Gematik Authenticator - Entwicklervariante mit Mockmodus

Anwendungen, die den Gematik Authenticator nutzen und integrieren wollen, sollten eine entwicklungsbegleitende Integrationsmöglichkeit erhalten. Daher wird der Authenticator in zwei unterschiedlichen Varianten ausgeliefert. Zum einen als Variante für den Produktivbetrieb und zum anderen als Entwicklervariante. Die Entwicklervariante bietet einen sogenannten Mockmodus und mehr Loggingmöglichkeiten, um so die Einbindung in das eigene Projekt einfacher und effizienter zu gestalten.

Für die Entwicklervariante muss die ***gematik Authenticator Setup - Mock Version X.X.X.exe*** aus dem gematik Sharepoint heruntergeladen werden.

Mockmodus

Innerhalb der Entwicklervariante ist ein sogenannter Mockmodus integriert, welcher die Verbindung eines Konnektors simulieren kann. Somit ist es möglich, Funktionstests ohne physisch vorhandenen Konnektor, Kartenterminal und Smartcards durchzuführen.

Hinweis: *Nutzen Sie die Entwicklervariante nicht in der Produktivumgebung, da sonst vertrauliche Informationen geloggt werden - die aktuelle Produktiv-Version finden Sie ebenfalls im gematik Sharepoint!*

Funktionsweise

Für den Mockmodus werden nur Software-Komponenten benötigt und keine physischen Karten (HBA, SMC-B), Kartenterminals oder Konnektoren.

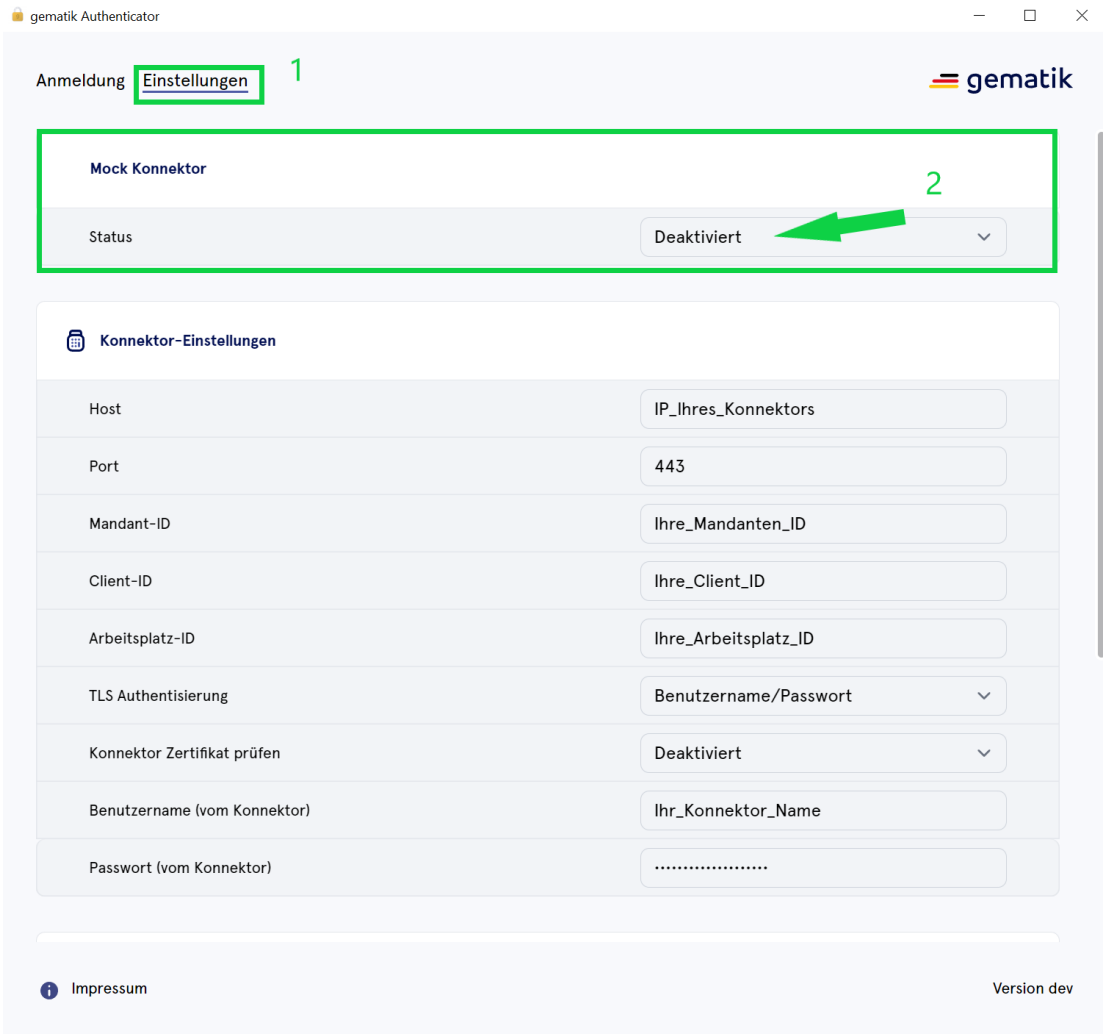
Innerhalb des Authenticators erfolgt der Zugriff auf die Konnektor-Funktionen über zusätzlich implementierte Funktionen im Code. Der Authenticator erstellt im Mockmodus mittels *jose-Library* unterschiedliche JWTs. Diese Tokens werden mittels der in den Einstellungen hinterlegten privaten Schlüssel (HBA oder SMC-B) signiert (JWS) und enthalten das Kartenzertifikat. Der Authenticator kann nun die physische Hardware (Konnektor, Kartenterminal, Smartcards) simulieren.

