

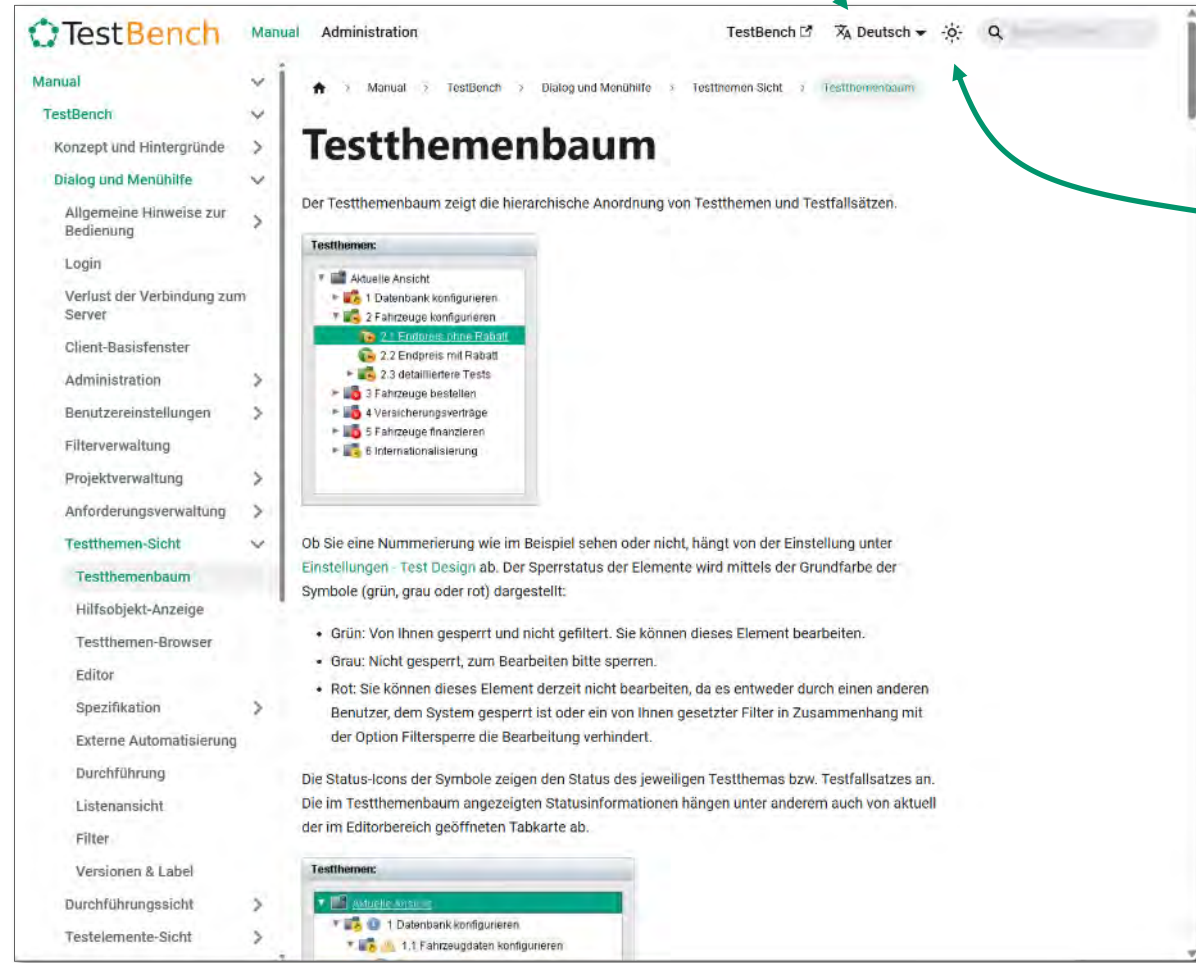
Neuerungen in Version 4.0

Patchlevel 4.0.39 BETA – 10.02.2026

Online Hilfe

- Die Online Hilfe wird im Browser geöffnet und kann unabhängig vom TestBench Client geöffnet und gelesen werden.
- Neue Funktionen
 - Umschalten der Sprachen innerhalb der Hilfe
 - Light und Dark Mode
 - Umfassende Volltextsuche
 - Glossar

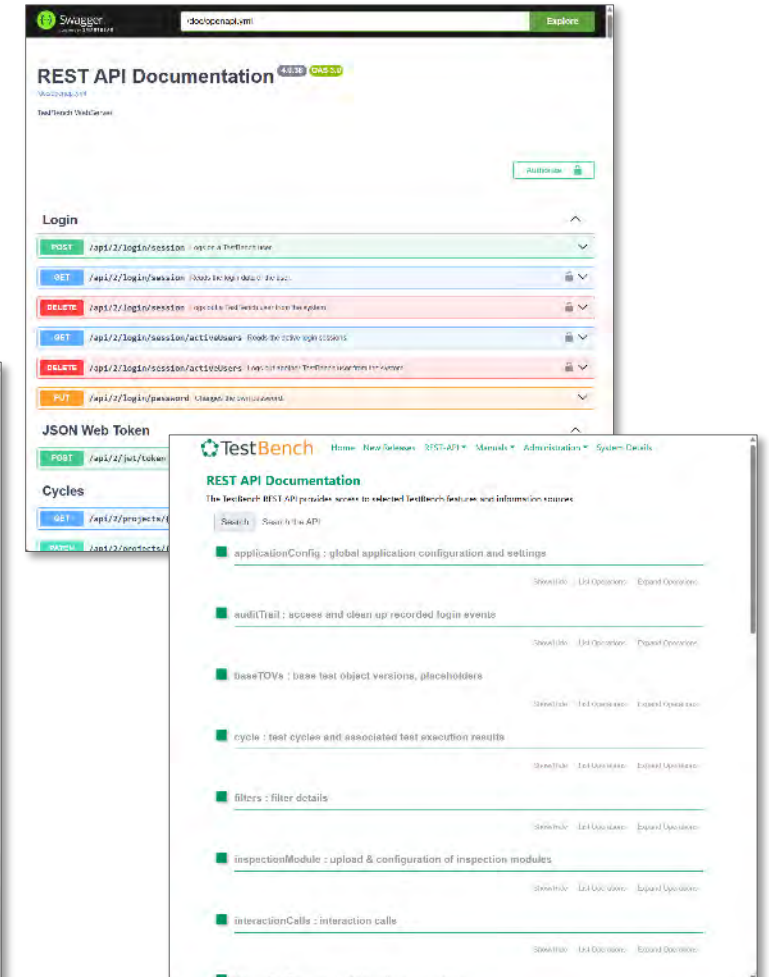
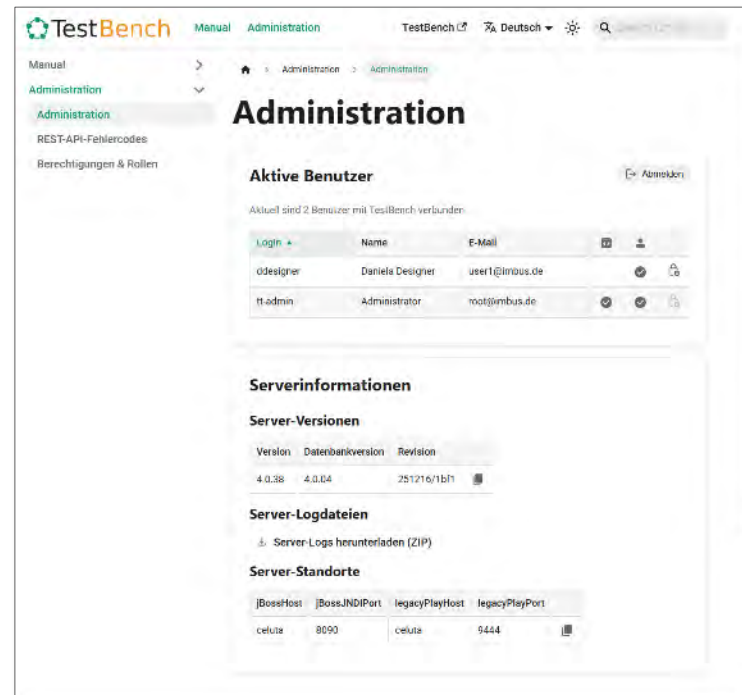
Sprache umschalten



*Dark Mode
Light Mode*

Administrationsseiten

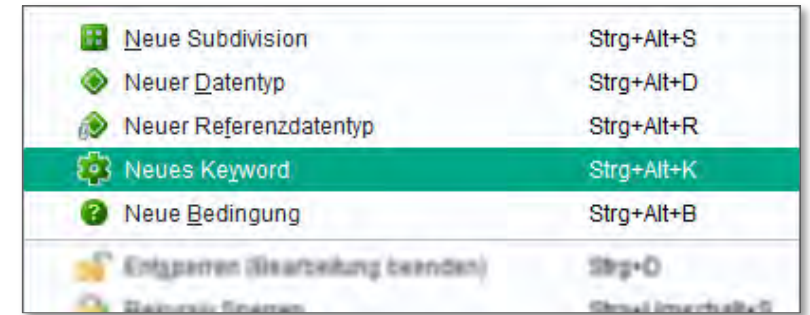
- Die Administrationsseiten im Backend enthalten wichtige Informationen:
 - Angemeldete Benutzer, Systemaccounts
 - Informationen zum Server
 - Server-Logdateien
 - Server-Versionen
 - Lizenzinformationen
- Dokumentation des REST-API
- Neuerungen der Versionen



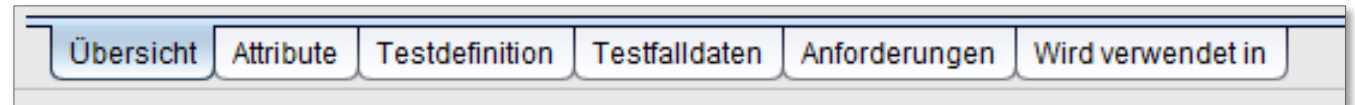
TestBench Nomenklatur

Anpassung an die Norm ISO 29119-5 Keyword-Driven Testing

Bisheriger Begriff	Neuer Begriff
Interaktion	Keyword
Keyword	Marker
Testfälle	Testfalldaten
Definition	Testfalldefinition



