セッション操作

Spring Frameworkのsessionスコープの Beanを利用する。

sessionスコープBeanの実装

sessionスコープのBeanは次の様に作成する。

```
@Component //  <1>
@SessionScope //  <2>
public class SampleSessionScopedDto implements Serializable  //  <3>

    private static final long  serialVersionUID  1L;

    private String  foo;

    private  BazDto  bε//  <4>

    //  getter/setter  省略

}
```

- <1> @Componentを付与し、Validatorをコンポーネントスキャン対象にする
- <2> @SessionScopeを付与し、Beanのスコープをセッションスコープにする
- <3> Serializable実装する
- <4> 含まれるDTOもSerializeが実装されていることを確認

sessionスコープBeanの利用方法

sessionスコープのBeanを利用して、オブジェクトをセッションに格納・取得する場合は、
次の様にsessionスコープのBeanを、ControllerにInjectして利用する。

```
@Controller
@RequestMapping("foo/DBAP0010")
public class DBAP0010PController  {

    @Autowired;
    private  SampleSessionScopedDto  sampleSessionScopedDt//  <1>

    @GetMapping("init")
    public String  init(Model  model)  {
        //  ...
                    model.addAttribute(sampleSessionSc//  <2to);
        //  ...
        return "view  nam";
          }
}
```

- <1> @Autowiredでインジェクションする
- <2> View(JSP)に渡す場合はModel#addAttribute(Object)で渡す
    - JSP上では ${sampleSessionScopedDto.foo} という形で参照する

sessionスコープBeanの利用方法 (画面共通部品向け)

広範囲に渡って横断的に存在するセッション情報を画面共通部品から参照する場合、
上記のようにControllerから明示的に渡す事も出来るが、却って保守性を下げることになるので、
次の様にspring:evalタグを使ってJSP上から直接sessionスコープのBeanを参照しても良い。

```
<spring:eval var="fooDto" expression="@fooDto" /> <!--1-->
```

${f:h(fooDto.bar)} <!--2-->

- <1> sessionスコープBeanをspring:evalタグで参照する
  - @bean名 という形で参照する
  - FooDtoの場合、Bean名は fooDto になる
- <2> sessionスコープBeanから値を取得する

参考

- 4.3. セッション管理 — TERASOLUNA Server Framework for Java (5.x) Development Guideline 5.3.0.RELEASE documentation