

LIFERAY PORTLETを読み解く

日本Liferayユーザグループ
林田 敦

自己紹介

□ 林田 敦

- 社会人1年目

- 某社でOSSに関する業務に従事

- FB : <http://www.facebook.com/atsushi.hayashida.5>

□ OSS活動

- 学生時代は日本Androidの会神戸支部に所属

- 先輩に連れられて丸の内MongoDB勉強会に参加

アジェンダ

1. Liferay Portalの簡単なお紹介
2. Liferay Portletを読み解く

始める前に・・・

ice break

皆さんのリテラシを教えてください

- Liferayに触ったことありますか？
 1. 情強だから開発できる
 2. 情弱だから触ったことない
 3. Liferayって何？
 4. 情強とか情弱って何？

セミナーの趣旨

- 3番が多いことを想定しています
 - Liferayって何？そもそもOSS自体あまり知らない・・・という方が対象
 - インストール方法から、ゆっくり解説していきます
- 目標
 - Liferay Portletの中身をざっと理解する
 - Liferay Portletを自作するための足がかりを得る
 - そのために、スクリーン上で一緒にソースを読みます

セミナーの注意点

- Liferay初心者の、Liferay初心者による、Liferay初心者のためのセミナー
 - 初心者おいてけぼりなセミナーはやりたくない！
 - 初心者でもOSSを盛り上げたい！
 - 情強な方々は、情弱な我々を救ってください

Liferay Portalの簡単なご紹介

1st section

Liferay Portalって何？

□ ポータル？

- wwwにアクセスするときの入口となるウェブサイトのこと

- 例: Yahoo! JAPAN、Google...

- Liferayは、企業情報ポータル等で使われます

- 企業内の散らばっているデータを集約表示したり

- ユーザや役職に応じた最適画面を作成したり

□ もちろんオープンソースです

- LGPLライセンス

Liferay Portalって何？

□ 特徴

□ 多機能

- 90以上のポートレット(部品群)

□ 高い操作性

- マウスを使った直観的な操作が可能
- デザインのアレンジが容易

□ 高い拡張性

- ポートレット(部品)単位での開発が可能
- JSR286 (Javaポートレットの標準仕様) 準拠
- SDKが用意されている

Liferay Portalを構築してみる

- 今回の環境
 - Windows7(64bit版)
 - Liferayはバージョン6.1.1を利用

Liferay Portalを構築してみる

- ローカルでLiferay Portalを構築してみよう
 1. www.liferay.comから「liferay-portal-6.1.1-ce-ga2」(Bundled with Tomcat)のzipファイルをダウンロードして解凍
 2. 解凍したディレクトリ中の「tomcat-7.0.27/bin/startup.bat」を起動
 3. 以上

