

Expression Language

Summary

Web Flow uses EL for data model and action execution. Here we will look at EL and define the flow.

Description

Supported EL Implementation

Unified EL

Set to use Unified EL by default. jboss-el is set to basic implement.

Note : web container generally supports el-api. It is like the case of tomcat 6.\

OGNL

OGNL is another el provided by SWF2. If adding to class path, it automatically find and use.

EL Compatibility

Unified EL and OGNL has similar grammar. Let's just use unified EL if possible.

Using EL

In case of using EL in the flow,

- In accessing data provided by the client, the input properties or request parameter
- In accessing internal data structure as flowScope
- Method calling in the spring bean
- In deciding creator

The view shown by Flow access the flow data structure using EL.

Expression Type

Standard eval Expression

The most general method is the eval expression. In this case, `${}` or `#{}` cannot be used. This example is called by `nextPage()` in `searchCriteria`.

```
<evaluate expression="searchCriteria.nextPage()" />
```

Expression Template

Next is the "template" expression and `${}` can be used as following.

```
<view-state id="error" view="error-${externalContext.locale}.xhtml" />
```

Replace the locale result set in `externalContext` and create as `error-result.xhtml`.

Special EL Variable

Scope

- `flowScope`: the object that is allocated to flow variable and has Flow scope. All objects saved in Flow range should be Serializable by default.

```
<evaluate expression="searchService.findHotel(hotelId)" result="flowScope.hotel" />
```

- viewScope: allocated to view variable and has range within view-state. Therefore, it can be referred only in view-state. All objects should be Serializable.

```
<on-render>
```

```
<evaluate expression="searchService.findHotels(searchCriteria)" result="viewScope.hotels" result-type="dataModel" />
```

```
</on-render>
```

- requestScope: allocated to request variable. Share within one Flow.

```
<set name="requestScope.hotelId" value="requestParameters.id" type="long" />
```

- flashScope: allocated to flash variable. When Flow starts, it is allocated. The area that is summarized upon termination of flow and is cleared after the view is shown. Object should be Serializable.

```
<set name="flashScope.statusMessage" value=""Booking confirmed"" />
```

- conversationScope

Allocated to conversation variable. Allocated when the topmost Flow starts and summarized when the topmost Flow is terminated. Share in children flow of the topmost Flow. Saved in HTTP session and should be serializable to cope with the situation when session copying is performed.

```
<evaluate expression="searchService.findHotel(hotelId)" result="conversationScope.hotel"/>
```

Context

- flowRequestcontext: express current Flow request. RequestContext API.
- messageContext: receive flow execution message including error or success message and can access the context for making. See MessageContext.

```
<evaluate expression="bookingValidator.validate(booking, messageContext)" />
```

- flowExecutionContext: express current Flow status. FlowExecutionContext API.
- externalContext: can access the external environment including user session property. ExternalContext API.

```
<evaluate expression="searchService.suggestHotels(externalContext.sessionMap.userProfile)" result="viewScope.hotels" />
```

Others

- requestParameters: access the request parameter from user

```
<set name="requestScope.hotelId" value="requestParameters.id" type="long" />
```

- currentEvent: access current Event objects

```
<evaluate expression="booking.guests.add(currentEvent.guest)" />
```

- currentUser: access the authenticated Principal

```
<evaluate expression="bookingService.createBooking(hotelId, currentUser.name)" result="flowScope.booking" />
```

- resourceBundle: manage message resource

```
<set name="flashScope.successMessage" value="resourceBundle.successMessage" />
```

- flowExecutionUrl: access context-relative URI for current flow execution view-state

Scope Searching Algorithm

In allocating variables in certain scope, the scope should be stated.