Ausschnitt Kontextmenü der Testelemente



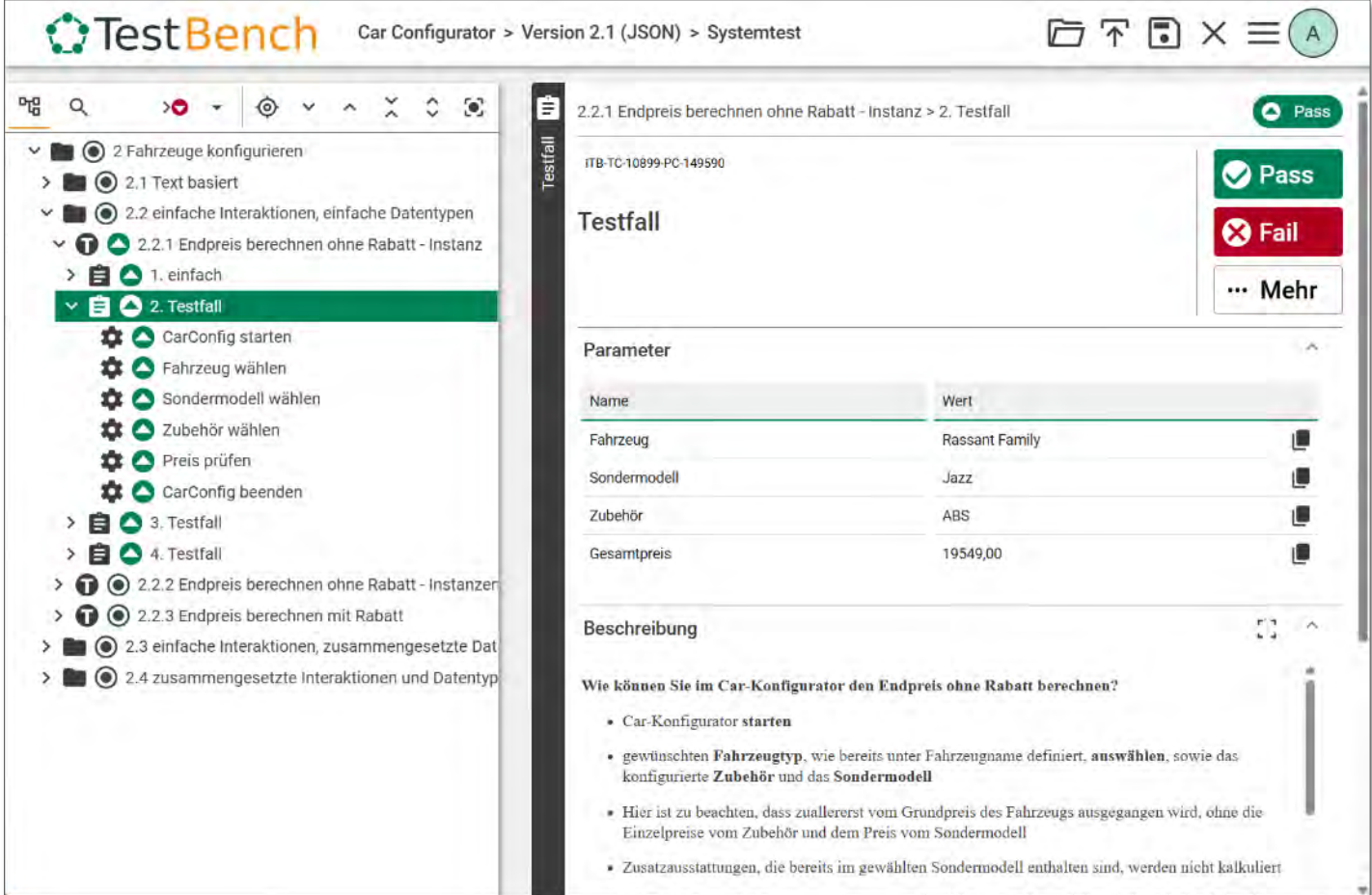
Tabkarten eines Testfallsatzes

In technischen Exportformaten, z.B. dem XML-Komplettbericht oder dem Projektexport/-Import, wurden keine Änderungen vorgenommen, so dass die Rückwärtskompatibilität erhalten bleibt.

Aus Java iTORX wird Web iTORX

Bestehender iTORX wird zu einer Progressive Web App (PWA)

- Startet im Web-Browser
- Öffnet Archive oder wird aus dem Client aufgerufen
- PWA = lässt sich wie eine App über den Browser installieren, aktualisiert sich trotzdem automatisch



The screenshot displays the TestBench web interface. The top navigation bar shows 'TestBench Car Configurator > Version 2.1 (JSON) > Systemtest'. The main content area is titled '2.2.1 Endpreis berechnen ohne Rabatt - Instanz > 2. Testfall' and shows a 'Pass' status. A table of parameters is visible:

Name	Wert
Fahrzeug	Rasant Family
Sondermodell	Jazz
Zubehör	ABS
Gesamtpreis	19549,00

Below the table, a description section titled 'Beschreibung' provides instructions on how to calculate the end price without discount in the car configurator.

Wie können Sie im Car-Konfigurator den Endpreis ohne Rabatt berechnen?

- Car-Konfigurator **starten**
- gewünschten **Fahrzeugtyp**, wie bereits unter Fahrzeugname definiert, **auswählen**, sowie das konfigurierte **Zubehör** und das **Sondermodell**
- Hier ist zu beachten, dass zuallererst vom Grundpreis des Fahrzeugs ausgegangen wird, ohne die Einzelpreise vom Zubehör und dem Preis vom Sondermodell
- Zusatzausstattungen, die bereits im gewählten Sondermodell enthalten sind, werden nicht kalkuliert

Zusammengesetzte Datentypen in Web iTORX (1/2) TestBench by imbus

Werte der zusammengesetzten Repräsentanten werden angezeigt

Ausgangssituation:

Namen von Repräsentanten zusammengesetzter Datentypen werden auf der übergeordneten Ebene nicht angezeigt, sondern die detaillierten Werte der Unterdatentypen.

Zusammengesetzter Datentyp in TestBench



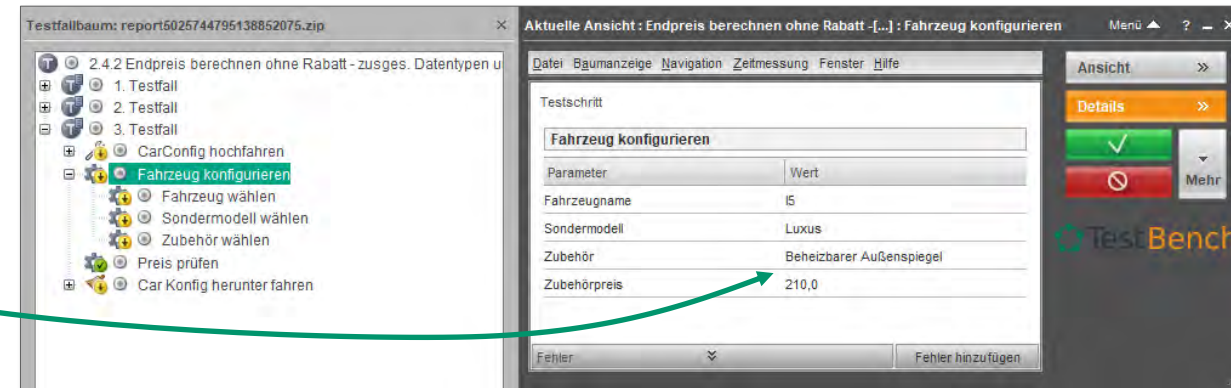
Äquivalenzklassen und Repräsentanten		
Name	Beschreibung	
1 <gültig>	gültiges Zubehör	
2 <ungültig>	ungültiges Zubehör	
*		

Name	Zubehör	Zubehörpreis
1 -	-	0,0
2 ABS mit Preis	ABS	990,0
3 Radio mit Preis	Radio	470,0
4 Lederlenkrad mit Preis	Lederlenkrad	360,0
5 Beheizbarer Außenspiegel mit Preis	Beheizbarer Außenspiegel	210,0
6 Fensterheber hinten mit Preis	Fensterheber hinten	190,0

Verwendung in Testsequenz



- Endpreis berechnen ohne Rabatt - zusges. Datentypen und Interaktionen
 - CarConfig hochfahren ()
 - CarConfig starten ()
 - Basisdaten laden ()
 - Fahrzeug konfigurieren (Fahrzeugkonfiguration.Fahrzeug.Fahrzeugname, Fahrzeugkonfiguration.Sondermodell.Sondermodell, Fahrzeugkonfiguration.Zubehörteile)
 - Fahrzeug wählen (Fahrzeugkonfiguration.Fahrzeug.Fahrzeugname)
 - Sondermodell wählen (Fahrzeugkonfiguration.Sondermodell.Sondermodell)
 - Zubehör wählen (Fahrzeugkonfiguration.Zubehörteile.Zubehör)
 - Preis prüfen (Fahrzeugkonfiguration.Gesamtpreis)
 - Car Konfig herunter fahren ()
 - Vom System abmelden ()
 - CarConfig beenden ()



Testfallbaum: report5025744795138852075.zip

Aktuelle Ansicht: Endpreis berechnen ohne Rabatt - [...] : Fahrzeug konfigurieren

Testschritt: Fahrzeug konfigurieren

Parameter	Wert
Fahrzeugname	I5
Sondermodell	Luxus
Zubehör	Beheizbarer Außenspiegel
Zubehörpreis	210,0

Detaillierte Werte in iTORX

Zusammengesetzte Datentypen in Web iTORX (2/2) TestBench by imbus

Werte der zusammengesetzten Repräsentanten werden angezeigt

Neu:

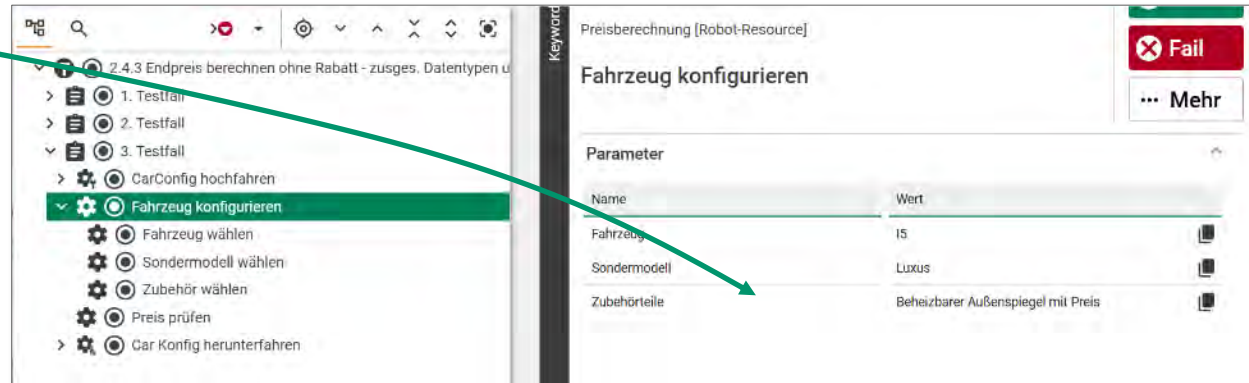
Web iTORX zeigt die Namen von Repräsentanten zusammengesetzter Datentypen auf der übergeordneten Ebene an.

