium/espace/Normes_1C/tech/build
- build.properties
- build.xml

build.properties

- Les informations sur le projet
  - app.name.home=CMonCanal
  - app.version=1.5
  - app.name.deploy=esup-canal-MonCanal
- Les répertoires
  - quickstart.home = /esupdev/esupdev-2.3/uPortal_2-3-1-quick-start
  - tomcat.home = ${quickstart.home}/Tomcat_5-0-18
  - uportal.home = ${quickstart.home}/uPortal_rel-2-3-1
  - deploy.home = ${quickstart.home}/webapps/uPortal
  - deploy.tree = org/esupportail/portal/channels
  - properties.tree = org_esup

build.xml

- all
- prepare
- clean
- undeploy
- deploy
- compil
- javadoc
- dist
- buildtar
- buildzip

Conventions de codage ESUP Portail

Normes Esup portail

- Conventions de codage
- Règles de nommage des packages
- Règles de nommage des fichiers de configuration
- Lecture des fichiers de configuration
- Règles de nommage des livrables ESUP

Conventions de codage

- Licence (Voir fichier license)
- Cartouche explicatif (Voir annexe1)
- Code pour javadoc
- Dans un package, la classe principale comporte le même nom que le package.
- Nom de classe : CNomCanal
- _variableprivee
- Utiliser get et set pour accéder aux variable
- http://www.esup-portail.org/consortium/espace/Normes_1C/recommandations/Conve

Règles de nommage des packages

- Package spécifique à une université
  - fr.univxxx.portal.channels.CCanal
  - fr.univxxx.portal.utils.badgeuses
- Package commun au projet esup
  - org.esupportail.portal.channels.CCanal
  - org.esupportail.portal.utils.db

Règles de nommage des fichiers de configuration

- <deploy_home>/web-inf/classes/properties/channels/org_esup/CCanal/CCanal.xml
- <deploy_home>/web-inf/classes/properties/channels/fr_univxxx/CCanal/CCanal.xml

Lecture des fichiers de configuration

- Utilisation de DIGESTER
- Voir exemple fourni

Règles de nommage des livrables ESUP

- Cas d'un produit ESUP Portail : Esup-phpcas-0.4.4[-RCx]
- Cas d'un produit existant patché par ESUP Portail : Cas-server-2.0.11-esup-y[-RCx]
- Contenu du package :

- README
- INSTALL
- UPGRADE
- ChangeLog
- LICENCE
- pubchan_<nom_du_module>.xml
- db/
- source/
- build/
- dist/
- docs/
- docs/api
- lib/
- properties/
- tests/
- webpages/

Création d'un canal uPortal

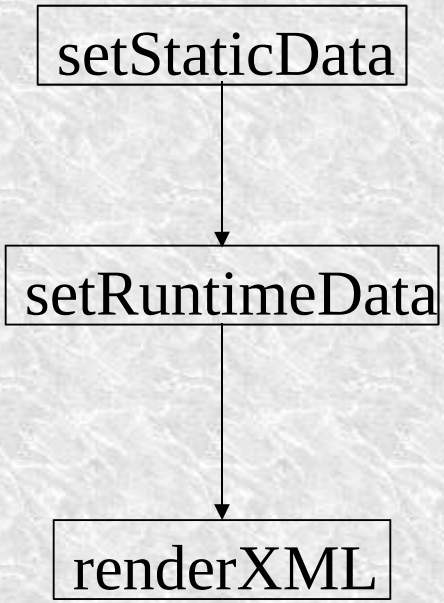
Canal uPortal

- Présentation des principales interfaces
- Cycle de vie d'un canal
- Les différentes méthodes à implémenter
- Exemple simple : Hello World
- Exemple avec récupération du login
- Exemple avec demande d'un ticket cas
- Exercice

