

Bericht zur technischen Revision zum Prozess KOMM24EinsichtAuskunftBestandsverzeichnis

Technische Revision Ticket: ZER-2783

Prüfer: Piet Schumacher

Version der Berichtsvorlage: a32e2064255010d1995f8d090c2414385908843b. Keine uncommittete Änderungen vorhanden.

Erstellt am: 2026-05-18, 09:44

Gesamturteil

Bei der technischen Revision wurden keine offensichtlich die Sicherheit und den Betrieb der Prozess-Plattform beeinträchtigenden Aspekte festgestellt.

Der Prozess kann daher in Betrieb genommen werden.

Ergänzenden Anmerkungen

- Die Call Activity Bootstrap ruft einen Unterprozess mit der ID m6000526.KOMM24FrameworkBootstrap auf.
- Die Call Activity Bootstrap ruft einen Unterprozess mit der ID m6000526.KOMM24FrameworkBootstrap auf.
- Die Call Activity Datenverarbeitung & Abschluss ruft einen Unterprozess mit der ID m6000526.KOMM24FrameworkSubDatenverarbeitungUndAbschluss auf.
- Die Parameter leistung, region werden aus der URL übernommen. Durch mutwillige Manipulation dieses Parameters kann der Benutzer den Prozess fehlsteuern, z.B. den Antrag bei einem falschen Sachbearbeiter einreichen.

Geprüfte Dateien

Prozessmodell(e)

Mandant: Kommunale_Prozesse Prozessname: KOMM24EinsichtAuskunftBestandsverzeichnis Version: v1.1.0

Prozessdefinitionsdateien:

- KOMM24EinsichtAuskunftBestandsverzeichnis.bpmn20.xml
- KOMM24EinsichtAuskunftBestandsverzeichnis-OAA.bpmn20.xml

Formular(e)

Formularname	Version	Mandant
KOMM24EinsichtAuskunft-StrassenBestandsverzeichnis	v1.1.0	Kommunale_Prozesse
KOMM24EinsichtAuskunft-StrassenBestandsverzeichnisAn-liegensklaerung	v1.1.0	Kommunale_Prozesse

Liste externer Server

- Gesteuert durch die Prozessinstanzvariable udServiceBaseUrl, die als Output der Call Activity m6000526.KOMM24FrameworkBootstrap gesetzt wird.

Geprüfte Aspekte

Erfüllt

Auflistung aller aufgerufenen externen Server in den Formularen ✓

Erklärung: Um feststellen zu können, ob Daten eventuell an Unberechtigte abgeleitet werden, wird im Bericht vermerkt, falls URLs von Servern in den Formularen aufgerufen werden.

Formular verwendet nur erlaubte Proxys ✓

Erklärung: Formulare dürfen nur die erlaubten Proxys 'proxy-ext-01.pzd-svn.de:8080', '10.101.1.127:8080' (= Verwaltungsnetz in processEngineConfig), '10.101.48.10:8080' (in Sachsen) oder '10.127.255.25:8080' (in BaWü) verwenden, da der Internetzugriff (der wiederum nur über ein Proxy möglich ist) von den Hostern getrackt werden können muss.

Keine externen Objekte in Prozessinstanzvariablen ✓

Erklärung: Es dürfen keine externen Objekte in Prozessinstanzvariablen gespeichert werden, da diese Variablen nicht in-memory gespeichert, sondern in eine Datenbank geschrieben werden. Bei diesem Schreibvorgang werden die Objekte serialisiert. Externe Objekte laufen dabei das Risiko zu sehr großen JSON-Blöcken serialisiert zu werden, deren Speichern viele Ressourcen verbrauchen würde. Explizit verboten sind: Logger, org.slf4j.Logger

Prozess verursacht keinen exzessiven Ressourcenverbrauch ✓

Erklärung: Da bei einem exzessiven Ressourcenverbrauch die gesamte Plattform (und damit auch andere Prozesse) ausgebremst werden, muss dies unterbunden werden. Die technische Revision prüft auf offensichtliche Fehler, wie z. B. erkennbare Endlosschleifen.

