

# Req-1 (API req): SO API extension to retrieve PNF workflow (SO-2540)

- GET

/onap/so/infra/workflowSpecifications/{version:[vV][1]}/workflows?pnfModelVersionId={UUID}

- Path Parameters

- version: String, Required
- pnfModelVersionId: String, Required

- pnfModelVersionId is the uuid of pnf resource.

```
@Path("onap/so/infra/workflowSpecifications")
@OpenAPIDefinition(info = @Info(title = "onap/so/infra/workflowSpecifications",
    description = "Queries of Workflow Specifications"))
@Component
public class WorkflowSpecificationsHandler {

    @Autowired
    private ResponseBuilder builder;

    @Autowired
    private CatalogDbClient catalogDbClient;

    private static Logger logger = LoggerFactory.getLogger(WorkflowSpecificationsHandler.class);
    private static final String ARTIFACT_TYPE_WORKFLOW = "workflow";
    private static final String NATIVE_WORKFLOW = "native";

    @Path("/{version:[vV]1}/workflows")
    @GET
    @Operation(description = "Finds Workflow Specifications", responses = @ApiResponse(
        content = @Content(array = @ArraySchema(schema = @Schema(implementation = Response.class))))
    @Transactional

    public Response queryWorkflowSpecifications(@QueryParam("vnfModelVersionId") String vnfModelVersionId,
        @PathParam("version") String version) throws Exception {
```

Existing one for VNF

# Req-2(API req): SO API to support custom PNF software upgrade workflow (SO-2071)

- POST

https://{serverRoot}/{apiRoot}/{version:[vV][1]}/serviceInstances/{serviceInstanceId}/**pnfs/{pnfInstanceId}**/workflows/{workflowUuid}

- serverRoot = https://{FQDN}:{Port}
- apiRoot = onap/so/infra/instanceManagement

- Path Parameters

- version: String, Required
- serviceInstanceId: String, Required
- **pnfInstanceId**: String, Required
- workflowUuid: String, Required

- Body parameters

- Body: String, Required  
(“requestDetails” is mandatory)

```
@Path("/onap/so/infra/instanceManagement")
@OpenAPIDefinition(info = @Info(title = "/onap/so/infra/instanceManagement",
    description = "Infrastructure API Requests for Instance Management"))
public class InstanceManagement {

    private static Logger logger = LoggerFactory.getLogger(InstanceManagement.class);
    private static String uriPrefix = "/instanceManagement/";
    private static final String SAVE_TO_DB = "save instance to db";

    @Autowired
    private RequestsDbClient infraActiveRequestsClient;

    @Autowired
    private CatalogDbClient catalogDbClient;

    @Autowired
    private MsoRequest msoRequest;

    @Autowired
    private RequestHandlerUtils requestHandlerUtils;

    @POST
    @Path("/{version:[vV][1]}/serviceInstances/{serviceInstanceId}/pnfs/{pnfInstanceId}/workflows/{workflowUuid}")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces(MediaType.APPLICATION_JSON)
    @Operation(description = "Execute custom workflow", responses = @ApiResponse(
        content = @Content(array = @ArraySchema(schema = @Schema(implementation = Response.class))))))
    @Transactional
    public Response executeCustomWorkflow(String request, @PathParam("version") String version,
        @PathParam("serviceInstanceId") String serviceInstanceId, @PathParam("vnfInstanceId") String vnfInstanceId,
        @PathParam("workflowUuid") String workflowUuid, @Context ContainerRequestContext requestContext)
```

Existing one for VNF

# Body parameters for custom PNF SW Upgrade workflow

Field Name	Format	Mandatory	Examples	Notes
{				Root object
“requestDetails ”: {	Object	M		Contains all relevant request information if any inputs were required for the chosen workflowSpecification (workflowUuid)
“requestParameters”: {	Object	M		Contains all relevant request parameters
“userParams”:[	Array	M		Contains all relevant user parameters with key-value pairs
{“name”:	String	M	“targetSoftwareVersion”	The key of user parameter
“value”: }	String	M	“v2”	The value of user parameter
]				
“payload”: “xxx”	String	O		If specified, shall be parsed as json.
}				End of requestParameters
}				End of requestDetails
}				Root object end