



eHealthExperts

EHX GROUP

PUPPETRY

Handbuch

Puppetry Handbuch

Stand: **Release**

Version: 1.0.4

SW-Version: 1.2.*

Erstellt von: eHealth Experts GmbH
An den Römerhügeln 11a
82031 Grünwald

Inhaltsverzeichnis

Inhaltsverzeichnis	3
Einleitung.....	5
Mehrbenutzerszenario	5
Nachrichtenanalyse	6
Abgrenzung zu einem physischen Konnektor.....	6
Das Infomodell von Puppetry	6
Ein Terminal an mehreren Arbeitsplätzen	7
Verwaltung von SMC-Bs	7
Zugangsdaten für die TLS Verbindungen.....	8
Ereignisse.....	8
Installation.....	9
Systemvoraussetzungen.....	9
Distributionen.....	9
Installation.....	10
Starten des Programms	10
Inbetriebnahme.....	12
Betriebsmodi	12
Admin-Modus	12
Test-Modus	13
Seitenmenü	13
Hauptbereich.....	14
Filter.....	15
Live-Protokoll	16
Wiederkehrende Schaltflächen	18
Schaltflächen im Menüband	18
Schaltflächen im Hauptbereich.....	19
Schaltflächen in den Detailmasken des Hauptbereichs.....	21
Menü-Einträge	22
Admin-Modus	22
Aufrufkontext.....	22
Versicherte.....	22

Praxis.....	23
Authentifizierung.....	23
Datensicherung.....	24
Test-Modus.....	24
Aufrufkontext.....	24
Abonnenten.....	26
Ereignisse.....	26
Testlabor.....	26
Vom Modus unabhängige Menü-Einträge.....	32
Einstellungen.....	32
Status.....	33
Protokolle.....	33
Systeminfo.....	34
Benutzerkonfiguration.....	35
Glossar.....	36

Einleitung

Puppetry ist der Konnektor-Simulator aus dem Hause *eHealthExperts*. Er wurde mit dem Ziel entwickelt, die Schnittstellen des Anwendungskonnektors zum Primärsystem virtuell und spezifikationskonform abzubilden.

Puppetry richtet sich damit in erster Linie an die Hersteller von Primärsystemen und dient ihnen als Hilfestellung bei der Anpassung ihrer Produkte für die Integration in die *Telematikinfrastruktur* (TI). Die Nutzer haben mit *Puppetry* die Möglichkeit, einfach und bequem die Funktion eines Konnektors spezifikationskonform zu simulieren und dadurch die Kompatibilität ihrer Software zu testen. *Puppetry* dient darum in erster Linie dazu, dem Nutzer eine Spielumgebung für eine nichtlineare Auseinandersetzung zum Verhalten von Konnektor und Primärsystem zur Verfügung zu stellen.

Als weitergehende Hilfestellung bietet *Puppetry* auch ein vorkonfiguriertes Set von linearen Testfällen an, welche Schritt-für-Schritt durchgeführt werden können. Jeder einzelne Testschritt erlaubt eine technische Validierung der zu erwartenden Reaktionen des zu testenden Primärsystems. Zusätzlich kann durch den Nutzer eine manuelle Validierung in Hinblick auf die Erwartungshaltung an das Primärsystem erfolgen. Im Anschluss an die Testfälle werden durch *Puppetry* zusammenfassende Testreports erstellt, mit denen die spezifikationskonforme Funktionsweise des Primärsystems gegenüber der *gematik* dokumentiert werden kann.

Die derzeitige Ausbaustufe von *Puppetry* (Version 1.2.*) bildet alle notwendigen Schnittstellen für die Fachanwendung *Versichertenstammdatenmanagement* (VSDM) mit dem Stand *Online-Produktivbetrieb* (OPB1) an.

Mehrbenutzerszenario

Die Software *Puppetry* wird, vergleichbar mit der Software eines physischen Konnektors, als Serveranwendung ausgeführt. Der Zugriff auf die Anwendung erfolgt über eine separate Web-Oberfläche; Hier kann der Nutzer sowohl die Konfiguration von *Puppetry* anpassen als auch die TI simulieren. Die Nutzung dieser Serveranwendung kann auf zwei unterschiedliche Arten erfolgen:

- als Installation auf einem netzwerkfähigen Computer mit der Möglichkeit zum Zugriff durch verschiedene Nutzer von verschiedenen Arbeitsplätzen,
- als lokale Installation zum alleinigen Gebrauch an einem Einzelplatz-PC.

Obwohl *Puppetry* eine Serveranwendung ist, kann unter Umständen ein dezentraler Gebrauch durch mehrere Nutzer – wie bei einem echten Konnektor – Nebeneffekte verursachen: Verschiedene Nutzer derselben Installation könnten sich gegenseitig beeinflussen, denn Konfigurations-

und Zustandsänderungen wirken sich immer global (also auf alle Nutzer) über die gesamte Anwendung aus.

Dieser Effekt lässt sich für die Nutzer dadurch umgehen, dass pro Instanz ein separates Set an Testdaten verwendet wird. Alternativ kann Puppetry auch problemlos als Einzelarbeitsplatzanwendung betrieben werden – mit beliebig viele Instanzen auf verschiedenen Rechnern beziehungsweise auf verschiedenen Ports.

Nachrichtenanalyse

Puppetry bietet dem Nutzer die Möglichkeit, Nachrichten auf der Anwendungsebene zu analysieren. Es können *SOAP-Nachrichten*, *CETP-Eventnachrichten* und *DVD/SDS-Anfragen* einfach und bequem nachvollzogen werden. Probleme, die auf Netzwerkebene beziehungsweise beim TLS-Verbindungsaufbau auftreten, müssen jedoch durch separate Anwendungen (beispielsweise *Tcpdump*, *Wireshark* oder vergleichbare) analysiert werden.

Die Zertifikatsprüfung im Rahmen des TLS-Verbindungsaufbau wird in Puppetry explizit geloggt. Dadurch können Fehler in diesem Zusammenhang auch einfach in den Log-Dateien von Puppetry nachvollzogen werden.

Abgrenzung zu einem physischen Konnektor

Bei der Anwendung Puppetry handelt es sich um eine Simulation der Schnittstellen eines Anwendungskonnektors zum Primärsystem für die spezifische Fachanwendung VSDM in dem Stand von OPB1. Darüber hinaus vorhandene Funktionen und Konfigurationsmöglichkeiten eines physischen Konnektors stehen nicht im Fokus der Anwendung und werden dementsprechend nicht mit abgebildet.

Aufgrund dieser Tatsache ergeben sich Vereinfachungen bei der Konfiguration bezüglich des Umfangs und der Komplexität. Sämtliche Einstellung bezüglich der Telematikinfrastruktur entfallen in der Anwendung. Das Infomodell wurde im Zuge dessen soweit vereinfacht, dass es nur die Aspekte abbildet, die sich auf die Außenschnittstellen zum Primärsystem auswirken.

Das Infomodell von Puppetry

Das Infomodell eines Konnektors prägt sich an den Außenschnittstellen vor allem durch den Aufrufkontext aus. Daher beschränken sich die Konfigurationsmöglichkeiten bezüglich des Infomodells in Puppetry hauptsächlich auf diesen Aspekt. *Mandanten*, *Clientsysteme*, *Arbeitsplätze* und

Kartenterminals können allerdings in Puppetry nicht alle einzeln konfiguriert und einander zugeordnet werden. Stattdessen können mehrere *Aufrufkontexte* konfiguriert werden, die jeweils über die Kombination aus *Mandanten-ID*, *Clientsystem-ID* und *Arbeitsplatz-ID* identifiziert werden. Es kann mehrere Aufrufkontexte geben, mit demselben Mandanten beziehungsweise Clientsystem oder Arbeitsplatz. Nur das Tripel aus diesen drei Werten muss immer eindeutig sein.

Pro Aufrufkontext können zugehörige Terminals verwaltet werden. Die Terminalverwaltung umfasst jedoch nicht das Pairing und Zuordnen von Terminals. Sofern ein Terminal in einem Aufrufkontext vorhanden ist, gilt es automatisch als gepairt und zugleich auch diesem Aufrufkontext zugeordnet. Die Konfiguration der Eigenschaften eines Terminals erlaubt es festzulegen, ob es sich um ein Remote-Terminal und oder ein physisches Terminal handelt und ob es aktuell verbunden ist.

Ein Terminal an mehreren Arbeitsplätzen

Während ein Terminal in der Praxis mehreren Arbeitsplätzen zugewiesen werden kann, erfolgt in Puppetry eine solche direkte Zuordnung nicht – hier werden die Terminals unterhalb eines Aufrufkontextes verwaltet. Solange der Betrachtungsgegenstand eines Tests nur ein Arbeitsplatz bzw. ein Aufrufkontext ist, spielt diese Einschränkung keine Rolle. Um dennoch indirekt simulieren zu können, dass ein Terminal gleichzeitig mehreren Aufrufkontexten zugeordnet ist, erfolgt innerhalb von Puppetry eine Synchronisation mehrerer Terminals mit der gleichen *Terminal-ID*. Wird also in einem Aufrufkontext in ein Karten-Terminal eine Karte gesteckt, wird diese Karte auch in allen anderen Karten-Terminals mit derselben Terminal-ID unter anderen Aufrufkontexten gesteckt.

