



東京ITスクール

TOKYO IT SCHOOL

Struts の概要と動作

目次

1. はじめに	1
2. Struts について	1
3. MVC モデルと Struts	3
4. Struts の構成要素	4
5. Struts 開発環境	7
6. Struts を動かす	10

1. はじめに

本講座では、J2EE フレームワークの主流の 1 つである Struts を学習していきます。この資料では、Struts の概要と MVC モデルを学び、実際に簡単なアプリケーションを作成して Struts の動作を確認します。

2. Struts について

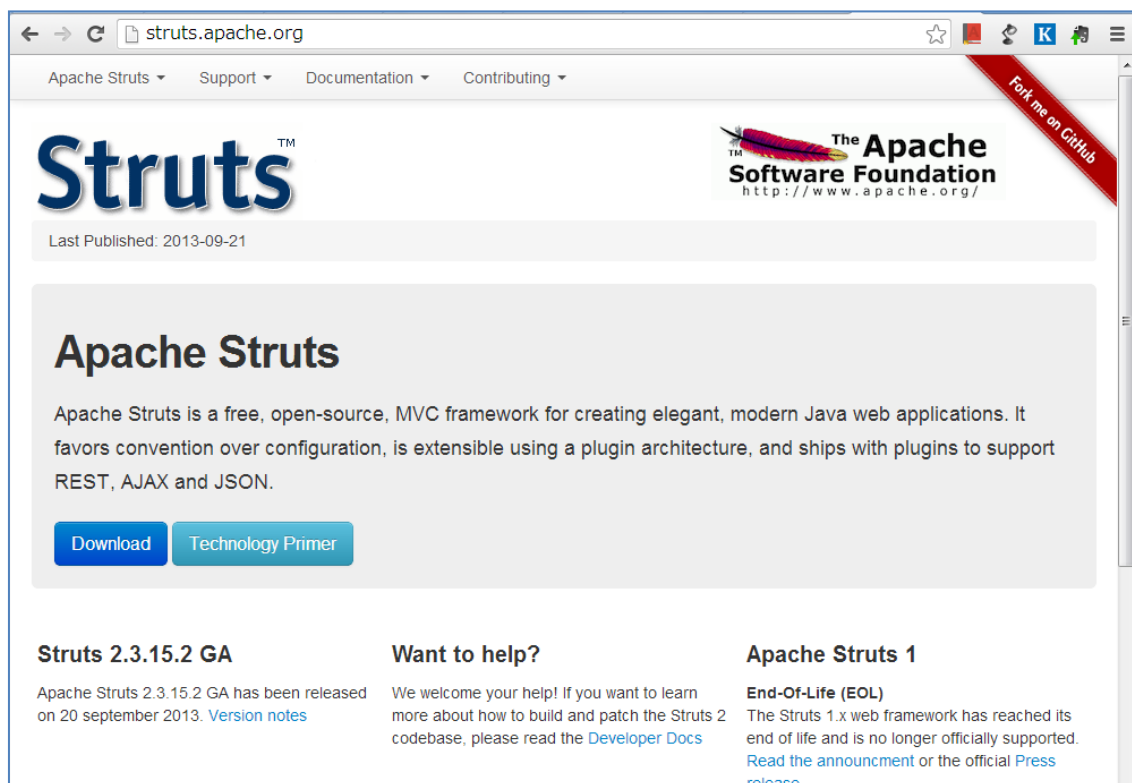
1 Struts とは

Struts はクレイグ・マクラナハン氏によって作成され、2000 年に The Apache Software Foundation の Jakarta Project に寄贈されました。その後同プロジェクトにおいて多くの改良が行われ、2004 年 3 月 18 日に The Apache Software Foundation の TLP(Top Level Project) になりました。そして今では、J2EE で Web アプリケーションを構築する際のフレームワークとして、最も一般的なものの 1 つになっています。

▼The Apache Software Foundation

【URL】 <http://www.apache.org>

The Apache Software Foundation の Struts ページ



2 Web アプリケーションと Struts

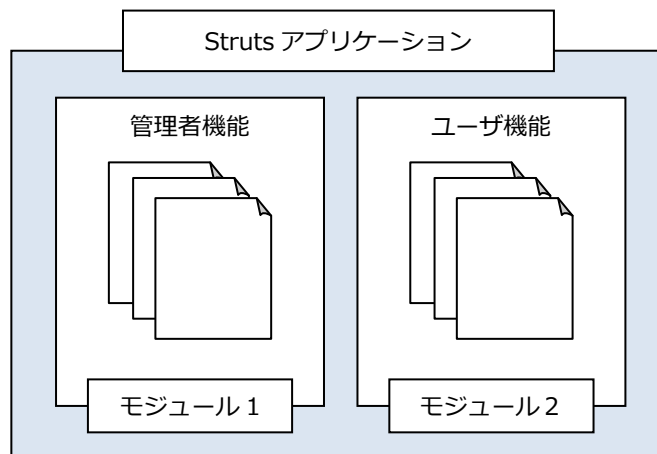
Struts は J2EE (Java2 Enterprise Edition) を用いて Web アプリケーションを作成するためのフレームワークの 1 つでサーブレット、JavaBeans、JSP、メッセージリソースファイルや XML など、多くの標準的技術から成り立っています。