PRACTICE

Liferayを実際に構築して、操作してみよう！

前提条件

- Liferay本体のzipはダウンロードして解凍済

操作内容

- startup.batを実行してLiferayを起動する
- 起動したLiferayをちょっとだけ触ってみる

確認事項

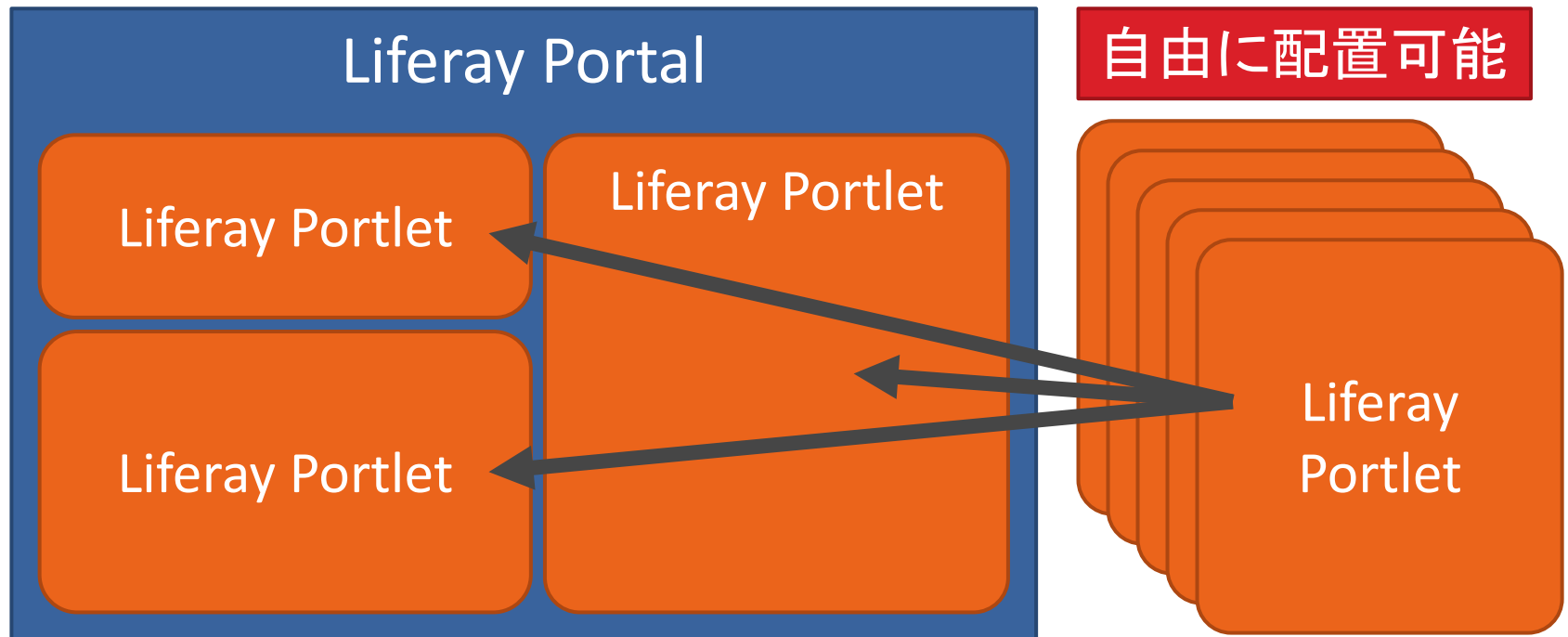
- Liferayが簡単に起動可能であること
- GUIで簡単に操作できること

Liferay Portletを読み解く

2nd section

Liferay Portletって何？

- Liferay Portalの画面を構成するパーツ
 - ガジェットみたいなもの



PRACTICE

Portletを操作してみよう！

操作内容

- Liferay上でPortletを追加する
- Portletの編集を行う

確認事項

- 多数のPortletが用意されていること
- Portletの追加や編集が簡単に行えること

自作ポートレットを作ってみる

□ まずはポートレットを作ってみよう

1. www.liferay.comから「liferay-plugins-sdk-6.1.1」(SDK)をダウンロードして解凍
2. コマンドプロンプトを起動し、SDK中の「portlets」ディレクトリ(ポートレットを作る場所)へ移動
3. コマンド「`create.bat hello-liferay "Hello Liferay"`」を実行
4. 以上

ポートレット名

ポートレットタイトル

自作ポートレットを作ってみる

- ですが、失敗します
 - 1回目 : antコマンド (Java用のメイクツール) が無い
 - 手動でantをインストール (ダウンロードして解凍後、PATH通す)
 - 2回目 : ECJ (Eclipseの独自Javaコンパイラ) が無い
 - 自動的にインストールされるので、もう一回コマンド実行
 - 3回目 : Tomcatのパスが違う
 - SDKの「build.properties」ファイルを以下のように修正

```
app.server.dir=${project.dir}/../bundles/tomcat-7.0.27  
app.server.deploy.dir=${app.server.dir}/webapps
```



```
app.server.dir=[Liferayのインストールディレクトリパス]/tomcat-7.0.27  
app.server.deploy.dir=[Liferayのインストールディレクトリパス]/deploy
```

