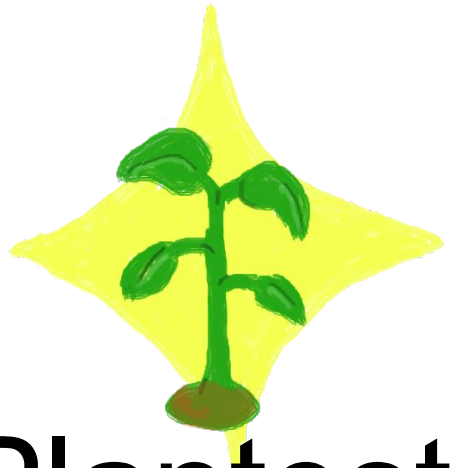
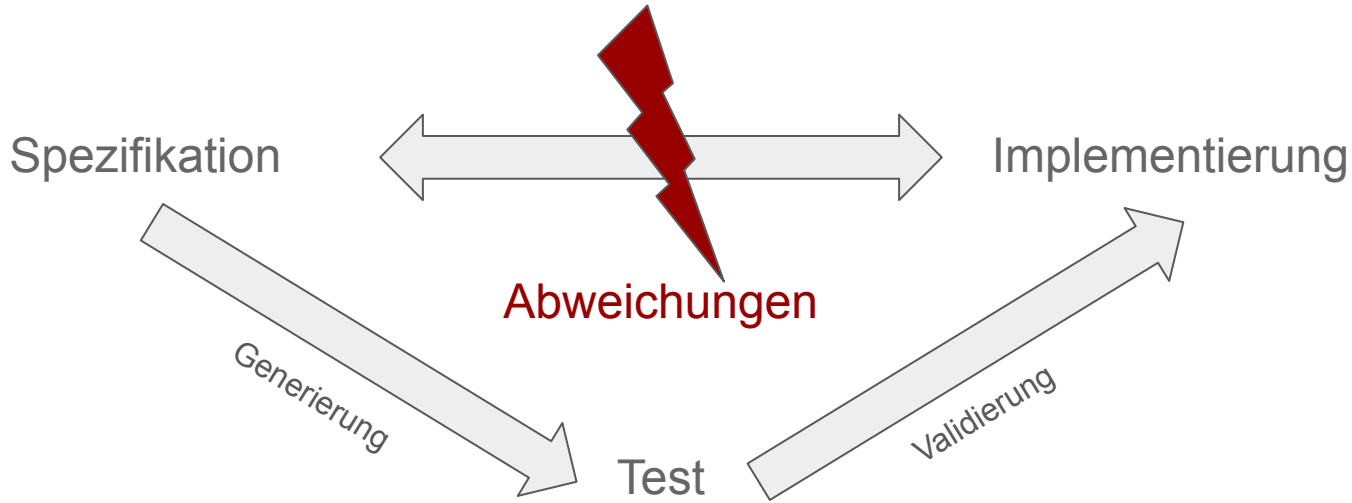