Verwaltung von SMC-Bs

Eine Registrierung und Zuordnung von SMC-Bs zu den einzelnen Mandanten/Aufrufkontexten ist in Puppetry nicht notwendig. Alle angelegten SMC-Bs sind immer auch allen existierenden Mandanten zugeordnet. Um dennoch die Reaktion des Primärsystems testen zu können, wenn eine dem Mandanten nicht zugeordnet SMC-B genutzt wird, gibt es an der Karte einen Parameter. Über diesen Parameter kann angegeben werden, dass die Karte keinem Mandanten zugeordnet ist. Wenn diese Karte genutzt wird, erhält man eine Fehlernachricht mit dem entsprechenden Fehlercode.

Zugangsdaten für die TLS Verbindungen

Gemäß Konnektor-Spezifikation sind die Zugangsdaten (Benutzername und Passwort für Basic Authentication; die Zertifikate für die Client-Authentifizierung) den Clientsystemen zugeordnet. So ist es auch in Puppetry umgesetzt. Für alle Clientsystem-IDs, die in der Verwaltung der Aufrufkontexte angelegt wurden, können separate Zugangsdaten definiert werden. Darüber hinaus können Standardzugangsdaten festgelegt werden. Diese gelten für alle Clientsystem, für die keine speziellen Zugangsdaten festgelegt wurden.

Ereignisse

Puppetry löst alle Ereignisse, die durch die Zustandsänderung an der Puppetry-Oberfläche entstehen, spezifikationskonform aus. So sorgt das Stecken einer Karte in Puppetry auch dafür, dass ein entsprechendes *CardInsert*-Ereignis ausgelöst wird. Genauso sorgt das Trennen der virtuellen Verbindung zur TI auch dafür, dass die entsprechenden Ereignisse ausgelöst werden.

Allerdings lassen sich nicht alle Ereignisse, die der Konnektor werfen kann, durch Interaktionen an der Oberfläche indirekt auslösen (zum Beispiel *EC_Time_Sync_Pending_Warning*). Um die Reaktion des Primärsystems dennoch auch auf solche Anwendungsfall unbezogene Ereignisse testen zu können, bietet Puppetry die Möglichkeit alle spezifizierten Events zusätzlich auch manuell auslösen zu können.

Installation

Systemvoraussetzungen

Die Verwendung von Puppetry stellt folgende Mindestvoraussetzungen an das System des Nutzers:

- mindestens 2GB freier Arbeitsspeicher
- ein *Java 8* fähiges Betriebssystem
- *Java 8 64-Bit*
- *Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8*
- *Firefox ESR (Version 45 oder höher)*
 - Textkodierung *Unicode*
 - Zoomfaktor *100%* (für eine optimale Nutzung)
- **Mindestfenstergröße *1024 × 768px***

Für eine optimale Nutzung von Puppetry empfiehlt sich die Einhaltung der oben genannten Mindestmaße. Puppetry kann auch bei kleineren Bildschirmgrößen verwendet werden, jedoch wird das Interface in diesem Fall in der Größe nicht weiter minimiert, sondern gescrollt.



Es ist prinzipiell möglich, die Web-Oberfläche von Puppetry auch mit anderen aktuellen Browsern, wie *Chrome*, *Edge* und *Safari* zu verwenden, jedoch wurden diese nicht vollständig auf die Kompatibilität zu Puppetry getestet. Eine Nutzung von Puppetry in Kombination mit diesen Browsern geschieht somit ohne Gewährleistung.

Distributionen

Um Puppetry nutzen zu können, muss die Software zunächst installiert werden, dafür stehen zwei unterschiedliche Distributionen bereit:

1. *puppetry*

Diese Distribution beinhaltet die Puppetry-Laufzeitumgebung sowie alle benötigten Konfigurationsdateien. Diese Distribution ist auf allen Betriebssystemen lauffähig, welche die Systemvoraussetzungen erfüllen. Für die Nutzung dieser Distribution müssen auf dem System die folgenden zusätzlichen Komponenten vorab installiert werden:

- Java 8
- Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8

Für die Installation von Java 8 und JCE folgen sie bitte der Installationsanleitung des Herstellers.

2. *puppetry inkl. OpenJDK (Windows 64-Bit)*

Diese Distribution beinhaltet neben der eigentlichen Installation bereits eine aktuelle Version des OpenJDK JRE. Diese Distribution ist ausschließlich unter Windows-Betriebssystemen mit 64-Bit Unterstützung lauffähig.

Installation

Nach Auswahl der geeigneten Distribution muss diese in das gewünschte Installationsverzeichnis entpackt werden. Dies kann zum Beispiel unter Windows der folgende Pfad sein:

`C:\Puppetry\`



Für den Start von Puppetry wird eine Lizenzdatei benötigt. Diese muss vor dem Start der Anwendung in das Installationsverzeichnis kopiert werden. Der Name der Lizenzdatei muss dem Schema *puppetry*.lic* entsprechen.

Starten des Programms

Das Installationsverzeichnis beinhaltet ein Batch-Script zur Start der Puppetry-Laufzeitumgebung. Unter Windows wird für den Start die Datei *start.bat*, unter Linux/Unix die Datei *start.sh* verwendet.

Beispiel für Windows:

`C:\Puppetry\start.bat`

Standardmäßig ist die Web-Oberfläche über alle Netzwerkadressen des Rechners mit dem Port 8080 erreichbar. Soll die Oberfläche nur über eine bestimmte Netzwerkadresse beziehungsweise über einen anderen Port erreichbar sein, kann dies über Kommandozeile-Parameter angegeben werden.

Beispiel für Windows:

```
C:\Puppetry\start.bat <host> <port>
```

Nach dem Start der Puppetry Laufzeitumgebung erfolgt der Zugriff auf die Web-Oberfläche via Browser. Dazu muss im Browser die folgende URL eingegeben werden:

```
http://localhost:8080/
```

beziehungsweise

```
http://<host>:<port>/
```

Inbetriebnahme

Nach dem erfolgreichen Start der Anwendung kann die Web-Oberfläche im Browser aufgerufen werden. Die Web-Oberfläche von Puppetry besteht grundsätzlich aus zwei Bereichen, dem Hauptbereich und dem Seitenmenü. Zusätzlich wird Puppetry in zwei Anwendungsbereiche unterteilt, die beiden Betriebsmodi *Test-* und *Admin-Modus*.

Betriebsmodi

Die Konnektor-Simulation Puppetry verfügt über zwei Betriebsmodi. Der jeweils für den Nutzer aktive Modus wird über dem Seitenmenü eingeblendet und zusätzlich durch ein eindeutiges Farbschema kenntlich gemacht. Beim Start von Puppetry wird standardmäßig der Test-Modus mit dem Menü-Eintrag *Aufrufkontext* geladen. Der Nutzer kann zwischen den beiden Betriebsmodi nach Belieben wechseln.

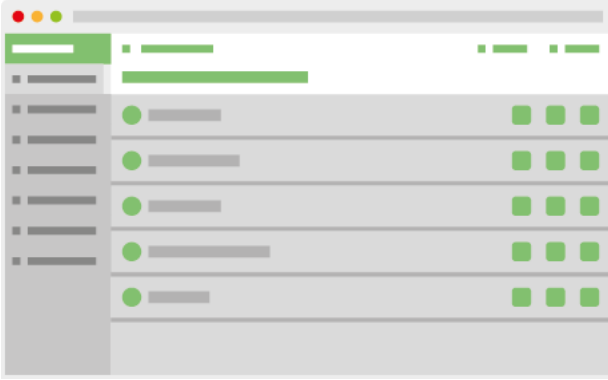
Admin-Modus



Befindet sich Puppetry im Admin-Modus, verwendet Puppetry ein blaues Farbschema.

Wünscht der Nutzer die vordefinierten Testdaten zu bearbeiten, muss dieser hierzu in den Admin-Modus wechseln.

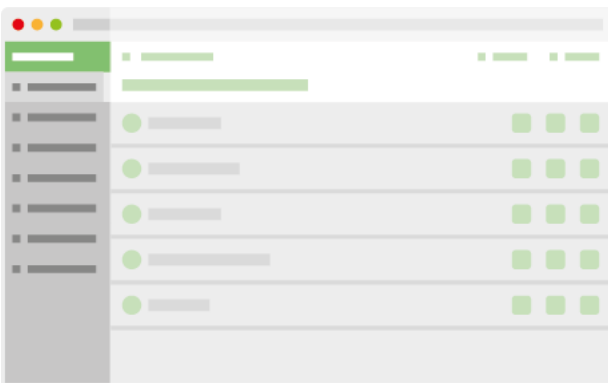
Test-Modus



Der Test-Modus dagegen verwendet ein grünes Farbschema, um eine schnelle visuelle Differenzierung zu gewährleisten.

Ziel des Test-Modus ist es, dem Nutzer die Möglichkeit zu geben, verschiedene Prozesse rund um einen spezifikationskonformen Konnektor nachzustellen.