Présentation des Principales interfaces

Nom de l'interface	But
IChannel	L'interface de base
ICacheable	Cette interface permet d'obtenir une fonctionnalité de cache. Elle permet donc d'éviter un traitement lourd pour afficher toujours la même chose.
IPrivileged	Interface technique , utiliser IPrivilegedChannel
IPrivilegedChannel	Permet d'accéder aux données internes du portail
IMultithreadedChannel	Permet l'ecriture d'un canal multi-threadé (une seule instance partagée entre les utilisateurs)

Cycle de vie d'un canal



Les différentes méthodes

GetRuntimeProperties

- public ChannelRuntimeProperties getRuntimeProperties()
{

return new ChannelRuntimeProperties() ;

}
- Permet au canal de signaler au portail s'il doit s'afficher ou pas en renvoyant un objet ChannelRuntimeProperties. A la création de cet objet la propriété pilotant l'affichage est vrai par défaut.

ReceiveEvent

- public void receiveEvent(PortalEvent ev) {}
- Le portail informe le canal des actions de l'utilisateur en appelant cette méthode. Pour cela le portail passe un objet PortalEvent dont l'appel à la méthode getEventNumber permet au canal de savoir si l'utilisateur a cliqué sur l'un des boutons suivants : "about", "detach", "edit", "help" ou si l'utilisateur supprime le canal ou si l'utilisateur ferme sa session uPortal.
- Par exemple, si l'on gère une page "about", c'est là que l'on va détecter que l'utilisateur demande son affichage. On mémorise alors l'information pour l'utiliser au moment de l'affichage par renderXML.

```
if (ev.getEventNumber() == PortalEvent.ABOUT_BUTTON_EVENT) {
    mode = MODE_INFO;
```

SetStaticData

- public void setStaticData(ChannelStaticData sd)
- ChannelStaticData permet de retrouver des informations sur la configuration du canal ou l'utilisateur courant.
- Cette méthode est appelée aussitôt après le constructeur et peut servir, comme ce dernier, à initialiser des variables.

SetRuntimeData

- public void setRuntimeData(ChannelRuntimeData rd) {

  this.runtimeData = rd ;

  }
- ChannelRuntimeData permet de retrouver des informations d'exécution comme les paramètres d'un lien cliqué dans le canal.
- Cette méthode est appelée juste avant l'affichage.

RenderXml

- public void renderXML(ContentHandler out) throws PortalException

- Cette méthode génère le XML qui sera traité par uPortal pour l'affichage.

Exemples

Exemple simple : Hello World

- Points intéressants
  - Récupération de paramètres (POST ou GET)
  - Gestion d'un mode « A Propos »
  - XSL adaptés au navigateur
  - Passage de paramètres au XSL

Exemple simple : Hello World

- Type IChannel
- Composition
  - ChelloWorld.java : classe java
  - ChelloWorld.ssl : orientation vers la bonne feuille de style
  - normal_explorer.xsl : feuille de style pour IE
  - normal_netscape.xsl : feuille de style pour mozilla
  - about.xsl : feuille de style pour le mode « A Propos »

Exemple simple : Hello World

- Récupération de paramètres (POST ou GET)
  - La récupération se fait en général dans la méthode setRuntimeData
  - On utilise runtimeData.getParameter

```
public void setRuntimeData(ChannelRuntimeData rd) {
  // Most of the processing is usually done here.
  this.runtimeData = rd;

  // process the form submissions
  if (runtimeData.getParameter("submit") != null) {
     name = runtimeData.getParameter("name");
     name_prev = name;
  }

  if (runtimeData.getParameter("clear") != null) {
     name_prev = "";
  }

  if (runtimeData.getParameter("back") != null) {
     mode = NORMAL_MODE;
  }
}
```

Le portail passe un objet ChannelRuntimeData qui contient les paramètres

Récupération de paramètres spécifiques

Exemple simple : Hello World

- Gestion d'un mode « A Propos »
  - La récupération se fait dans la méthode receiveEvent
  - On utilise un objet de type PortalEvent
  - On utilise aussi la constante PortalEvent.ABOUT_BUTTON_EVENT pour identifier le clique sur «A Propos»

```
public void receiveEvent(PortalEvent ev) {
    if (ev.getEventNumber() == PortalEvent.ABOUT_BUTTON_EVENT) {
        mode = ABOUT_MODE;
    }
}
```