Plantestic

Modellbasierte Software-Entwicklung | SoSe 2020 | Einführung

Motivation


Spezifikation
Implementierung
Abweichungen
Generierung
Test
Validierung

Pipeline

PlantUML
Diagramm

Xtext
Grammatik

QVT™

2x QVT
Transformation

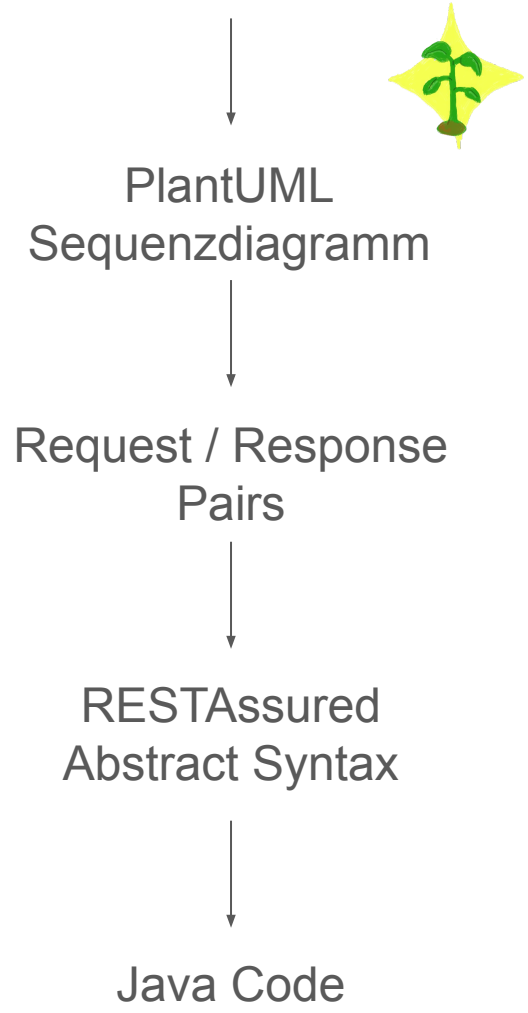
Pipeline

Code-Generierung

REST-Test

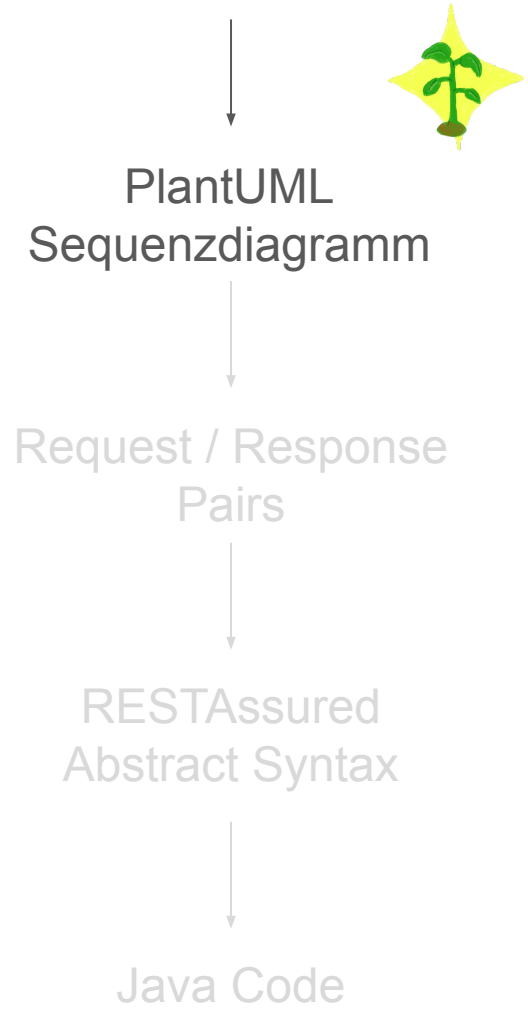
Mock Server

Pipeline


PlantUML
Sequenzdiagramm
Request / Response
Pairs
RESTAssured
Abstract Syntax
Java Code

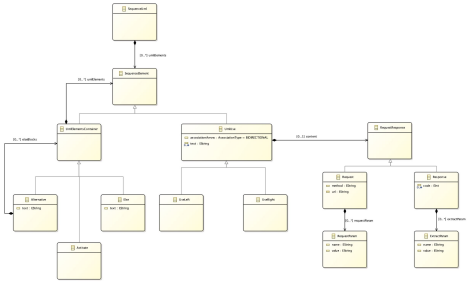
Pipeline

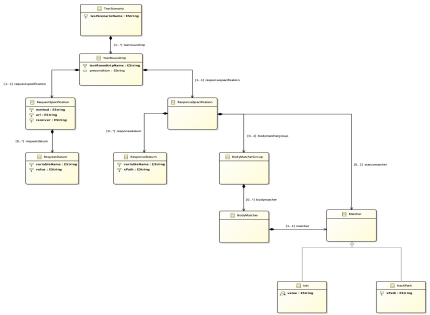
- Parser-Generierung mit XText
- Lesen der PlantUML-DSL aus der Grammatik

