

Message Store Interceptor

An interceptor to store a [ValidationAware](#) action's messages / errors and field errors into HTTP Session, such that it will be retrievable at a later stage. This allows the action's message / errors and field errors to be available longer than just the particular HTTP request.

If no session exists, nothing will be stored and can be retrieved later. In other terms, the application is responsible to open the session.

In the **STORE** mode, the interceptor will store the [ValidationAware](#) action's message / errors and field errors into HTTP session.

In the **RETRIEVE** mode, the interceptor will retrieve the stored action's message / errors and field errors and put them back into the [ValidationAware](#) action.

In the **AUTOMATIC** mode, the interceptor will always retrieve the stored action's message / errors and field errors and put them back into the [ValidationAware] action, and after Action execution, if the [Result](#) is an instance of [ServletRedirectResult](#), the action's message / errors and field errors into automatically be stored in the HTTP session..

The interceptor does nothing in the **NONE** mode, which is the default.

The operation mode could be switched using:

1. Setting the interceptor parameter eg.
xml
2. Through request parameter `allowRequestParameterSwitch` must be 'true' which is the default

Parameters

- `allowRequestParameterSwitch` - To enable request parameter that could switch the operation mode of this interceptor.
- `requestParameterSwitch` - The request parameter that will indicate what mode this interceptor is in.
- `operationMode` - The operation mode this interceptor should be in (either **STORE**, **RETRIEVE**, **AUTOMATIC**, or **NONE**). **NONE** being the default.

Extending the Interceptor

There is no known extensions.

Examples

xml

With the example above, `submitApplication.action` will have the action messages / errors / field errors stored in the HTTP Session. Later when needed, (in this case, when `applicationFailed.action` is fired, it will get the action messages / errors / field errors stored in the HTTP Session and put them back into the action.