

The [REST API](#) is part of the product and API documentation is available with [Swagger UI](#) in [VisionX](#), e.g.:
`http://localhost/swagger/?url=http://localhost/app/webcontent/Appname/workflow.w.json`

The port may be different in your environment if port 80 is not available, e.g. `localhost:8080`.

If you want to start a workflow with Java directly, it's no problem:

```
import java.math.BigDecimal;

import org.junit.Test;
import org.restlet.data.ChallengeScheme;
import org.restlet.representation.Representation;
import org.restlet.resource.ClientResource;

import com.sibvisions.rad.server.http.rest.JSONUtil;
import com.sibvisions.util.type.StringUtil;

/**
 * Simple JUnit Test for workflow REST API.
 */
public class TestStartWorkflowRestService
{
    /**
     * Tests {@link WorkflowAccess#startWorkflow(Object, Object...)} via
     REST API.
     *
     * @throws Exception if test fails
     */
    @Test
    public void startWorkflow() throws Exception
    {
        ClientResource cres = new
ClientResource("http://localhost/services/rest/Appname/Session/object/workfl
owAccess/startWorkflow");
        cres.setChallengeResponse(ChallengeScheme.HTTP_BASIC, "admin",
"admin");

        //starts the workflow "MyWorkflow" without parameters
        Representation res = cres.post(new Object[] {"MyWorkflow"});

        System.out.println(StringUtil.toString(JSONUtil.getObject(res)));
    }
}
```

Note that the URL is different in the Java call because the Swagger UI was used from VisionX, and the manual call assumes that the application runs deployed with an application server like Tomcat.

From:

<http://doc.sibvisions.com/> - **Documentation**

Permanent link:

http://doc.sibvisions.com/workflow/rest_api

Last update: **2020/06/12 15:01**

