

Web 演習の課題である「公欠システム」についてのヒント集です。

## 1. カレンダーコントロールについて

カレンダーの表示については、HTML5 より存在するカレンダーコントロールを使うとらくちんです。

例) `<input type="date" name="date">`

↓こんな感じで表示されます

公欠日



※ただし、エクリプスの内部ブラウザは HTML 5 に対応していないので、ただのテキストになります。動作確認するときは、FireFox やクロームなどで確認しましょう。

また、上記の例であればサーバレット側で入力情報を取得するのは

```
//値の取得
String dateString = request.getParameter("date");
```

で OK です。

例えば、入力画面で 2019 年 10 月 1 日を選ぶと、変数 dateString には  
“2019-10-01” という形でハイフン区切りの文字列で値が格納されます。

## 2. 日付の変換について

ビーンズに格納するときに String→Date の変換が必要になります。

日付の変換は SimpleDateFormat を使うとできます。以下例です。

```
//文字列→日付の変換
String dateValue = "2019-11-01";
try{
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Date date = sdf.parse(dateValue);
}catch(ParseException e){
    //エラー時の処理
}
```

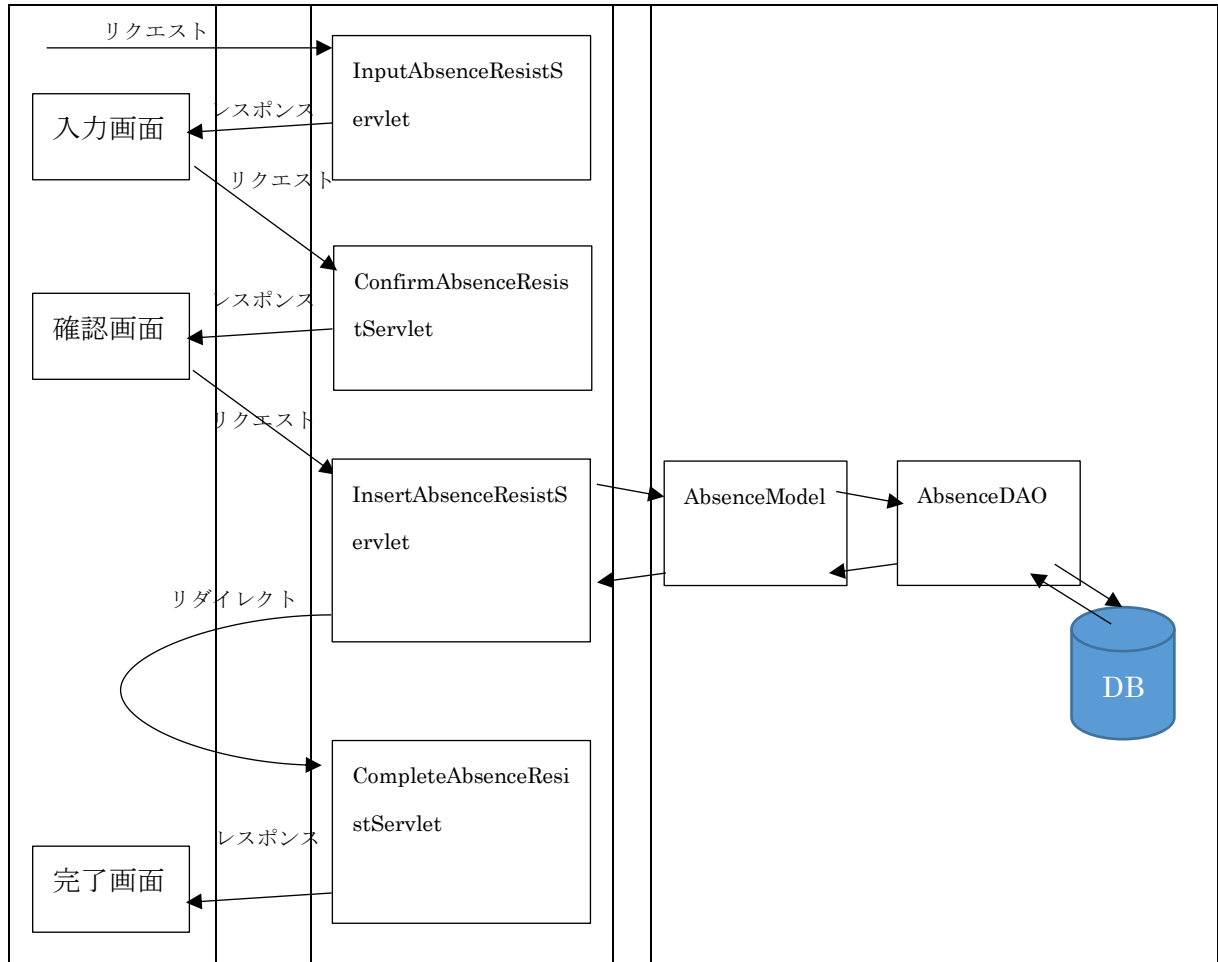
詳細は調べてみよう！

### 3. 登録処理の流れについて

登録の処理の画面遷移は以下ようになります。

登録画面→確認画面→完了画面

この流れを MVC 含めて考えると



DB に値を挿入するタイミングは確認画面で登録ボタンを押したとき、つまり InsertAbsenceResistServlet で行います。

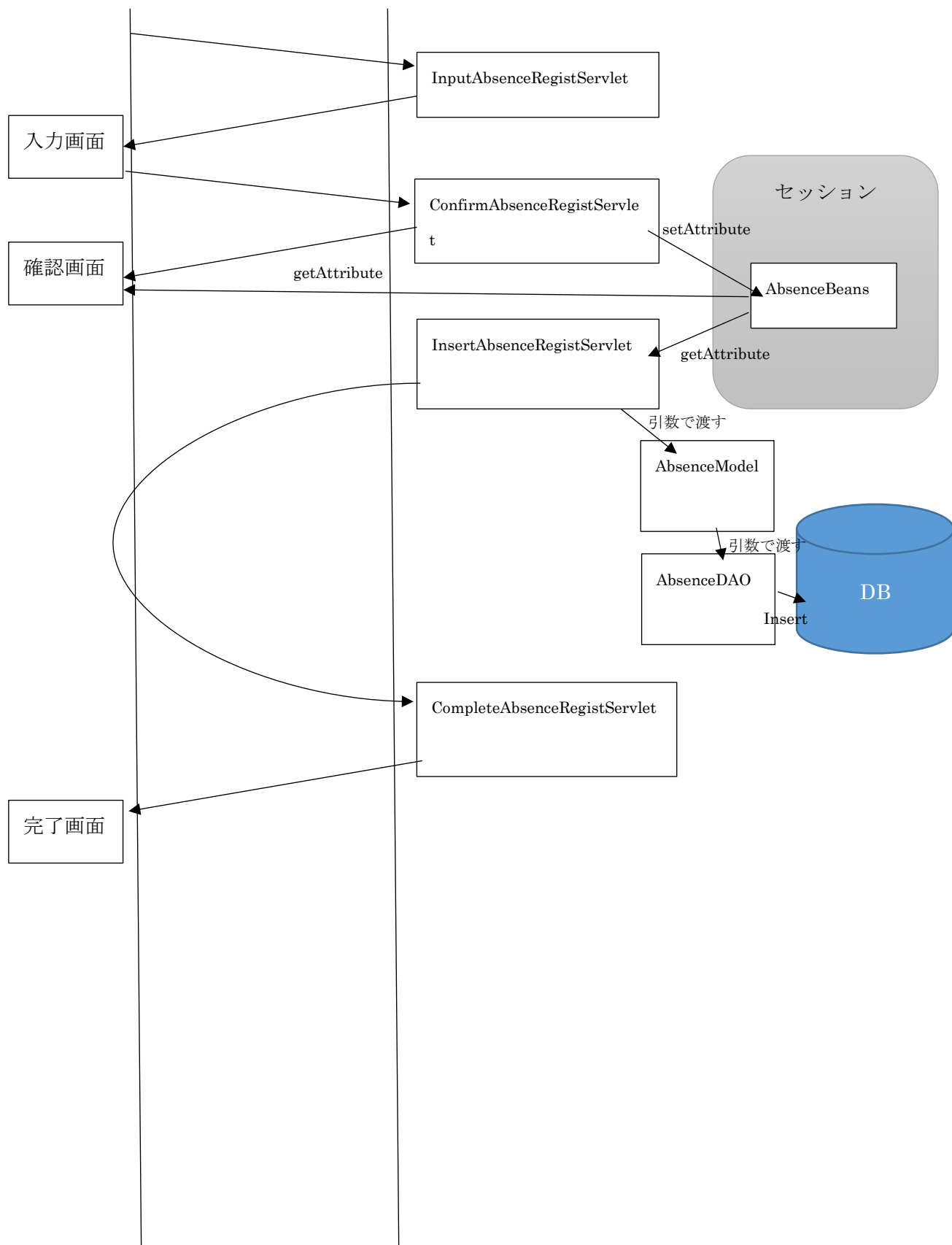
挿入するデータは最初の入力画面で入力した値です。言い方を変えると**入力画面で入力した値を完了画面表示直前で挿入**するのです