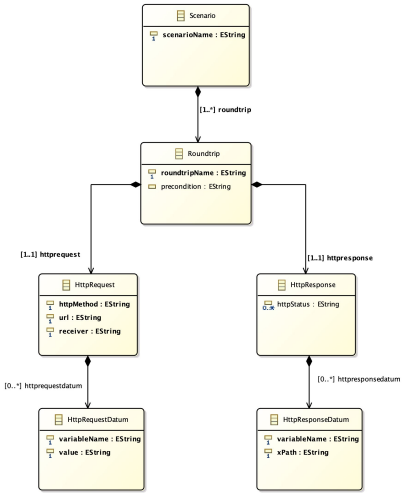


Pipeline

- Model-to-Model-Transformation mit QVTO







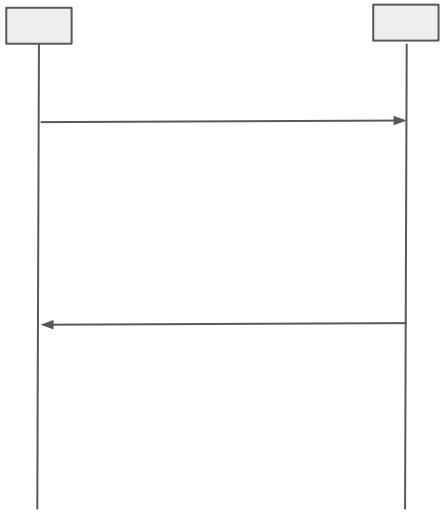
PlantUML
Sequenzdiagramm

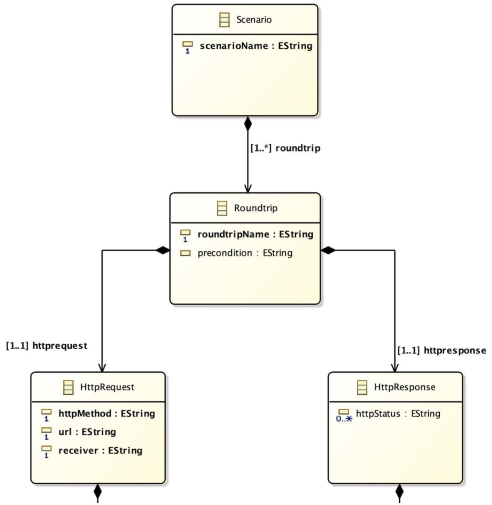
Request / Response
Pairs

RESTAssured
Abstract Syntax

Java Code

Pipeline





PlantUML Sequenzdiagramm

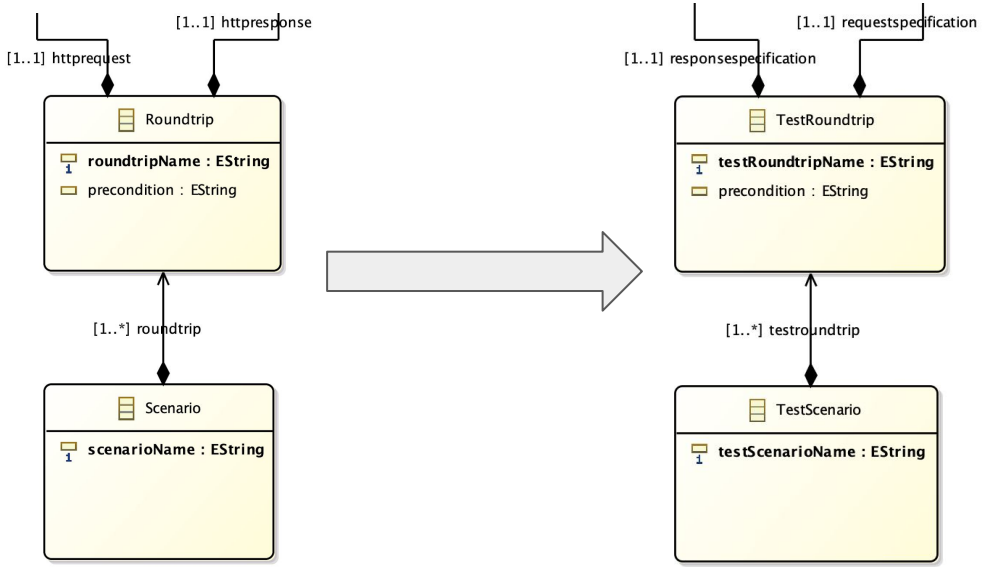
Request / Response Pairs

RESTAssured Abstract Syntax

Java Code

Pipeline

- Model-to-Model-Transformation mit QVTO



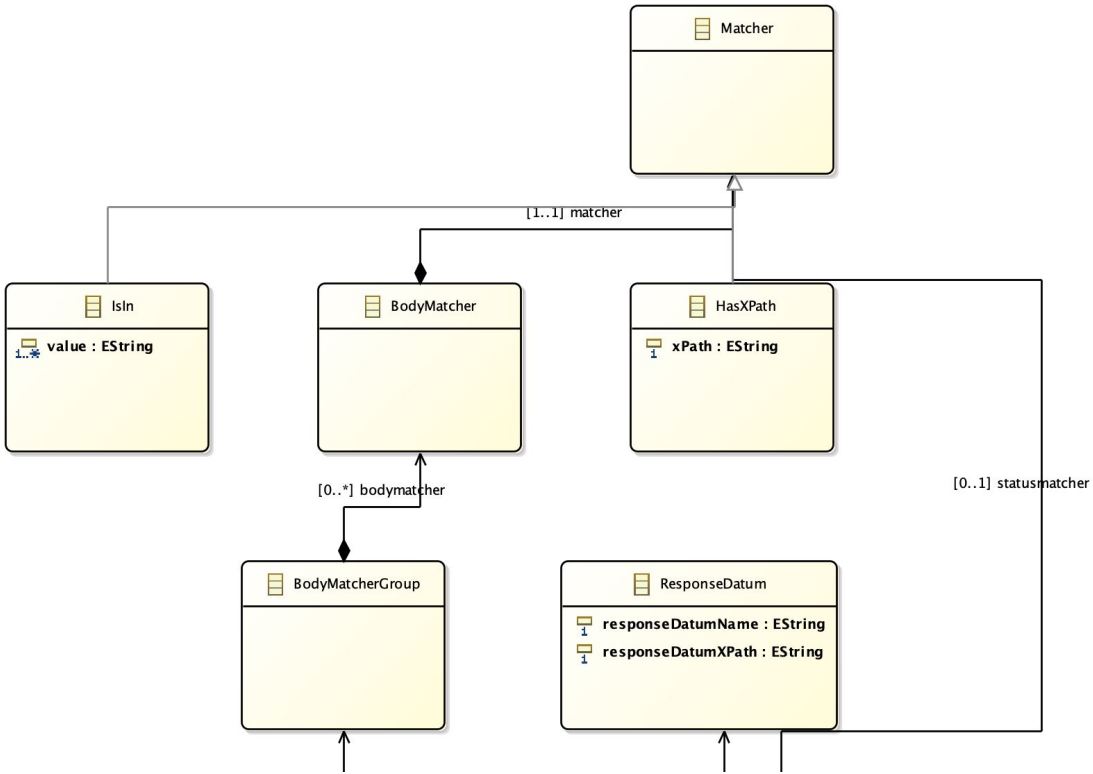
PlantUML Sequenzdiagramm

Request / Response Pairs

RESTAssured Abstract Syntax

Java Code

Pipeline



PlantUML Sequenzdiagramm

Request / Response Pairs

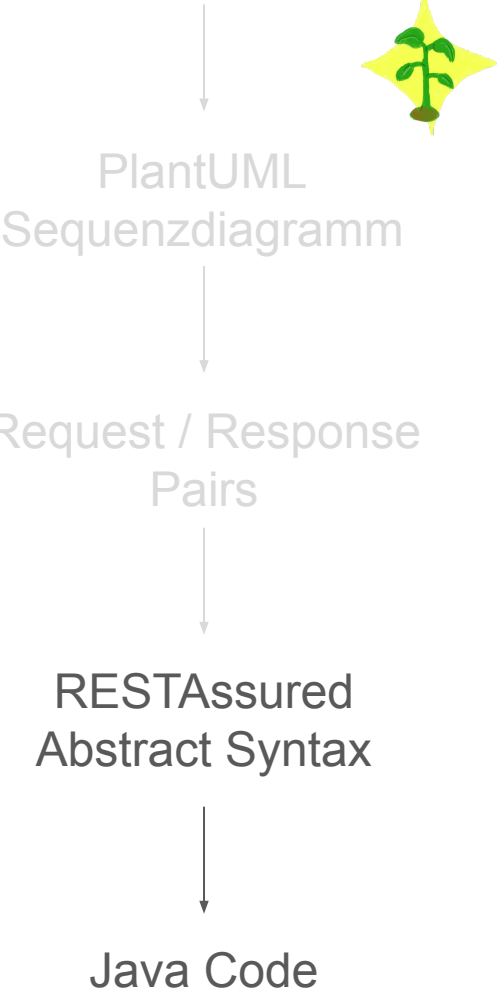
RESTAssured Abstract Syntax

Java Code

Pipeline

- Model-to-Text-Transformation mit Acceleo
- Modularisierter Acceleo Code
- Ausführbarer Code

```
Response [testRoundtrip.testRoundtripName/] = RestAssured.given()
        [testRoundtrip.requestspecification.generateGivenPart()/]