フレームワークとは、どんなアプリケーションにも共通に必要なデータの入出力やエラー処理、画面遷移などの基本的な仕組みを提供するものです。

Struts が提供する機能は以下の通りです。

- ・ リクエスト値の JavaBean プロパティへの自動設定機能
- ・ リクエスト値の検証（入力チェック）機能
- ・ ファイルアップロード機能
- ・ JSP 開発を支援するカスタムタグライブラリ
- ・ レイアウト機能
- ・ ロケールに合わせた国際化機能
- ・ 仮想パスによる画面遷移管理機能

また、Struts のメリットとして、モジュールによるアプリケーションの分割が挙げられます。アプリケーションが肥大してくると設計、実装、保守のすべての面において管理が難しくなりますが、Struts ではモジュールという管理単位を導入することにより、モジュール単位で設計、開発、保守を行うことができます。



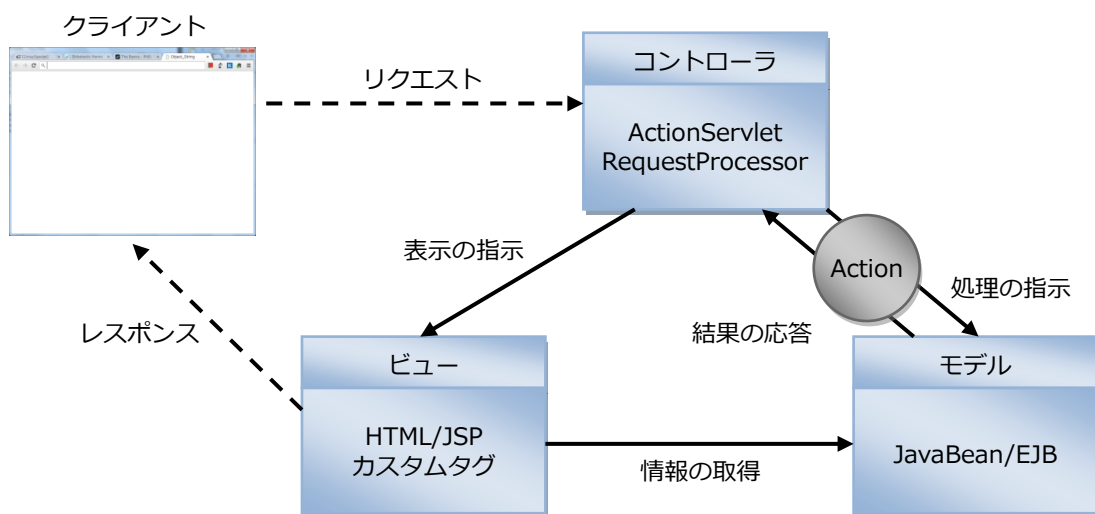
3. MVC モデルと Struts

Struts では MVC モデルを採用しています。MVC モデルとはアプリケーションを次の 3 つに分けて設計、開発していく手法です。

モデル	アプリケーションで取り扱うデータの管理及びデータに対するロジック
ビュー	画面等へのモデルの表示
コントローラ	モデルとビューの制御

Struts は、上記の MVC モデルを基にした以下のような MVC モデルを採用しています。

【Struts における MVC モデル】



Struts を用いたアプリケーションにおいて、モデルは開発者が実装する JavaBean もしくは EJB、ビューは JSP および Struts のカスタムタグ、コントローラは ActionServlet と RequestProcessor、Action となります。

ユーザからのリクエストは、まず ActionServlet に渡されます。ActionServlet はリクエストの URL から呼び出すモジュールを判断し、そのモジュールの RequestProcessor へリクエストを転送します。そして RequestProcessor は、あらかじめ設定された内容に応じて実行する Action を決定し、Action を介してモデルである JavaBean (もしくは EJB) のロジックを実行します。

モデルの処理が終了すると、RequestProcessor は Action を介して結果 (どの JSP ページを表示に使用するか) を取得します。ActionServlet は取得した情報をで、ビュー (JSP) にフォワードを行います。遷移先の JSP では、session や request スコープに登録された JavaBean からデータを取得し、表示を行います。ユーザは最終的に JSP ページによって生成された結果を、レスポンスとして取得します。

4. Struts の構成要素

続いて、Struts のコアクラスと設定ファイルについて解説します。Struts は 25 のパッケージ、300 近くのクラスで構成されています。ここではその中でも特に重要なコアクラスと、設定ファイルを簡単に紹介します。

