| ただ一疋の青 | い猫のかげ |
|--|---|
| ーeclipseによるJavat | ナーブレットの作成- |
| 岐阜経済大学 経営学部 経 | 营情報学科 井戸 伸彦 |
| 来歴: 0.0版 2004 0.1版 2005年2月14日 一部を 1.1版 2008年1月30日 一部 スライド(| ₹4月18日 eclilpse3.01,lomboz3.01に修正 Sをeclilpse3.3,tomcat6に修正 ひ結成 |
| スノート、 はじめに (1)環境設定 (1.1)なにがやりたいのか? (1.2)プラグイン | (4) Tomcatの起動とJSPの閲覧 (4.1) Tomcatの起動 (4.2) JSPの閲覧 (5) Javaサーブレット |
| (1.3)環境設定 (1.4)サーバ設定 (2)プロジェクト (2.1)~(2.5)動的Webプロジェクトの作成 | (5.1)~(5.6)Javaサーブレットの作成 (5.7)サーブレットの閲覧 (6)その他 (6.1) web.xmlファイルの中身 |
| (2.6),(2.7)プロジェクトのサーバへの追加 (3) J S P の作成 (3.1) ~ (3.5) JSPの作成 | (6.2) web.xmlファイルの編集 (6.3)コンソールでのエラー表示 (6.4)デバッグモード、ログファイル |

はじめに

- ■本スライドは、lombozがプラグインされたeclipse、および、Tom catを用いてJavaサーブレットを作成する手順について記しま す。
- ■このスライドでは、手順のみを記していき、編集したファイルの 意味については次の2つのスライドにて勉強することにします。
 - 「ウィークエンド・シャッフル JSP入門 」
 - •「されど我らが日々 サーブレット入門 」
- ■Jdk1.6 / Tomcat6.0 / eclipse3.3が、次の文書のとおりにインス トール済みであることを前提としています。
 - 「jd k 1 . 6 のインストール(Windows)」
 - 「Tomcat6のインストール(Windows)」
 - 「eclipse3.3のインストール(Windows)」

■岐阜経済大学では、全WindowsPCに、eclipse3.3、Tomcat6は インストールされています。本スライドでは、これらでの環境で の操作を記します。



(1.2) プラグイン

- ■学内でLinun PC、Windows PCにインストールした eclipseには、WTPがプラグインされています。
- ■プラグインとは、eclipseに機能を追加することであると 考えておいてください。

■WTPをプラグインすることにより、eclipseでは、スライ ド(1.0)に示したようなJavaサーブレット(JSP)の作成 を容易に行う環境を提供することが出来ます。WTPは J2EEプロジェクト作成支援プラグインとして、標準の位 置づけにあります。



(1.3)環境設定

■WTPを利用する際には、環境設定が必要となります。 ■環境設定は、スライド「eclipseを用いたJavaアプリ ケーションの作成」で説明した、データの置き場所ごと、 すなわち、各ユーザごとに必要となります。これを(1.4) で説明します。



(1.4) サーバ設定(1/4)

新規(N)

閉じる(0)

ファイルを開く()...

- ■[ファイル] -[新規]
 - [その他]をクリック(
- ■「新規:ウィザードを選択」に τ.

🛢 Java EE – Eclipse ブラットフォーム ファイル(E) 🔎 編集(E) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) Alt+Shift+N 🕋 プロジェクト 🌱 その他(②)... Ctrl+W 🛢 新規 ウィサードを選択 新規サーバーの定義 ウィザード(W): フィルター入力 📺 ''' 🔽 📿 🖄 🗖 <u> 〔…)</u> サーバーン 1部1サーバー ┍╺╪┝╧┑╭╪╪╴╴└┉ 44 \ . - Y T - - KAT $\dot{\chi} \land > (\underline{N})$ < 戻る(B) \bigcirc

(1.5)サーバ設定(2/4)