Zusammengesetzter Datentyp in TestBench

Zubehöerteile				
Äquivalenzklassen und Repräsentanten				
	Name	Beschreibung	Name	Zubehörpreis
1	<gültig>	gültiges Zubehör	-	0,0
2	<ungültig>	ungültiges Zubehör	2 ABS mit Preis	990,0
*			3 Radio mit Preis	470,0
			4 Lederlenkrad mit Preis	360,0
			5 Beheizbarer Außenspiegel mit Preis	210,0
			6 Fensterheber hinten mit Preis	190,0

Verwendung in Testsequenz

Testfall
Endpreis berechnen ohne Rabatt - zusges. Datentypen und Interaktionen
CarConfig hochfahren ()
CarConfig starten ()
Basisdaten laden ()
Fahrzeug konfigurieren (Fahrzeugkonfiguration.Fahrzeug.Fahrzeugname, Fahrzeugkonfiguration.Sondermodell.Sondermodell, Fahrzeugkonfiguration.Zubehöerteil)
Fahrzeug wählen (Fahrzeugkonfiguration.Fahrzeug.Fahrzeugname)
Sondermodell wählen (Fahrzeugkonfiguration.Sondermodell.Sondermodell)
Zubehör wählen (Fahrzeugkonfiguration.Zubehöerteil.Zubehör)
Preis prüfen (Fahrzeugkonfiguration.Gesamtpreis)
Car Konfig herunter fahren ()
Vom System abmelden ()
CarConfig beenden ()



Preisberechnung [Robot-Resource]

Fahrzeug konfigurieren

Parameter

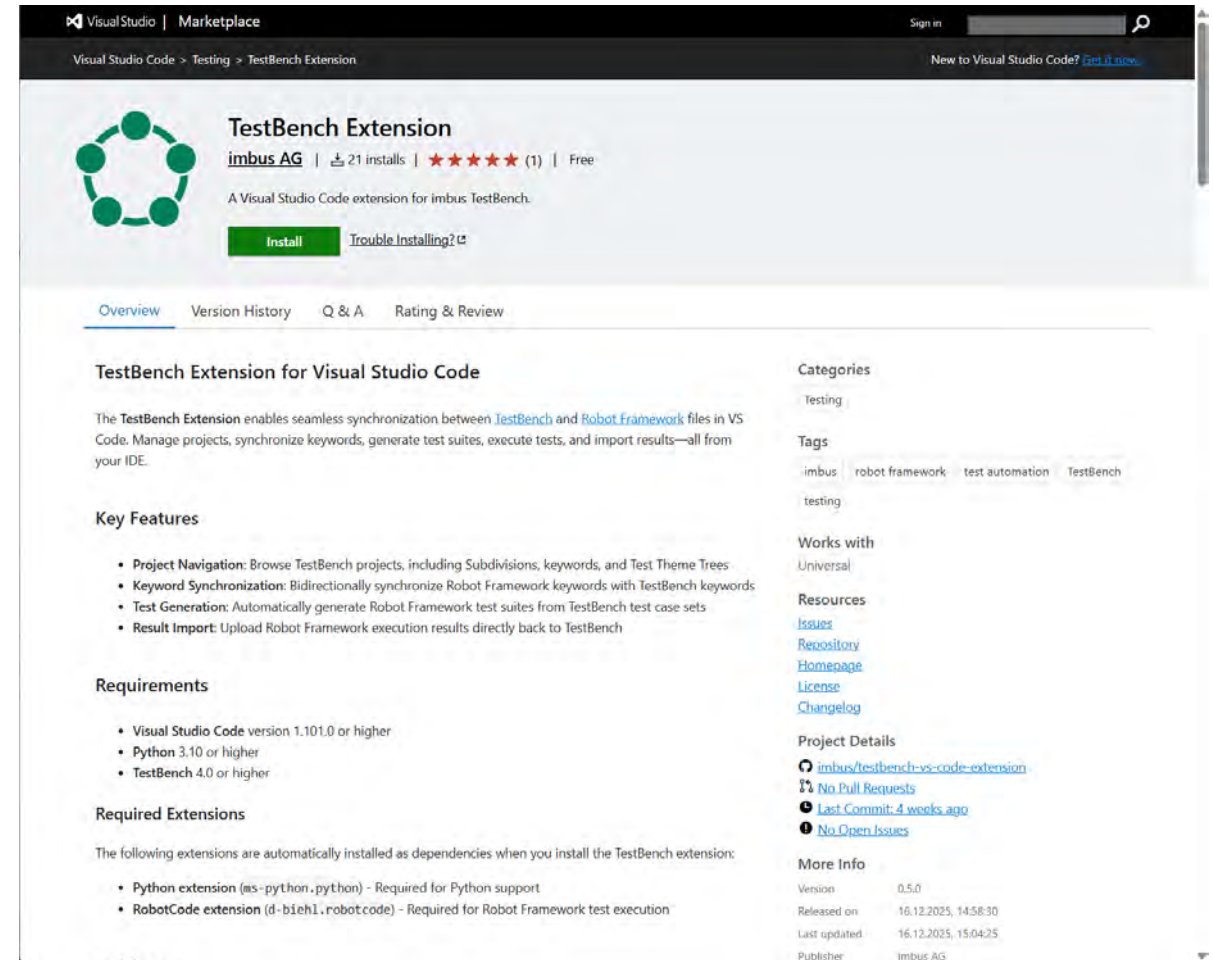
Name	Wert
Fahrzeug	IS
Sondermodell	Luxus
Zubehöerteile	Beheizbarer Außenspiegel mit Preis

Name des Repräsentanten

Visual Studio Code Extension

Spezifikation = Automatisierung

- Fachliche Spezifikation von Keyword-Driven Tests und deren Testdaten in TestBench
- Technische Automatisierung, Codierung und Debugging der Keywords in VS Code Extension
- Fachtester arbeiten ausschließlich in TestBench
- Testautomatisierer arbeiten ausschließlich in VS Code



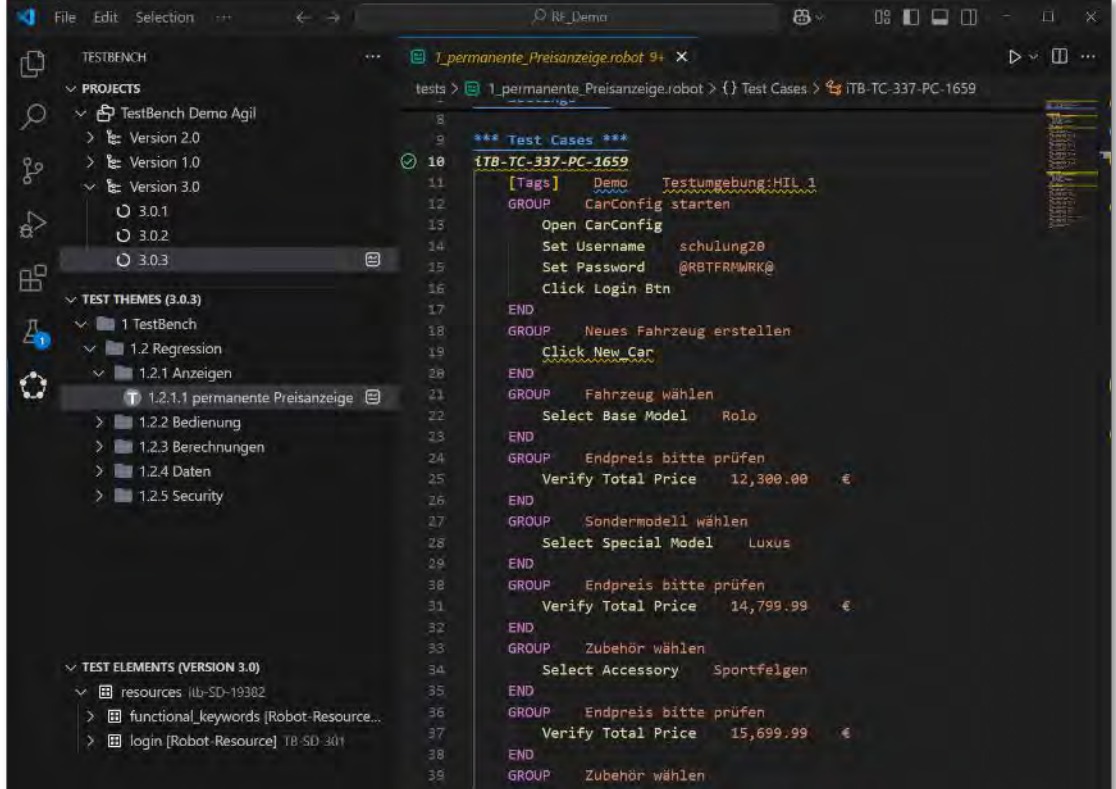
Visual Studio Marketplace:

<https://marketplace.visualstudio.com/items?itemName=imbus.testbench-extension>

Visual Studio Code Extension

Integration von Robot Framework basierter Testautomatisierung

- TestBench Tests werden als Robot Framework Tests generiert
- Keywords werden in VS Code implementiert
- neue Keywords werden mit TestBench synchronisiert (Keywords können auf beiden Seiten angelegt werden)
- Namen und Beschreibungen werden mit TestBench synchronisiert