Im Code ist keine zentrale Klasse für den Mockmodus angelegt, sondern die notwendigen Funktionen sind überall dort implementiert, wo eine Interaktion mit einem Konnektor stattfinden würde.

Nutzungshinweise

Um den Mockmodus nutzen zu können, starten Sie die zuvor heruntergeladene ***gematik Authenticator Setup - Mock Version X.X.X.exe*** und öffnen Sie die Einstellungsseite (**1**).

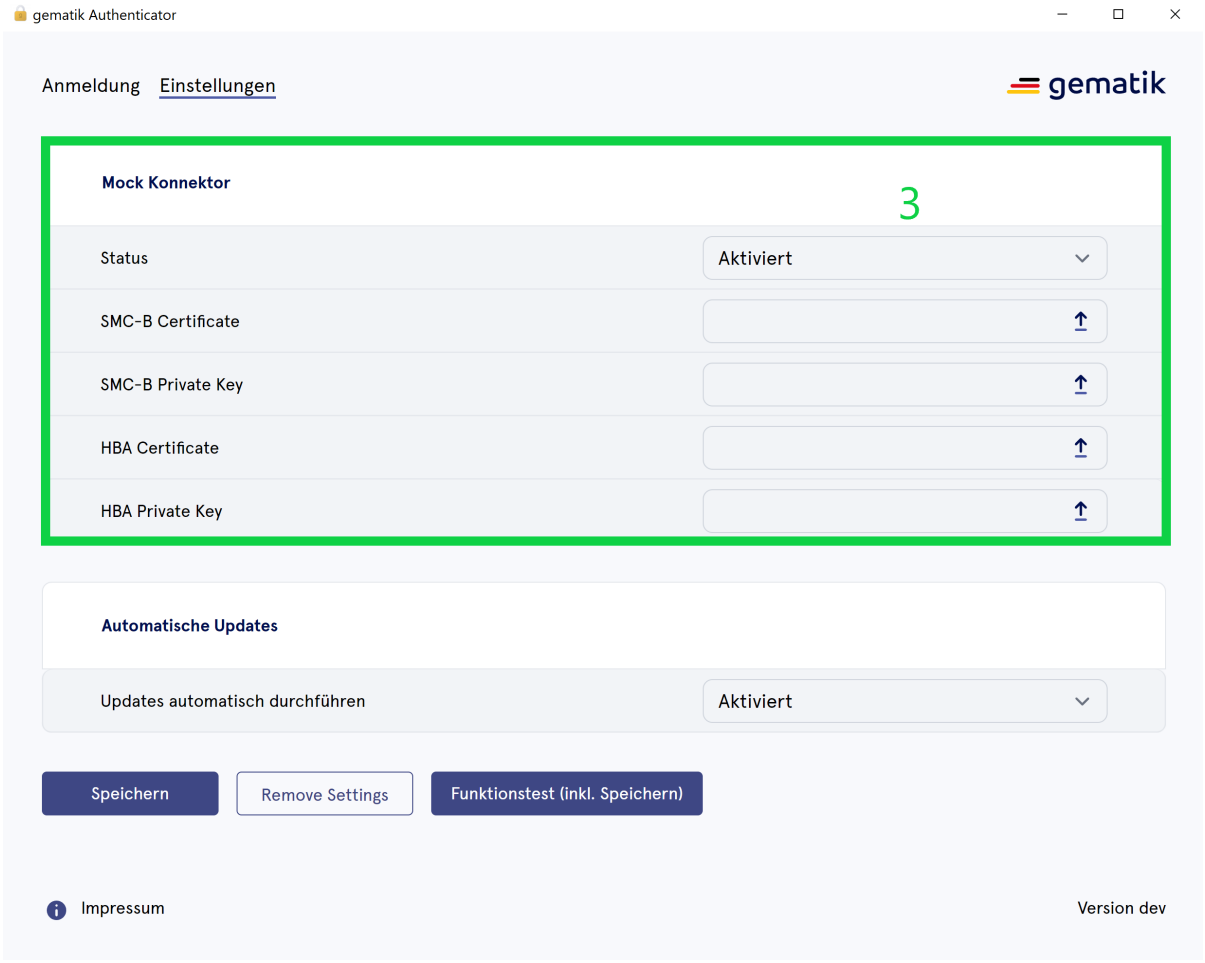
Im Gegensatz zur Produktiv-Version befindet sich hier nun ein weiteres Feld **Mock Konnektor** - dieser ist per Default deaktiviert. Klicken Sie in das Drop-Down-Menü (**2**), um den Mockmodus zu aktivieren.