■「新規サーバー:新規サー バーの定義」にて、

[Apache]

-[Tomcat v6.0 Server]を クリック()。 ■[次へ]をクリック()。

| 😂 新規サーバー 📃 🗖 🔀 | | |
|---|--|--|
| 新規サーパーの定義 作成するサーバーのタイプを 選択します | | |
| サーバーのホスト名(<u>H</u>): localhost 追加サー | | |
| サーバーのタイプを選択(<u>S</u>): フィルター入力 | | |
| 🕞 🦻 Apache | | |
| 📕 Tomcat v3.2 Server | | |
| Tomcat v4.0 Server | | |
| Tomcat v4.1 Server | | |
| Tomcat v5.0 Server | | |
| lomcat vb.5 Server | | |
| | | |
| Publishes and runs J2EE and Java EE v configurations to a local Tomcat server. | | |
| ⑦ <戻る團 (次へ>№) | | |
| | | |



| 🖨 新規サーバー 🔲 🗖 🔀 | |
|--|--|
| Tomcat サーパー インストール・ディレクトリーを指定してください | フォルダの参照 Tomcat インストール・ディレクドリーを選択してくださ 3 |
| 名前(M): Apache Tomcat v6.0 Tomcat インストール・ディレクトリー(D): C:¥Program Files¥Apache Software Foundation¥Tomcat 6.0 参照(B) くる | G V1 1/C1 -> G V1 1/C1 -> G V1 1/C1 -> G V1 1/C1 -> G V1 1/C1 V2 V2) G □ -bu Frozeran Files G □ -bu Frozeran Files G □ Adobe G □ Adobe G □ Adobe G □ Accele Software Foundation G □ Commun Files G □ Commun Fil |
| JRE(」): ワークベンチ・デフォルト JRE ✓ インストール済みの JREQ | 2 7#ルダ: Tomcat 60 新しい7#ルダの作成(型) OK キャンセル |
| | |

(1.7)サーバ設定(4/4)

■「新規サーバー:プロ ジェクトの追加および 除去」にて、「終了」を クリック(■プロジェクトエクスプ ローラー中に、サー バーが表示()され ることを確認。



<u>(2)プロジェクト</u>

■プロジェクトの作成は、スライド「eclipseによる」avaア プリケーションの作成」でも行いました。その時作成したのは、"Javaプロジェクト"でした。

■今回は、"動的Webプロジェクト"を作成します。これは、スライド(1.1)のようなことが行えるプロジェクトであると理解しておいてください。

(2.1)動的Webプロジェクトの作成(1/5)



💼 👝 サンプル

< 戻る(B)

(?)

/次へ >(ℕ)

3

(2.2)**動的W**ebプロジェクトの作成(2/5)

■「新規動的Webプロジェ クト」にて、[プロジェクト 名]に適当な名前(右図 では"myWeb")と入力 ()。

ここでのプロジェクト名 は、後で作成したページ にアクセスする際のUR Lの一部となる。スライド (4.2)参照。

■[次へ]をクリック()。



(2.3)**動的Webプロジェクトの作成**(3/5)

■「新規動的Webプロ ジェクト:プロジェクト・ ファセット」にて、次の 2つがチェックされて いることを確認()。

- Java
- 動的Webモジュール

■[次へ]をクリック()。

| ● 新規動的 Web プロジェクト | | |
|--|--------------|---------------|
| プロジェクト・ファセット このプロジェクトで有効にするファセットを選択してください。 | | |
| 構成: Apache Tomcat v6.0 デフォル 🔽 (保管(S) | 」 『削 | 除(<u>D</u>) |
| プロジェクト・ファセット | バージ | ∋) |
| | 60 | _ |
| → Java 永続化 | 1.0 | Ť |
| JavaServer Faces | 1.1 | - |
| | 1.2.3 25 | - |
| (< ランタ (< ランタ | といっ いてムを表 | 示® |
| ⑦ 〈戻る(B) 次へ >(N) 、2(87(F) |) [*1 | ンセル |

