

 無料電子ブック

学習

jsf

Free unaffiliated eBook created from
Stack Overflow contributors.

#jsf

.....	1
1: jsf	2
.....	2
.....	2
Examples	2
JSF	2
.....	2
.....	2
2: Ajax	3
Examples	3
.....	3
.....	3
Ajax on javascript	4
.....	5
3: JSF 2Flash	7
.....	7
Examples	7
Flash	7
4: JSF	9
.....	9
Examples	9
.....	9
Bean	10
5: JSF	11
.....	11
Examples	11
.....	11
1	11
6: JSF	13
.....	13
.....	13
.....	13

Examples.....13

.....13

 facelets.SKIP_COMMENTS.....13

..... **15**

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [jsf](#)

It is an unofficial and free jsf ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jsf.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: jsfをいめる

JavaServer FacesJSFは、HTMLフォームベースのWebアプリケーションをするためににされるモデルビュープレゼンターフレームワークです。コンポーネントとレンダリングキットをして、ステートフルHTMLビューをFaceletsまたはJSPタグをしてし、バックングビーンをしてモデルデータとアプリケーションロジックにすることができます。

バージョン

バージョン	リリースノート	
2.3	Arjan Tijmsによる	2017-03-28
2.2		2013-05-21
2.1		2010-11-22
2.0		2009-07-01
1.2		2006-05-11
1.1		2004-05-27
1.0		2004-03-11

Examples

JSFのインストール

コンテンツはgood 'ol [JSF wikiページ](#)にされました

こんにちは

コンテンツはgood 'ol [JSF wikiページ](#)にされました

コンテンツはgood 'ol [JSF wikiページ](#)にされました

オンラインでjsfをいめるをむ <https://riptutorial.com/ja/jsf/topic/916/jsfをいめる>

2: Ajax

Examples

ビューをにする

ajaxリクエストをし、ビューののみをします。

Bean.java

```
@ManagedBean
@ViewScoped
public class Bean {

    public Date getCurrentDate() {
        return new Date();
    }

}
```

sample.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
<h:head />
<h:body>
    <h:form>
        <h:commandButton value="Execute ajax">
            <f:ajax render="output" />
        </h:commandButton>
        <p>
            <h:outputText id="output" value="Ajax date: #{bean.currentDate}" />
        </p>
        <p>
            <h:outputText id="output2" value="Non-Ajax date: #{bean.currentDate}" />
        </p>
    </h:form>
</h:body>
</html>
```

フォームパーツをしてリクエストをく

フォームののみをするようにします。 `text1` はリスナーのとしてされますが、 `text2` はされませ `text2` 。

Bean.java

```
@ManagedBean
@ViewScoped
```

```

public class Bean {

    private String text1;

    private String text2;

    public String getText1() {
        return text1;
    }

    public void setText1(String text1) {
        this.text1 = text1;
    }

    public String getText2() {
        return text2;
    }

    public void setText2(String text2) {
        this.text2 = text2;
    }

    public void listener() {
        System.out.println("values: " + text1 + " " + text2);
    }

}

```

sample.xhtml

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
<h:head />
<h:body>
    <h:form>
        <h:inputText id="my_input" value="#{bean.text1}" />
        <h:inputText value="#{bean.text2}" />
        <h:commandButton value="Execute ajax">
            <f:ajax execute="@this my_input" listener="#{bean.listener}" />
        </h:commandButton>
    </h:form>
</h:body>
</html>

```

Ajax on javascript イベント

ユーザーがフィールドにするたびにがされます。

Bean.java

```

@ManagedBean
@ViewScoped
public class Bean {

```

```
public Date getCurrentDate() {
    return new Date();
}

}
```

sample.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
<h:head />
<h:body>
  <h:form>
    <h:inputText>
      <f:ajax event="keyup" render="output" />
    </h:inputText>
    <p>
      <h:outputText id="output" value="Ajax date: #{bean.currentDate}" />
    </p>
  </h:form>
</h:body>
</html>
```

デレイ

ミリでされたでリクエストをします。つまり、のリクエストがキューにられたにのリクエストがした、のリクエストはスキップされます。これはJSF 2.2からです

Bean.java

```
@ManagedBean
@ViewScoped
public class Bean {

    public Date getCurrentDate() {
        return new Date();
    }

}
```

sample.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
<h:head />
<h:body>
  <h:form>
    <h:inputText>
      <f:ajax event="keyup" render="output" delay="2000" />
    </h:inputText>
    <p>
```

```
        <h:outputText id="output" value="Ajax date: #{bean.currentDate}" />
    </p>
</h:form>
</h:body>
</html>
```

オンラインでAjaxをむ <https://riptutorial.com/ja/jsf/topic/4916/ajax>

3: JSF 2のFlashスコープ

FlashのはRuby on Railsからされ、ライフサイクルでされたユーザービューでオブジェクトをすをします。 Railsのとの、フラッシュの1つのは、じユーザーセッションでされたのビューにされ、そのクリアされます。「のビュー」はのビューとじビューIDをつがあることにすることができ