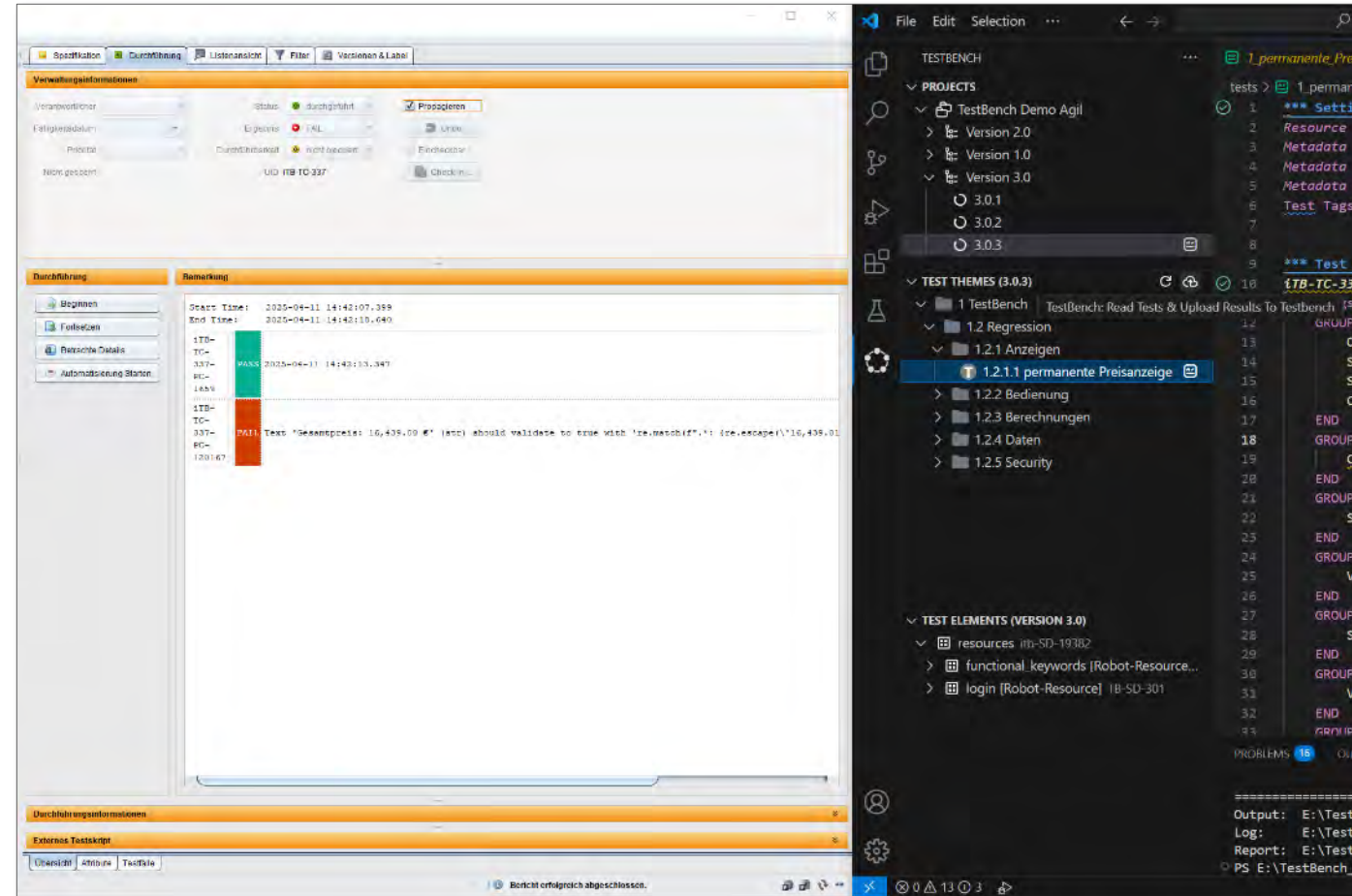


```
*** Test Cases ***
ITB-TC-337-PC-1659
  [Tags] Demo Testumgebung:HIL 1
  GROUP CarConfig starten
  Open CarConfig
  Set Username schulung20
  Set Password @RBTFRMWRK@
  Click Login Btn
  END
  GROUP Neues Fahrzeug erstellen
  Click New Car
  END
  GROUP Fahrzeug wählen
  Select Base Model Rolo
  END
  GROUP Endpreis bitte prüfen
  Verify Total Price 12,300.00 €
  END
  GROUP Sondermodell wählen
  Select Special Model Luxus
  END
  GROUP Endpreis bitte prüfen
  Verify Total Price 14,799.99 €
  END
  GROUP Zubehör wählen
  Select Accessory Sportfelgen
  END
  GROUP Endpreis bitte prüfen
  Verify Total Price 15,699.99 €
  END
  GROUP Zubehör wählen
```

VS Code Extension – weitere Funktionen

Testergebnisse direkt zurück speichern

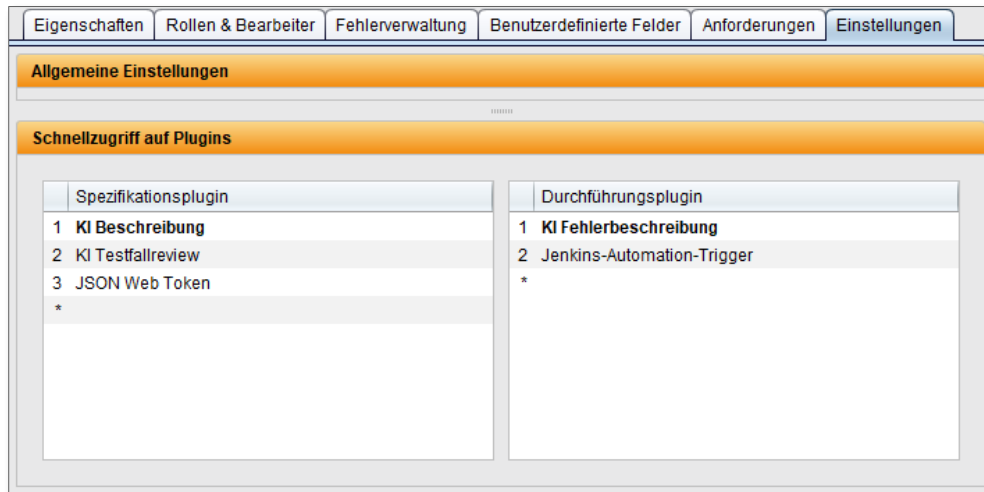
- Tests können in VS Code durchgeführt(*) werden
- Testergebnisse können direkt in TestBench zurückgespeichert werden



(*) über freie RobotCode Extension

Schnellzugriff auf Plugins

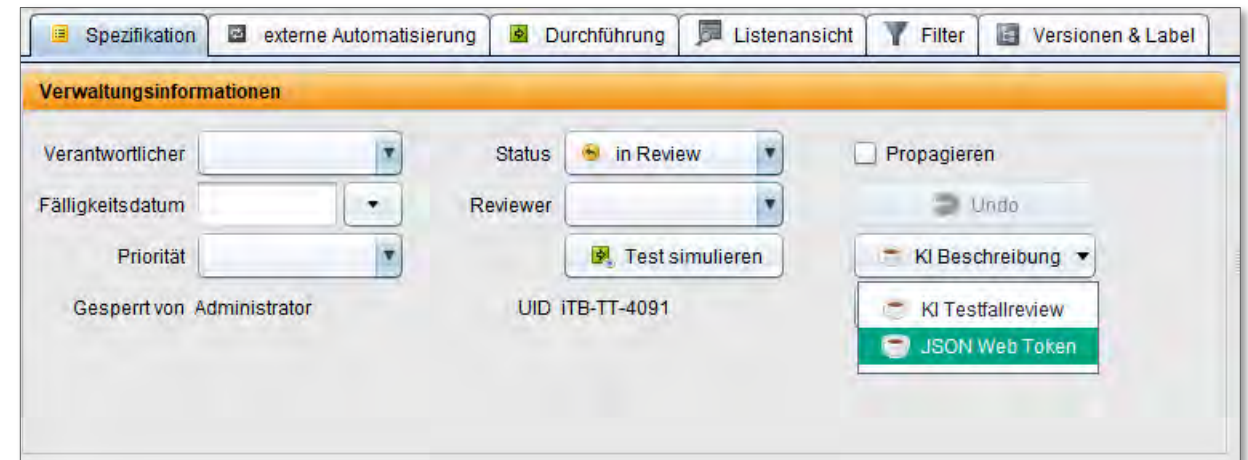
- Die mit dem letzten Release eingeführten Schaltflächen zum Aufrufen von Plugins wurden so erweitert, dass hinter jeder Schaltfläche mehrere Plugins konfiguriert werden können:



Definition der Schaltflächen



Schaltflächen der Durchführung



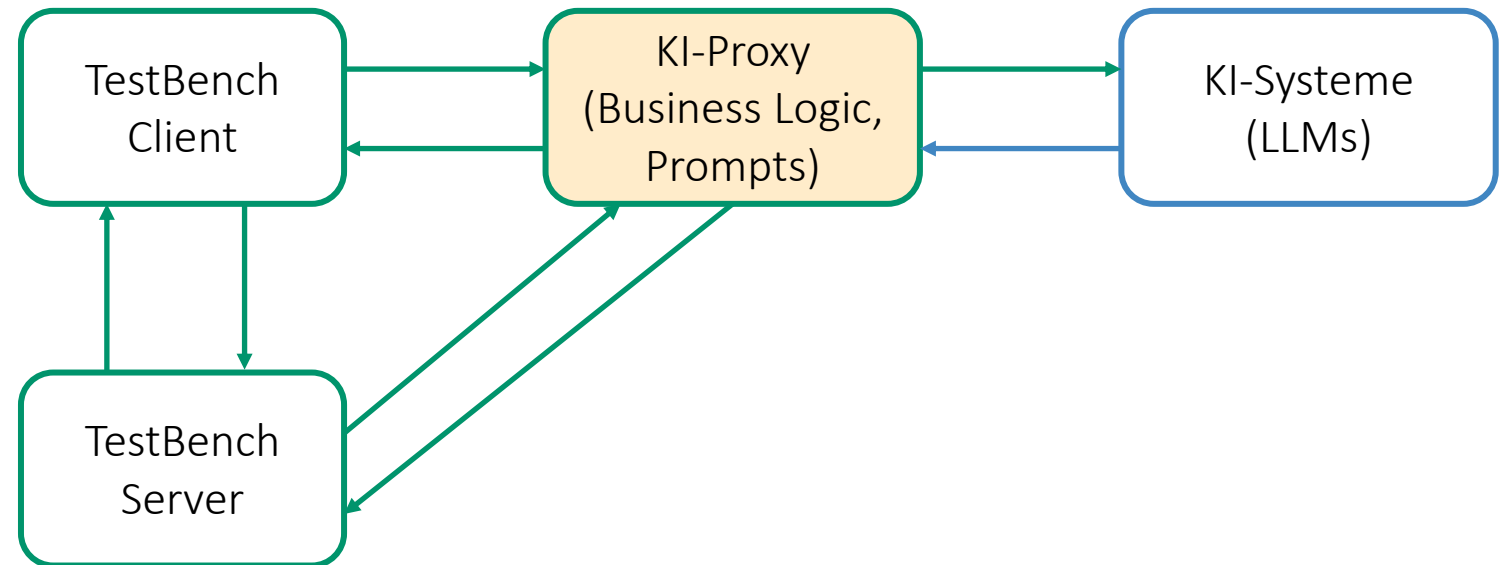
Schaltflächen der Spezifikation

TestBench KI-Assistenten

Aufbau des KI-Service (KI-Proxy)