    .when()
        [testRoundtrip.requestspecification.generateWhenPart()/]
    .then()
        [testRoundtrip.responsespecification.generateThenPart()/];
```



Pipeline

```
public void test() throws Exception {

    try {
        String evaluatableAltConditionString = substitutor.replace( source: "${Templating.testCondition} ==
            'SomeValue'");
        assert engine != null;
        Boolean evaluatedCondition = (Boolean) engine.eval(evaluatableAltConditionString);
        if(evaluatedCondition) {
            paramsMap.put("variableName", "value");
            paramsMap.put("variableName2", "value2");

            try {
                Response roundtrip = RestAssured.given() RequestSpecification
                        .auth().basic(substitutor.replace( source: "${TestReceiver.username}"), substitutor.replace
                            ( source: "${TestReceiver.password}")) RequestSpecification
                        .param( parameterName: "variableName", substitutor.replace
                            ( source: "${variableName}")) RequestSpecification
                        .param( parameterName: "variableName2", substitutor.replace
                            ( source: "${variableName2}")) RequestSpecification
                    .when() RequestSpecification
                        .get( path: substitutor.replace( source: "${TestReceiver.path}") + substitutor.replace
                            ( source: "/test/${Templating.id}")) Response
                    .then() ValidatableResponse
                        .assertThat() ValidatableResponse
                            .statusCode(IsIn.isIn(Arrays.asList(200))) ValidatableResponse
                            .and().extract().response();
                paramsMap.put("variableName", roundtrip.jsonPath().getString( path: "itemA"));
                paramsMap.put("variableName2", roundtrip.jsonPath().getString( path: "itemB"));

            } catch (Exception exception) {
```