1 ActionServlet

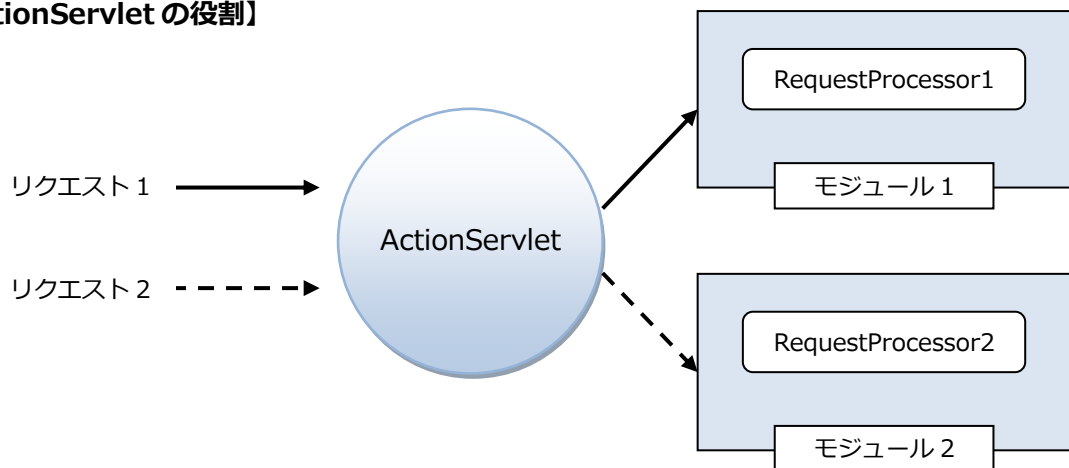
ActionServlet は Struts フレームワーク内で唯一のサーブレットであり、クライアントからのリクエストを最初に受け取り処理します。ActionServlet の主な役割は、受け取ったリクエストに応じた RequestProcessor を呼び出すことです。

ActionServlet はインスタンス化されるときに、init()メソッド内で次のような処理を行います。

- web.xml から初期化パラメータを読み込みます。このとき ActionServlet は、struts-config.xml の場所を取得します。
- struts-config.xml を読み込みます。HTTP リクエストのパスと Action のマッピング、メッセージリソースの読み込み、プラグインのロードなどが行われます。

その後 HTTP リクエストの URL からモジュールを判定して、適切な RequestProcessor の呼び出しを行います。つまり、コントローラとして処理のほとんどは RequestProcessor が担っていることになります。

【ActionServlet の役割】



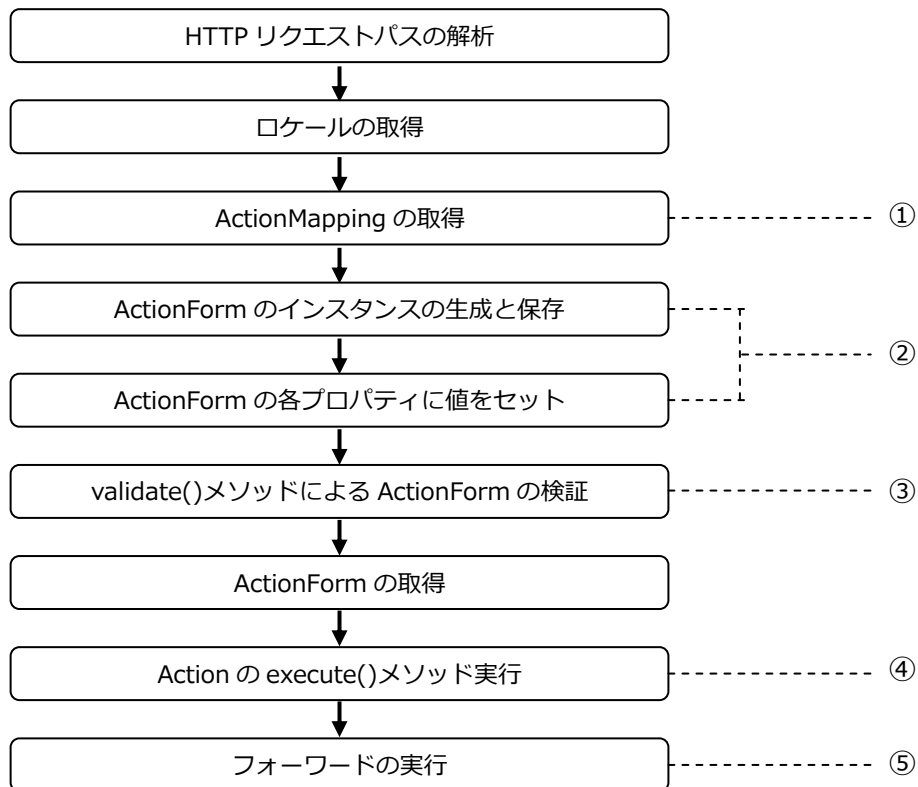
2 RequestProcessor

RequestProcessor は最も重要なクラスの 1 つで、Struts におけるコントローラの中心的な役割を担います。RequestProcessor はモジュールに対して 1 つのインスタンスが作成され、各モジュールに対するリクエストを Struts 設定ファイルの内容に基づいて処理します。