(2.4)**動的Webプロジェクトの作成**(4/5)

■「新規動的Webプロジェク 🖶 新規動的 Web フロジェクト ト:Webモジュール」にて、次 Web モジュール の3つの入力を確認。 Web モジュール設定を構成します。 • コンテキストルート(コンテキスト・ルート ♦myWeb ww.Web スライド(2.2)で入力した名前 コンテンツ・ディレクトリー: 2 コンテンツディレクトリー(WebContent Java ソース・ディレクトリー: 3 ♦WebCOntent src • Javaソースディレクトリー(▼配備記述子の生成 ♦ SrC ■[終了]をクリック()。 次へ >(N) < 戻る(<u>B</u>) \bigcirc .終了<u>(F)</u> (2.5)動的Webプロジェクトの作成(5/5)
■「プロジェクトエクスプローラー」に、作成したプロジェクトが表示されるのを確認。

| 😂 Java EE - Eclipse ブラットフォーム |
|--|
| ファイル(E) 編集(E) ナビゲート(<u>N</u>) 検索(<u>A</u>) ブ |
| ! 📬 = 🔚 🖹 🕸 = 🜔 = 🌯 = ! 😤 🛛 |
| 🔁 プロジェクト・エクスプローラー 🛛 🖓 🦳 🗖 |
| 🖻 🔄 🛜 🏹 |
| 📮 🞥 myWeb |
| 👜 😕 Java リソース: src |
| 🗄 🗁 build |
| 🖃 🥟 WebContent |
| 🖮 🥭 META-INF |
| 🖪 👝 WEB-INF |
| |

(2.6)プロジェクトのサーバへの追加(1/2)

■ワークスペース下側の「サーバービュー」のタグをク リック()。

■[localhostのTomcatv6.0サーバ]の欄を右クリック() して、[プロジェクトの追加および削除]をクリック()。



(2.7) プロジェクトのサーバへの追加(2/2)

- ■「プロジェクトの追加および削除」にて、
 - [使用可能なプロジェクト]中の"myWeb"を選択(
 - ●[追加]をクリック()。「構成プロジェクト」を確認()。
 - •[終了]をクリック()。



(3.1) J S P の作成(1/4)

■"Hello!"と表示する JSPを作成します。 ■「パッケージ・エク スプローラー」中の "WebContent" (を右クリックし、[新 規]-[JSP]をクリック



(3.2)JSPの作成(2/4)

■「新規JavaServer Page」に て、[ファイル名]の欄にファイ ル名を入力(、右図では "hello.jsp")。

ここでのJSPのファイル名は、後で作成したページにア クセスする際のURLの一部 となる。スライド(4.2)参照。 ■「次へ]をクリック()。 ●新規 JavaServer Page JavaServer ページ 新規 JavaServer Page を作成します。 親フォルダーを入力または選択(E): mvWeb/WebContent 🟠 🗇 🔿 🖃 🔁 myWeb 🕞 .settings 🖻 🗁 build 🥞 src 🗄 🥟 WebContent 💼 📂 サーバー ファイル名(M): hello.jsp 拡張 >>(<u>A</u>))次へ >(<u>N</u>) 、 終了(F) (?) く 戻る(B)

(3.3) J S P の作成(3/4)