Prozesse schreiben keine verbotenen Prozessinstanzvariablen ✓

Erklärung: Prozesse können Prozessinstanzvariablen schreiben. Dabei muss aber sichergestellt werden, dass keine Variablen, welche bereits vom Serviceportal reserviert wurden mit anderen/falschen Werten überschrieben werden. Dies bedeutet, dass es grundsätzlich verboten ist, diese Variablen zu setzen. Als reserviert gelten folgende Variablen: `startTime`, `startedBy`, `startedByName`, `startedByUser`, `startedByPersoenlichesKontoId`, `processInstanceId`, `processEngineConfig`, `processName`, `processType`, `prozesssteuerung`, `region`, `regionName`, `ags`, `leistung`, `lebenslage`, `taetigkeit`, `prozessIntegrationConfigurationId`, `parameter`, `startParameter`. Es gibt jedoch Ausnahmen, wenn das System die entsprechende Variable nicht selbstständig setzt und der korrekte Wert verwendet wird (z. B. wenn in einem Hauptprozess eine CallActivity aufgerufen wird und dabei die Variable `startedBy` übergeben wird. In dem Fall muss aber auch der vom Hauptprozess verwendete Wert für `startedBy` unverändert weitergegeben werden.)

Prozessmodelle nutzen nicht zu viele teure Operationen ✓

Erklärung: Es dürfen nur wenige als teuer bekannte Operationen im Prozessmodell durchgeführt werden, da diese die gesamte Plattform ausbremsen könnten. Aktuell sind als teure Operationen bekannt: Lesen von Prozessparametern und Lesen von Informationen von externen Servern.

Auflistung aller aufgerufenen externen Server in den Prozessen ✓

Erklärung: Um feststellen zu können, ob Daten eventuell an Unberechtigte abgeleitet werden, wird im Bericht vermerkt, falls URLs von Servern in den Prozessen aufgerufen werden.

Die Klasse `ClientUtil` wird nicht verwendet ✓

Erklärung: Die Klasse `ClientUtil` ist nicht Bestandteil der öffentlichen API für Prozessentwickler. Mit ihr könnte ein tiefgreifender Zugriff auf die Interna des Serviceportal durchgeführt werden. Darum muss die Verwendung bei der technischen Revision abgelehnt werden. Ausnahme: Die Funktion `ClientUtil.getClientUtil().getFormOperationClient().getFormAndMappingWithoutContentById` darf genutzt werden.

Dynamische Formular-IDs müssen Einschränkungen unterliegen ✓

Erklärung: Wird die ID eines Formulars nicht statisch angegeben, sondern dynamisch ermittelt, so muss sichergestellt sein, dass sich die Dynamik entweder auf eine feste kleine Positivliste beschränkt (z. B. es wird

zwischen 2 festen Formularen gewählt) oder es muss im Prozessmodell ein fester Namespace für diese Formulare definiert werden, damit die Zusammengehörigkeit zwischen Formular und Prozess klar festgestellt werden kann.

Es werden keine globalen Listeners definiert ✓

Erklärung: Es dürfen keine globalen Listeners verwendet werden, da diese für alle Prozesse der Serviceplattform gelten würden.

Es wird vor unerwarteten Imports gewarnt ✓

Erklärung: Prozessmodellierer können auf die Klassen aller Libraries zugreifen, die von der Prozessplattform verwendet werden. Das ist potentiell problematisch, denn es können im Rahmen der Weiterentwicklung der Plattform ohne Ankündigungen Änderungen an diesen Klassen vorgenommen werden, was zu Fehlern in Prozessen führen kann. Das technische Revision Team pflegt eine Liste an "erwarteten Imports" für die das Plattformteam Stabilität beim Änderungen gewährleistet. Imports, die sich nicht in dieser Liste befinden, werden als "Ergänzende Anmerkung" in den Bericht zur technischen Revision aufgenommen.

Formular-ID ist konsistent zum Namen des Formulars im Admincenter ✓

Erklärung: Die ID des Formulars im Admincenter muss der ID des Formulars in der Formulardefinitionsdatei (die JSON-Datei) entsprechen, da ansonsten eine fehlerfreie Zuordnung innerhalb der Serviceplattform nicht mehr möglich ist.

Formulare dürfen keine CSS Styles definieren ✓

Erklärung: Formulare dürfen keine eigenen CSS Klassen schreiben, da diese potentiell die Klassen des Serviceportals überschreiben können. Das Verwenden vom 'style' Attribut in einem HTML-Tag ist aus Sicht der technischen Revision okay, werden aber von der Serviceplattform beim Rendern des Formulars entfernt (und sind dadurch sinnlos). Das Verwenden von bereits existierenden CSS-Klassen ist ebenfalls okay - es wird allerdings seitens des Serviceportals keine Stabilität garantiert. D.h. die Definition der Klasse kann sich während der Entwicklung ändern und sogar komplett entfernt werden. Darum wird in einem solchen Fall eine ergänzende Anmerkung geschrieben.