Seitenmenü



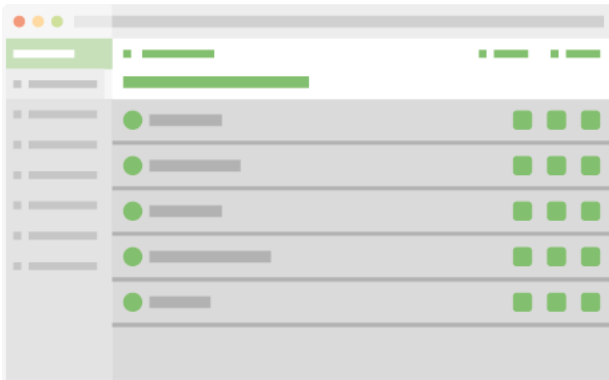
Für den schnellen Zugriff auf alle relevanten Funktionen verfügt Pupperty über ein links angeordnetes Seitenmenü. Dieses kann über das Menü-Icon ein- und ausgeklappt werden. Beim Start von Pupperty befindet sich das Seitenmenü im ausgeklappten Zustand.

Folgende Menü-Einträge sind Bestandteil der beiden Betriebsmodi:

	Test	Admin	Beschreibung
Aufrufkontext	✓*	✓**	* Konfigurationen der <u>verfügbaren</u> Aufrufkontexte. **Erstellen und Verwalten von Aufrufkontexten.
Versicherte		✓	Erstellen und Verwalten von Versichertenkarten (eGK, KVK, ...).
Praxis		✓	Erstellen und Verwalten von Praxiskarten (SMC-B, HBA, ...).
Abonnenten	✓		Übersicht der am Ereignisdienst angemeldeten Abonnenten.
Ereignisse	✓		Auslösen vordefinierter Ereignisse.

Testlabor	✓		Durführung von Testfällen.
Authentifizierung	✓	✓	Verwaltung der Zugangsdaten für die Client-Authentifizierung.
Einstellungen	✓	✓	Konfigurationen des Konnektors (Verbindung und Dienstverzeichnis), des Ereignisdienstes und der Anzeigeeinstellungen (Live-Protokoll, Aktualisierung).
Status	✓	✓	Konfiguration vom Konnektor (ein/aus) und Verbindungsstatus (online/offline).
Protokolle	✓	✓	Zugriffsmöglichkeit auf die Protokolldateien (Basis-Log, SOAP-Log, Schnittstellen-Log), die während der Nutzung des Programmes generiert wurden.
Datensicherung		✓	Erstellung von Datensicherung und Wiederherstellung einer vorherigen Konfiguration.
Systeminfo	✓		Übersicht der Versionsnummer und der Lizenz.

Hauptbereich



Der Hauptbereich von Puppetry befindet sich im rechten Bereich des Browserfensters.

Das Hauptfenster von Puppetry unterteilt sich in zwei Bereiche: dem oberen Menüband und dem Inhaltsbereich. Innerhalb dieses Bereiches werden alle wichtigen Aktionen der Software durchgeführt.

Zum Wechsel zwischen den beiden Betriebsmodi wird die Schaltfläche *Modus wechseln* im Menüband verwendet. Diese Funktion ist jederzeit verfügbar. Ebenfalls im Menü-Band befindet sich die Schaltfläche *Live-Protokoll*.

Filter

Unterhalb des Menübandes wird bei Listenansichten ein Filter eingeblendet, der die Filterung der Einträge erlaubt. Hinter dem Eingabefeld des Filters wird die Gesamtzahl der Einträge beziehungsweise im gefilterten Zustand die Anzahl der dargestellten Einträge angezeigt. Sobald eine Filterung vorgenommen wird, werden Einträge entsprechend der Eingabe reduziert.

Die Filterung erfolgt über das *Label* und die *Beschreibung*. Die angegebenen Filterwerte werden als *reguläre Ausdrücke* für die Filterung ausgewertet. Ohne Sonderzeichen verhält sich die Filterung wie eine normale Textsuche. Unter Verwendung von Sonderzeichen ($?^*+[]^{\$}()$) können darüber hinaus auch komplexere Filterungen vorgenommen werden. Dabei werden Sonderzeichen innerhalb des Sucheintrages immer als regulärer Ausdruck gewertet. Um diese Zeichen auch in einer einfachen Textsuche verwenden zu können, muss diesen immer ein Backslash \ vorangestellt werden.

Beispiele für mögliche Filterwerte und deren Bedeutung:

- `eGK`
Es werden alle Einträge mit dem Wert „eGK“ gefiltert.
- `eGK \ (`
Es werden alle Einträge mit dem Wert „eGK (“ gefiltert.
- `eGK|PIN`
Es werden alle Einträge mit dem Wert „eGK“ **oder** dem Wert „PIN“ gefiltert.
- `(?=.*eGK)(?=.*PIN)`
Es werden alle Einträge mit dem Wert „eGK“ **und** dem Wert „PIN“ gefiltert.
- `\d`
Es werden alle Einträge gefiltert, die eine Nummer enthalten.
- `(Prüfung).*(nachweis)`
Es werden alle Einträge gefiltert, die als Wert „Prüfung“ folgend von einer beliebigen Anzahl von Zeichen und abschließend dem Wert „...nachweis“ enthalten.
- `(Prüfung).?(nachweis)`
Es werden alle Einträge gefiltert, die als Wert „Prüfung“ folgend von einem beliebigen Zeichen und abschließend dem Wert „...nachweis“ enthalten.







Live-Protokoll





Zur Kontrolle der korrekten Funktionsweise der Konnektor-Simulation kann über die Schaltfläche *Live-Protokoll* jederzeit die zentrale Log-Datei von Puppetry eingesehen werden. Das Live-Protokoll steht dabei sowohl im Test-Modus als auch im Admin-Modus zur Verfügung.








Die Darstellung erfolgt über einen nichtmodalen Dialog. Die Position und Größe dieses Dialogs kann vom Nutzer geändert werden. Wird das Browserfenster so sehr verkleinert, dass der Dialog nicht mehr vollständig dargestellt werden kann, kann es beim Vergrößern des Browserfensters zu Darstellungsproblemen kommen. Sollte das Live-Protokoll außerhalb des sichtbaren Bereichs dargestellt werden, kann das Darstellungsproblem dadurch behoben werden, indem das Live-Protokoll über entsprechende Schaltflächen neu eingeblendet oder die Seite gänzlich neu geladen wird.

Das Live-Protokoll verfügt über eine Titel-Leiste mit verschiedenen Aktionen, die die Darstellung des Fensters beeinflussen:

Icon	Funktion
	Automatisches Scrollen bei Aktualisierungen eingeschaltet.
	Automatisches Scrollen bei Aktualisierungen ausgeschaltet.
	Live-Protokoll auf die Größe der Titel-Leiste reduzieren.
	Live-Protokoll ausklappen.
	Live-Protokoll in linke untere Ecke minimieren.
	Live-Protokoll wiederherstellen.

	Live-Protokoll maximieren.
	Live-Protokoll schließen.

Die Ausgabe des Live-Protokolls lässt zusätzlich eine Reduzierung der Ausgabe in Bezug auf die vier Kategorien *Ereignisse*, *Aufrufe*, *Fehler* und *Hinweise* zu. Sie erfolgt über die An- oder Abwahl der jeweiligen Checkboxes im oberen Menüband des Fensters. Zusätzlich besteht die Möglichkeit, die Log-Ausgabe über die Angabe eines Freitextfilters einzuschränken.







Icon	Typ	Beschreibung
	Anfrage	Empfangener SOAP-Request.
	Antwort	Versendete SOAP-Response.
	Dienstverzeichnis	Anruf des Dienstverzeichnisdienstes.
	Ereignis	Versendetes Ereignis.
	Ereignis	Ausgelöstes aber nicht versandtes Ereignis.
	Fehler	Versendeter SOAP-Fault
	Testfall	Ausführungsschritt eines Testfalls

Wiederkehrende Schaltflächen

Puppetry verwendet für die Schaltflächen des Programms eigens erarbeitete Icons.










Schaltflächen im Menüband





Icon	Beschreibung
	Öffnet und schließt das Seitenmenü.
	Öffnet das Live-Protokoll.
	Wechselt zwischen Admin- und Test-Modus.
	Fügt der Liste einen neuen Eintrag hinzu.
	Lädt einen gematik Testbericht als ZIP-Archiv herunter.
	Lädt eine Testfallübersicht als ZIP-Archiv herunter.