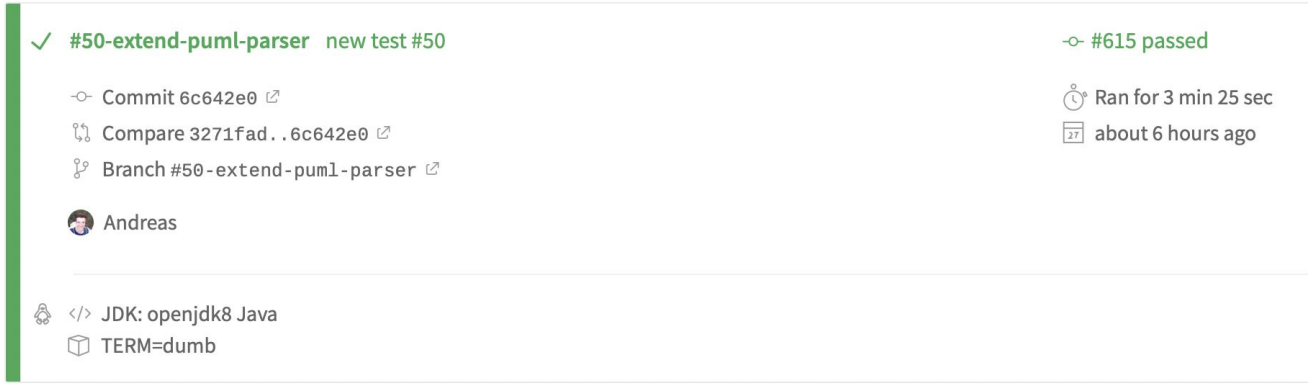
PlantUML Sequenzdiagramm

Request / Response Pairs

RESTAssured Abstract Syntax

Java Code

Pipeline



RESTAssured
Abstract Syntax

Java Code

Java Runtime
Compilation

Test gegen
Mock-Server

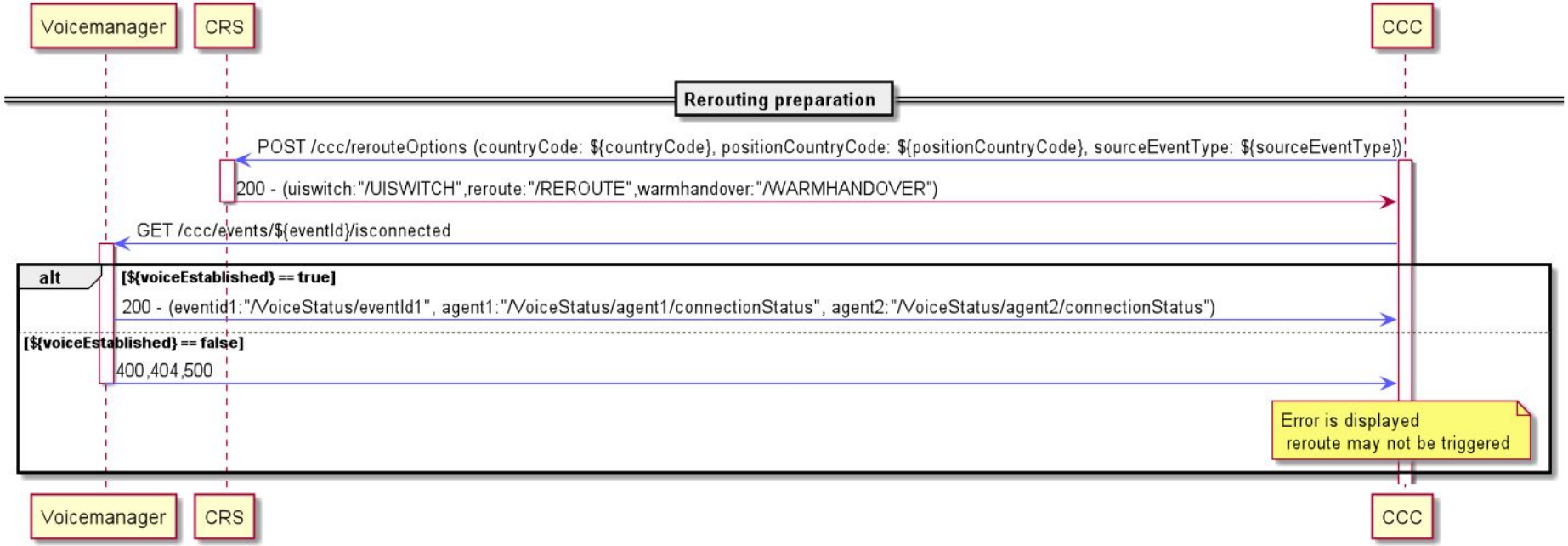
Pipeline

- Allgemeingültig
- Easy to use
- Übergabe von Parametern
- Mächtige Condition-Auswertung

Ausführung ohne Eclipse

```
plantestic>./gradlew run --args="rerouting.puml"
```