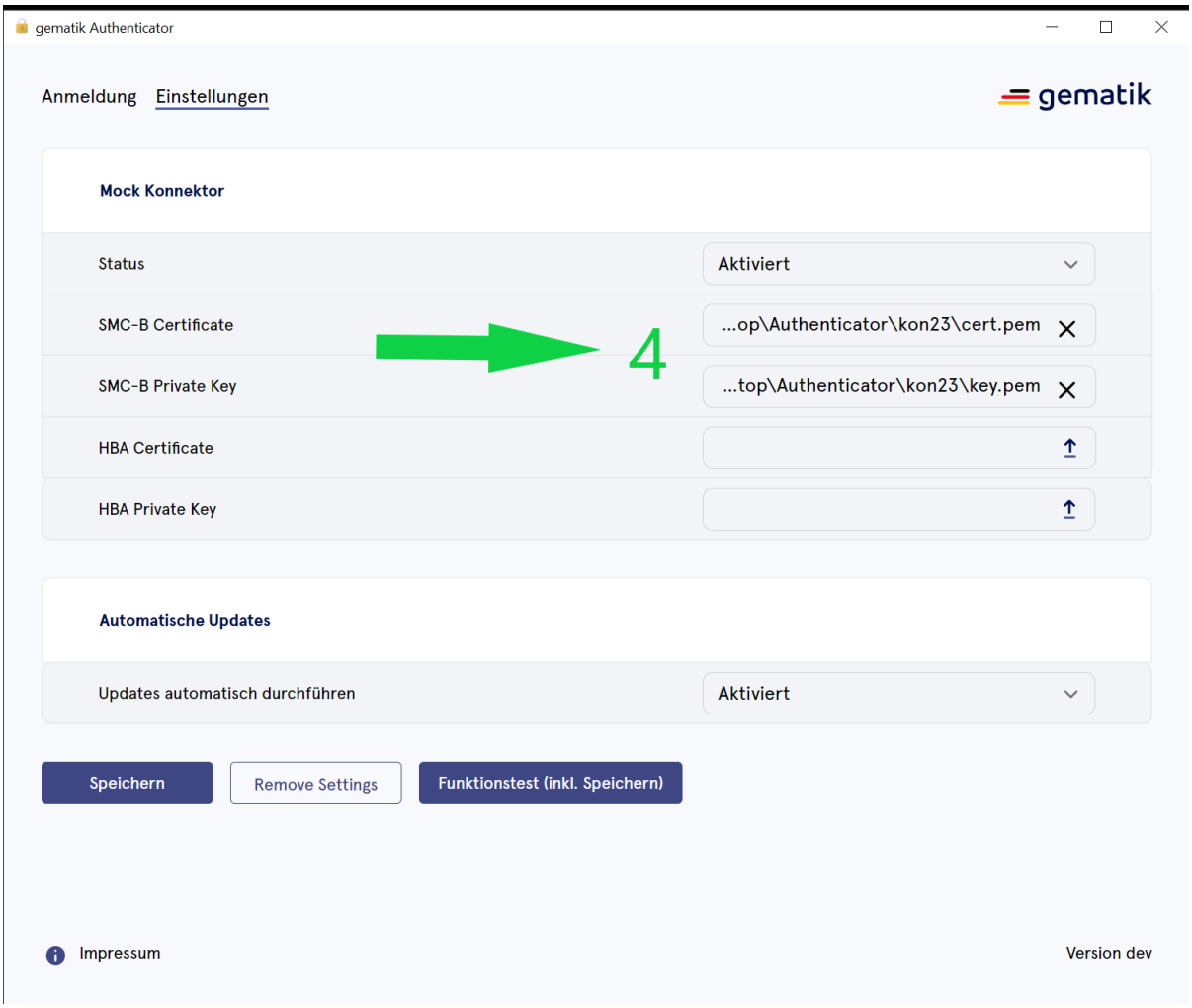
Ist der Mockmodus aktiviert, verändert sich die Einstellungsmaske und es erscheinen vier neue Möglichkeiten für die Dateieingabe (3).

Sie haben nun die Möglichkeit, Zertifikate für die Nutzung einer SMC-B Karte (Zertifikat & privater Schlüssel) oder die Nutzung einer HBA Karte (Zertifikat & privater Schlüssel) hochzuladen. Beispielzertifikate (gültige, abgelaufene, zurückgezogene, ...) werden vom Authenticator Team der gematik GmbH im gematik Sharepoint bereitgestellt, um so sämtliche Testszenarien abzudecken - Sie können jedoch jederzeit Ihre Zertifikate und privaten Schlüssel verwenden, um die Mockmodus-Funktion zu nutzen.

Hinweis: *Achten Sie darauf, dass die Zertifikate im .pem-Format vorliegen, da es sonst zu einer Fehlermeldung kommt. Liegen Ihnen beispielsweise die benötigten Zertifikate nur im p12-Format vor, finden Sie hier eine mögliche Vorgehensweise, diese ins benötigte .pem-Format zu exportieren: Umwandlung einer p12-Datei in PEM-Format*



Laden Sie die gewünschten Zertifikate hoch (4), so wie im folgenden Beispiel (für z. B. Nutzung SMC-B) und achten Sie darauf, dass diese im .pem-Format vorliegen.