Objet qui stocke les événements

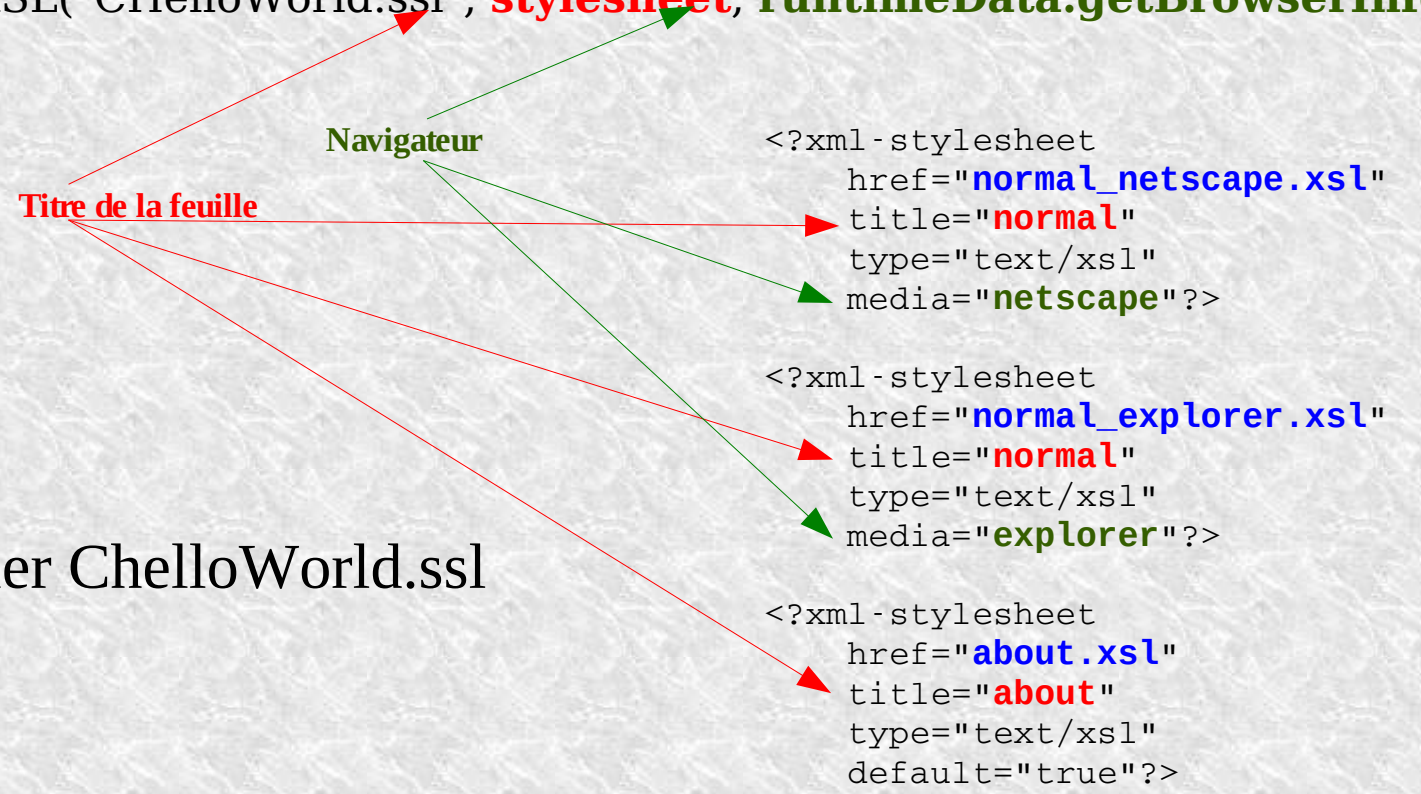
Constante pour le boutton A Propos

  - NB : Il existe d'autre événements :
    - UNSUBSCRIBE : désincription du canal
    - EDIT_BUTTON_EVENT : boutton edit
    - HELP_BUTTON_EVENT : boutton aide
    - DETACH_BUTTON_EVENT : boutton détacher

