

# Event - CustomRestRule

🌐 Sprache: **Deutsch** · [English](#)

## 📖 Event - CustomRestRule

Eigene REST-Endpunkte in Nuclos bereitstellen – mit @Path, Kontext und sauberer Fehlerbehandlung.

REFERENZ

ENTWICKLER

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

### Auf dieser Seite

- [Wann greift die Regel?](#)
- [Struktur & Aufruf](#)
- [Kontext](#)
- [Fehlerbehandlung](#)
- [Verwandte Seiten](#)

## Wann greift die Regel?

Eine **CustomRestRule** stellt eine eigene **REST-Schnittstelle** bereit. Sie wird direkt über den REST-Service aufgerufen, ist unabhängig von Businessobjekten und muss *nicht* über den Regelmanager zugewiesen werden. Die Berechtigung erfolgt im Unterformular „REST-Regeln“ der jeweiligen Benutzergruppe.

## Struktur & Aufruf

Die Klasse implementiert CustomRestRule. Endpunkte werden über @Path-Annotationen definiert. Alle CustomRest-Regeln haben den URL-Präfix /rest/custom/.

```

package example.rest;

import java.util.Arrays;
import java.util.List;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;

import org.nuclos.api.rule.CustomRestRule;

@Path("example")
public class CustomRestWithPathTestRule implements CustomRestRule {

    @GET
    @Path("customers")
    @Produces("application/json")
    public List<Customer> customers() {
        return Arrays.asList(
            new Customer(1, "Mustermann"),
            new Customer(2, "Meier")
        );
    }

    public static class Customer {
        private int id;
        private String name;

        public Customer(final int id, final String name) {
            this.id = id;
            this.name = name;
        }

        public int getId() {
            return id;
        }

        public String getName() {
            return name;
        }
    }
}

```

Aufruf von `http://<host>/nuclos/rest/custom/example/customers` liefert:

```

[
  {
    id: 1,
    name: "Mustermann"
  },
  {
    id: 2,
    name: "Meier"
  }
]

```

## Kontext

Über ein Context-Objekt erhältst du Zugriff auf z.B. den aufrufenden Benutzer oder die Sprache:

```
@Inject
protected Provider<CustomRestContext> context;

// Dazu müssen noch drei weitere Klassen importiert werden:

import javax.inject.Inject;
import javax.inject.Provider;
import org.nuclos.api.context.CustomRestContext;
```

## Fehlerbehandlung

Nicht abgefangene Exceptions meldet der Container (Tomcat) als HTTP 500. Für eigene Antworten die Exceptions mit try/catch behandeln und passende Response-Objekte liefern:

```

package example.rest;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.Response;
import org.nuclos.api.rule.CustomRestRule;
import org.nuclos.api.exception.BusinessException;

@Path("Test")
@Rule(name="HandleException", description="Demonstrates how to handle exception in CustomRest")
public class HandleException implements CustomRestRule {

    @GET
    @Path("response/{parameter}")
    @Produces("application/json")
    public Response parameterTest(@PathParam("parameter") String parameter) {
        try {
            if( this.functionToThrowException(parameter) ) {
                return Response.status(Response.Status.FORBIDDEN).entity(new Message("Dieser User hat keine
Berechtigung für das Ausführen der Regel.")).build();
            }
        } catch (BusinessException e) {
            // may also log error
            return Response.status(Response.Status.BAD_REQUEST).entity(new Message(e.getMessage())).build();
        }

        return Response.status(Response.Status.NO_CONTENT).build();
    }

    private boolean functionToThrowException(String parameter) throws BusinessException {
        if (parameter == null) {
            throw BusinessException("No parameter given");
        }

        return "failure".equals(parameter);
    }

    // Any DTO class for response, will be JSON converted
    public static class Message {

        String message;

        public Message(String message) {
            this.message = message;
        }

        public String getMessage() {
            return message;
        }
    }
}

```

## Verwandte Seiten

### Regeln (serverseitig)

Grundlagen.

[Öffnen](#)