Starker Fokus auf Flexibilität und Anpassbarkeit

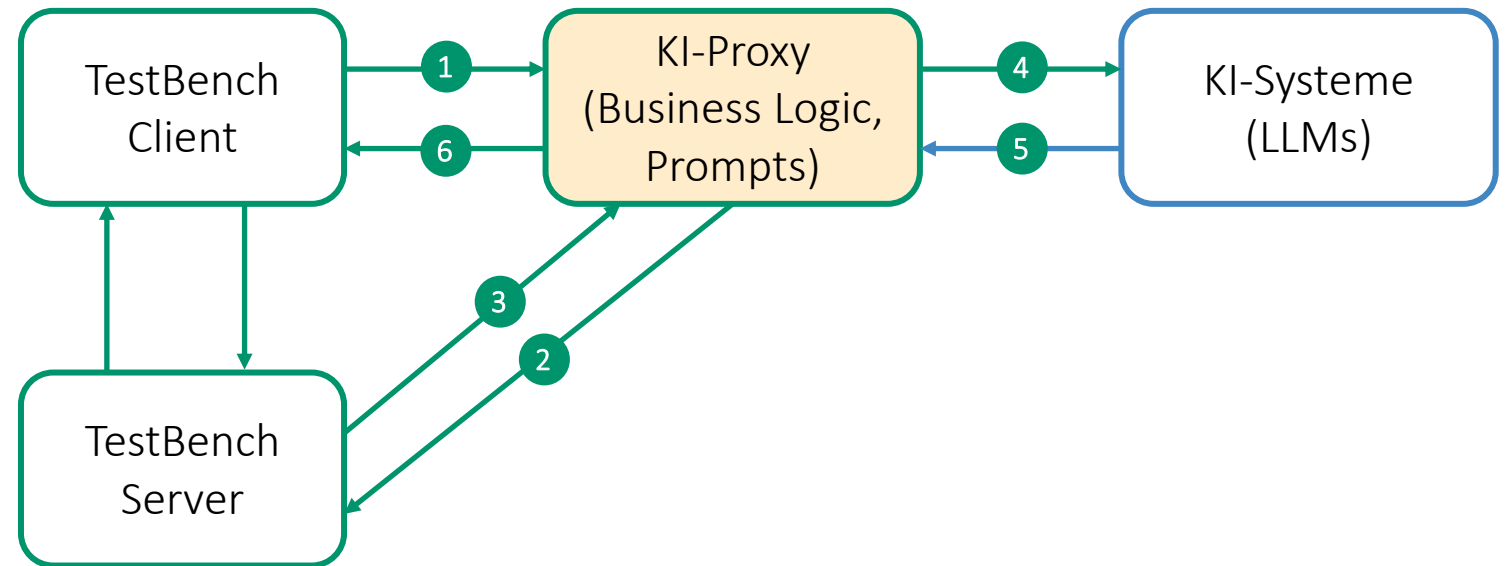
- Prompts modular anpassbar und konfigurierbar je nach Kunden- / Projektbedürfnissen
- Quelloffen
- An verschiedene KI-Systeme anbindbar: lokale Systeme, private Clouds, öffentliche Clouds
- Standard:
öffentliches OpenAI ChatGPT



TestBench KI-Assistenten

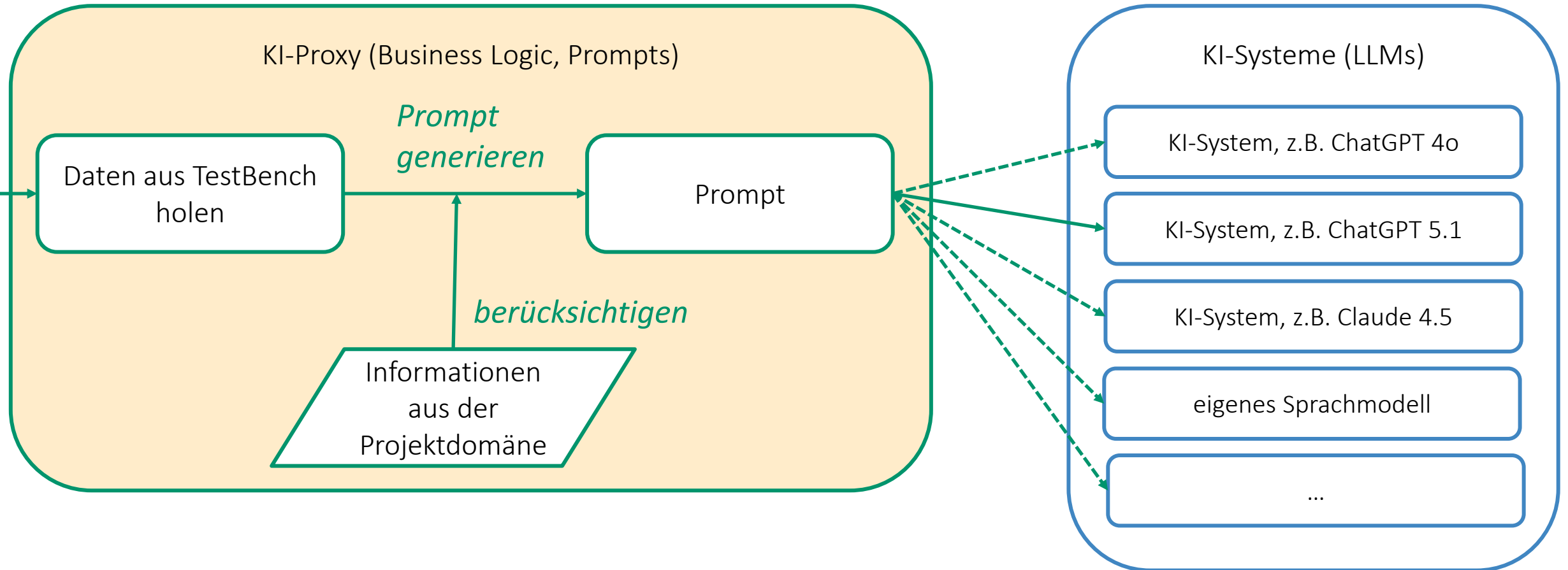
Interner Ablauf

1. Client startet die Anfrage für einen bestimmten Use Case an die KI-Bridge im KI-Proxy und liefert grundlegende Informationen, wie z.B. die ID eines Testfallsatzes.
2. Die KI-Bridge holt sich die zur Erzeugung der Prompts usw. notwendigen Informationen vom TestBench-Server.
3. Der TestBench-Server liefert die angefragten Daten an die KI-Bridge.
4. Die KI-Bridge erzeugt die Anfrage, das Prompt und schickt diese an die konfigurierte KI.
5. Die KI-Bridge verarbeitet die Antwort der KI
6. Die KI-Bridge bereitet die Antwort der KI auf und schickt das Ergebnis an den Client.



TestBench KI-Assistenten

Innere Architektur

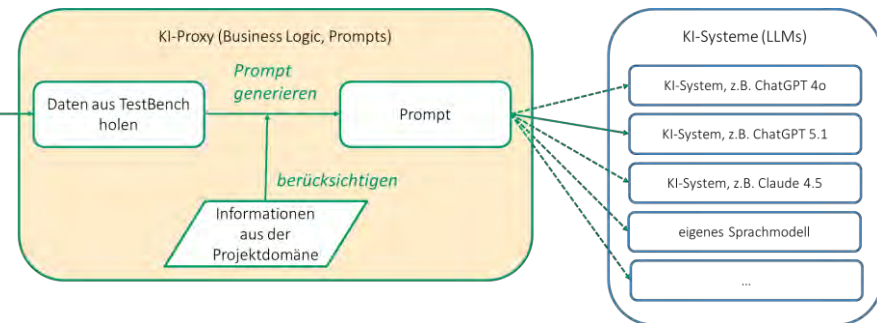


TestBench KI-Assistenten

KI-Funktionen (Use Cases) flexibel anpassbar und erweiterbar

- Immer bessere und spezialisiertere Sprachmodelle machen immer neue KI-Lösungen möglich
- Use Cases nutzen jeweils optimales^(*) Sprachmodell, statt alle das gleiche
- Integration von Domänenwissen liefert bessere Ergebnisse
- Einsatz eigener KI-Experten

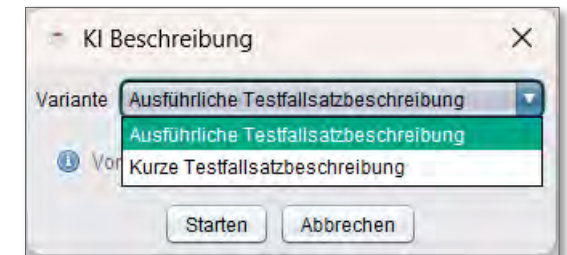
^(*) z.B. hinsichtlich Fähigkeiten, Lizenzierung, Lokation, Schutzbedürfnis von Daten



TestBench KI-Assistenten

Generierung der Testfallsatz-Beschreibung

- Erzeugung der allgemeinen Beschreibung aus Testsequenz und Testfalldaten
- Für jeden Assistenten können Varianten hinterlegt werden:
Hier: Ausführliche und kurze Testfallsatzbeschreibung



Allgemeine Beschreibung

KI Beschreibung - 2025-12-01 14:24:06

Dieser Testfallsatz prüft (dass die Preisanzeige im CarConfig nach Auswahl des Fahrzeugs, des Sondermodells und sukzessiver Zubehörartikel fortlaufend und korrekt aktualisiert wird)

Überprüfungen:

- Endpreis nach Fahrzeugauswahl entspricht dem erwarteten Wert.
- Endpreis nach Wahl des Sondermodells entspricht dem erwarteten Wert.
- Endpreis nach jedem Zubehörschritt (1 und 2) entspricht dem erwarteten Wert.
- Preisanzeige aktualisiert sich bei jedem Schritt und bleibt über den Ablauf hinweg erhalten.

Testfälle:

- iTB-TC-337-PC-1659: Fahrzeug Rolo → 12,300.00; Sondermodell Luxus → 14,799.99; Zubehör Sportfelgen → 16,049.00; weiteres Zubehör Lederlenkrad → 16,059.99.
- iTB-TC-337-PC-120167: Fahrzeug Minigolf → 15,000.00; Sondermodell Gomera → 16,413.00; Zubehör Fußmatten → 16,439.00; weiteres Zubehör Beheizbare Aussenspiegel → 16,649.00.

Von KI generierte Antworten können ungenau sein.

TestBench KI-Assistenten

Review von Testsequenzen

- Review von Testfallsätzen, ob diese mit der Implementierung übereinstimmen und einem festgelegten Regelwerk genügen.
- Regelwerk kann im Prompting des Assistenten hinterlegt werden.