Exemple simple : Hello World

- XSL adaptés au navigateur
  - Dans la méthode renderXML :

```
xslt.setXSL("CHelloWorld.ssl", stylesheet, runtimeData.getBrowserInfo
  ());
```



- Dans le fichier ChelloWorld.ssl

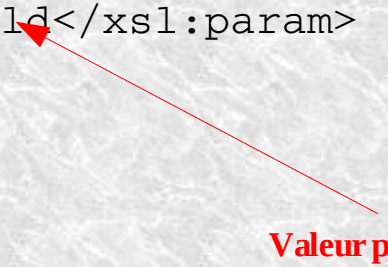
Exemple simple : Hello World

- Passage de paramètres au XSL
  - Dans la méthode renderXML :


```
xslt.setStylesheetParameter("name_prev", name_prev);
```

  - **Dans les feuilles de styles XSL :**

```
<xsl:param name="name_prev">world</xsl:param>
```



Hello World

Voir : CHelloWorld.zip

Exemple avec récupération du login

- **Récupération de l'objet IPerson**
  - Utilisation de l'objet **ChannelStaticData staticData**

```
IPerson userConnect = null;
// Get the person object
userConnect = staticData.getPerson();
```

- **Récupération du login**
  - Utilisation de l'objet IPerson

```
String login="";
// Get the login of the person
login = (String) userConnect.getAttribute(IPerson.USERNAME);
```

Méthode permettant de récupérer un attribut

Constant qui contient le nom de l'attribut pour le nom d'utilisateur

récupération du login

Voir : CGetLogin.zip

Exemple avec demande ticket CAS

- **Utilisation d'une librairie eSup-Portail**

```
import org.esupportail.portal.utils.*;
import org.esupportail.portal.utils.CASExceptions.*;
```

- Récupération du ticket

```
String ticket="";
String Service = "imap://univ.fr";
ticket = CAS.get_pt(staticData, Service);
```

Exemple avec demande ticket CAS

- **Exceptions remontées**
  - `CASTempException : Exception de type temporaire`
  - `CASPermException : Exception de type permanente`
  - `CASGenericException : Exception de type générale (plus large)`
- **Récupération du code et du message**
  - `CodeException=exp.getExceptionCode();`
  - `MsgException=exp.getMessage();`
- **Constante des codes**
  - `CASGenericException.CAS_BAD_PGT`
  - `CASGenericException.CODE_NO_SERVICE_CALL`
  - `CASGenericException.CODE_NO_CAS_AUTH`
  - `CASGenericException.CODE_NO_CAS_SECURITY_CONTEXT`
  - `CASGenericException.CODE_CAS_IO_ERROR`
  - `CASGenericException.CODE_CAS_INTERNAL_ERROR`
  - `CASGenericException.CODE_CAS_BAD_PGT`
  - `CASGenericException.CODE_CAS_INVALID_REQUEST`
  - `CASGenericException.CODE_CAS_NOT_VALID_PT`

récupération du ticket

Voir : CGetTicket.zip

Publication du canal

Publication d'un canal

- Utiliser le gestionnaire de canaux
- Utiliser un fichier XML
  - ant uportal.pubchan -Dchannel=Canal.xml

Exercice

Exercice

- Prendre l'exemple CHelloWorld
  - déployer
  - publier à l'aide du fichier xml
  - visualiser le résultat
- Modifier pour récupérer le login de l'utilisateur
  - modifier le source
  - modifier la feuille de style (enlever le formulaire+afficher le login)
  - compiler, deployer, visualiser