Nun können Sie einen Funktionstest durchführen, um die Erreichbarkeit der IDPs zu testen. Innerhalb des Mockmodus wird nur die Konnektor-Kommunikation gemockt, die Erreichbarkeit der jeweiligen IDPs ist nicht von der Mockmodus-Einstellung betroffen.

Hinweis: *Der Funktionstest wird derzeit um Testszenarien für die hochgeladenen Zertifikate erweitert, sodass Sie demnächst noch mehr direktes Feedback erhalten.*

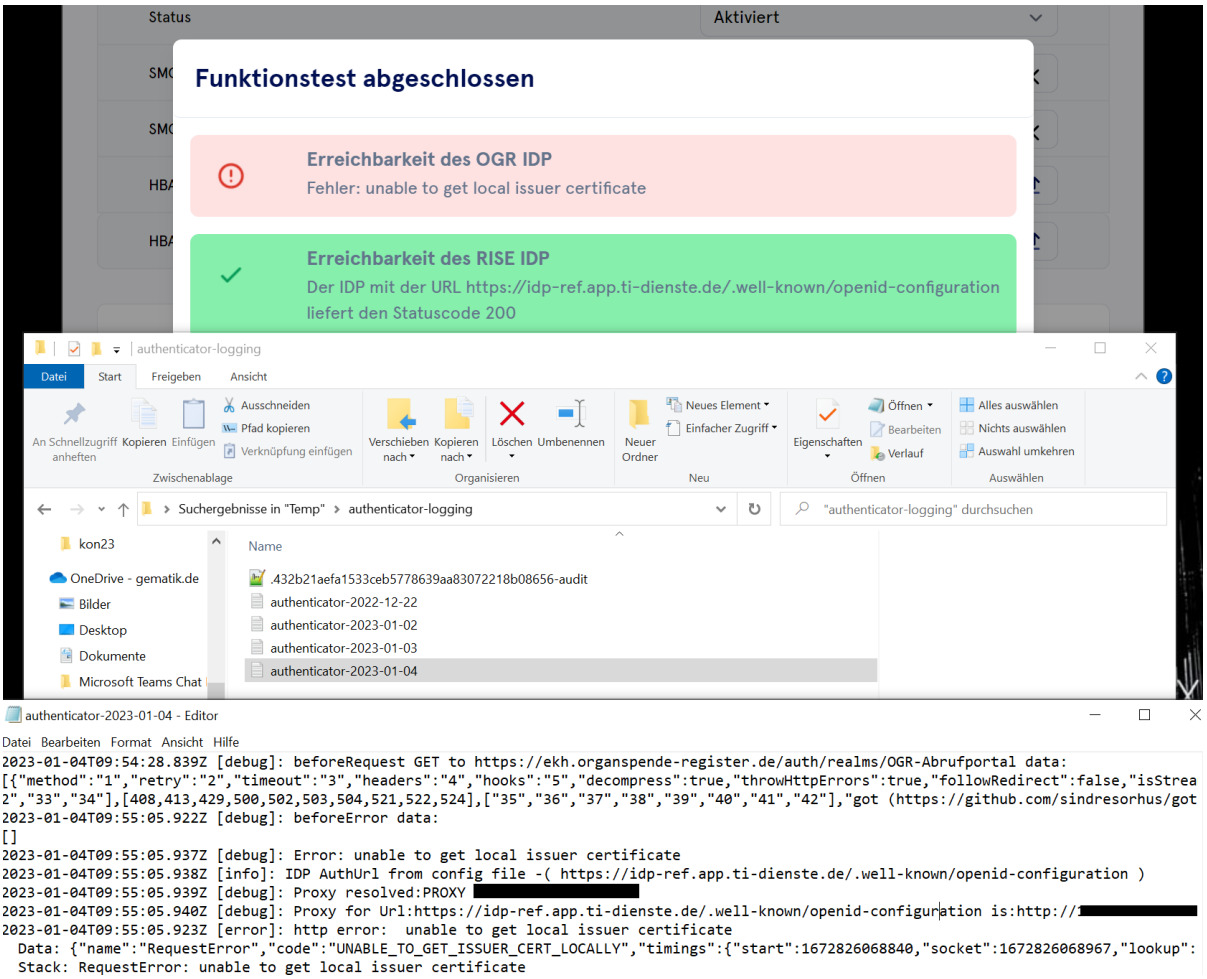
Erweitertes Logging

Innerhalb der Entwicklervariante wird der ganze Netzwerk-Traffic des Authenticators innerhalb eines zentralen Logfiles geloggt. Dieses Logfile umfasst die Requests und Responses, die innerhalb des Authenticators stattfinden.

Das Logfile wird automatisch angelegt und befindet sich unter ***C:/Users/{Nutzerverzeichnis}/AppData/Local/Temp/authenticator-logging /authenticator-${datum}.log***

Achten Sie daher darauf*, die Entwicklervariante nicht in der Produktivumgebung zu verwenden, da sonst vertrauliche Informationen geloggt werden - die aktuelle Produktiv-Version finden Sie ebenfalls im gematik Sharepoint!*

Treten - wie im folgenden Beispielausschnitt - Fehler auf, können Sie mit Hilfe des angelegten Logfiles leichter mögliche Ursachen untersuchen.



Exkurs Fachbegriffe (Kleines Glossar)

Um den Einstieg mittels dieser Dokumentation zu erleichtern, werden nachfolgend genutzte Fachbegriffe, die in dieser Dokumentation Anwendung finden, kurz erläutert.

HBA

Der elektronische Heilberufsausweis (HBA) ist der zentrale Zugang zu wichtigen Telematikinfrastruktur-Anwendungen. Er ist z.B. für das Auslesen und Signieren des Notfalldatensatzes notwendig. Benötigt wird er außerdem, um Arztbriefe, Befunde, E-Rezepte und elektronische Arbeitsunfähigkeitsbescheinigungen (eAU) rechtssicher elektronisch zu signieren.