Reviewanmerkungen

KI Review - 2025-12-04 14:42:26

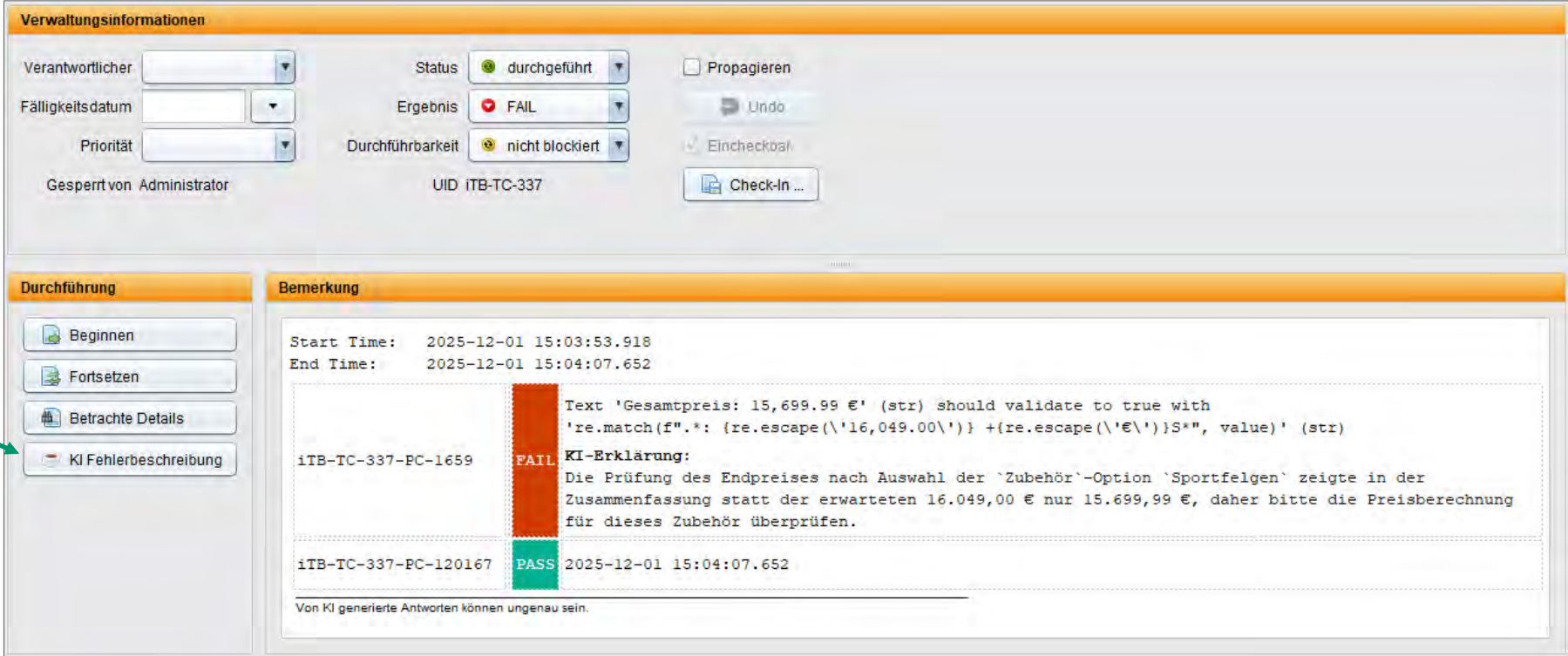
- Mehrere Schritte mit der Bezeichnung „prüfe Endpreis“ sind nicht im Infinitiv formuliert.
- Es fehlt ein Schritt mit `step_type:check`.
- Schritte namens „prüfe Endpreis“ weisen auf eine Validierung hin, sind jedoch mit `step_type:flow` versehen.
- Alle bereitgestellten Testfallsatzbeschreibungen enthalten den Hinweis „Von KI generierte Antworten können ungenau sein.“ und sind deshalb unzulässig.
- Die Testfallsatzbeschreibung „KI Beschreibung - 2025-12-02 13:41:01“ besteht nur aus dem Testziel und enthält keine Abschnitte „Überprüfungen“ und „Testfälle“.

Von KI generierte Antworten können ungenau sein.

TestBench KI-Assistenten

Erklärung von Fehlermeldungen aus der Automatisierung

- Übersetzung der technischen Details einer Fehlermeldung in „fachliche“ Sprache
- Die ursprüngliche Fehlermeldung bleibt erhalten



The screenshot displays the TestBench interface. At the top, under 'Verwaltungsinformationen', there are fields for 'Verantwortlicher', 'Fälligkeitsdatum', 'Priorität', 'Status' (set to 'durchgeführt'), 'Ergebnis' (set to 'FAIL'), 'Durchführbarkeit' (set to 'nicht blockiert'), and 'UID' (ITB-TC-337). Below this, there are buttons for 'Propagieren', 'Undo', 'Eincheckbar', and 'Check-In ...'. The main area is divided into two sections: 'Durchführung' on the left and 'Bemerkung' on the right. The 'Durchführung' section has buttons for 'Beginnen', 'Fortsetzen', 'Betrachte Details', and 'KI Fehlerbeschreibung'. The 'Bemerkung' section shows the start and end times of the test run and a list of test results. One result is highlighted in red, indicating a failure. The failure message is: 'Text 'Gesamtpreis: 15,699.99 €' (str) should validate to true with 're.match(f\".*: {re.escape('\\16,049.00\\')} +{re.escape('\\€\\')}S*\", value)' (str)'. Below this, a 'KI-Erklärung' (AI explanation) is provided: 'Die Prüfung des Endpreises nach Auswahl der `Zubehör`-Option `Sportfelgen` zeigte in der Zusammenfassung statt der erwarteten 16.049,00 € nur 15.699,99 €, daher bitte die Preisberechnung für dieses Zubehör überprüfen.' Another result is highlighted in green, indicating a pass.

Test Case ID	Result	Time
iTB-TC-337-PC-1659	FAIL	2025-12-01 15:04:07.652
iTB-TC-337-PC-120167	PASS	2025-12-01 15:04:07.652

Von KI generierte Antworten können ungenau sein.

Assistenten werden über Plugins aufgerufen

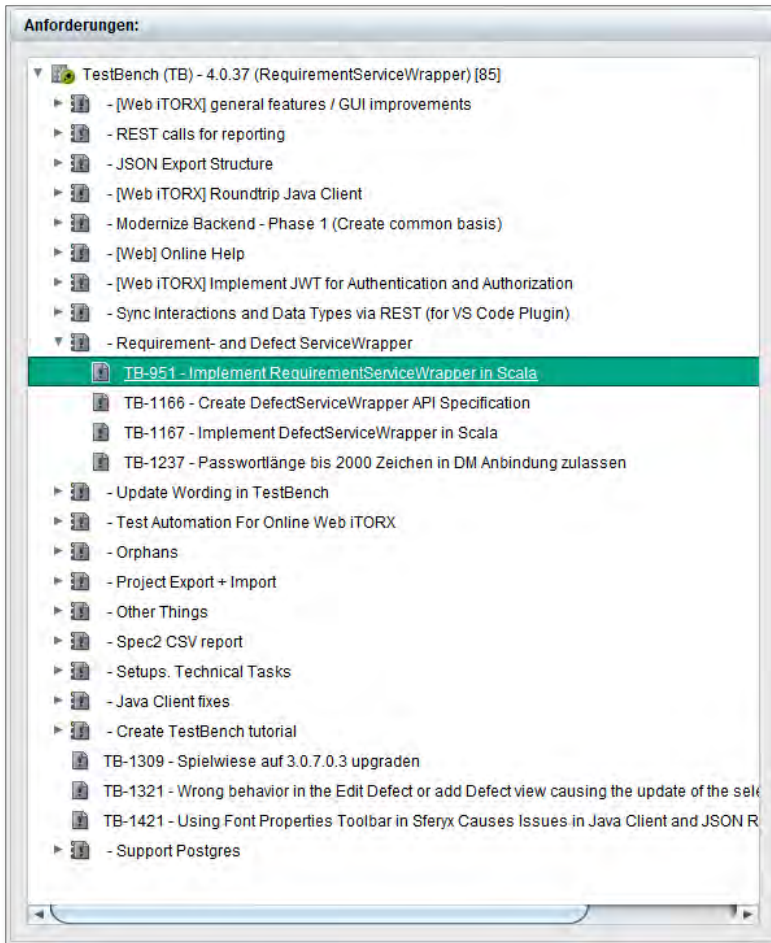
Anforderungsmanagement-Anbindung

Modernisierung der Anbindung von Anforderungsmanagement-Systemen

- Neue REST-API für Anforderungsmanagement-Proxy
- Erste Implementierungen: XLSX-Dateien, JSON-Line-Dateien, Atlassian Jira
- Neue Funktionen:
 - Abbildung von Anforderungsbäumen, z.B. bei Epics und deren User Stories
 - Änderungshistorien
- Quelloffene Implementierung

Anforderungsmanagement-Anbindung

Beispiel: Atlassian Jira



Version	Datum	Geändert von	Kommentar
10.0	2025-11-25 10:01:16		resolution: 'None' → 'Done'; status: 'IN TEST' → 'Done'
9.3	2025-11-25 09:39:38		Technical criteria: 'System Test(/) Branch(-) F-Branch' → 'System Test(/) Branch(/) F-Branch'
9.2	2025-11-24 21:00:25		Technical criteria: 'System Test(-) Branch(-) F-Branch' → 'System Test(/) Branch(-) F-Branch'
9.1	2025-10-20 14:39:11		Technical criteria: '(?) Integration Test System Test(-) Branch(-) F-Branch' → 'System Test(-) Branch(-) F-Branch'
9.0	2025-09-30 14:52:41		status: 'IN REVIEW' → 'IN TEST'
8.0	2025-09-30 14:52:37		status: 'IN DEVELOPMENT' → 'IN REVIEW'
7.0	2025-09-12 16:56:19		status: 'IN REVIEW' → 'IN DEVELOPMENT'
6.0	2025-08-13 10:12:56		status: 'IN DEVELOPMENT' → 'IN REVIEW'
5.5	2025-07-28 16:26:34		Acceptance criteria: '(/) RequirementServiceWrapper is refactored from Java to Scala (see [https://git.im...
5.4	2025-07-28 16:26:23		Acceptance criteria: '(/) RequirementServiceWrapper is refactored from Java to Scala (see [https://git.im...
5.3	2025-07-28 08:53:34		Acceptance criteria: '(/) RequirementServiceWrapper is refactored from Java to Scala (see [https://git.im...
5.2	2025-07-28 08:41:30		Acceptance criteria: '(-) RequirementServiceWrapper is refactored from Java to Scala(-) Update to {{Req...
5.1	2025-07-24 10:44:33		Acceptance criteria: '(-) RequirementServiceWrapper is refactored from Java to Scala(-) Update to {{Req...
5.0	2025-07-10 15:08:52		Story Points: 'None' → '5'; status: 'Planned' → 'IN DEVELOPMENT'
4.2	2025-07-10 12:23:37		Acceptance criteria: '(-) RequirementServiceWrapper is refactored from Java to Scala(-) Update to {{Req...
4.1	2025-07-10 12:18:14		Acceptance criteria: '(-) RequirementServiceWrapper is refactored from Java to Scala(-) Update to {{Req...
4.0	2025-07-10 10:22:38		status: 'Defined' → 'Planned'
3.1	2025-07-10 10:12:11		Technical criteria: '----- Backend Test Criteria ----- Integration Test(-) Branch(-) F-BranchBlue...
3.0	2025-07-10 10:10:44		status: 'To Discuss' → 'Defined'
2.0	2025-07-04 13:48:09		status: 'Draft' → 'To Discuss'
1.3	2025-07-03 15:46:46		Acceptance criteria: 'None' → '(-) RequirementServiceWrapper is refactored from Java to Scala(-) Update...
1.2	2025-07-03 15:12:13		description: 'None' → 'Implement the {{RequirementWrapper2}} API in Scala.'
1.1	2025-07-03 15:08:49		summary: 'Open and include RM Service Wrapper API in TestBench Core' → 'Implement Requirements...
1.0	2025-01-23 17:14:41		Issue created by I on 2025-01-23 17:14:41

