Der ePb-Server

Spezifikation der Schnittstelle

von

Sebastian Süß, Ingo Graser und Burkhardt Renz

Projektgruppe e.P.b.
Technischer Bericht Fachhochschule
Gießen-Friedberg

Version 1.0 – März 2005

Inhaltsverzeichnis

Vorbemerkung

Das vorliegende Dokument beschreibt die Schnittstellen eines ePb-Servers.

Die Schnittstellen wurden auf der Basis des Vorentwurfs zum neuen Personenstandsgesetz entwickelt, wie er im Jahr 2004 vorlag. Die aktuelle Fassung des Gesetzesentwurfs vom Beginn 2005 ist noch nicht berücksichtigt. Dieser aktuelle Entwurf hat Auswirkungen auf die Spezifikation, im Wesentlichen jedoch auf Details und nicht auf das Prinzip der Schnittstelle.

Kapitel 1

Aufgaben eines ePb-Servers

1.1 Zweck

Ein ePb-Server verwaltet Personenstandseinträge. Er ist der Treuhänder, der die elektronischen Einträge sicher verwahrt.

Der Begriff „sicher“ ist in doppeltem Sinn zu verstehen:

- Die elektronischen Personenstandseinträge sind durch eine digitale Signatur unterschrieben, deren Gültigkeit und Überprüfbarkeit langfristig garantiert werden muss.
- Der ePb-Server sorgt für die sichere Speicherung der Einträge und erlaubt nur Berechtigten den Zugriff auf die gespeicherten Informationen.

1.2 Mitspieler

Ein *ePb-Server* verwaltet die Personenstandseinträge eines oder mehrerer Standesämter. Jedes Standesamt wird durch seine Standesamtsnummer identifiziert. Die Standesamtsnummer ist bundesweit eindeutig.

Ein *ePb-Client* ist ein Anwendungsprogramm im Standesamt, das Zugriff auf einen ePb-Server hat. Ein ePb-Client hat auf genau einen ePb-Server Zugriff.

Ein *ePb-Anwender* ist ein Anwender an einem ePb-Client. Der ePb-Anwender kann für ein oder mehrere Standesämter zuständig sein und innerhalb eines Standesamt für eines oder mehrere Personenstandsregister. Mit der Anmeldung am ePb-Client wird gleichzeitig die Anmeldung am ePb-Server ermöglicht. Diese Anmeldung erfolgt mit den Anmeldedaten des ePb-Anwenders, die für jede Person individuell sind.

Ein spezieller ePb-Client ist der *ePb-AdminClient*, der zur Administration des ePb-Servers dient. Ein ePb-Anwender am ePb-AdminClient wird *ePb-Administrator* genannt. Er verwendet in dieser Rolle spezielle Anmeldedaten, die sich von denen des normalen ePb-Anwenders unterscheiden – auch dann, wenn dieselbe Person als ePb-Anwender und als ePb-Administrator tätig ist. Im Folgenden ist mit einem ePb-Anwender stets ein normaler ePb-Anwender gemeint.

Ein weiterer spezieller ePb-Client ist der *ePb-RevisionsClient*, der dazu dient, die Aktionen des ePb-Servers zu überwachen. Der ePb-Anwender am ePb-RevisionsClient wird *ePb-Revisor* genannt. Auch für diese Rolle werden spezielle Anmeldedaten verwendet.

ePb-Server können miteinander verbunden sein. Aus Sicht eines Standesamt unterscheiden wir also den *lokalen ePb-Server*, der die Einträge des Standesamts verwaltet und *entfernte ePb-Server*, die Einträge anderer Standesämter verwalten.

Entfernte ePb-Server können nur Einsicht in Personenstandseinträge erhalten, sie können die Einträge nicht archivieren oder verändern. Die Berechtigung für die Einsichtnahme wird vom lokalen ePb-Server erteilt, sie gilt für den ePb-Server. Der entfernte ePb-Server entscheidet, welcher seiner ePb-Anwender Zugriff auf einen anderen ePb-Server hat. In Sachen Berechtigung der ePb-Anwender vertraut der lokale ePb-Server also dem entfernten Server. Um gleichwohl die Zugriffe überwachen zu können, gibt der entfernte ePb-Server bei jedem Zugriff an, welcher seiner ePb-Anwender ihn ausführt.

1.3 Funktionen

1.3.1 Archivieren von Einträgen

Lokale ePb-Anwender können neue Personenstandseinträge beim ePb-Server *archivieren* und vorhandene Einträge *fortführen*. Der ePb-Client muss garantieren, dass jede Fortführung mit dem ePb-Server synchron durchgeführt wird, damit der ePb-Server stets den aktuellen Stand des Eintrags kennt. Umgekehrt garantiert der ePb-Server, dass bei der Suche nach einem Eintrag stets der aktuelle Stand mit allen Fortführungen gefunden wird.

