

# Workflow Exercise

Workflow Exercise	1
Create a workflow	2
Create AutomatedWorkflow Beans	2
Introduce workflow into the Lifecycle	4

## Create a workflow

[Should be contained in TelcotrainingCore essential data impex]

INSERT\_UPDATE

```
WorkflowTemplate;code[unique=true];name[lang=de];name[lang=en];owner(uid);description[lang=d  
e];description[lang=en];activationscript;visibleForPrincipals(uid);;  
;TelcoCustomerRegistration;TelcoCustomer Registrierung;TelcoCustomer  
Registration;admin;;;admingroup;;;
```

INSERT\_UPDATE

```
WorkflowActionTemplate;code[unique=true];name[lang=de];name[lang=en];description[lang=de];de  
scription[lang=en];principalAssigned(uid);workflow(code)  
[unique=true];sendEmail;emailAddress;rendererTemplate(code);actionType(code)  
;NPR_Confirm;Bestätigung von einem Administrator;Confirmation by an  
administrator;;;admingroup;TelcoCustomerRegistration;false;;;start  
;NPR_End;Ende;End;;;admingroup;TelcoCustomerRegistration;false;;;end
```

INSERT\_UPDATE

```
AutomatedWorkflowActionTemplate;code[unique=true];name[lang=de];name[lang=en];description[l  
ang=de];description[lang=en];principalAssigned(uid);workflow(code)  
[unique=true];sendEmail;emailAddress;rendererTemplate(code);actionType(code);jobHandler  
;NPR_Confirm_Positive;TelcoCustomer bestätigt;TelcoCustomer  
confirmed;;;admingroup;TelcoCustomerRegistration;false;;;normal;registrationConfirmationActio  
nJob  
;NPR_Confirm_Negative;TelcoCustomer nicht bestätigt;TelcoCustomer  
declined;;;admingroup;TelcoCustomerRegistration;false;;;normal;registrationDeclineActionJob
```

INSERT\_UPDATE

```
WorkflowDecisionTemplate;code[unique=true];name[lang=de];name[lang=en];actionTemplate(code  
);description[allownull=true];;;;;  
;NPR001;TelcoCustomer bestätigen;Confirm TelcoCustomer;NPR_Confirm;,,,,;  
;NPR002;TelcoCustomer nicht bestätigen;Decline TelcoCustomer;NPR_Confirm;,,,,;  
;NPR003;Ende;End;NPR_Confirm_Positive;,,,,;  
;NPR004;Ende;End;NPR_Confirm_Negative;,,,,;
```

INSERT\_UPDATE WorkflowActionTemplateLinkTemplateRelation;source(code)

```
[unique=true];target(code)[unique=true];andConnectionTemplate;qualifier;,,,,;  
;NPR001;NPR_Confirm_Positive;false;WorkflowActionTemplateLinkTemplateRelation;,,,,;  
;NPR002;NPR_Confirm_Negative;false;WorkflowActionTemplateLinkTemplateRelation;,,,,;  
;NPR003;NPR_End;false;WorkflowActionTemplateLinkTemplateRelation;,,,,;  
;NPR004;NPR_End;false;WorkflowActionTemplateLinkTemplateRelation;,,,,;
```

## Create AutomatedWorkflow Beans

After that create the beans corresponding to the automated actions.

```
public abstract class AbstractCustomerRegistrationActionJob implements  
AutomatedWorkflowTemplateJob {
```

```
    @Autowired  
    private ModelService modelService;
```

@Autowired

private EmailService mailService;

protected TelcoCustomerModel getAttachedTelcoCustomer(final WorkflowActionModel action)

```
{
    final List<ItemModel> attachments = action.getAttachmentItems();
    if (attachments != null)
    {
        for (final ItemModel item : attachments)
        {
            if (item instanceof TelcoCustomerModel)
            {
                return (TelcoCustomerModel) item;
            }
        }
    }
    return null;
}
```

public void setModelService(final ModelService modelService)

```
{
    this.modelService = modelService;
}
```

protected ModelService getModelService()

```
{
    return this.modelService;
}
```

public void setMailService(final EmailService mailService)

```
{
    this.mailService = mailService;
}
```

protected EmailService getMailService()

```
{
    return this.mailService;
}
```

}

-----  
@Component

public class RegistrationConfirmationActionJob extends AbstractCustomerRegistrationActionJob {

@Override

public WorkflowDecisionModel perform(WorkflowActionModel workflowActionModel) {

final TelcoCustomerModel telcoCustomer =

getAttachedTelcoCustomer(workflowActionModel);

if(BooleanUtils.isFalse(telcoCustomer.getVerified()))

```
{
    telcoCustomer.setVerified(Boolean.TRUE);
    getModelService().save(telcoCustomer);
}
```

```

// getMailService().createActivationEmail()

for (final WorkflowDecisionModel decision : workflowActionModel.getDecisions())
{
    return decision;
}
return null;
}
}

```

-----

```

@Component
public class RegistrationDeclineActionJob extends AbstractCustomerRegistrationActionJob {
    @Override
    public WorkflowDecisionModel perform(WorkflowActionModel workflowActionModel) {
        final TelcoCustomerModel telcoCustomer =
getAttachedTelcoCustomer(workflowActionModel);

        if(BooleanUtils.isFalse(telcoCustomer.getVerified()))
        {
            telcoCustomer.setVerified(Boolean.FALSE);
            getModelService().remove(telcoCustomer);
        }

// getMailService().createActivationEmail()

for (final WorkflowDecisionModel decision : workflowActionModel.getDecisions())
{
    return decision;
}
return null;
}
}

```

-----

in Telcotrainingcore-spring.xml you make the component scan for the package containing the workflows.

```

<context:component-scan base-package="com.vertusa.telco.core.workflow"/>

```

## Introduce workflow into the Lifecycle

```

public class TelcoCustomerValidationInterceptor implements
ValidateInterceptor<TelcoCustomerModel> {

```

```

    @Autowired
    private WorkflowService workflowService;

```

```

@Autowired
private WorkflowTemplateService workflowTemplateService;
@Autowired
private WorkflowProcessingService workflowProcessingService;
@Autowired
private UserService userService;
@Autowired
private ModelService modelService;

@Autowired
private ValidationService validationService;
@Autowired
ReloadableResourceBundleMessageSource telcoTrainingCoreMessageSource;
@Autowired
I18NService i18NService;

@Override
public void onValidate(TelcoCustomerModel model, InterceptorContext ctx) throws
InterceptorException {

    if(validationService.validate(model).isEmpty())
    {
        final WorkflowTemplateModel workflowTemplate = this.workflowTemplateService
            .getWorkflowTemplateForCode("TelcoCustomerRegistration");

        final WorkflowModel workflow = this.workflowService.createWorkflow(workflowTemplate,
model, userService.getAdminUser());
        modelService.save(workflow);
        for (final WorkflowActionModel action : workflow.getActions())
        {
            modelService.save(action);
        }

        this.workflowProcessingService.startWorkflow(workflow);
    }else throw new
InterceptorException(telcoTrainingCoreMessageSource.getMessage("telco.customer.adhhar.numb
er.length",null,i18NService.getCurrentLocale()));

}
}

```