自作ポートレットを作ってみる

- これで、ポートレットが作成できる
 1. コマンド「`create.bat hello-liferay "Hello Liferay"`」を実行
 2. 「`portlets/hello-liferay-portlet`」ディレクトリができあがる(これがポートレットの実体)
 3. 以上

PRACTICE

Portletを作ってみよう！

前提条件

- antコマンドとECJはインストール済
- build.propertiesファイルは修正済 (tomcatのパス)

操作内容

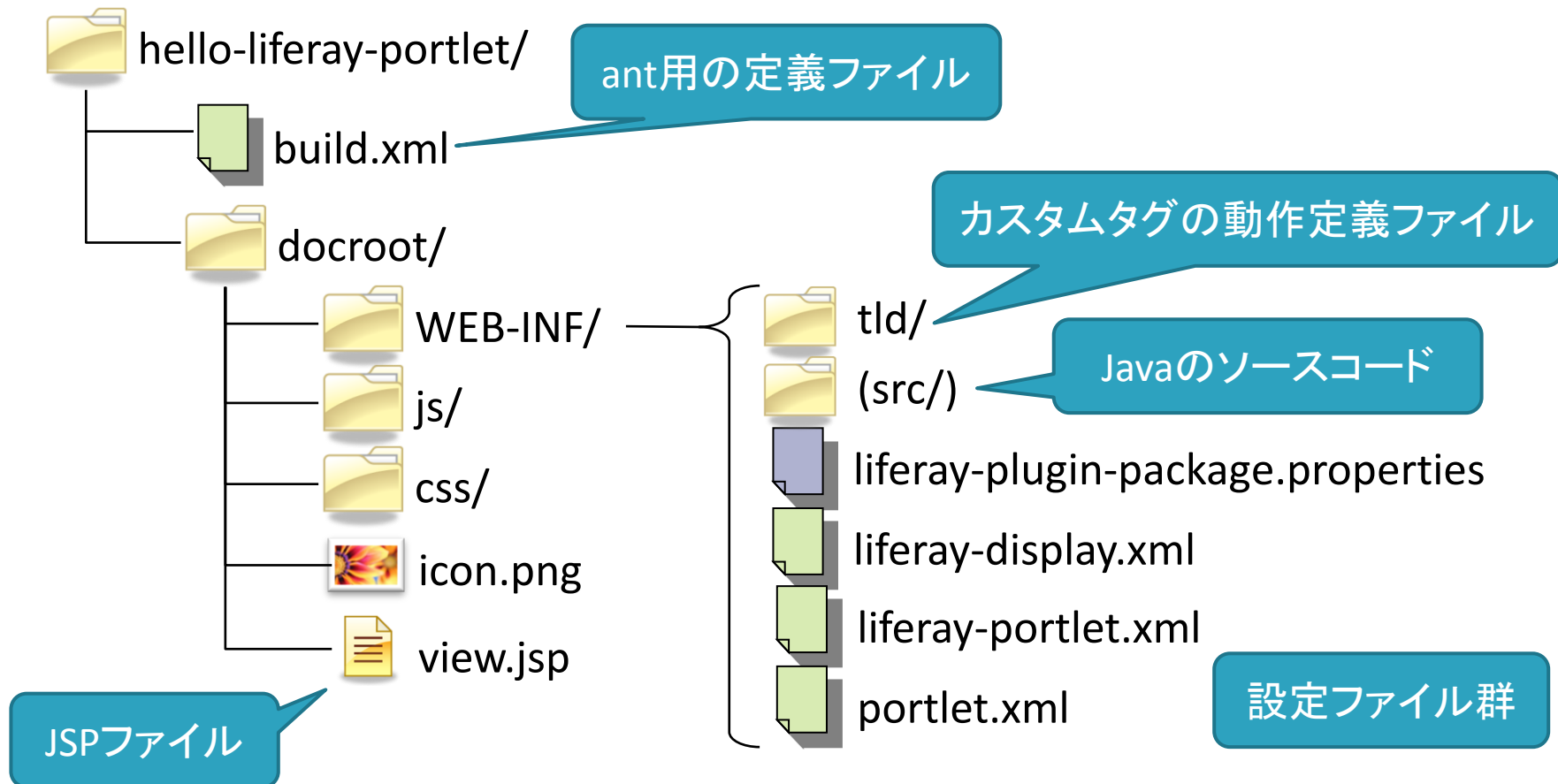
- hello-liferay-portletを作成する
 - SDKで「create.bat hello-liferay “Hello Liferay”」を実行

