

# deutsches forschungsnetz



---

---

---



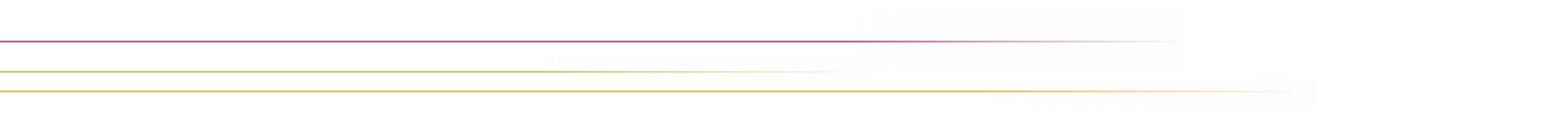
Shibboleth IdP 3.x

Spring Web Flow

AAI- und Shibboleth-Workshop | Berlin, 18./19. Februar 2019

Wolfgang Pempe ([pempe@dfn.de](mailto:pempe@dfn.de))

Steffen Hofmann ([steffen.hofmann@fu-berlin.de](mailto:steffen.hofmann@fu-berlin.de))



- ▶ Offenes Java Framework (<https://spring.io/>) für Entwicklung in Java EE
- ▶ Im Shibboleth IdP 3.x kommt Spring in zweierlei Hinsicht zum Tragen:
  - ▶ Spring-basierte Konfiguration: „generalized Spring bean file format“ (für die Dateien, die nicht oder nur in Ausnahmefällen bearbeitet werden), ansonsten Hybrid-Varianten
  - ▶ **Spring Web Flow:** „mechanism used to manage IdP profile logic, authentication processing, and essentially all of the orchestration of function in the system. [...] The Spring Web Flow mechanism is used to execute ProfileActions, which make up the bulk of the IdP's processing model“  
(<https://wiki.shibboleth.net/confluence/display/IDP30/GeneralArchitecture>)

# Spring Web Flow

- ▶ <http://projects.spring.io/spring-webflow/>
- ▶ Basiert auf Spring MVC (Model View Control)
- ▶ Flow: Abfolge von Schritten, die einen bestimmten Geschäftsprozess definieren, bei Shib IdP in erster Linie sog. Profile Actions (SSO, SLO etc.)
- ▶ Status („state“): einzelner Schritt im Flow
- ▶ Daten können von einem State zum nächsten weitergereicht werden

## Was ist ein Flow?

- ▶ Es gibt einen klaren Startpunkt.
- ▶ Es wird eine Abfolge definierter Schritte/Status durchlaufen.
- ▶ Ereignisse steuern den Statusübergang, also den Übergang von einem Status zu einem anderen Status.
- ▶ Es gibt jeweils einen klaren Endzustand

# State

- ▶ State: ein Status/Zustand innerhalb der Abfolge
- ▶ Jeder State hat innerhalb einer Flow-Definition eine eindeutige **id**, z.B. „ValidateUsernamePassword“
- ▶ Arten von States
  - ▶ View State
  - ▶ Action State
  - ▶ Subflow State
  - ▶ Decision State
  - ▶ End State

## <view-state>

- ▶ Anzeige einer Webseite
- ▶ Fortsetzung durch Absenden eines Formulars oder Link

## <action-state>

- ▶ Abarbeitung 1..n Expressions
- ▶ Aufruf von Methoden von bereits instanziierten Beans
- ▶ Neuinstanziierung von Klassen
- ▶ ...



## <subflow-state>

- ▶ Subflows sind Flows, die aus anderen Flows heraus aufgerufen werden können.
- ▶ Lassen sich aus verschiedenen Flows heraus aufrufen
- ▶ Enden in der Regel durch den Rücksprung in den jeweils aufrufenden Parent-Flow

## <decision-state>

- ▶ einfache Ablaufsteuerung anhand von Kontextattributen
- ▶ letztlich bedingte Anweisung (if ... then ... else)

<end-state>

- ▶ Definiertes Ende eines Flows

# Events

- ▶ Ein Event besteht u.a. aus
  - ▶ id (z.B. „proceed“, „success“, „NoCredentials“)
  - ▶ source (irgendein Object als Quelle)
  - ▶ attributes (Map mit beliebigen Objekten als Attribute)
- ▶ siehe `org.springframework.webflow.execution.Event`