■「新規JavaServer Page: JSPテンプレートの選択」にて、 [終了]をクリック()。



(3.4) JSPの作成(4/4) ■エディターにて、何か適当なメッセージ(、下図では "ようこそ from hello.jsp")を編集。

■保管ボタンをクリック()。

| 🖨 Java EE – myWeb/We | ebContent/hello.jsp - Eclipse ブラットフォ |
|--|---|
| ファイル(E) 編集(E) リファク! | タリング(C) ソース(S) ナビゲート(N) 検索(A) ブ |
| | > • • • • • • • • • • • • • • • • • • • |
| L 70919 2 2 - D | 📄 hello.jsp 🛛 📃 🗖 |
| E S S ▼ myWeb myWeb Dava リソース: sro Duild WebContent Methojsp Prob META-INF Duild META-INF Duild Prob META-INF Duild Prob META-INF Duild Prob META-INF Duild Duild Prob META-INF Duild D | <pre><%@ page language="java" pageEncoding="windows <!DOCTYPE html PUBLIC "-/ <html> <head> <meta http-equiv="Content <title>Insert title here< </title></head> <body> </body> </html></pre> |

(4)Tomcatの起動とJSPの閲覧

- ■スライド(1.1)に記したように、作成したJSPはTomcat 上で動作します。したがって、次の手順で作成したJS Pを閲覧します。
 - ●Tomcatの起動(
 - •WebブラウザからのJSPの閲覧(



(4.1)Tomcatの起動

■「サーバービュー」のタグをクリック()。 ■[localhostのTomcatv6.0サーバ]の欄を選択()。 ■右側の、[サーバーを始動]ボタンをクリック()。



(4.2)JSPの閲覧

■Webブラウザを立ち上げ、次のURLにアクセス。

• http://localhost:8080/myWeb/hello.jsp





| 🕘 Insert | title here – Microsoft Internet Explorer 📃 | |
|------------------|--|-----|
| ファイル(E) | 編集(E) 表示(V) お気に入り(A) ツール(T) へ [≫] | 2 |
| アドレス(<u>D</u>) | 🔄 http://localhost:8080/myWeb/hello.jsp 🔽 🛃 移動 | リンク |
| ようこそ | t, from hello.jsp | < > |

JSPを変更したり新規に作っても、一度立ち上げた Tomcatを、立ち上げなおす必要はありません。

(5.1) Java**サーブレットの作成**(1/6)

- ■「プロジェクトエクスプローラー」にて、
- ・作成した[myWeb]中の[Javaリソース:src]を右クリック
- ・[新規]-[その他]をクリック()。





(5.3) Javaサーブレットの作成(3/6)

■「サーブレット作成」に て、[クラス名]の欄にク ラス名を入力(、右図 では"HelloServlet")。 ■[次へ]をクリック(パッケージの利用方 法を知っている方は、 パッケージを指定しても ОΚ.

| 🛢 サーブレット | 作成 | | |
|---------------------------------------|--------------------------------|----|--|
| サ ーブレット 作 クラス・ファイルのす | # 成 范先をを指定します。 | S | |
| プロジェクト: フォルダー: | myWeb 🖍 | 参照 | |
| Java パッケージ: クラス名: | HelloServlet | 参照 | |
| スーパークラス: | javax.servlet.http.HttpServlet | 参照 | |
| □ 既存のサーブレットの使用 カラフィン・ HalleSemulat | | | |
| ⑦ 〈戻る(B) 次へ >(M) 終了(E) キャン | | | |



(5.5)Java**サーブレットの作成(5/6)**

- ■「サーブレット作成」 て、
- ·[doPost]のチェックを 外す()。
- ・[終了]をクリック()。

| 🛢 サーブレット1 | 作成 | |
|--|---|----------------|
| サーブレット 作 修飾子、実装す? ・スタブを指定して | ・ 成 るインタフェース、および生成するメソッド ください。 | S |
| 修飾子: | ✓ Public ■要約 ■最終 | |
| インターフェース | javax.servlet.Servlet | 追加 |
| 作成するメソッド・ | スタブの選択 | |
| マスーパーク | ラスからのコンストラクター | |
| ☑ 継承され: | を抽象メソッド | |
| | toString getServ | letInfo |
| | /doPutdoDelet | e |
| | | |
| Ŭ | | |
| | | ب ا |
| | | |