RequestProcessor の主な処理内容は以下の通りです。

- ・ リクエストされた URL に関連づけられた各種情報の取得
- ・ ActionForm の取得とリクエストパラメータの値の格納
- ・ Action の取得と実行
- ・ JSP や HTML へのフォワード

また、RequestProcessor の処理のおおまかな流れは以下の通りです。



① HTTP リクエストの URL に対応した ActionMapping 情報を、struts-config.xml の内容から検索します。ActionMapping とは HTTP リクエストパスと Action、およびそれに関連付けられる ActionForm の関係を定義したものです。

② ActionForm の設定が行われます。①で取得した ActionMapping 情報からインスタンスを生成すべき ActionForm が決定され、インスタンスがスコープに登録された後、HTTP リクエストパラメータをプロパティにセットします。

③ ActionForm にセットされたリクエストパラメータの検証（入力チェック）を行います。

- ④ Action の execute() メソッドが実行され、このメソッドからビジネスロジック層へ処理が引き継がれます。
- ⑤ 他のリソースへ処理が引き渡されます。他のサーブレット、JSP、HTML など、引き渡し先はさまざまです。フォーワードかリダイレクトかについては、struts-config.xml で指定します。

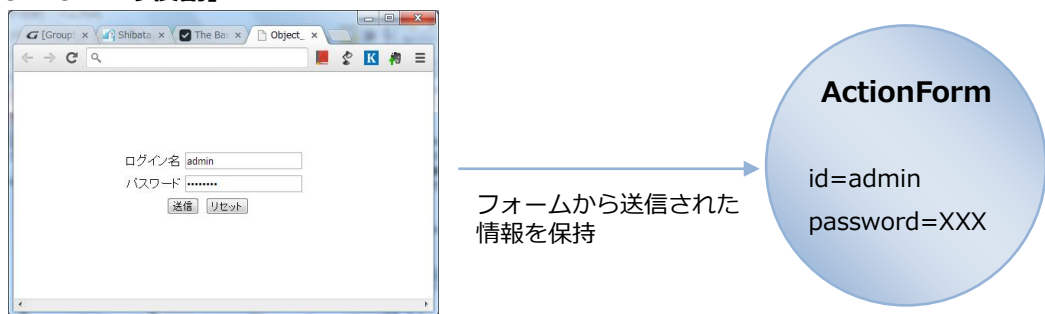
3 Action

Action はリクエストとモデルの橋渡しとしての役割を担います。具体的にはモデルにアクセスし、ビューに渡すデータを準備するというのが主要な役割です。

4 ActionForm

ActionForm はモデルとビューの間に位置し、フォームから送られた値を Action へ転送する役割を担います。ユーザーが入力した値の入力チェックも行います。

【ActionForm の役割】



5 Struts 設定ファイル

Struts の各種情報を定義するファイルで、struts-config.xml という XML ファイルです。基本的にモジュール 1 つに対して、1 つの Struts 設定ファイルを作成します。リクエストされる URL と、Action や ActionForm との関連をファイルに設定します。また画面遷移もこのファイルで管理します。

