

Type custom : les interfaces de programmations ESUP Portail

[IChannel](#) | [IPrivileged](#) | [IMultithreaded](#) | [IServant](#) | [ICacheable](#) | [IMimeResponse](#) | [Interactions](#)

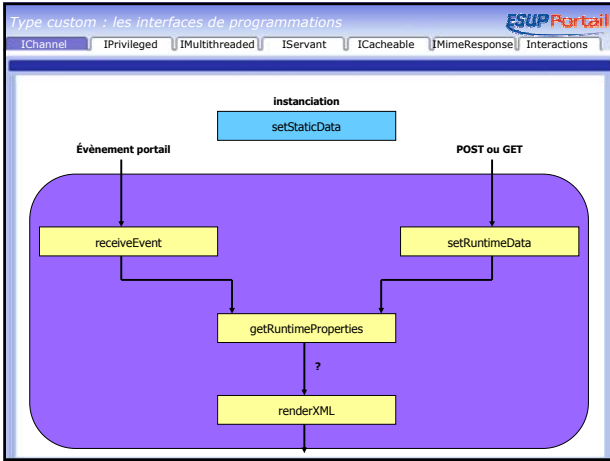
IChannel : canal standard, stateful

5 méthodes à implémenter :

- `public void setStaticData(ChannelStaticData sd)`
- `public void setRuntimeData(ChannelRuntimeData rd)`
- `public void receiveEvent(PortalEvent ev)`
- `public ChannelRuntimeProperties getRuntimeProperties()`
- `public void renderXML(ContentHandler out)`

Une implémentation de référence à étendre : BaseChannel

- **ChannelStaticData** : contient les données statiques du canal, son identifiant, un objet contenant les attributs de la personne connectée et un objet permettant d'accéder au contexte applicatif.
- **ChannelRuntimeData** : contient les données dynamiques du canal, c'est-à-dire les paramètres envoyés en POST ou en GET par le navigateur, l'URL de base du canal, l'URL de base des media, l'URL de base du worker de download.
- **ChannelRuntimeProperties** : indique au framework si il faut ou pas effectuer le rendu du canal



Type custom : les interfaces de programmations ESUP Portail

[IChannel](#) | [IPrivileged](#) | [IMultithreaded](#) | [IServant](#) | [ICacheable](#) | [IMimeResponse](#) | [Interactions](#)

IPrivileged : canal privilégié

1 seule méthode à implémenter :

- `public void setPortalControlStructures(PortalControlStructures pcs)`

Aucune implémentation de référence

IPrivilegedChannel : hérite de IChannel et de IPrivileged → 6 méthodes à implémenter

- **PortalControlStructures** : objet permettant d'accéder en lecture et en écriture aux objets `HttpServletRequest`, `HttpServletResponse`

Type custom : les interfaces de programmations ESUP Portail

IChannel | IPrivileged | **IMultithreaded** | IServant | ICacheable | IMimeResponse | Interactions

IMultithreadedChannel : canal standard, stateless

5 méthodes à implémenter :

- public void setStaticData(ChannelStaticData sd, String uid)
- public void setRuntimeData(ChannelRuntimeData rd, String uid)
- public void receiveEvent(PortalEvent ev, String uid)
- public ChannelRuntimeProperties getRuntimeProperties(String uid)
- public void renderXML(ContentHandler out, String uid)

Une implémentation de référence à étendre : BaseMultithreadedChannel

- uid est une chaîne de caractères permettant d'identifier l'instance de canal en cours d'exécution.
- BaseMultithreadedChannel propose une variable statique et des méthodes permettant de sauvegarder l'état de chaque instance de canal dans des Hashmap synchronisées.