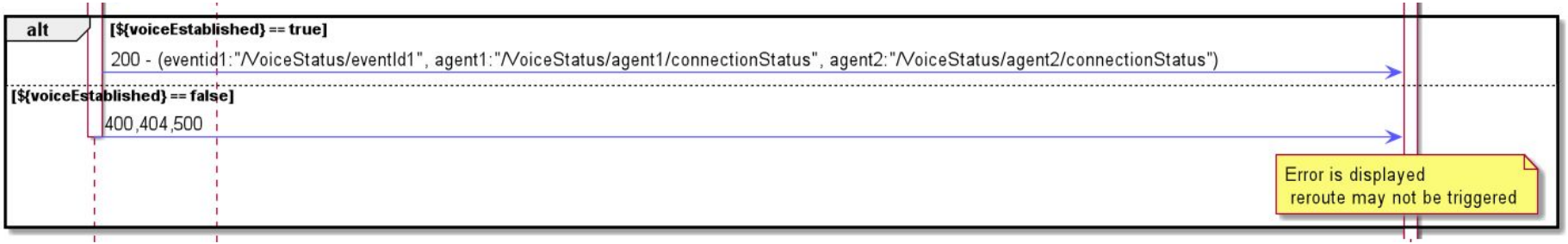
Input-Anforderungen



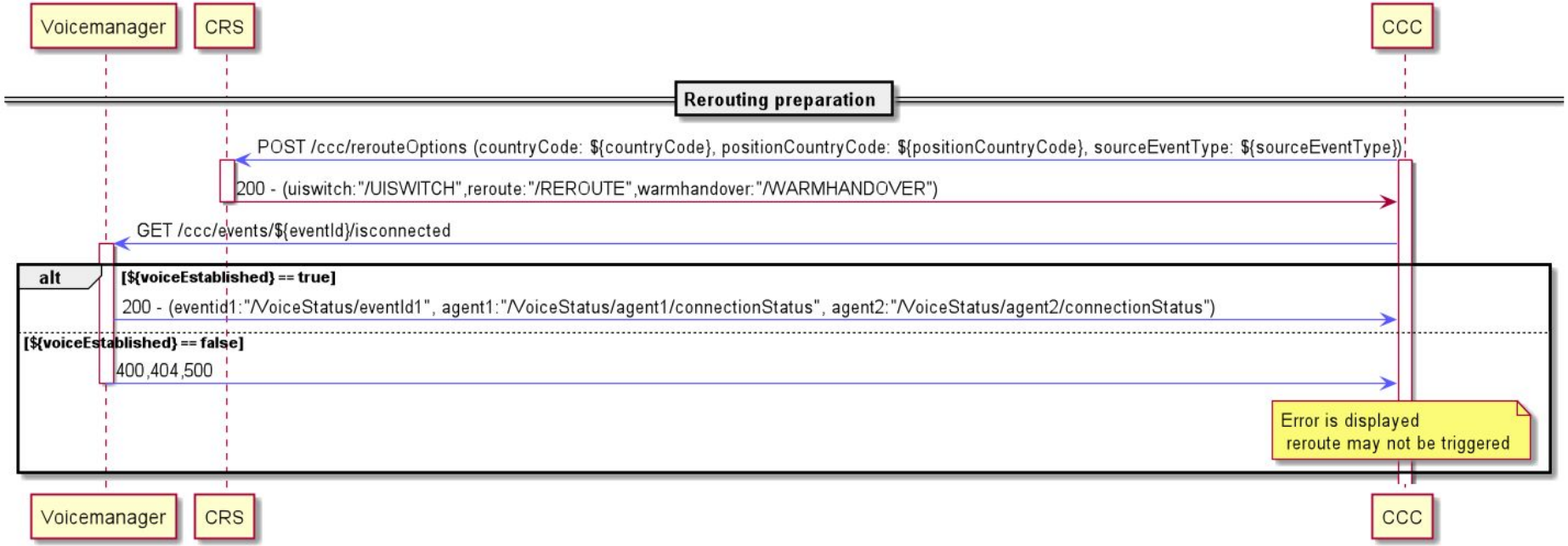
Templating

```
id = "123"
countrycode = "Ger"
positionCountryCode = "Muc"
sourceEventType = "Hallo"

[Voicemanager]
   username = "Test"
   password = "123test"

[CRS]
   username = "Test"
   password = "123test"


[CCC]
   username = "Test"
   password = "123test"
```

JavaScript-Evaluation



Input-Anforderungen



{demo}

Fragen?