JSFのでは、`<redirect />`をむ`<navigation-case>`でも、Flashのながされることをするがあります。は、じセッションであるGETのでも、フラッシュのながされることをするがあります。これにより、Facesアプリケーションは「Post / Redirect / Get」デザインパターンをにすることができます。

Examples

Flashスコープののデモンストレーション

Bean1

```
@ManagedBean
@ViewScoped
public class Bean1 implements Serializable {

    /**
     * Just takes the given param, sets it into flash context and redirects to
     * page2
     *
     * @param inputValue
     * @return
     */
    public String goPage2(String inputValue) {
        FacesContext.getCurrentInstance().getExternalContext().getFlash()
            .put("param", inputValue);
        return "page2?faces-redirect=true";
    }
}
```

page1.xhtml

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:c="http://java.sun.com/jsp/jstl/core">
<h:head />
<h:body>

    <!-- Sets the first flash param at the action method -->
    <h:form>
        <h:inputText value="#{inputValue}" />
        <h:commandButton action="#{bean1.goPage2(inputValue)}"
            value="Go Page 2" />
    </h:form>
</h:body>
</html>
```

```

</h:form>

<!-- Sets the second flash param -->
<c:set target="#{flash}" property="param2" value="Myparam2" />

<!-- Tries to retrieve both of the params.
Note none of them is displayed at the first page hit.
If page refreshed, the second param which has been already set, will be displayed -->
<p>Param1: #{flash['param']}</p>
<p>Param2: #{flash['param2']}</p>
</h:body>
</html>

```

Bean2

```

@ManagedBean
@ViewScoped
public class Bean2 implements Serializable {

    public String getParam() {
        /**
         * Takes the parameter from the flash context
         */
        return (String) FacesContext.getCurrentInstance().getExternalContext()
            .getFlash().get("param");
    }
}

```

page2.xhtml

```

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:f="http://java.sun.com/jsf/core">
<h:head />
<!-- This page just displays the received params -->
<h:body>
    <!-- Different ways to retrieve the params from the flash scope -->
    <p>Param1: #{bean2.param}</p>
    <p>Param1: #{flash.param}</p>
    <p>Param1: #{flash['param']}</p>
    <p>Param2: #{flash['param2']}</p>

    <!-- Keep the first param for next redirection -->
    #{flash.keep.param}

    <!-- Return to page1 and see how the first param is retained -->
    <h:button outcome="page1?faces-redirect=true" value="return to 1" />
</h:body>
</html>

```

オンラインでJSF 2のFlashスコープをむ <https://riptutorial.com/ja/jsf/topic/3906/jsf-2のflashスコープ>

4: JSF アノテーション

はこのウェブサイトからくのをる

- <http://www.jmdoudoux.fr/java/dej/chap-annotations.html>
- <http://docs.oracle.com/javaee/6/tutorial/doc/girch.html>

Examples

アノテーションの

なぜアノテーション

に、アノテーションをしてをにし、コードをよりかつきれいにする。

とはですか

Java 5アノテーションは、なでメタデータのをします。Javaにけられたこのメタデータは、コンパイルまたはでされるがあります。

のをにするようにJavaがされました。

- のとをにするのがJavaにされました。
- バイトコードはアノテーションのをにするようにされている。

はどこでできますか

はのものでできます。

パッケージ、クラス、インタフェース、コンストラクタ、メソッド、フィールド、パラメータ、またはそのものです。

アノテーションのカテゴリ

アノテーションには3つのカテゴリがあります。

- マーカー これらにはがありません

例えば `@Deprecated`、`@Override` ...

- の これらにはが1つしかありません

たとえば、`@MyAnnotation ("test")`

- なアノテーション これらのアノテーションにはがあります

`@MyAnnotation (arg1 = "test 3", arg2 = "test 2", arg3 = "test3")`

にたように、のアノテーションをできます

マネージド **Bean** スコープのアノテーション

マネージド **Bean** の

Bean をするには、アノテーション `@ManagedBean` が `@ManagedBean`

えば

```
@ManagedBean
public class Example {}
```

パッケージがです

```
import javax.faces.bean.ManagedBean;
```

マネージド **Bean** スコープ

アノテーションをして、**Bean** がされるスコープをします。

`@NoneScoped`, `@RequestScoped`, `@ViewScoped`, `@SessionScoped`, `@ApplicationScoped` マネージド **Bean** のスコープがあります。

- アプリケーション `@ApplicationScoped` アプリケーションスコープは、すべてのユーザーが Web アプリケーションとやりとりしてもされます。
- セッション `@SessionScoped` セッションスコープは、Web アプリケーションの HTTP リクエストにわたってします。
- ビュー `@ViewScoped` ビュースコープは、ユーザーが Web アプリケーションのページビューとするはされます。
- リクエスト `@RequestScoped` リクエストスコープは、Web アプリケーションの HTTP リクエストにします。
- なし `@NoneScoped` アプリケーションにしてスコープがされていないことをします。
- カスタム `@CustomScoped` ユーザーのスコープです。そのは、`java.util.Map` としてするがあります。カスタムスコープはあまりされません。

オンラインで JSF アノテーションをむ <https://riptutorial.com/ja/jsf/topic/7021/jsfアノテーション>

5: JSFテンプレート

JSFには、**facelets**タグと呼ばれるWebアプリケーションのレイアウトをするためのタグがされています。これらのタグは、のページのを1かですするためのをします。