Formulare schreiben keine verbotenen Prozessinstanzvariablen ✓

Erklärung: Formulare können über die target Attribute von Feldern (oder vom Formular selber) Prozessinstanzvariablen schreiben. Dabei muss aber sichergestellt werden, dass keine Variablen überschrieben werden, welche bereits vom Serviceportal belegt sind, da diese auch intern verwendet werden.

Logger dürfen keine personenbezogenen Daten oder zu viele Einträge schreiben ✓

Erklärung: Die verwendeten Log-Statements dürfen keine personenbezogenen Daten loggen, da Logs von vielen verschiedenen Stellen eingesehen werden können, die ggf. keinen Zugriff auf personenbezogene Daten haben sollten. Zusätzlich sollte die Anzahl der Log-Einträge nicht zu groß werden, da diese Schreibvorgänge in der Datenbank verursachen. (Als Größenordnung sollten 5 Log-Einträge bei einem erfolgreichen Prozessablauf nicht überschrieben werden.)

Logger müssen die korrekten Log-Level nutzen ✓

Erklärung: Die Log-Level "ERROR" und "FATAL" sind für Lognachrichten vorgesehen, die die Funktion der Serviceplattform als Ganzes beeinträchtigen und sind daher nicht für Prozesse vorgesehen, da deren Einflussbereich nicht über einen Prozess hinaus gehen sollte. Es existieren auch noch die Levels "DEBUG" und "TRACE", welche aber nicht von Prozessmodellierern über das Prozesslog-Tool eingesehen werden können. Deren Nutzung ist nicht empfohlen (da unnötig Logs geschrieben werden, welche die Festplatte der Plattform belasten) aber nicht verboten. Prinzipiell sind für Ereignisse aus Prozessen heraus nur die Levels "INFO" und "WARN" vorgesehen.

Logger nutzen korrektes Namensschema ✓

Erklärung: Logger müssen das korrekte Namensschema `de.seitenbau.serviceportal.prozess.PROZESSNAME_HIER` nutzen, damit alle Log-Nachrichten korrekt einem Prozess zugeordnet werden können. Außerdem können die Log-Nachrichten im Admincenter / über die Schnittstelle nur dann eingelesen werden, wenn sie den korrekten Präfix verwenden.

Prozessmodelle verwenden nur erlaubte Proxys ✓

Erklärung: Prozessmodelle dürfen nur die erlaubten Proxys 'proxy-ext-01.pzd-svn.de:8080', '10.101.1.127:8080' (= Verwaltungsnetz in processEngineConfig), '10.101.48.10:8080' (in Sachsen) oder '10.127.255.25:8080' (in BaWü) verwenden, da der Internetzugriff (der wiederum nur über ein Proxy möglich ist) von den Hostern getrackt werden können muss.

Servicetasks verwenden nur erlaubte Proxys ✓

Erklärung: Servicetasks dürfen nur die erlaubten Proxys 'proxy-ext-01.pzd-svn.de:8080', '10.101.1.127:8080' (= Verwaltungsnetz in processEngineConfig), '10.101.48.10:8080' (in Sachsen) oder '10.127.255.25:8080' (in BaWü) verwenden, da der Internetzugriff (der wiederum nur über ein Proxy möglich ist) von den Hostern getrackt werden können muss.

User Tasks - Akteure ✓

Erklärung: User Tasks in Prozessmodellen können unterschiedliche Akteure (Assignees) haben, die Zugriff auf Daten aus dem Prozess bekommen können. Es muss sichergestellt werden, dass dieser Zugriff nur berechtigten Personen ermöglicht wird.

Verwendung von Pools, Lanes und Subprozessen ✓

Erklärung: Pools und Subprozesse erhöhen die Komplexität der Prozessmodelle und erschweren die technische Revision dieser, daher ist die Verwendung dieser nicht empfohlen. Falls die Notwendigkeit für einen Subprozess besteht, wird empfohlen dies durch eine CallActivity zu realisieren.

Übernahme unsicherer Informationen ✓

Erklärung: Es werden keine in der URL definierten Informationen (z.B. leistungId) außer der Regionalisierung im Prozess so weiterverwendet, dass der Prozess fehlgesteuert wird. Dadurch soll verhindert werden, dass der Prozess über manipulierte Parameter fehlgesteuert oder missbraucht werden kann. Eine typische Fehlsteuerung besteht darin, dass Informationen an eine unpassende Stelle geschickt werden.