Les Web services

Web services

- Présentation des web services
- Axis
- Exemple
- Exercice

Web Services

- Fonctions distantes
  - Encapsulation XML des données
    - SOAP
    - XML-RPC
  - Transport des données en http (ou smtp)
  - Compatibilité entre différents langages
- Description à l'aide de WSDL (format XML)
- Déclaration à l'aide de UDDI

Axis

- Conteneur de Web Services
- Bibliothèque cliente de Web Services
- Outils
  - java2WSDL
  - WSDL2java
  - TCPMonitor

Conteneur de Web Services 1

- Axis tourne en tant que contexte Tomcat
- On déploie des classes java dans ce contexte (outil spécifiques pour le faire)
- Déploiement décrit dans un fichier WSDD (format XML)
  - Scope
  - Fonctions accessibles
  - Etc.

Exemple

Conteneur de Web Services 2

- Exemple de classe java

```java
package org.esupportail.formation.exemple.webservices;

/**
 * @author  lscri, Raymond Bourges
 *
 */
public class Calculator {
	private int nbCall = 0;

	public int add(int a, int b){
		nbCall++;
		return a+b;
	}

	public int getNbCall() {
		return nbCall;
	}
}
```

Conteneur de Web Services 3

- Fichier de déploiement

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
   xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
       <service name="Calculator" provider="java:RPC">
           <parameter name="className"
value="org.esupportail.formation.exemple.webservices.Calculator"
   />
           <parameter name="allowedMethods" value="*"/>
       </service>
</deployment>
```

Conteneur de Web Services 4

- Target ant de déploiement

```
<target name="deployWsdd" description="publication du Web Service">
        <java fork="true" dir="."
         classname="org.apache.axis.client.AdminClient">

                <classpath refid="classpath"/>

                <arg value="deploy.wsdd"/>

               <arg value="-lhttp://localhost:8080/axis/services"/>

        </java>

 </target>
```

- Ne pas oublier de copier la classe dans l'environnement tomcat de Axis

Conteneur de Web Services 5

- Le web service est accessible via
  - http://localhost:8080/axis/services/Calculator
- Le WSDL du service est accessible via
  - http://localhost:8080/axis/services/Calculator?wsdl

Ecrire un client 1

- Utilisation de WSDL2Java
  - Permet de générer des classes java clientes du WebService
- Target ant d'appel de WSDL2Java

```
<target name="wsdl2java">
  <java fork="true" dir="c:/tmp"
classname="org.apache.axis.wsdl.WSDL2Java">
    <classpath refid="classpath"/>
      <arg
value="http://localhost:8080/axis/services/Calculator?wsdl"/>
  </java>
</target>
```

Ecrire un client 2

- Placer les classes dans le code source de l'application client
- Utilisation de ces classes

```
public class Test {
  public static void main(String[] args) throws Exception {
    localhost.axis.services.Calculator.Calculator calc =
      new CalculatorServiceLocator().getCalculator();
   int tmp = calc.add(1, 2);
   System.out.print("Résulat --> "+tmp);
  }
}
```

Gestion des sessions

- Ajout au WSDD
  - <parameter name="scope" value="session"/>
- Modification du code client
  - ((CalculatorSoapBindingStub)calc).
    setMaintainSession(**true**);

TCP Monitor

- Outil pour observer le dialogue http
- On lance via ant
  - org.apache.axis.utils.tcpmon
- On modifie le code pour utiliser un autre port
  - getCalculator(**new** URL ("http://localhost:80/axis/services/Calculator"))

Exercice

- Installer AXIS
- Utiliser l'exemple donné
- Déployer les services
- Générer les classes clientes
- Tester avec les classes de test fournies

Annexe 1

```
/**
   *
   * Description :<br/>
   * ....
   * @version $Id : CExemple.java, V1.0,  15 juin 2004<br/>
   * Copyright (c) 2004 Esup Portail (www.esup-portail.org)<br/>
   * Classe(s) : CExemple<br/>
   * @author ...<br/>
   */
```