要するに、リクエスト(画面)をまたいで、入力した情報を持ちまわらなければなりません。

リクエストをまたいで情報を保持するときに使用するのは「セッション」でしたね。

そう、登録画面の流れはセッションを使用して実現します。

セッションを含めた流れを図にしたものは以下の通りです



## 登録処理のソースヒント

### ConfirmAbsenceResistServlet

```
@WebServlet("/absence_confirm")  
public class ConfirmAbsenceResistServlet extends HttpServlet {  
  
    @Override  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException  
    {  
        ///////////////////////////////////  
        //値の取得  
          
        ///////////////////////////////////  
        //ビーンズに値をセットする  
        AbsenceBeans absenceBeans = new AbsenceBeans();  
          
        ///////////////////////////////////  
        //文字列→日付の変換  
        try{  
              
            absenceBeans.setAbsenceDate(date);  
        }catch(ParseException e){  
            absenceBeans.setAbsenceDate(new Date());  
        }  
          
        //ユーザーIDを取得するためセッションから  
        //ログイン情報を取得する  
          
        //ビーンズにuserId(学籍番号)をセット  
          
        ///////////////////////////////////  
        //セッションに保存(本来は保存の前に入力チェックをする)  
          
        ///////////////////////////////////  
        //画面遷移  
          
    }  
}
```

## InsertAbsenceResistServlet

```
@WebServlet("/absence_insert")  
public class InsertAbsenceResistServlet extends HttpServlet {  
  
    @Override  
    protected void doPost(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException  
    {  
        ~~~~~  
        //セッションから値を取得  
        ~~~~~  
        ~~~~~  
        //登録処理  
        ~~~~~  
        ~~~~~  
        //セッション情報を削除  
        session.removeAttribute("absenceBeans");  
        ~~~~~  
        //完了画面へリダイレクト  
        ~~~~~  
    }  
}
```

## AbsenceModel

```
public class AbsenceModel {  
  
    /**  
     * 公欠届の一覧を取得する（全権取得）  
     *  
     * @return  
     */  
    public List<AbsenceBeans> getList(){  
        List<AbsenceBeans> list = null;  
  
        AbsenceDao absenceDao = new AbsenceDao();  
  
        try{  
            //DB接続  
            ~~~~~  
            ~~~~~  
            //一覧を取得  
            ~~~~~  
        }finally{  
            //DB切断  
            ~~~~~  
        }  
  
        return list;  
    }  
  
    /**  
     * 公欠届の新規挿入  
     *  
     * @param absenceBeans  
     */  
    public void insert(AbsenceBeans absenceBeans){  
        AbsenceDao absenceDao = new AbsenceDao();  
  
        try{  
            //DB接続  
            ~~~~~  
            ~~~~~  
            //情報を挿入する  
            ~~~~~  
        }finally{  
            //DB切断  
            ~~~~~  
        }  
    }  
}
```

## ○ログアウトについて

ログアウトはセッションを破棄すれば OK です

セッションを破棄するメソッドは、HttpSession が持っている invalidate メソッドを使用します。

また、ログアウト後はログイン画面に遷移するのが一般的なので、ログイン画面にリダイレクトします。

処理の手順としては

1. セッションオブジェクトを取得する
2. セッションを破棄するメソッドを呼び出す
3. ログイン画面にリダイレクト

となります

一覧表示について

一覧表示についての流れは以下の通りです

--	--	--	--	--