```
xmlns:h="http://xmlns.jcp.org/jsf/html"  
xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
```

Examples

テンプレートをする

1つのアプリケーションのテンプレートをする

/WEB-INFフォルダのにtemplate.xhtmlというのファイルをしします。テンプレートファイルにはフレームワークのみがアクセスできるようにしします。

/WEB-INF/template.xhtml

```
<!DOCTYPE html>  
<html lang="en"  
  xmlns="http://www.w3.org/1999/xhtml"  
  xmlns:h="http://xmlns.jcp.org/jsf/html"  
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets">  
  
  <h:head>  
    <title><ui:define name="title">Default title</ui:define></title>  
  </h:head>  
  
  <h:body>  
    <!-- Some styles that we might include for our whole application -->  
    <h:outputStylesheet name="css/template.css" />  
  
    <!-- Shared content for the application, e.g. a header, this can be an include -->  
    <div>  
      Application Header  
    </div>  
  
    <!-- The content we want to define in our template client -->  
    <div>  
      <ui:insert name="content" />  
    </div>  
  </h:body>  
</html>
```

このファイルは、アプリケーションのテンプレートとしてしします。ここで、viewディレクトリにのビューをしします。

/home.xhtml

```
<ui:composition template="/WEB-INF/template.xhtml"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets">

  <ui:define name="title">Home</ui:define>

  <ui:define name="content">
    Welcome to the application!
  </ui:define>
</ui:composition>
```

このクライアント・ビューの`template`を`template`。これは、JSFにテンプレートをするようにします。に、`<ui:define>`をして、テンプレートでされたにされるのコンテンツをします。クライアントで`/home.xhtml`にアクセスすると、がレンダリングされます。

オンラインでJSFテンプレートをむ <https://riptutorial.com/ja/jsf/topic/5033/jsfテンプレート>

6: JSFのコメント

き

マークアップとしてのJSFは、コードのコメントをサポートしていますが、のようなHTMLコメントコードをすとしてください。<- のボタンにコメントします。<hcommandButton value = "push" onclick = "アラート 'Hello';" />->それはもコメントしていません。これは、JSFがタグ<!--と-->にコメントされていても、このコードをデフォルトとしてするためです。JSFコードをコメントするには2つのソリューションがあります

- <ui:remove>コメントしたいJSFコード</ ui:remove>

は、Oracleのマニュアルをしてください。

- oracle.comの<ui:remove>
- facelets.java.netの SKIP_COMMENTS

Examples

タグをする

コメントするJSFコードのに<ui:remove>タグと</ui:remove>タグをするがあります。

```
<ui:remove>
  <h:outputLabel value="Yeah, I'm really commented" />
</ui:remove>
```

もちろん、このxmlnsをヘッダーのhtmlタグにするがあります。こののなをしてください

```
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets">

  <ui:remove>
    <h:outputLabel value="Yeah, I'm really commented" />
  </ui:remove>

</html>
```

faceletsをする.SKIP_COMMENTS

web.xmlにのようなタグをするがあります。

```
<context-param>
  <param-name>facelets.SKIP_COMMENTS</param-name>
  <param-value>>true</param-value>
```

```
</context-param>
```

これでのHTMLコメントタグ<!--および-->をできます<!--

```
<!--  
  <h:outputLabel value="Yeah, I'm really commented" />  
-->
```

web.xmlでされたfacelets.SKIP_COMMENTSのなはのとおりです

```
<html xmlns="http://www.w3.org/1999/xhtml"  
  xmlns:h="http://java.sun.com/jsf/html"  
  xmlns:ui="http://java.sun.com/jsf/facelets">  
  
  <!--  
    <h:outputLabel value="Yeah, I'm really commented" />  
  -->  
  
</html>
```

オンラインでJSFのコメントをむ <https://riptutorial.com/ja/jsf/topic/8164/jsfのコメント>

クレジット

S. No		Contributors
1	jsfをいめる	BalusC , Community , Emil Sierżęga , Pixelstix
2	Ajax	Xtreme Biker
3	JSF 2のFlashスコープ	Xtreme Biker
4	JSFアノテーション	Right leg , YCF_L
5	JSFテンプレート	BalusC , DimaSan , Emil Sierżęga , Xtreme Biker
6	JSFのコメント	albertoiNET