Eine detaillierte Definition finden Sie auf unserer gematik Homepage (HBA).

SMC-B

Bei der SMC-B handelt es sich um den elektronischen Ausweis für medizinische Einrichtungen: die Security Module Card Typ B.

Die Security Module Card Typ B ist eine institutionsbezogene Smartcard. Die SMC-B dient in der Telematikinfrastruktur als Sicherheitsmodulkarte. Sie repräsentiert mit ihren kryptographischen Schlüsseln und Zertifikaten eine Institution innerhalb der Telematikinfrastruktur. Der Namenszusatz "-B" ist historisch bedingt.

Eine detaillierte Definition finden Sie auf unserem Fachportal (SMC-B).

IDP

IDP steht für "Identity Povider" - Ein sogenannter Identitätsanbieter speichert und verwaltet die digitalen Identitäten von Benutzern, so kann dieser Benutzeridentitäten beispielsweise anhand von Kombinationen aus Benutzernamen und Passwort verifizieren.

JWT

JWT (JSON Web Token) ist ein Open Standard (RFC 7519), der eine sichere Art von Daten bereitstellt, die von einer Partei an eine andere übertragen werden können. Ein JWT besteht aus drei Teilen: einem Header, einem Payload und einer Signatur. Der Header enthält Informationen darüber, wie die Signatur generiert wurde, während der Payload Angaben darüber enthält, was in dem JWT enthalten ist. Die Signatur hilft, die Integrität der Daten im JWT zu gewährleisten, indem sie sicherstellt, dass sie nicht verändert wurden, während sie übertragen werden.

JWS

JWS (JSON Web Signature) ist ein Teil des JWT-Standards (RFC 7515) und bezieht sich spezifisch auf die Signatur in einem JWT. Die Signatur wird verwendet, um die Integrität der Daten im JWT zu gewährleisten und sicherzustellen, dass sie nicht verändert wurden, während sie übertragen wurden.