(5.6) Javaサーブレットの作成(6/6)

■エディターにて、"doGet"のメソッド中()に、次の文 を追加。

•response.getWriter().println("hello, from
HelloSevlet.");

■保管ボタンをクリック。



protected void doGet(HttpServletRequest request, HttpServletRe:
 // TODO Auto-generated method stub
 response.getWriter().println("hello, from HelloSevlet.");
}



Javaサーブレットを変更したり新規に作ると、Tomcat を立ち上げなおす必要がありますが、eclipseではこれ を自動的に行ってくれます。

(6.1)web.xmlファイルの中身

■前スライド(5.4)での入力により、web.xmlには、次のような設定が自動でなされます。

<servlet>

<description></description>

<display-name>HelloServlet</display-name>

<servlet-name>HelloServlet</servlet-name>

<servlet-class>HelloServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HelloServlet</servlet-name>

<url-pattern>/helloPage</url-pattern>

</servlet-mapping>

(6.2)web.xmlファイルの編集

- ■次のように"web.xml"を XMLエディターで開くと、右 図のような形で編集を行う ことができます。
- 上記のようにXMLエディタでファイルを開きたいとき: ・ファイルを右クリック()して、[アプリケーションから開く]-[XML Editor]をクリック()。





(6.3) コンソールでのエラー表示

■誤りのあるJSPやサーブレットが実行された場合、「コンソール・ビュー」でこれに関する情報が表示されます。



(6.4)デバッグモード、ログファイル

■下図のように[デバッグモードで始動]をクリック()すると、Javaアプリケーションと同様な方法で、JSPやサーブレットにブレークポイントを設定することができま



す。

■Tomcatのログファイルは、次のフォルダにあります。

C:¥Program Files¥Apache Software Foundation¥Tomcat 6.0¥logs



(7.1) My S Q L の利用 (1/3)

■MySQLを利用する際には、JDBCドライバが必要です。 これが次のフォルダにコピー済みであるとします。

- フォルダ C:¥Program Files¥Java¥jdk1.5.0_13¥jre¥lib¥ext
- •ファイル mysql-connector-java-5.1.5-bin.jar

| 🔁 プロジ. | ιクト・エクス 🕱 🦳 🗖 |
|----------------------------|-----------------------|
| | 🖃 🔄 🐩 🎽 |
| ∎ } ⊣ ∎ 1 | bum 新規(<u>N</u>) |
| | プロパティー(B) |

<u>(7.2)MySQLの利用 (2/3)</u>



| JARの選択 | | | ? 🗙 |
|--|--|--|--------------------|
| ファイルの場所型: | 🚞 ext | ~ | |
| していていていていていていた。 最近使ったファイル | dnsns.jar localedata.jar mysql-connector- sunjce_provider.ja sunpkcs11.jar | -java-5.1.5-bin.jar ar | |
| マイ ネットワーク | ファイル名(N): | mysql-connector-java-5.1.5-bin.jar | 開(()) |
| | ファイルの種類(工): | *.jar;*.zip | キャンセル |
| ●フロパティー: album | | | |
| フィルター入力 | Java のビルド・パス | ι. | ⇔ - ⇔ - |
| リソース BeanInfo パス | (伊 ソース(S) (戸 プ ビルド・パス上の JAR: ● ▲ Apache Tom | *ロジェクト(P) ■ ライブラリー(L) �� 順序および) およびクラス・フォルダー(T): ector-java-5.1.5-bin.jar - C:¥Pro ncat v6.0 [Apache Tomcat v6.0] 外部、IA | ビクスボート() の追加(」) |
| Java のビルド・パラ Javadoc ロケーショ JSP フラグメント ▼ | ■ ■ EAR ライブラリー ■ ■ JRE システム・ライブラリー [ire1.5.0_13] 変数の追加(V) | | Di自加(<u>)</u> ▼ |
| 0 | | ОК | キャンセル |