Type custom : les interfaces de programmations ESUP Portail

IChannel | IPrivileged | IMultithreaded | **IServant** | ICacheable | IMimeResponse | Interactions

IServant : canal esclave

2 méthodes à implémenter :

- public boolean isFinished()
- public Object [] getResults()

Aucune implémentation de référence

Permet l'utilisation du canal en mode esclave par un autre canal. Le canal appelant reprend automatiquement la main lorsque isFinished retourne 'true' et si le canal esclave a produit des résultats, le canal appelant peut les récupérer en appelant la méthode getResults

Type custom : les interfaces de programmations ESUP Portail

IChannel | IPrivileged | IMultithreaded | IServant | **ICacheable** | IMimeResponse | Interactions

ICacheable : canal standard, gestion d'un cache

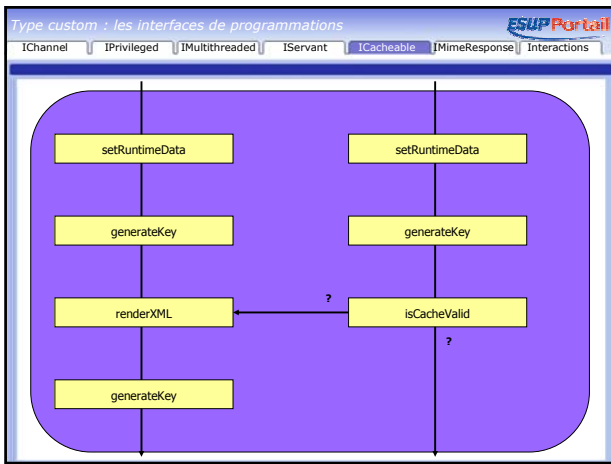
2 méthodes à implémenter :

- public ChannelCacheKey generateKey()
- public boolean isCacheValid(Object validity)

Aucune implémentation de référence

Suivant le type d'objet que l'on place dans le cache, on va pouvoir réaliser différents types de cache :

- Cache temporel (l'objet contient une heure / date)
- Cache utilisateur (l'objet contient le login utilisateur)
- Cache canal (l'objet contient le channelID)
- Cache applicatif (l'objet contient le nom de l'application)
- ...



Type custom : les interfaces de programmations ESUP Portail

IChannel | IPrivileged | IMultithreaded | IServant | ICacheable | IMimeResponse | Interactions

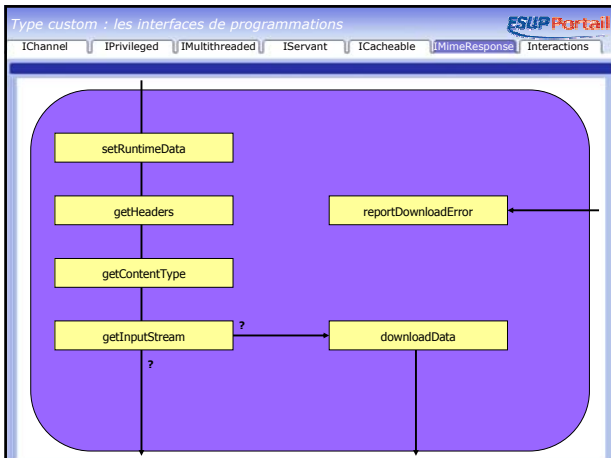
IMimeResponse : téléchargement de fichier


6 méthodes à implémenter :

- `public String getContentType()`
- `public InputStream getInputStream()`
- `public void downloadData(OutputStream out)`
- `public String getName()`
- `public Map getHeaders()`
- `public void reportDownloadError(Exception e)`

Aucune implémentation de référence

Permet le téléchargement d'un fichier du portail vers le navigateur du client



Type custom : les interfaces de programmations 

IChannel | IPrivileged | IMultithreaded | IServant | ICacheable | IMimeResponse | Interactions

Dans le code Java :

L'objet XSLT propose 3 méthodes :

```
setStylesheetParameter(String, String);
setStylesheetParameters(HashMap);
setStylesheetParameters(Hashtable);

xslt.setStylesheetParameter("paramname", "paramvalue");
```

Dans le code XSL :

```
<param name="paramname" />
...
<a href="http://www.toto.fr?param=${paramname}">lien</a>
```