確認事項

- 簡単にPortletが作成できること
 - SDKの「portlet」ディレクトリ以下に「hello-liferay-portlet」ディレクトリが自動的に作成される

Liferay Portletを読み解く

- 早速、自作ポータルレットの中身を覗いてみよう



Liferay Portletを読み解く

- 各設定ファイルの内容 (WEB-INFの中身)
 - liferay-plugin-package.properties
 - 本プラグインパッケージの情報
 - liferay-display.xml
 - インストール後に表示されるポートレットのカテゴリ定義
 - liferay-portlet.xml
 - Liferayポートレット情報定義 (cssやjs、iconなどへのPATH)
 - portlet.xml
 - ポートレット情報定義 (初期画面、タイトル、resourceなど)

PRACTICE

ソースを読んでみよう！

操作内容

- portletを構成する各ファイルのソースを読む
 - view.jsp
 - 設定ファイル群(～.xml、～.properties)

確認事項

- ファイル構成や各ファイルの意味を知る

自作ポートレットをデプロイしてみる

- デプロイって？
 - アプリを配置して、実際にポートレットを追加できる状態にすること
 - TomcatバンドルのLiferayの場合、Tomcatのwebappsディレクトリ
- ポートレットをデプロイする方法
 - コマンドプロンプトを起動し、SDK中の「portlets/hello-liferay-portlet」ディレクトリへ移動
 - コマンド「ant deploy」実行
 - 以上

自作ポータルレットをデプロイしてみる

- すると、Liferayにデプロイされる
 - antにより、Liferayの「deploy/」下にwarファイルとしてデプロイされる
 - Liferay起動中は、Liferayにより自動的に「tomcat-7.0.27/webapps/」下に展開される
 - 必要なファイル群も自動的に配置してくれる

PRACTICE

自作ポートレットをデプロイしてみよう！

操作内容

- Liferayを停止する
- SDK中の「portlets/hello-liferay-portlet」ディレクトリへ移動する
- コマンド「ant deploy」を実行する
- Liferayを起動する

確認事項

- warファイルがLiferayの「deploy/」下にデプロイされること
- Liferay起動後、ポートレットがLiferayの「tomcat-7.0.27/webapps/」下に展開されること

PRACTICE

自作ポートレットを画面に追加してみよう！

操作内容

- Liferay上で自作ポートレットを画面に追加する

確認事項

- 自作ポートレットも同じように画面に追加できること

カスタマイズしてみる(その①)

□ 要件

□ ログインユーザ名を表示したい

- 例:「Hello, a-hayashida!」



ログインユーザ名

カスタマイズしてみる(その①)

□ 方法

1. view.jspに以下のコードを追加
 - `<%@ taglib uri="http://liferay.com/tld/theme" prefix="liferay-theme" %>`
 - `<liferay-theme:defineObjects />`
 - `Hello, <%= user.getScreenName() %>!
`
2. 以上

カスタマイズしてみる(その①)

- Tips: ログインユーザ名の取得
 - ▣ liferay-theme.tldからソースコードを追ってみる
 - defineObjectsで呼び出されるDefineObjectsTagクラス
 - liferay-portal-src-6.1.1-ce-ga2¥util-taglib¥src¥com¥liferay¥taglib¥theme¥DefineObjectsTag.java
 - DefineObjectsTagクラスでは、JSPのpageContextにログインユーザのインスタンスを「user」としてsetAttributeする
 - `pageContext.setAttribute("user", themeDisplay.getUser());`