Änderungshistorie einer User Story in Jira

Anforderungsbaum mit Epics mit ihren User Stories, Tasks und Bugs

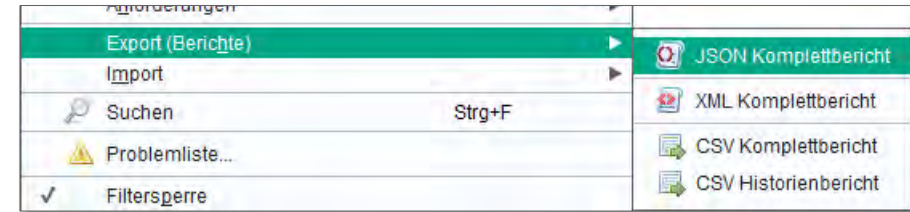
Fehlermanagement-Anbindung

Modernisierung der Anbindung von Fehlermanagement-Systemen

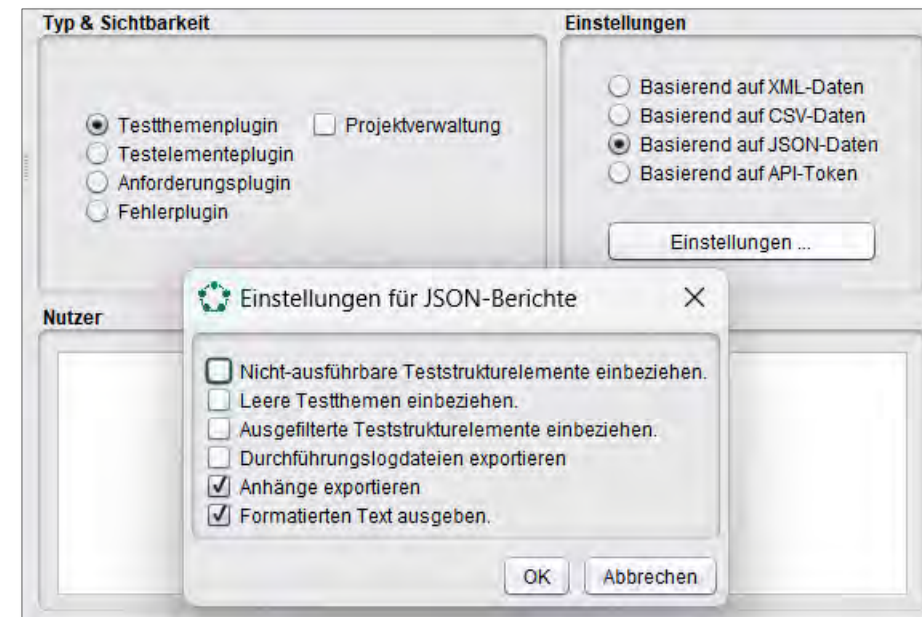
- Neue REST-API für Fehlermanagement-Proxy
- Erste Implementierungen: XLSX-Dateien, JSON-Line-Dateien, Atlassian Jira
- Neue Funktionen:
 - Anlegen von Bugs aus TestBench auch für Jira
 - Synchronisation aller Custom Fields (= Benutzerdefinierte Felder in TestBench)
- Quelloffene Implementierung

JSON als neues Export-/Import-Format

- JSON ist ein weiteres Export- und Importformat der TestBench.
- Web iTORX und weitere neue Integrationen nutzen dieses Format.
- Bei der Testdurchführung werden keine Protokolldateien erzeugt, sondern der Bericht wird um die Testergebnisdaten ergänzt.
- TestBench stellt daher einen neuen JSON-basierten Plugin-Typ auf Basis eines JSON Komplettberichts zur Verfügung.



JSON Komplettbericht im Berichtsmenü



Plugin-Definition auf Basis von JSON-Daten

Neue Authentifizierung über JSON Web Token

- Ein JSON Web Token (JWT) ist ein nach [RFC 7519](#) genormter Access Token
- Allgemeine Struktur: `xxxxx.yyyyy.zzzzz`
 - Header xxxxx: **Metadaten**, wie Signaturalgorithmus und Token-Typ
 - Payload yyyyy: **Nutzdaten** (sog. *Claims*), wie Ressource, Benutzer, Zielsystem, Ablaufdatum usw.
 - Signatur zzzzz: **Signatur** über Header und Nutzdaten, mit Secret oder Private Key erzeugt
- Signatur verhindert Manipulation

Ein JWT ist ein signiertes, aber nicht verschlüsseltes Datenpaket aus Header, Payload und Signatur, das Identitäts-Claims sicher überträgt.

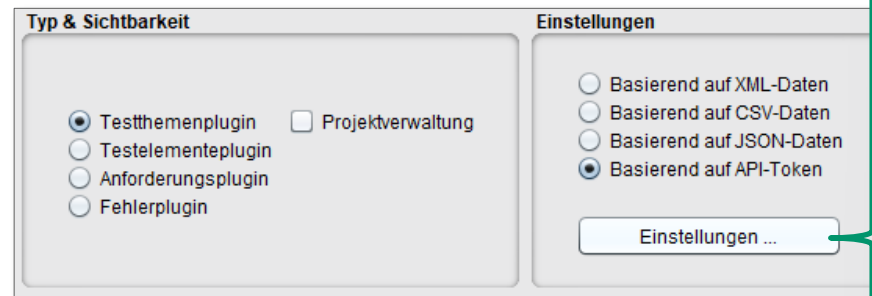
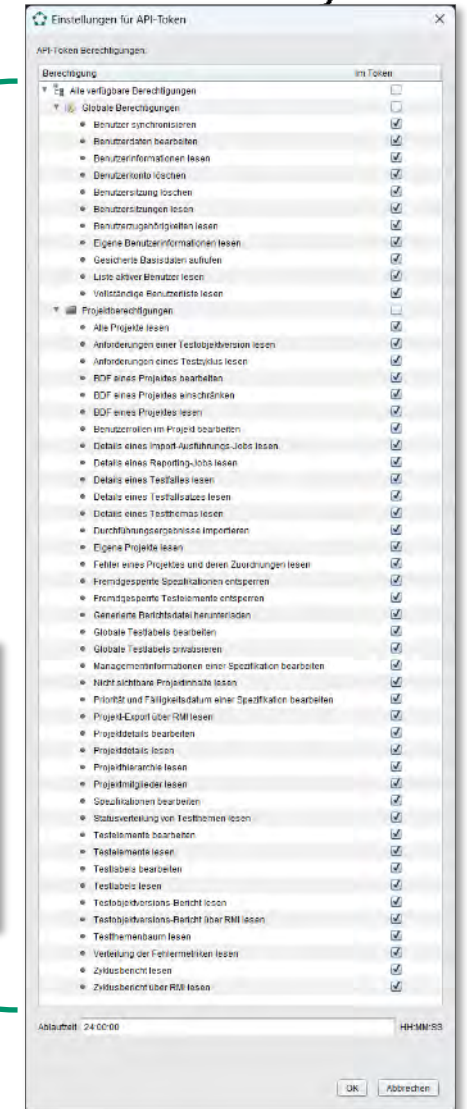
Nutzen des JSON Web Token

- Ermöglicht es, Daten sicher zwischen TestBench und anderen Systemen auszutauschen.
- Enthält **alle wichtigen Informationen**, um auf Daten des TestBench Servers zuzugreifen.
- Bestimmt, wer auf welche Daten wie lange zugreifen darf.
- Es ist kein Austausch von Authentifizierungsdaten, wie Login und Passwort, nötig.

- Web iTORX ist über JWT an TestBench angebunden.
- Web iTORX kann über eine URL, in die der JWT eingebunden ist aufgerufen werden und lädt die im Token hinterlegten Daten aus dem TestBench Server.
Dies erlaubt eine Integration in Systeme, wie z.B. Jira oder Jenkins über einen Link mit der entsprechenden URL.

Plugin für JSON Web Token

- Um JSON Web Tokens nutzen zu können stellt TestBench einen neuen Plugin-Typ zur Verfügung: **Basierend auf API-Token**
- In der Definition des Plugins wird angegeben, welche Rechte der Token weitergibt.
- Anbindung an externe Systeme kann ohne Weitergabe hochprivilegierter Zugangsdaten erfolgen.



Beschreibung der möglichen Berechtigungen und Rollen.

Plugin-Definition auf Basis eines API-Tokens mit Liste der möglichen Berechtigungen

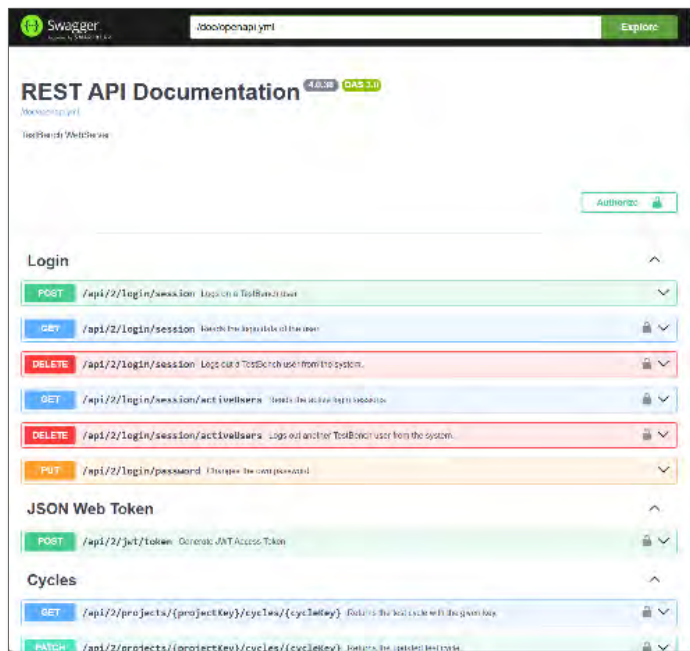
Plugins kennen den Aufrufkontext

- Die Schnittstelle aller Plugins wurde um Informationen erweitert, die den Kontext des Aufrufs beschreiben, z.B.:
 - Name und Typ des Plugins
 - Key von Testobjektversion und Testzyklus
 - ID des Knotens, von dem aus der Aufruf erfolgt