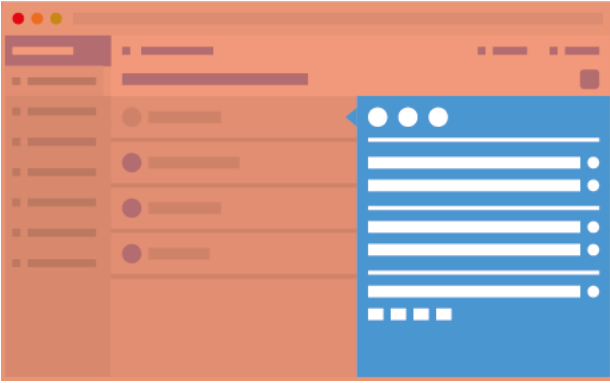
Schaltflächen im Hauptbereich



Icon	Beschreibung
Allgemeine Schaltflächen	
	Übernimmt die vorgenommenen Änderungen.
	Verwirft die vorgenommenen Änderungen.
	Lädt die aktuelle Konfigurationsdatei (.puppetbox) herunter.
	Stellt den Auslieferungszustand der Konfiguration wieder her.
	Einspielen einer externen Konfigurationsdatei (.puppetbox).
Schaltflächen der Listeneinträge	
	Der Eintrag ist gesperrt, er kann weder bearbeitet noch gelöscht werden.
Kontextspezifische Auswahlmöglichkeiten für Listeneinträge	
	Wechselt zur Terminalansicht eines Eintrages.
	Überschreibt das Standardverhalten eines Aufrufkontextes.
	Deaktiviert einen Eintrag.

	Aktiviert einen Eintrag.
	Trennt die Terminal-Verbindung.
	Stellt die Terminal -Verbindung wieder her.
	Öffnet die Detailmaske im Admin-Modus für eine Karte.
	Entfernt eine zuvor gesteckte Karte.
	Löscht einen bestehenden Eintrag aus der Liste.
	Lädt eine PDF-Datei herunter.
	Lädt eine Log-Datei herunter.
	Startet einen Testlauf.

Schaltflächen in den Detailmasken des Hauptbereiches



Icon	Beschreibung
	Schließt die Detailmaske ohne Änderungen.
	Speichert vorgenommene Änderungen in der Detailmaske.
	Klont einen bestehenden Eintrag.
	Bietet einen Hilfetext via Tooltip zu einem spezifischen Parameter an.
	Auswahl einer Datei.
	Generiert ein Zertifikat.
	Download des ausgewählten Eintrages.
	Löschen des ausgewählten Eintrages.

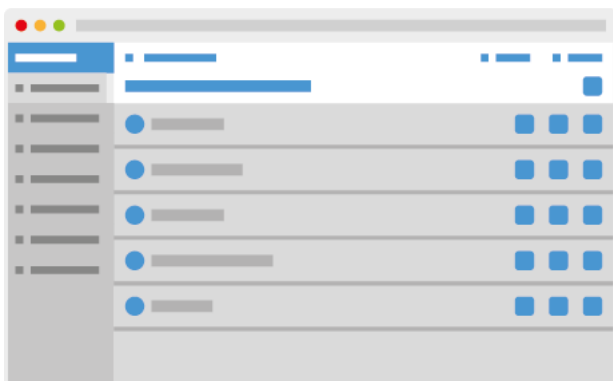
Menü-Einträge

Nachfolgend werden die einzelnen Menü-Einträge von Puppetry und die dahinter verbundenen Konfigurations- bzw. Analysefunktionen genauer beschrieben.

Admin-Modus

Der Admin-Modus dient der Konfiguration der Testdaten, die später im Test-Modus verwendet werden sollen. Bei diesen handelt es sich um die *Aufrufkontexte*, die *Versichertenkarten*, die *Praxiskarten* und die *Authentifizierungsdaten*.

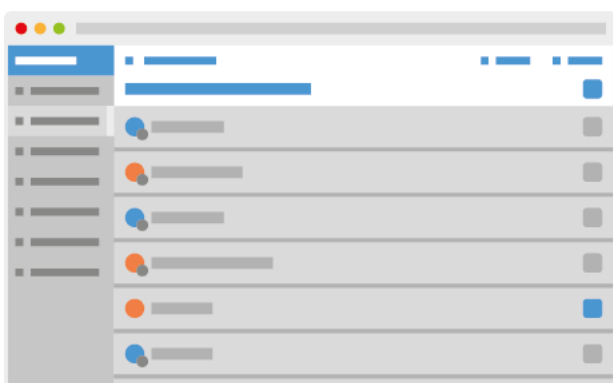
Aufrufkontext



Über den Menü-Eintrag *Aufrufkontexte* im Admin-Modus gelangt man zu einer Listenansicht der im Hauptbereich konfigurierten Aufrufkontexte. Es können neue angelegt bzw. bestehende angepasst und gelöscht werden. Pro Aufrufkontext wird das Tripel aus *Mandanten-ID*, *Clientsystem-ID* und *Arbeitsplatz-ID* angegeben. Pro Aufrufkontext können die diesem Aufrufkontext zugeordneten

Terminals verwaltet werden.

Versicherte

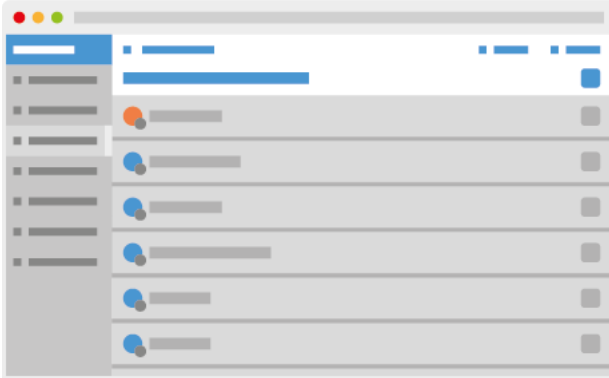


Über den Menü-Eintrag *Versicherte* gelangt der Nutzer zu der Verwaltung der Versichertenkarten (eGK und KVK). Jeder Karte kann ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden. Für die Inhalte gibt es Eingabefelder, die eine vereinfachte Bearbeitung der Versichertenstammdaten (VSD) erlauben. Darüber hinaus ist es auch möglich, generierte XML-Daten für die persönlichen

Versichertendaten (PD), die allgemeinen Versicherungsdaten (VD), die geschützten Versichertendaten (GVD) und den Prüfungsnachweis zu überschreiben und somit auch Datensätze zu

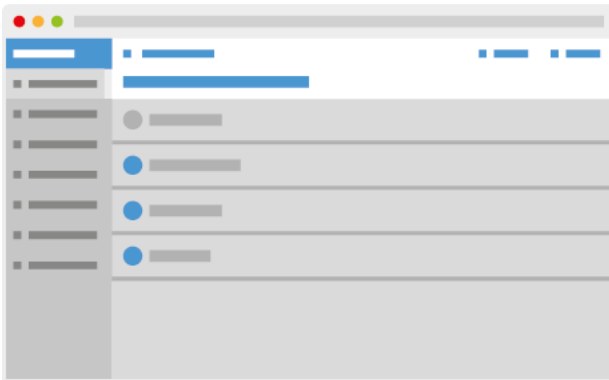
erzeugen, die nicht über die eigentlichen Konfigurationsparameter von Puppetry hergestellt werden können. Eine Beschreibung der einzelnen Parameter kann auf der Web-Oberfläche jeweils über den Hilfe-Button angezeigt werden.

Praxis



Über den Menü-Eintrag *Praxis* gelangt der Nutzer zu der Verwaltung der Praxiskarten (*SMC-B*, *HBA* und *vHBA*). Jede Karte kann ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden. Eine Beschreibung der einzelnen Parameter kann auf der Web-Oberfläche jeweils über den Hilfe-Button angezeigt werden.

Authentifizierung



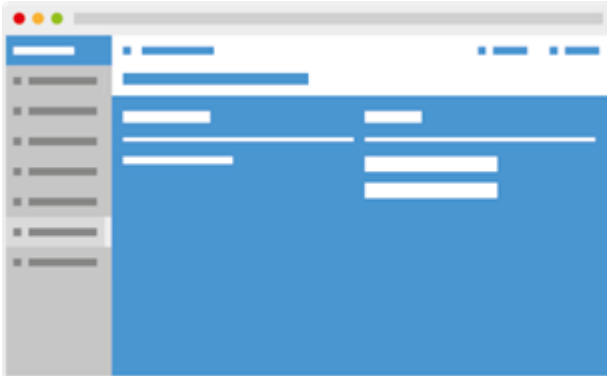
Über den Menü-Eintrag *Authentifizierung* gelangt der Nutzer zu der Verwaltung der Zugangsdaten für die Client-Authentifizierung. Pro Clientsystem-ID, die in einem der konfigurierten Aufrufkontexte angelegt wurde, wird hier automatisch ein entsprechender Eintrag erstellt. Darüber hinaus gibt es *Standard-Authentifizierungszugangsdaten*. Diese werden immer für die Clientsysteme verwendet, solange sie nicht explizit überschrieben worden sind. Die Zugangsdaten sind dann relevant, wenn als Authentifizierungsmechanismus *TLS* mit *Basic Authentication* oder *TLS* mit *Zertifikatsprüfung* verwendet wird.

Für die Nutzung der *Basic Authentication* muss ein Benutzername und ein Passwort hinterlegt werden. Für die *Zertifikatsprüfung* muss hingegen entweder ein Truststore hochgeladen werden oder ein Zertifikat durch Puppetry generiert werden, welches das Primärsystem dann ebenfalls verwendet.