5. Struts 開発環境

ここでは Struts を動作させるために必要な環境を構築する方法について説明します。本講座では、次のような環境の下で開発を進めます。

OS	Windows7
JDK	JDK1.6
サーブレットコンテナ	Tomcat6.0
フレームワーク	Struts1.3.1

必要なソフトは以下の通りです。

Pleiades All in One Eclipse3.7 Full All in One Edition

オープンソースで提供されている IDE（統合開発環境）です。

プラグイン（拡張ライブラリ）を導入することで、様々な機能を追加できます。

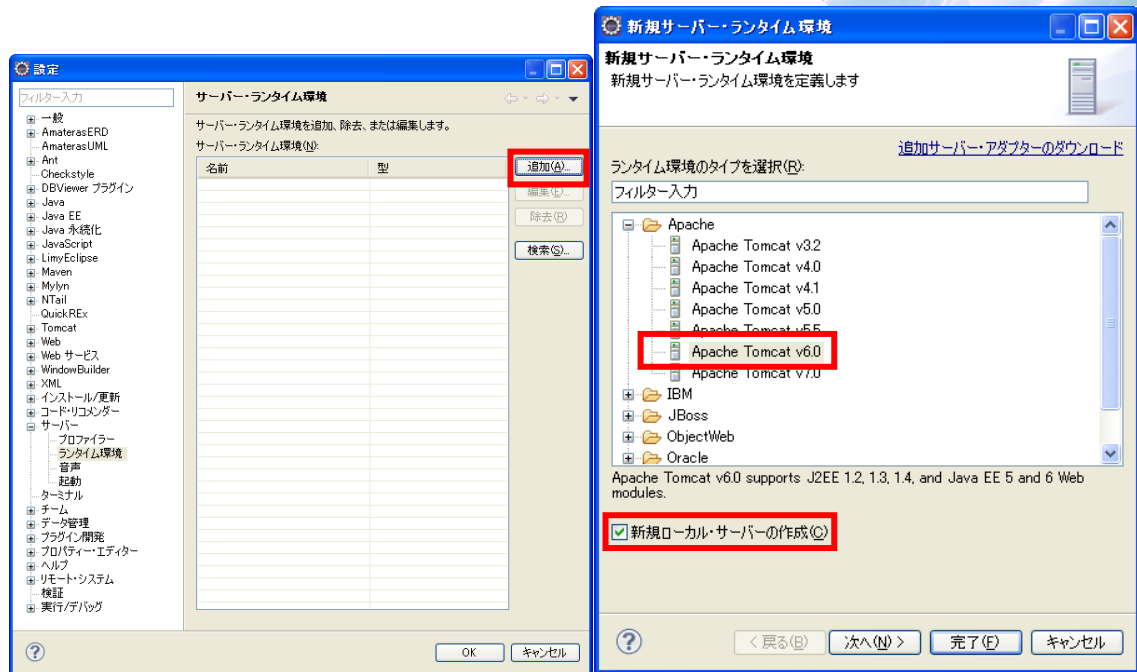
Amateras StrutsIDE プラグイン

Struts プロジェクトの作成機能を Eclipse に追加するためのプラグインです。

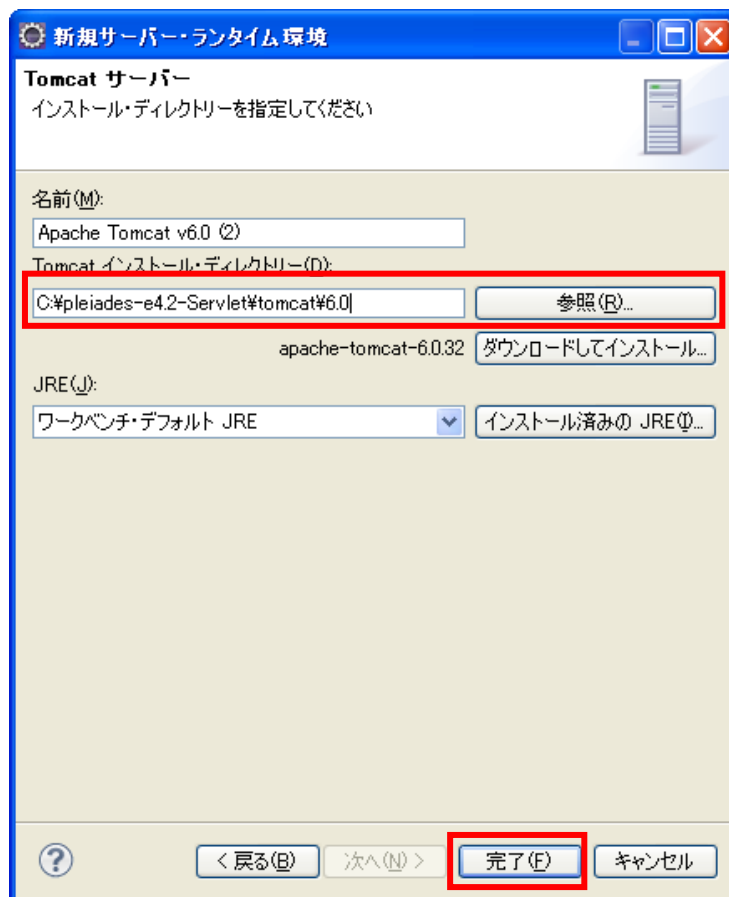
Struts のプロジェクトに必要となる Jar ファイルを自動的に設定してくれます。

1 Eclipse のインストールと設定

- ① Eclipse を以下のサイトからダウンロードして下さい。
<http://mergedoc.sourceforge.jp/>
- ② 本講座では、Eclipse のバージョンは「Pleiades All in One 3.7 Indigo」の Full All in One を使用します。ダウンロードしたファイルが zip の場合は解凍ソフトを使用し、実行ファイルの場合はダブルクリックすることで解凍されます。
- ③ 解凍したフォルダを任意のディレクトリに配備します。
※日本語の名前が付いているフォルダに解凍するとうまく動作しない可能性があります。
- ④ Eclipse を起動後、Web アプリケーションを作成するための設定を行います。メニューから「ウィンドウ→設定」と選択し、開いたウィンドウ内左側のリストの中から「サーバー」の左側にある+を押下します。
- ⑤ 展開後に出てくる「ランタイム環境」を選択してください。
- ⑥ 右側の上にある、「追加」ボタンを押下します。
- ⑦ 小さなウィンドウが開かれサーバーの名前と Version が選択肢として現れますので、「Apache Tomcat6.0」を選択後、「新規ローカルサーバーの作成」にチェックを入れ「次へ」ボタンを押下します。