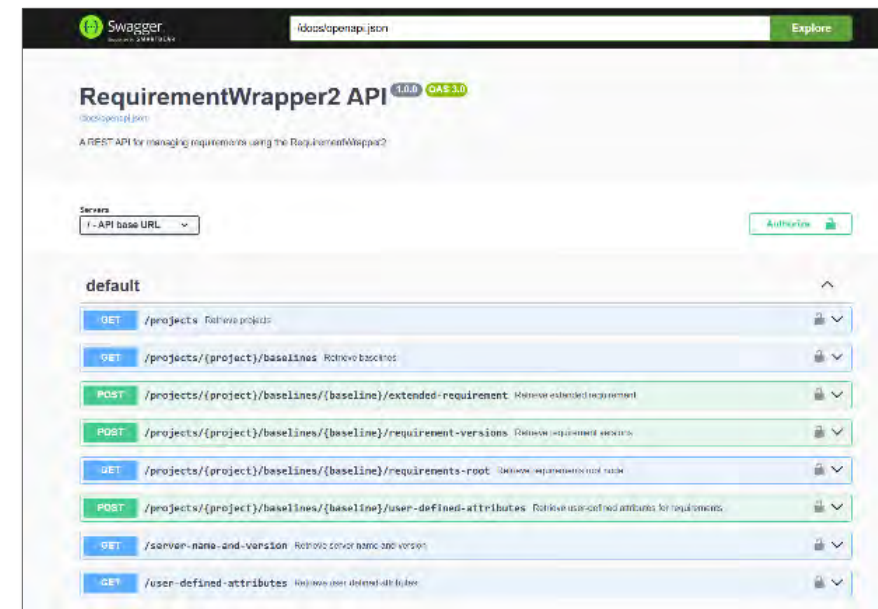
Visible context	Execution context
Plugin name: <input type="text" value="JSON Web Token"/>	Project key: <input type="text" value="3"/>
User: <input type="text" value="tt-admin"/>	TOV key: <input type="text" value="13"/>
Encoding: <input type="text" value="—"/>	Cycle key: <input type="text" value="43"/>
Tree type: <input type="text" value="TESTTHEMES"/>	Tree root ID: <input type="text" value="ITB-TC-337"/>
	Tree root key: <input type="text" value="4611686020000020668"/>
	Element type: <input type="text" value="TESTCASESET"/>
	Tree type: <input type="text" value="TESTTHEMES"/>
	Data type: <input type="text" value="API"/>
	Filtering: <input type="text" value="—"/>

Erweitertes REST-API

- Die Ebene des Application Server besteht jetzt aus drei Servern. Der dritte, neue Server stellt ein stark erweitertes REST-API zur Verfügung.
- Die neuen Services zur Anbindung von Anforderungs- und Fehlermanagementsystemen und der KI-Proxy stellen ebenfalls neue REST-APIs zur Verfügung.



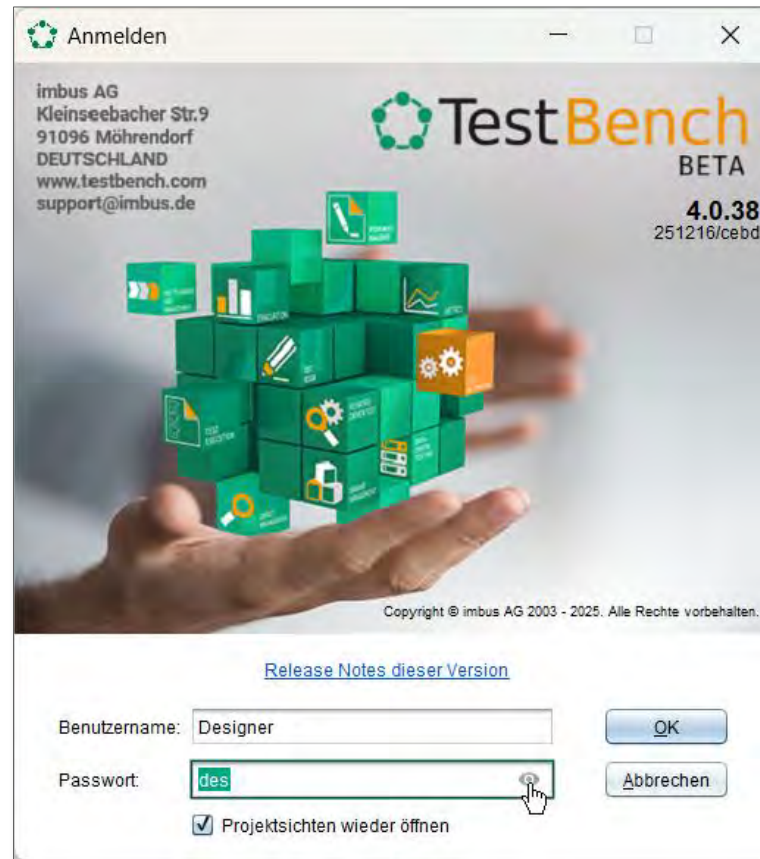
REST-API des neuen Servers



REST-API des Anforderungsmanagement Proxies





Passwort im Anmeldefenster anzeigen



In der Anmeldemaske kann über Anklicken des Auges im Password-Feld das Passwort temporär sichtbar gemacht werden:



Tastaturkürzel zum Anlegen von Elementen

Das Anlegen von Teststrukturelementen(*) kann jetzt auch über Tastenkürzel erfolgen:

 Neues <u>T</u> estthema	Strg+Alt+T
 Neuer Test <u>f</u> allsatz	Strg+Alt+S
 Ent <u>l</u> sperren (Bearbeitung beenden)	Strg+D
 Rekursiv <u>S</u> perren	Strg+Umschalt+S

 <u>N</u> eue Subdivision	Strg+Alt+S
 Neuer <u>D</u> atentyp	Strg+Alt+D
 Neuer Referenzdatentyp	Strg+Alt+R
 Neues Key <u>w</u> ord	Strg+Alt+K
 Neue <u>B</u> edingung	Strg+Alt+B
 Ent <u>l</u> sperren (Bearbeitung beenden)	Strg+D

(*) Teststrukturelemente sind: Testthemen, Testfallsätze, Subdivisionen, Keywords, Datentypen und Bedingungen



Zeit für

Fragen!

TestBench Demo 2 (Version 5.0, Demo) - Testthemen-Sicht

Spezifikation Durchführung Listenansicht Filter Versionen & Label

Test simulieren

Vorbedingungen

Aktionen und erwartete Reaktionen

Testsequenz	1. Parameter	Bemerkung
1 CarConfig starten		
2 Fahrzeug erstellen	Fahrzeug	
3 prüfe Endpreis	Endpreis_nach_Fahrzeug	
4 Sondermodell wählen	Sondermodell	
5 prüfe Endpreis	Endpreis_nach_Sondermodell	
6 Zubehör wählen	Zubehör - 1	
7 prüfe Endpreis	Endpreis_nach_Zubehör - 1	
8 Zubehör wählen	Zubehör - 2	
9 prüfe Endpreis	Endpreis_nach_Zubehör - 2	
10 CarConfig beenden		

Nachbedingungen

Parameterliste

Datentyp	Parametername	Standardwert
1 Fahrzeug	Fahrzeug	
2 Endpreis	Endpreis_nach_Fahrzeug	
3 Sondermodell	Sondermodell	

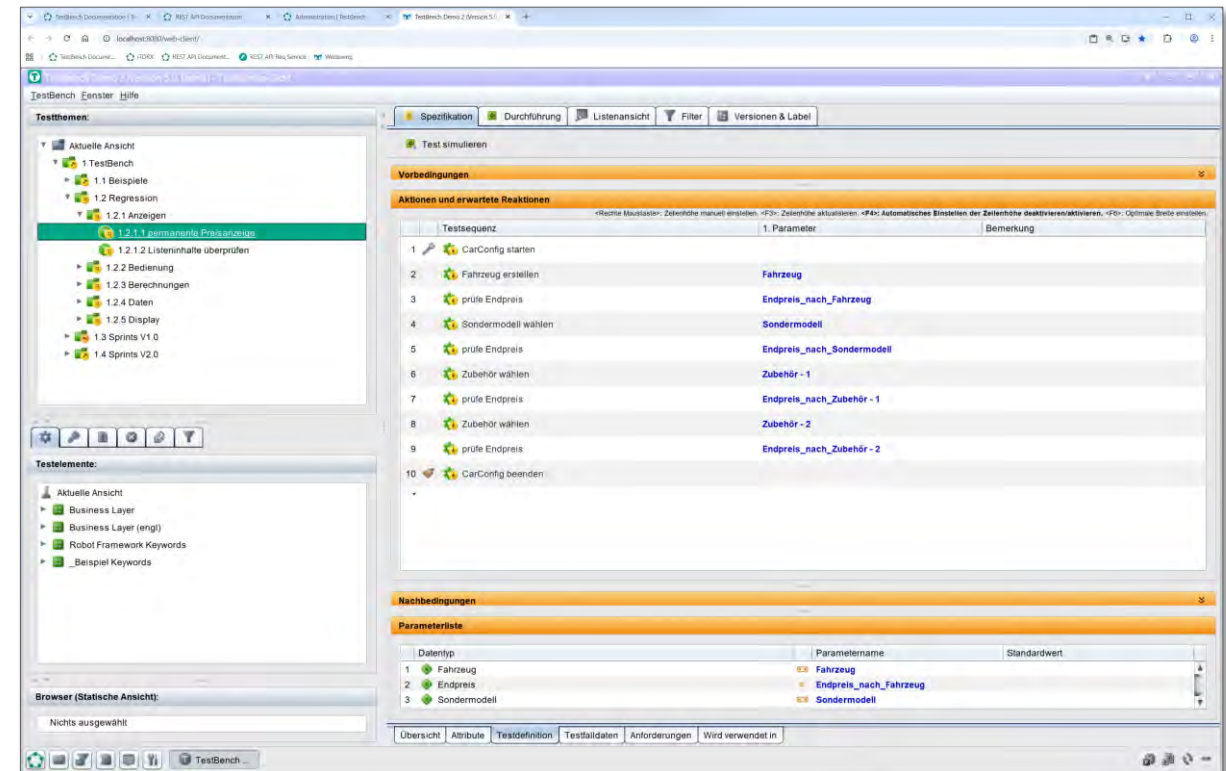
Übersicht Attribute Testdefinition Testfalldaten Anforderungen Wird verwendet in

TestBench...

TestBench Client im Browser?

Technologie: Webswing

- Client läuft vollständig im Browser
- Einzelne Fenster werden Browser-Tabs oder –Fenster
- Java-Client kann parallel genutzt werden
- Weiterer Server auf Ebene des Application-Servers



KONTAKT

imbus AG

Kleinseebacher Str. 9
91096 Möhrendorf
DEUTSCHLAND
Tel. +49 9131 7518-0
info@testbench.com
www.testbench.com