Das Format der Keystores bzw. Truststores ist gemäß Konnektor Spezifikation PKCS12. Die Passwörter (Datei und Schlüssel) müssen für Puppetry immer *123456* sein.

Datensicherung



Im Menü-Eintrag *Datensicherung* kann der Nutzer den aktuellen Zustand seiner Konfiguration von Puppetry sichern, wiederherstellen oder eine neue Konfiguration einspielen. Um den aktuellen Zustand von Puppetry zu sichern, muss der Nutzer die Backup-Datei (.puppetbox) seiner aktuellen Konfiguration herunterladen. Bei Bedarf kann er diese Datei über die entsprechende

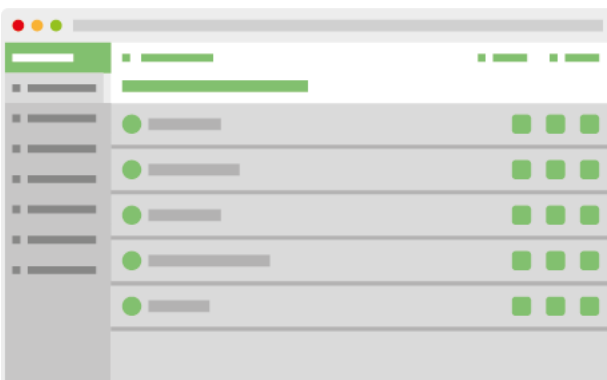
Funktion wieder in Puppetry einspielen. Zudem ist es möglich, fremde Konfigurationen einzuspielen. In diesem Fall werden die eigenen Testdaten überschrieben.

Sollen beispielsweise nur die gematik Testfälle ausgetauscht werden, sollte dies durch das direkte Austauschen der entsprechenden Dateien im Dateisystem erfolgen.

Test-Modus

Der Test-Modus enthält sämtliche Menü-Einträge, die für Zustandsänderungen der simulierten Umgebung relevant sind. So können hier beispielsweise Karten gesteckt und gezogen werden, Verbindungen getrennt, Fehler provoziert, Zeitverhalten manipuliert und Ereignisse ausgelöst werden. Darüber hinaus können in diesem Modus die linearen Testfälle ausgeführt werden.

Aufrufkontext



Über den Menü-Eintrag *Aufrufkontexte* im Test-Modus gelangt man zu einer Listenansicht im Hauptbereich. Bei den dort gelisteten Einträgen handelt es sich um die im Admin-Modus konfigurierten Aufrufkontexte. Pro Aufrufkontext kann in die jeweilige *Live-Terminal-Ansicht* dieses Kontextes gewechselt werden. Dazu muss auf das Terminal-Icon eines Eintrages geklickt werden.

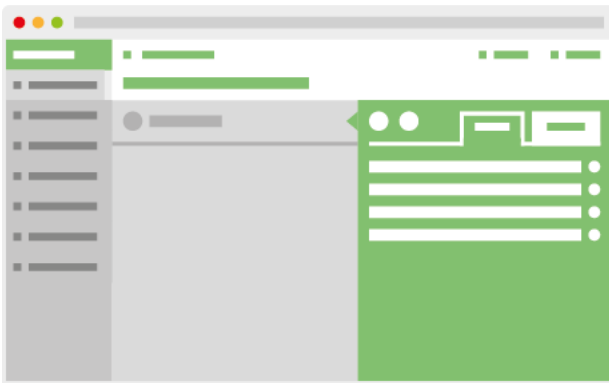


In der *Live-Terminal-Ansicht* hat der Nutzer die Möglichkeit, in angezeigten Terminals (analog zu einem physischen Terminal) Karten zu stecken oder zu entfernen. Dazu stehen dem Nutzer die vordefinierten Versicherten- und Praxiskarten zur Auswahl.

Das Stecken der Karte erfolgt per Drag'n'Drop. Im rechten Bereich kann eine Karte ausgewählt

und auf einen der freien grauen Slots im linken Bereich gezogen werden. Dadurch wird diese Karte in den jeweiligen Slot gesteckt. Soll die Karte aus dem Terminal gezogen werden, kann hierfür die entsprechende Schaltfläche des Terminals verwendet werden.

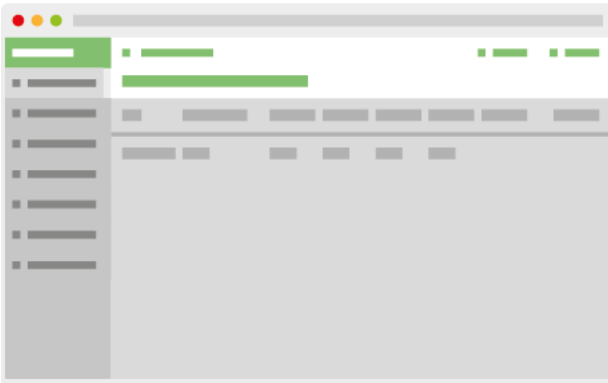
In der Live-Terminal-Ansicht kann auch die physikalische Trennung der Verbindung vom Kartenterminal zum Konnektor simuliert werden.



Pro Aufrufkontext kann zu der Verwaltung des *Übersteuertes Antwortverhalten* gewechselt werden. Diese Übersteuerung gilt entsprechend nur für den aktuellen Aufrufkontext. Der Nutzer hat die Möglichkeit, das Antwortverhalten für verschiedene Methoden zu manipulieren. So kann eingestellt werden, ob ein SOAP-Fehler oder eine technisch manipulierte Antwort statt der eigentlichen Antwort zurückgegeben werden soll. Die Antwort lässt sich dabei auch in Gänze durch eine benutzerdefinierte Antwort ersetzen. Zusätzlich kann die Ausführung einer Methode auch verzögert werden. Eine solche Verzögerung kann vor oder nach der fachlichen Bearbeitung der Methode erfolgen.

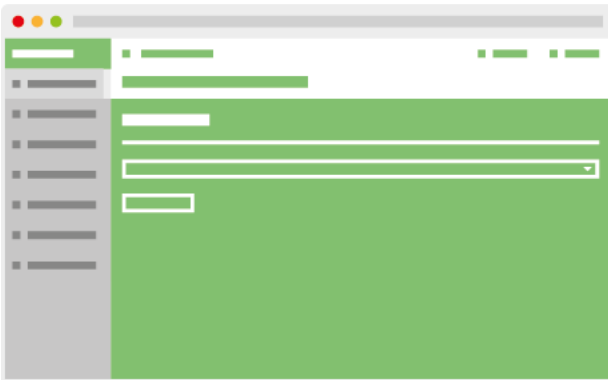
In der Listenansicht der Aufrufkontexte können diese mit einem Klick deaktiviert oder aktiviert werden. Ein deaktivierter Aufrufkontext ist über die Konnektor-Schnittstelle nicht nutzbar. Somit können Aufrufkontexte einfach deaktiviert werden, ohne die zugrundeliegende Konfiguration entfernen zu müssen.

Abonnenten



Unter dem Menü-Eintrag *Abonnenten* können alle Abonnenten des Ereignisdienstes eingesehen werden, die sich über die Ereignisdienst-Schnittstelle des Konnektors bei Puppetry registriert haben. Pro Abonnent wird angezeigt, mit welchen Parametern dieser sich registriert hat. Zusätzlich ist es möglich, bestehende Abonnenten zu entfernen. Somit lässt sich zum Beispiel eine Situation simulieren, in welcher der Konnektor aufgrund von Netzwerk-Problemen dreimal Ereignisse nicht erfolgreich abschicken konnte und daher den Abonnenten aus seiner Liste entfernt hat.

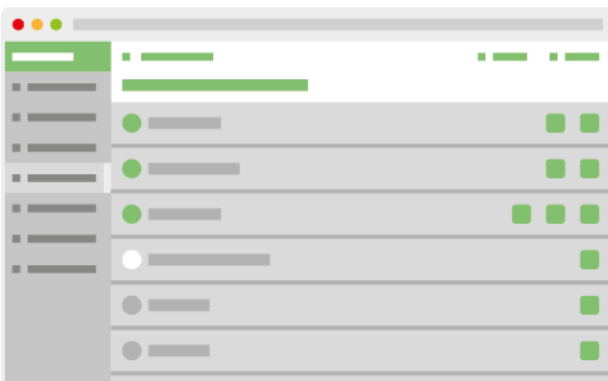
Ereignisse



Unter dem Menü-Eintrag *Ereignisse* hat der Nutzer die Möglichkeit, alle spezifizierten Ereignisse manuell auszulösen. Für jedes Ereignis sind verschiedene Konfigurationsparameter möglich (siehe Konnektor-Spezifikation).

Die Werte eines zuvor ausgelösten Ereignisses werden gespeichert und sind beim nächsten Auswählen des Ereignisses vorausgewählt.