- ⑧ 「Tomcat インストールディレクトリ」部分右にある「参照」ボタンを押下し、Eclipse を解答したフォルダ内にある「tomcat」フォルダの中の「6.0」フォルダを選択し「完了」ボタンを押下します。



- ⑨ JSP ファイルのエンコードの設定を行います。「ウィンドウ→設定→Web→JSP ファイル」のエンコードを「ISO 10646/Unicode(UTF-8)」に設定します。

以上で Eclipse の環境構築は終了となります。

2 Amateras StrutsIDE プラグイン

Struts プラグインの動作には GEF プラグインと EclipseHTMLEditor プラグインが必要となります。

① GEF プラグインのインストール方法

Eclipse を起動し、「ヘルプ→新規ソフトウェアのインストール」を選択します。

開いたウィンドウ内、作業対象と書いてある部分からプルダウンで「-すべての使用可能なサイト-」を選択します。

リストが展開されたら、「モデリング」の項目を探し、左側の▶をクリックして、展開します。

中にある、「グラフィカル編集フレームワーク GEF SDK」にチェックを入れ、「次へ」をクリックしプラグインをインストールします。

途中、同意を求められる部分と再起動を促す部分がありますが、指示に従ってインストールして下さい。

② StrutsIDE と EclipseHTMLEditor のインストール

StrutsIDE と EclipseHTMLEditor を Amateras の Web ページからダウンロードします。以下のサイトにアクセスし、ダウンロードタブを開きます。

<https://sourceforge.jp/projects/amateras/>

※バージョンは、両方共に 2.0.7 のものをダウンロードして下さい。

ダウンロードしたファイルを、Eclipse をインストールしたフォルダ内にある plugins に移動し、Eclipse を再起動します。

以上で Struts 開発環境の構築は完了となります。

6. Struts を動かす

それでは、Struts の機能の詳細を確認する前に、まずは Struts の動作を体感してみましょう。
今回は、ログインフォームに ID とパスワードを入力し、ログインボタンを押すことで画面遷移をし、次の画面では、入力した ID とパスワードが表示されるという簡単な Web アプリケーションを作成してみます。

■ログイン画面



■ログイン結果



アプリケーション作成の流れは次の通りです。

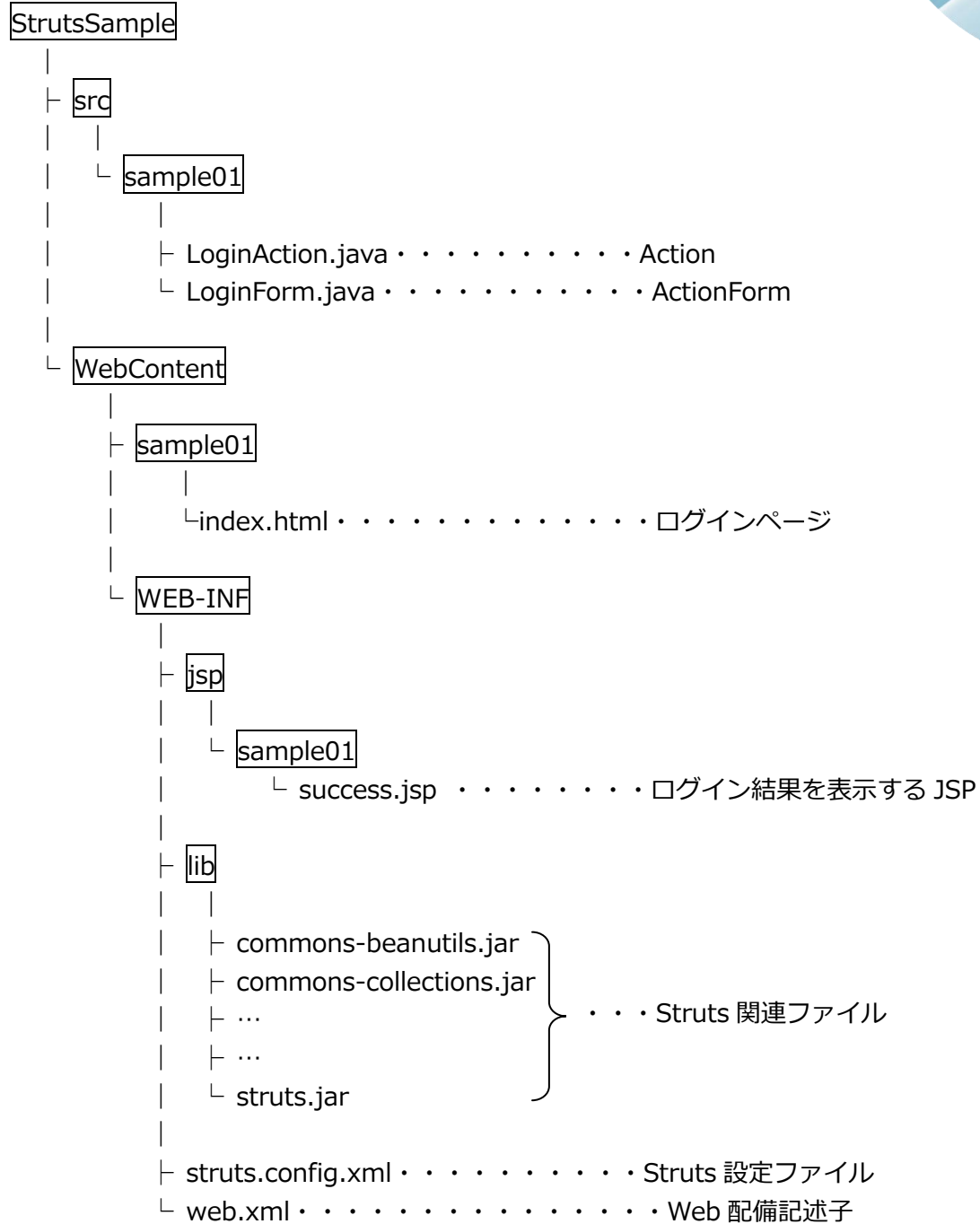
- (1) プロジェクト作成
- (2) 設定ファイル (web.xml、struts-config.xml) 作成
- (3) Action、ActionForm 作成
- (4) JSP、HTML 作成
- (5) ビルドとデプロイ
- (6) Tomcat 起動