PRACTICE

ポートレットをカスタマイズしてみよう！①-1

前提条件

- Liferay本体のソースはダウンロード済み

操作内容

- liferay-theme.tldのソースを確認
- defineObjectsで呼び出されるDefineObjectsTag.javaのソースを確認

確認事項

- pageContextにログインユーザを「user」としてsetAttributeしていること
 - `pageContext.setAttribute("user", themeDisplay.getUser());`

PRACTICE

ポートレットをカスタマイズしてみよう！①-2

前提条件

- view.jspは修正済

- <%@ taglib uri="http://liferay.com/tld/theme" prefix="liferay-theme" %>
- <liferay-theme:defineObjects />
- Hello, <%= user.getScreenName() %>!

操作内容

- view.jspのコードを確認する
- デプロイしてポータルで確認する

確認事項

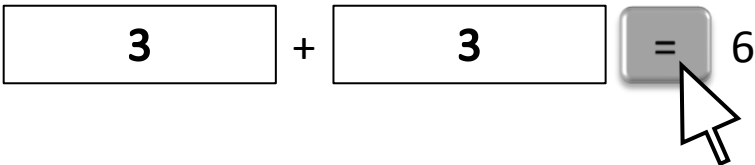
- JSPをカスタマイズすることで、ポートレットの画面をカスタマイズすることが可能であること

カスタマイズしてみる(その②)

□ 要件

□ 足し算したい

■ 例: + =



カスタマイズしてみる(その②)

□ 方法

1. calc-portletの作成
2. Actionクラスの作成
 - calc-portlet/docroot/WEB-INF/src/Calc.javaを作成
3. JSPの修正
 - view.jspの修正
4. 設定ファイルの修正
 - portlet.xmlを修正

カスタマイズしてみる(その②)

□ Tips: Actionクラスの利用

□ Actionクラス

■ doViewメソッド

- ポートレットを表示する際に実行される

■ processActionメソッド

- JSPのFormからActionとして呼ばれた際に実行される

□ portlet.xmlの変更点

■ <portlet-class>でActionクラスを指定する

- JSPからは<portlet:actionURL />で取得できる

カスタマイズしてみる(その②)

- Tips: メソッド間のデータ受け渡し
 - ActionResponseとRenderRequestを利用する
 - `void ActionResponse.setRenderParameter(String key, String value)`
 - `String RenderRequest.getParameter(String key)`

PRACTICE

ポートレットをカスタマイズしてみよう！②

前提条件

- calc-portletは作成済
- Actionクラスは作成済
- view.jsp、portlet.xmlは修正済

操作内容

- Actionクラス、view.jsp、portlet.xmlのコードを読む
- デプロイしてポータルで確認する

確認事項

- ActionクラスのdoViewメソッドとprocessActionメソッドの実装方法
- Actionクラスを利用したポートレットが動作すること

おわりに

ending

以上、お疲れ様でした！

- 本セミナーで学んだこと
 - Liferay Portalの概要と構築方法
 - Liferay Portletの基本
 - 作成する方法
 - ファイル構成
 - ファイルの内容
 - Liferay Portletのカスタマイズ
 - JSPのみのカスタマイズ
 - Actionクラスを使うカスタマイズ

- これであなたもLiferay Portlet開発者です！

LIFERAY PORTLETを読み解く

ご清聴ありがとうございました！

日本Liferayユーザグループ

林田 敦

参考資料

- LGPLライセンス概要(引用:wikipedia)
 - 社内など私的組織内部や個人で(private)利用するにあたってのソースコード改変、再コンパイルには制限がない。
 - LGPLで頒布されたプログラムを再頒布する際にはソースコードを公開する必要がある。