Testlabor



Unter dem Menü-Eintrag *Testlabor* kann der Nutzer die vordefinierten linearen Testfälle durchführen. Das Testlabor enthält eine Übersicht der vorhandenen *Testsets* (z.B. des *gematik Testfallkatalogs*). Durch einen Klick auf das Testset werden dessen Details angezeigt. Hier besteht die Möglichkeit eine Testfallübersicht über die im Set enthaltenen Testfälle und deren jeweiligen Status zu erhalten. Die Testfallübersicht kann als PDF heruntergeladen werden. Das PDF enthält die Anzahl der erfolgreichen, der fehlerhaften und der nicht unterstützten sowie der nicht

durchgeführten Testfälle und die Einzelergebnisse der Testfälle im Testset. Alternativ kann dazu auch eine Excel-Liste heruntergeladen werden, die neben der Zusammenfassung auch die Definitionen der Testfälle enthält. Um alle Testfallergebnis PDFs und die Testfallübersicht als Paket herunter zu laden, bietet Puppetry eine ZIP-Datei an. Mit einem Klick auf das rechtsbündig angezeigte Icon wird die Übersicht der im Set enthaltenen Testfälle angezeigt.

Handelt es sich um einen gematik Testfallkatalog, enthält der Detailbereich des Testsets einen Abschnitt für den gematik Testbericht. Hier können die Werte *Primärsystem Name*, *Primärsystem Version*, *Aktenzeichen* der Leistungserbringerorganisation, *gematik ZLS* und der *Name des Testers* gepflegt werden. Diese Werte werden bei der Generierung des Testberichts und der Testfallergebnis-PDFs verwendet. Für die Generierung des Testberichts müssen mindestens Primärsystem Name, Primärsystem Version und der Name des Testers angegeben sein. Der offizielle *Testbericht* kann als reines PDF oder als Paket mit allen Testfallergebnis-PDFs in Form einer ZIP-Datei heruntergeladen werden. Zusätzlich können im Detailbereich des Testsets auch alle bisherigen Testfallergebnisse des Testsets komplett zurückgesetzt werden.



In der Übersicht der Testfälle besteht die Möglichkeit, durch einen Klick auf einen Testfall dessen Details anzuzeigen. Zu diesen Details gehören der Name des Testfalls, eine Kurzbeschreibung, sowie eine ausführliche Beschreibung. Handelt es sich um einen als optional markierten Testfall (von einer SOLL- oder KANN-Anforderung abgeleitet) im Testset, kann der Nutzer die-

sen als durch sein Primärsystem nicht unterstützt kennzeichnen. Diese Kennzeichnung erfolgt über eine Checkbox „Testfall wird durch das Primärsystem nicht unterstützt.“ in der Detailansicht des jeweiligen Testfalls. Wurde der Testfall bereits durchgeführt, wird zusätzlich der Zeitpunkt der letzten Testausführung angezeigt. Im Menüband unterhalb der Schaltfläche „Modus wechseln“ kann durch einen Klick auf das jeweilige Icon direkt eine Testfallübersicht als auch der Testbericht (nur innerhalb des gematik-Testfallkatalogs) als ZIP-Datei heruntergeladen werden.

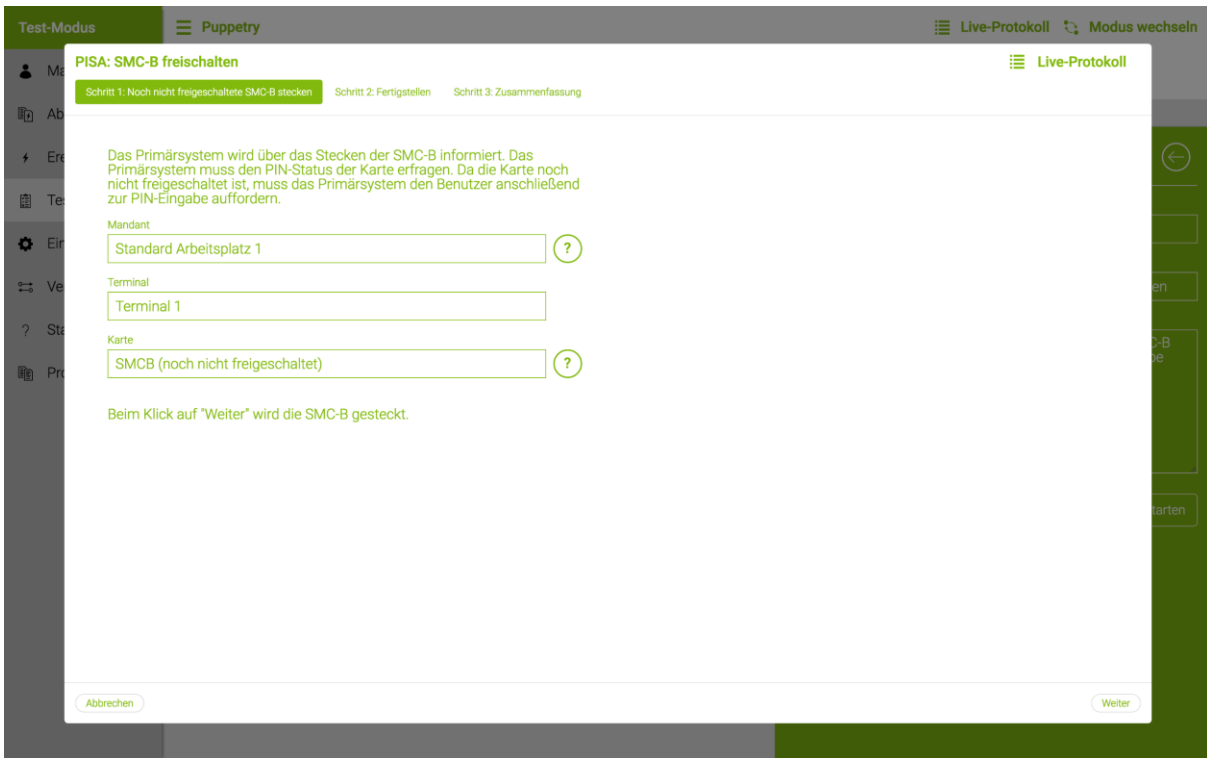
Für jeden durchgeführten Testfall kann das Testfallergebnis und die dazugehörigen Log-Dateien des letzten Testdurchlaufes abgerufen werden. Zusätzlich ist es möglich, eine Zusammenfassung über alle Testfälle als PDF-Bericht herunterzuladen. In diesem Bericht ist jeder Testfall einzeln mit seinem letzten Ergebnis aufgeführt.

Nach dem Start eines Testfalls wird dem Nutzer ein Pop-Up angezeigt, welches ihn Schritt-für-Schritt durch den Testfall führt. Es wird immer der Name des Testfalls (die Kurzbeschreibung kann via Tooltip auf dem Namen eingeblendet werden) und der aktuelle Testfallschritt angezeigt. Die Anzahl der Schritte ist vom jeweiligem Testfall abhängig. Nach Abschluss des Testfalls wird ein Testfallergebnis erstellt. Dieses Testfallergebnis umfasst im PDF eingebettet auch sämtliche Log-Dateien (diese können in einem entsprechenden PDF-Reader eingesehen werden).

Zu Beginn eines Testfalls wird automatisch ein initialer Zustand hergestellt. Das heißt, es werden alle Karten gezogen bzw. die für den Test relevanten Karten gesteckt und ggf. Einstellungen wie der zu verwendende Authentifizierungsmechanismus angepasst. Nach Beendigung des Testfalls, wird wieder der Zustand vor der Ausführung des Testfalls hergestellt. Also die in dem Testfall benutzten Karten werden gezogen und die vor der Testfallausführung steckenden Karten werden wieder gesteckt. Auch die im Rahmen des Testfalls angepassten Einstellungen werden zurückgesetzt (inkl. ggf. notwendigem Neustart von Puppetry).

Bei einem Testfall kann es zu einer technischen Validierung durch Puppetry kommen (beispielsweise, ob eine bestimmte Methode vom Primärsystem aufgerufen wurde). In der Regel ist es möglich, eine Validierung erneut anzustoßen, sofern die Methode nicht innerhalb des dafür vorgesehenen Zeitfensters vom Primärsystem aufgerufen wurde. Neben der technischen Validierung kann der Nutzer aufgefordert werden, eine Überprüfung des Verhaltens am Primärsystem vorzunehmen. In diesem Fall bietet Puppetry bei dem zugehörigen Eintrag eine Checkbox an, über die die Beobachtung zum erwarteten Verhalten am Primärsystem aktiv bestätigt werden muss.

Die nachfolgenden Bilder zeigen einen beispielhaften Ablauf eines Testfalls:



Test-Modus Pupperty Live-Protokoll Modus wechseln

PISA: SMC-B freischalten

Schritt 1: Noch nicht freigeschaltete SMC-B stecken Schritt 2: Fertigstellen Schritt 3: Zusammenfassung

Das Primärsystem wird über das Stecken der SMC-B informiert. Das Primärsystem muss den PIN-Status der Karte erfragen. Da die Karte noch nicht freigeschaltet ist, muss das Primärsystem den Benutzer anschließend zur PIN-Eingabe auffordern.

Mandant: ?

Terminal:

Karte: ?

Beim Klick auf "Weiter" wird die SMC-B gesteckt.