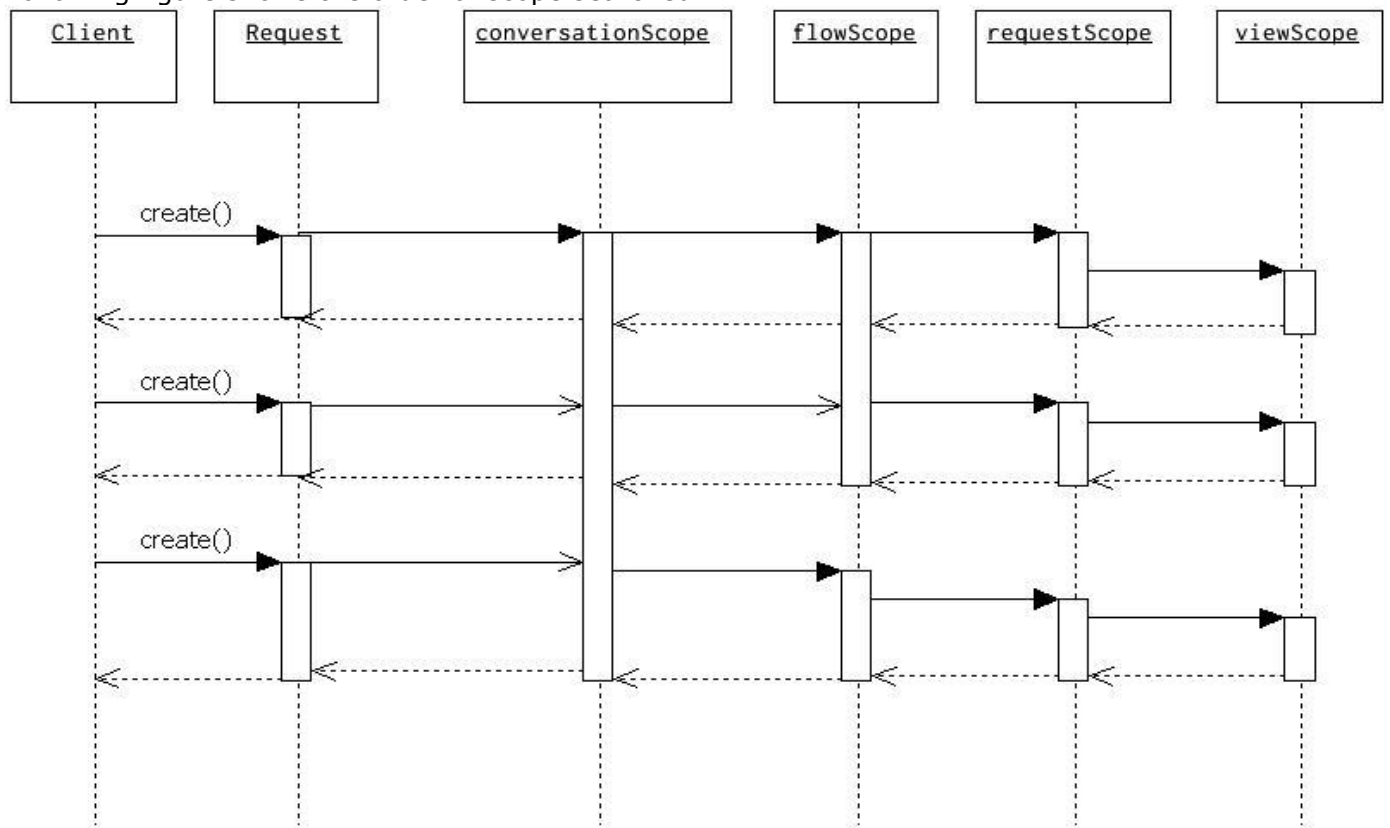
```
<set name="requestScope.hotelId" value="requestParameters.id" type="long" />
```

In accessing variables in certain scope, the scope does not have to be stated.

```
<evaluate expression="entityManager.persist(booking)" />
```

If the scope is not indicated like booking, scope searching algorithm operates. This algorithm will search for the order in the scope of request→flash→view→flow→conversation. EvaluationException occurs if not.

Following figure shows the order of scope searched.



Reference

- [Spring Web Flow reference 2.0.x](#)
- Spring Web-Flow Framework Reference beta with Korean (by Park Chan Wook)
- Pro Spring 2.5(Apress) - Chapter 18 Spring Web Flow