import java.util.List;

public record WorkflowMetadata(

String category,

List<String> tags,

Trigger trigger,

ExternalRefs externalReferences,

RuntimeConfig runtimeConfig

) {

public record Trigger(String type, String expression) {}

public record ExternalRefs(String gitlabPipelineId, String documentId) {}

public record RuntimeConfig(int timeoutSeconds, int retries) {}

}

import jakarta.persistence.AttributeConverter;

import jakarta.persistence.Converter;

import com.fasterxml.jackson.databind.ObjectMapper;

@Converter(autoApply = false)

public class WorkflowMetadataConverter implements AttributeConverter<WorkflowMetadata, String> {

private static final ObjectMapper objectMapper = new ObjectMapper();

@Override

public String convertToDatabaseColumn(WorkflowMetadata attribute) {

try {

return objectMapper.writeValueAsString(attribute);

} catch (Exception e) {

throw new IllegalArgumentException("Fout bij serialiseren van WorkflowMetadata", e);

}

}

@Override

public WorkflowMetadata convertToEntityAttribute(String dbData) {

try {

return objectMapper.readValue(dbData, WorkflowMetadata.class);

} catch (Exception e) {

throw new IllegalArgumentException("Fout bij deserialiseren van WorkflowMetadata", e);

}

}

}

import jakarta.persistence.\*;

import java.util.UUID;

@Entity

@Table(name = "workflows")

public class Workflow {

@Id

private UUID id;

private String name;

private String status;

@Convert(converter = WorkflowMetadataConverter.class)

@Column(columnDefinition = "jsonb")

private WorkflowMetadata metadata;

// getters/setters of record, afhankelijk van je stijl

}