Prüfergebnis: Prüfung wiederholen

Die Aktion wurde einer automatischen Prüfung durch Pupperty unterzogen:

- Die Methode 'GetPinStatus' wurde nicht innerhalb des erwarteten Zeitintervalls von 3 Sekunden durch das Primärsystem aufgerufen. ❌
- Die Methode 'VerifyPin' wurde nicht innerhalb des erwarteten Zeitintervalls von 3 Sekunden durch das Primärsystem aufgerufen. ❌
- Der PIN-Status 'RETRY_COUNTER' der zu prüfenden Karte entspricht nicht dem erwarteten PIN-Status 'NO_ERROR'. ❌

Wurde die Aktion aus Sicht des Primärsystems erfolgreich abgeschlossen?

Ja, die Aktion wurde aus Sicht des Primärsystems erfolgreich abgeschlossen ✓

Hier haben Sie die Möglichkeit einen zusätzlichen Kommentar einzugeben.

Zur Dokumentation des Ergebnisses haben Sie hier die Möglichkeit Dateien hochzuladen.

Dateien auswählen via Klick oder via Drop

Abbrechen Weiter

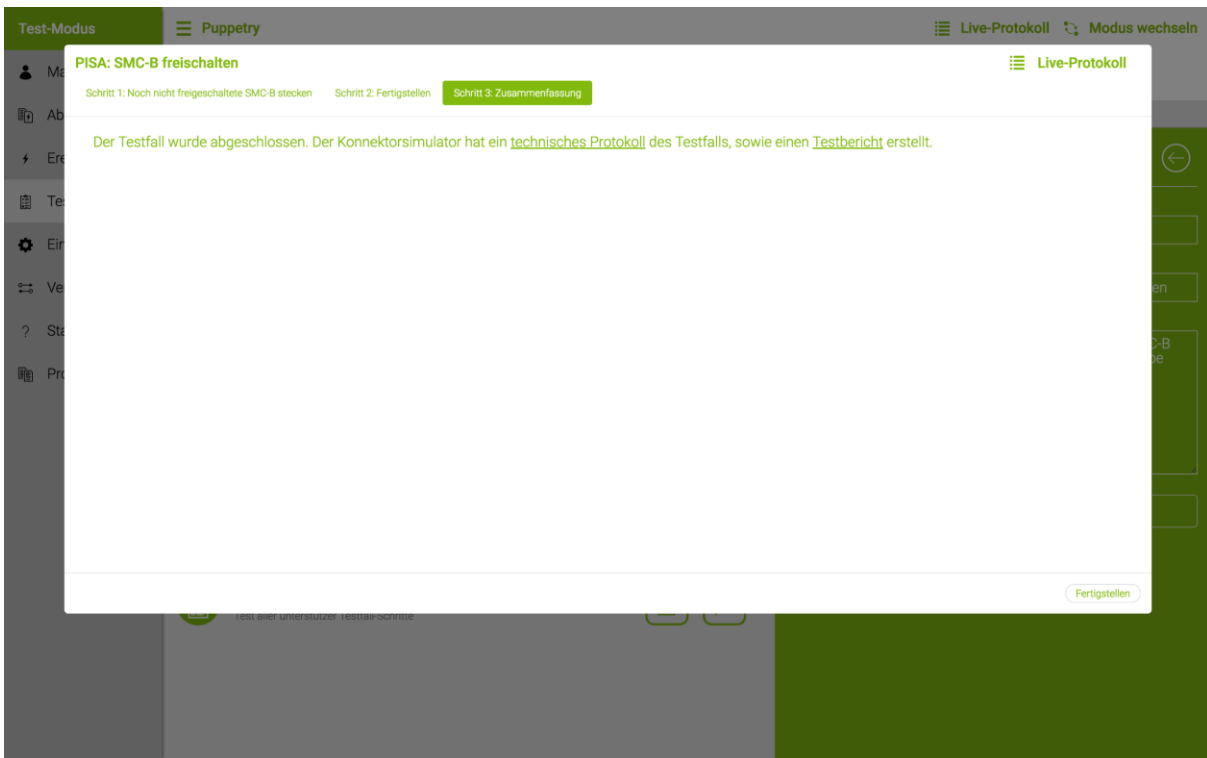
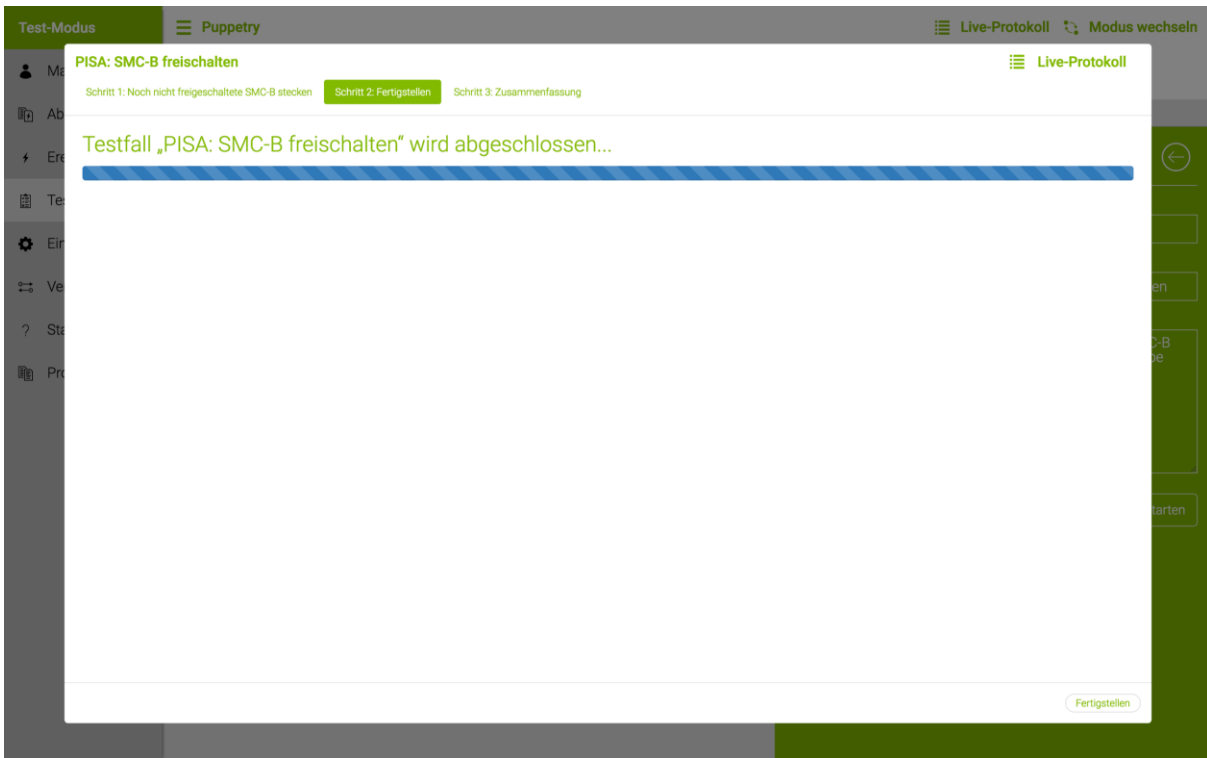
Test-Modus Pupperty Live-Protokoll Modus wechseln

PISA: SMC-B freischalten

Schritt 1: Noch nicht freigeschaltete SMC-B stecken Schritt 2: Fertigstellen Schritt 3: Zusammenfassung

Klicken Sie auf "Weiter" um den Testfall abzuschließend und eine Zusammenfassung des Testfalls zu erstellen. Das System wird anschließend auf den ursprünglichen Zustand zurückgesetzt.

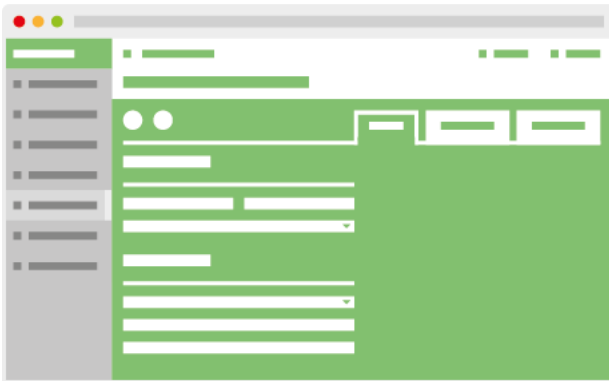
Abbrechen Weiter



Vom Modus unabhängige Menü-Einträge

Neben den Menü-Einträgen, die nur im Admin-Modus beziehungsweise im Test-Modus zu sehen sind, gibt es Menü-Einträge, die dem Nutzer immer zur Verfügung stehen, da sie für beide Betriebsmodi relevant sind.

Einstellungen



Im Menü-Eintrag *Einstellungen* können Anpassungen bezüglich der Konnektor-Schnittstelle, des Ereignisdienstes und der Anzeigeeinstellung der Web-Oberfläche vorgenommen werden.