作成するファイルの一覧を以下に示します。

ファイル名	解説	ロケーション
web.xml	アプリケーションの設定ファイル	/WEB-INF/
struts-config.xml	Struts 設定ファイル	/WEB-INF/
LoginAction.java	入力された ID やパスワードを処理する Action	/src/sample01/
LoginForm.java	リクエストパラメータで送られてきた ID やパスワードを受け取る ActionForm	/src/sample01/
success.jsp	ログイン結果を示す JSP	/WEB-INF/jsp/sample01/
index.html	ID やパスワードの入力フォーム	/sample01/

Eclipse 上から見たディレクトリ構成を以下に示します。

【ディレクトリ構成】



1 プロジェクトの作成

① プロジェクトの設定

Eclipse で新規にプロジェクトを作成します。

- ・ Eclipse のファイルタブ→新規→その他→Web→動的 WEB プロジェクト

プロジェクト名 : StrutsSample

ターゲット・ランタイム : Apache Tomcat v6.0

② サーバーに追加

サーバービューの「ローカル・ホスト の Tomcat v6.0 サーバー」を右クリックで「追加および除去」を選択します。

次に「StrutsSample」を選択して、追加ボタンを押下します。

③ Struts サポートの追加

作成した「StrutsSample」プロジェクトの「WebContent」を右クリックし、「新規→その他」と進みます。

開いたウィンドウから、「Amateras→Struts→Struts サポートの追加」を選択し、「次へ」を押下します。

「Web アプリケーションのルート」の値が「/StrutsSample/WebContent」であることを確認し完了ボタンを押下します。

2 設定ファイルの編集

Struts を動かすための準備として、次の 2 つの設定ファイルを編集します。

(1) web.xml の編集

web.xml は Web アプリケーションの要素および配置情報をサーブレットコンテナに伝えま
る役割を持つ重要なファイルです。この中に Struts を使用するための記述を行います。

次のように、「/WebContent/WEB-INF/web.xml」を編集します。

【web.xml】

```
<?xml version="1.0" encoding="Shift_JIS"?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4">

  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
```

```
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>

</web-app>
```

Struts を利用した Web アプリケーションで使用されるサーブレットは、org.apache.struts.action.ActionServlet だけです。servlet 要素では、このサーブレットを action という名前で登録します。

このサーブレットは struts 設定ファイルに書かれた内容に従って動作するので、その設定ファイル名をサーブレット初期値として設定します。ここではファイル名を struts-config.xml とし、web.xml と同じディレクトリに配置しています。load-on-startup 要素は、Tomcat の起動と同時に起動するサーブレットを指定するものですが、ここでは ActionServlet を起動しています。

servlet-mapping 要素では拡張子が「do」の URL でアクセスされた際に、ActionServlet が呼び出されるように設定しています。

(2) struts-config.xml の作成

次に Struts 設定ファイルの設定を行います。このファイルは Struts を使用したアプリケーションの設計図に当たります。

「/WebContent/WEB-INF/」の直下に次のようなファイルを作成してください。

【struts-config.xml】

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!DOCTYPE struts-config PUBLIC
  "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
  "http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">

<struts-config>
  <form-beans>
    <form-bean name="sample01_loginForm" type="sample01.LoginForm"/>
  </form-beans>

  <action-mappings>
    <action path="/sample01/login"
      type="sample01.LoginAction"
      name="sample01_loginForm"
      scope="request">
```