# Events und States

- ▶ In jedem State werden Events erzeugt:
  - ▶ view-state: POST- oder GET-Parameter **\_eventId**
  - ▶ action-state: die Ergebnisse der aufgerufenen Expressions mithilfe von **<evaluate>**
  - ▶ subflow-state: das Ergebnis des Subflows
  - ▶ decision-state: der Wert von **then** oder **else**
  - ▶ end-state: die **id** des end-state ist gleichzeitig die **id** des Events

## Events im view-state, Beispiele

```
<form method="post">  
    <input type="submit" value="confirm,, name="_eventId"/>  
    <input type="submit" value="ablehnen,, name="_eventId_cancel"/>  
</form>  
  
<a href="{flowExecutionUrl}&_eventId=confirm">confirm via url</a>
```

# Evaluate

- ▶ letztlich das meist genutzte Element in einem Flow
- ▶ an vielen Stellen anwendbar
- ▶ besonders häufig in action-state oder bei Eintritt in view-state
- ▶ Aufruf von `org.springframework.webflow.action.EvaluateAction`
- ▶ Es wird immer ein Event zurückgegeben!
- ▶ Attribute des Elements:
  - ▶ `expression`: Aufruf einer Klassenmethode
  - ▶ `result`: das Ergebnis kann im Datenmodell des Flows gespeichert werden
  - ▶ `result-type`: Konvertierung (cast) eines eventuellen Rückgabewertes

# Transition

- ▶ Statusübergänge, Ziel ist also ein neuer State
- ▶ Werden in der Regel an Ereignisse gebunden
- ▶ „state-Level“ Transitions innerhalb eines States und nur dort gültig
- ▶ „flow-Level“ Transitions sind global-transitions, die für alle States in einem Flow gelten
- ▶ Eine Transition wird ausgelöst, sobald ein passendes Event existiert!



## Flow-Definition

- ▶ Das Zusammenspiel der Status (State) und der Statusübergänge (Transition) wird in der Regel in einer XML-Datei festgelegt
- ▶ beim IdP an \*-flow.xml zu erkennen

# Gerüst XML-Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<flow xmlns="http://www.springframework.org/schema/webflow"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.springframework.org/
schema/webflow http://www.springframework.org/schema/webflow/
spring-webflow-2.0.xsd">

</flow>
```

Beispiel:

`${idp.home}/system/flows/intercept/expiring-password-flow.xml`

# Scopes

- ▶ Flash-Scope
- ▶ Request-Scope
- ▶ View-Scope
- ▶ Flow-Scope
- ▶ Conversation-Scope

## Flash-Scope

- ▶ Scope wird mit Eintritt in Flow erzeugt
- ▶ Werte bleiben nur bis zum Verlassen des nächsten View-State im Kontext, wird also nach einer Anzeige geleert
- ▶ Auf diese Weise können Fehlermeldungen und andere Inhalte auf eine Folgeseite übertragen werden
- ▶ Werte bleiben bei einem Refresh im Browser erhalten
- ▶ Scope wird mit Flow-Ende zerstört.

## Request-Scope

- ▶ wird mit dem HTTP-Request erzeugt
- ▶ wird mit der Anzeige einer View bzw. der HTTP-Response zerstört

## View-Scope

- ▶ wird erzeugt, wenn in view-state eingestiegen wird  
(`<on-entry>`)
- ▶ wird bei Anzeige der View (Render) zerstört

## Flow-Scope

- ▶ Werte bleiben nur innerhalb des jeweiligen Flows erhalten
- ▶ Werte sind von Parent- und Subflows abgetrennt
- ▶ Scope wird beim Verlassen des Flows zerstört

## Conversation-Scope

- ▶ Umfasst alle Flows
- ▶ Werte können an jeder Stelle auch im z.B. Subflow manipuliert werden, daher ist Vorsicht geboten
- ▶ **Hiermit werden vom IdP alle wichtigen Daten während eines Authentifizierungsdurchlaufs zusammengefasst transportiert**



# Vielen Dank! Fragen? Kommentare?

# DFN

## ► Kontakt

### ► DFN-AAI Team

E-Mail: [aai@dfn.de](mailto:aai@dfn.de)

Tel.: +49-30-884299-9124

Fax: +49-30-884299-370

Adresse:

DFN-Verein, Geschäftsstelle

Alexanderplatz 1

D-10178 Berlin