Für die *Konnektor-Schnittstelle* kann konfiguriert werden, über welches Interface und welchen HTTPS Port diese erreichbar ist. Gleiches gilt für die Dienste, jedoch kann hier zusätzlich

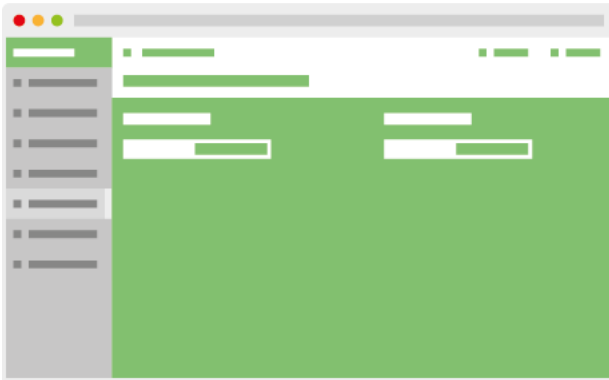
eingestellt werden, über welchen Authentifizierungsmechanismus die Konnektor-Dienste erreichbar sind.

Für den *Dienstverzeichnisdienst* kann eingestellt werden, ob dieser weiterhin über HTTP erreichbar ist, während die Konnektor-Dienste nur über eine TLS-gesicherte Verbindung angebunden sind. Funktional kann für den Dienstverzeichnisdienst das Antwortverhalten gesteuert werden, es kann eine unterschiedliche Dienstversion oder eine angepasste Dienst-URL provoziert werden.

Für den *Ereignisdienst* des Primärsystems kann gleichfalls der Authentifizierungsmechanismus festgelegt, sowie für den Ereignisdienstclient das Kommunikationsverhalten verändert werden. Da das konkrete Kommunikationsverhalten von dem verwendeten Konnektor abhängt, kann somit der Umgang des Primärsystems mit verschiedenen möglichen Verhalten getestet werden. Um die Kommunikation des Ereignisdienstes unabhängig von der Nutzung des Systeminformationsdienstes (Subscribe) testen zu können, kann eine statische Ereignis-Endpunkt-Adresse angegeben werden. Ist diese Adresse eingerichtet, werden alle auftretenden Ereignisse – unabhängig von Abonnements – immer an diese Adresse geschickt. Um das Erneuern von Abonnements zu testen, kann die Gültigkeitsdauer eines Abonnements angepasst werden (Standard 25 Stunden). Zur Simulation von Netzwerkproblemen besteht die Möglichkeit zur Unterdrückung von Ereignis-Nachrichten an das Primärsystem.

Für die Web-Oberfläche kann zusätzlich noch eingestellt werden, in welchem Intervall das Live-Protokoll aktualisiert und in welchem Intervall die Live-Terminal-Anzeige aktualisiert wird (z.B. um über das programmatische Ziehen eine Karte informiert zu werden).

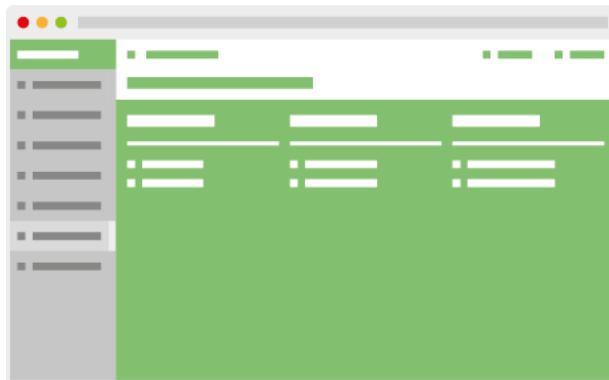
Status



Unter dem Menü-Eintrag *Status* kann der Nutzer den Status der Konnektor-Dienste und den Status der virtuellen Verbindung zur Telematikinfrastruktur ändern. Wird der Konnektor-Status auf *AUS* gestellt, stellt dies die Situation nach, in der ein physischer Konnektor ausgeschaltet beziehungsweise vom Strom genommen wird. Wird der Verbindungsstatus auf *OFFLINE* gestellt, stellt

dies die Situation nach, in der bei einem physischen Konnektor die Kabelverbindung zur Telematikinfrastruktur unterbrochen wird.

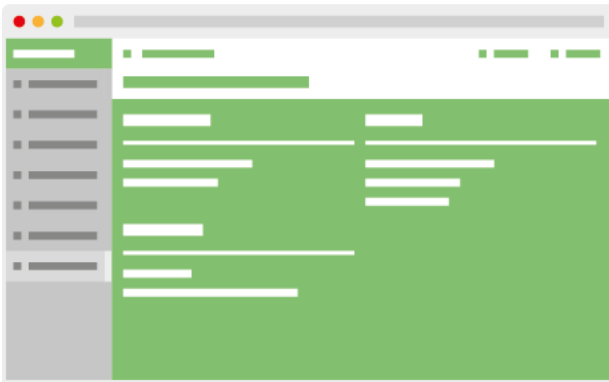
Protokolle



Unter dem Menü-Eintrag *Protokolle* können die zur Laufzeit durch Puppetry angelegten Log-Dateien *Basis* (enthält alle Log-Einträge der Puppetry-Software), *SOAP* (enthält alle SOAP-Nachrichten) und *Schnittstelle* (entspricht den Einträgen im Live-Protokoll) heruntergeladen werden. Zu jedem Protokoll-Typ kann die aktuelle Log-Datei oder eine komprimierte Version, die zusätz-

lich alle vergangenen Log-Dateien beinhaltet, heruntergeladen werden.

Systeminfo



Unter dem Menü-Eintrag *Systeminfo* erhält der Nutzer genauere Informationen zur der aktuellen Installation von Puppetry. Dazu gehört die Version der Software, der Konfiguration und die Lizenzinformationen.

Benutzerkonfiguration

Das Konfigurationsverzeichnis von Puppetry befindet sich im Unterordner *conf* des Installationsverzeichnisses – hier erfolgt in den spezifischen Unterverzeichnissen die Ablage von Testdaten- und Testfallkonfigurationen (*testData.xml* und *testCases.xml*).

Im Auslieferungszustand befindet sich im Verzeichnis *gematik* die geschützte Konfiguration, welche für den Konformitätstest notwendig ist. Im Verzeichnis *user* werden die vom Nutzer mittels Puppetry-Interface erstellten Testdaten abgelegt. Weitere Unterverzeichnisse enthalten die *testData.xml* oder *testCases.xml* Dateien für die Testdaten und Testfallkonfigurationen. Ein eigenes Unterverzeichnis mit einer *testCases.xml* Datei wird im *Testlabor* jeweils als ein eigenständiges *Testsets* angezeigt. Dabei ergibt sich der Name des *Testsets* aus dem Namen des Unterverzeichnisses.

Wenn verschiedene Testdatenkonfigurationsdateien verwendet werden, ist es wichtig, dass die gewählten technischen IDs der Konfigurationsobjekte über alle Konfigurationsdateien hinweg eindeutig sind. Sollten mehrere Testdatenkonfigurationsdateien vorhanden sind, die die gleichen IDs verwenden, blockiert dies den Start von Puppetry. Eine entsprechende Fehlermeldung wird dann beim Ausführen des Programmes angezeigt. Verzeichnisse oder Testdaten sollten darum niemals innerhalb des Verzeichnisses lediglich kopiert werden, sondern bedürfen auch immer einer Anpassung durch den Nutzer.

Glossar

Konnektor

Der Konnektor koordiniert und verschlüsselt die Kommunikation zwischen Clientsystem, eGK, HBA/SMC und zentraler Telematikinfrastruktur. Er stellt damit das Bindeglied zwischen diesen Komponenten auf Leistungserbringerseite bzw. eKiosk und Telematikinfrastruktur dar. Der Konnektor ist ein Produkttyp.

SMC-B (Institutionskarte)

Die Institutionskarte entspricht technisch weitgehend dem Heilberufsausweis (HBA), bezieht sich jedoch auf eine organisatorische Instanz des Gesundheitswesens (z.B. Praxis, Apotheke, Krankenhaus). Die Institutionskarte wird auch als Security Module Card Typ B (SMC-B) bezeichnet.

Terminal (Kartenterminal)

Migrationsfähige (entsprechen der aktuellen eHealth-Kartenterminal Spezifikation [gemSpec_KT]) Kartenlesegeräte, welche zusätzlich eine USB- bzw. V24-Schnittstelle unterstützen, sowie mit einem Upgrade ohne Austausch der Geräte zu einem vollwertigen LAN-fähigen „eHealth-KT“ aufgerüstet werden können. Kartenlesegeräte auf dieser Basis müssen an der V.24- und/oder USB-Schnittstelle mindestens den „Basis Command Set (BCS)“ unterstützen.

Primärsystem

Ein IT-System, das bei einem Leistungserbringer eingesetzt wird – z.B. eine Praxisverwaltungssoftware (PVS), ein Krankenhausinformationssystem (KIS) oder eine Apothekensoftware (AVS) – und sich unter dessen administrativer Hoheit befindet. Das Primärsystem ist kein Bestandteil der TI-Plattform.

Quelle: **gematik Glossar**

https://www.gematik.de/cms/media/dokumente/gemGlossar_V400.pdf