```
<forward name="success" path="/WEB-INF/jsp/sample01/success.jsp"/>
</action>
</action-mappings>
</struts-config>
```

struts-config.xml については後ほど詳しい説明を行いますが、ここでは<action>タグの path 属性に指定した URL (http://localhost:8080/StrutsSample/sample01/login.do) にアクセスすると、type 属性に指定した LoginAction が実行され、最終的には<forward>タグの path 属性に指定した success.jsp が表示されるという流れだけを頭に入れておいてください。

設定ファイルの編集は以上となります。

3 Action クラスと ActionForm クラスの作成

Struts アプリケーションで必要となるクラスとして最も基本となる Action クラスと ActionForm クラスの作成を行います。2 つのクラスを作成する前に「/src/」の直下に「sample01」というパッケージを作成しておきましょう。

まずは ActionForm を作成します。先ほど作成したパッケージの直下に LoginForm という名前のクラスを作成しましょう。ActionForm は org.apache.struts.action.ActionForm を継承する必要があります。

ActionForm の役割は、リクエストパラメータの受け取りです。リクエストパラメータ名に対応するプロパティと、それらに対する getter/setter メソッドを作成すると値が自動的にプロパティにセットされます。

作成する LoginForm を次に示します。

【LoginForm.java】

```
package sample01;

import org.apache.struts.action.ActionForm;

public class LoginForm extends ActionForm {

    // リクエストパラメータ名と同じプロパティ
    private String id = null;
    private String password = null;

    // getter/setter メソッド
    public String getId() {
        return id;
    }

    public String getPassword() {
        return password;
    }

    public void setId(String id) {
        this.id = id;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

続いて、Action を作成します。作成場所は ActionForm と同様です。Action の役割は、データベースへの問い合わせなど、ビジネスロジックを呼び出すことと、処理が終了した際に JSP や HTML にフォワードを行い、HTTP レスポンスとして利用者へ応答を返し一連の処理を完了させることです。

Action を作成する際は、org.apache.struts.action.Action を継承します。このクラスでは execute() メソッドをオーバーライドし、このメソッドの中にリクエストパラメータに対する処理などを記述します。

今回は、引数で渡される ActionForm を先ほど作成した LoginForm にキャストし、LoginForm にセットされたリクエストパラメータの値を取得します。続いて、次の画面で表示する文字列を request スコープに登録します。最後に、フォワード先の JSP を指定する ActionForward インスタンスを返します。

作成する LoginAction を次に示します。

【LoginAction.java】

```
package sample01;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class LoginAction extends Action {
    public ActionForward execute(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws Exception {

        // ActionForm を LoginForm にキャスト
        LoginForm loginForm = (LoginForm) form;

        // フォームに入力された ID を取得する
        String id = loginForm.getId();

        // フォームに入力されたパスワードを取得する
        String password = loginForm.getPassword();
        String message = " ID : " + id + " パスワード : " + password;

        // message を request スコープに登録する
        request.setAttribute("message", message);
    }
}
```

```
// ActionForward を返却  
return mapping.findForward("success");  
}  
}
```

4 HTML、JSP の作成

最後に、ID とパスワードを入力する index.html と、入力された文字列を表示させる success.jsp を作成します。index.html は「/WebContent/sample01/」の直下に、success.jsp は「/WebContent/WEB-INF/jsp/sample01/」の直下に作成します。

まずは、index.html の作成です。id と password の input 要素の name 属性の値は LoginForm のプロパティと一致している必要がありますので注意してください。

【index.html】

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
    <title>ログイン画面</title>  
  </head>  
  <body>  
    <h2>ログイン画面</h2>  
    <form action="../sample01/login.do" method="post" >  
      ID: <input type="text" name="id"><br>  
      PASSWORD: <input type="password" name="password"><br>  
      <input type="submit" value="ログイン">  
    </form>  
  </body>  
</html>
```

続いて、success.jsp を作成します。LoginAction でリクエストにセットした message の値を参照しています。

【success.jsp】

```
<%@ page contentType="text/html; charset=UTF-8" %>  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
    <title>ログイン結果</title>
```

```
</head>
<body>
  <h2>ログイン結果</h2>
  <%= (String)request.getAttribute("message") %>
</body>
</html>
```

5 サンプルを実行する

それでは、作成したアプリケーションの動作を確認しましょう。本来は作成したプログラムをコンパイル（ビルド）と Tomcat への配備（デプロイ）が必要ですが、これらの作業は Eclipse が自動的に行うため割愛します。

それでは、Tomcat を起動します。（すでに Tomcat が起動している場合は、再起動してください。

Tomcat が起動したら、Web ブラウザから次の URL にアクセスし、動作を確認しましょう。

<http://localhost:8080/StrutsSample/sample01/index.html>

■ ログイン画面



■ ログイン結果



Struts アプリケーションの動作確認は以上となります。