Der ePb-Server kann auch dazu verwendet werden, nur die *Namenverzeichnisse* zu verwalten, ohne dass die Einträge selbst auf dem ePb-Server gehalten werden. Diese Funktion bietet die Möglichkeit Namenverzeichnisse älterer Register für die Suche im ePb-Server zu verwenden.

Ältere Einträge können eingescannt und dann im Archiv des ePb-Servers archiviert werden. Dazu sind spezielle ePb-Clients denkbar.

Es gibt somit zwei Möglichkeiten, wie Einträge aus der Sicht des ePbs geführt werden:
papiergeführter Eintrag: Zu einem Eintrag sind die Suchdaten und evtl. Eintragsdaten im ePb-Server archiviert. Der Vorgang wird im Standesamt auf Papier geführt. Das ePb dient nur als Namenverzeichnis.
elektronisch geführter Eintrag: Zu einem Eintrag sind Suchdaten, evtl. Eintragsdaten und das Dokument als PDF vorhanden. Der Vorgang wird im ePb geführt. Fortführungen werden ab sofort elektronisch durchgeführt.

1.3.2 Suchen von Einträgen

Wenn zu einem Eintrag die Schlüsselinformationen (Standesamtsnummer, Register, Eintragsjahr und Eintragsnummer) bekannt sind, kann man eine *Punktsuche* durchführen. Der ePb-Server wird den Eintrag aus seinem Bestand oder aus dem eines entfernten Servers bereitstellen.

Sind die Schlüsselinformationen nicht bekannt, kann eine *unscharfe Suche* durchgeführt werden, die eine Liste von Eintragsdaten liefert, die die Schlüsselinformationen der fraglichen Einträge enthalten. Diese können dann im zweiten Schritt zur Punktsuche verwendet werden.

Der ePb-Server liefert bei der Punktsuche stets die aktuellste Version des Eintrags mit allen Fortführungen. Zusätzlich besteht die Möglichkeit, alle Versionen eines Eintrags –vom Ersteintrag bis zur aktuellsten Fortführung– beim ePb-Server anzufragen.

Die Einträge werden beim Archivieren im ePb-Server mit der digitalen Signatur des ePb-Anwenders versehen. Diese Signatur wird überprüft und der ePb-Server sorgt für ihre langfristige Gültigkeit. Der ePb-Client kann dem Server also vertrauen. D.h. wenn ein Eintrag vom ePb-Server geliefert wird, ist garantiert, dass er die unveränderte aktuelle Fassung des Eintrags ist. Um die digitale Signatur und alle Zeitstempel zur Erhaltung ihrer Gültigkeit zu überprüfen, hat der ePb-Client die Möglichkeit, die Sicherheitsinformationen zu einem Eintrag anzufordern.

1.3.3 Administration des ePb-Servers

Zur Administration gehört das Einrichten und Verwalten von

- Kennungen und Zertifikaten für ePb-Anwender, ePb-Administratoren und ePb-Revisor,
- Kennungen und Zertifikaten für entfernte ePb-Server, denen der Zugriff erlaubt ist,

- Berechtigungen, die lokale ePb-Anwender auf entfernten ePb-Servern haben,
- Zertifikaten von Trustcentern

Diese Aufgaben werden mit dem ePb-AdminClient wahrgenommen.

Darüberhinaus gibt es weitere Aufgaben der Administration, etwa die Datensicherung und die Installation von Updates. Diese Aufgaben werden nicht mit dem ePb-AdminClient durchgeführt und nicht in dieser Spezifikation beschrieben.

1.3.4 Protokollieren von Aktionen auf dem ePb-Server

Die Aktionen auf dem ePb-Server werden protokolliert, um eine missbräuchliche Verwendung der Informationen nachvollziehbar zu machen. Man spricht von *Revision*.

Zur Revision gehört

- die Konfiguration der Protokollinformation
- die Abfrage von Protokollinformationen in Bezug auf bestimmte ePb-Anwender, auf bestimmte entfernte ePb-Server
- die Abfrage von Protokollinformationen in Bezug auf bestimmte Einträge
- die Abfrage von Protokollinformationen in Bezug auf bestimmte Aktionen

1.4 Verwendungsweise

Die Schnittstelle des ePb-Servers wird als eine Reihe von *Web Services* beschrieben, die entsprechend des obigen Abschnitts in sechs Kategorien eingeteilt sind.

Der ePb-Server stellt diese Web Services im sogenannten *Dokumentenstil* bereit.

Die ePb-Clients und entfernten ePb-Server wickeln *Konversationen* mit dem ePb-Server ab, die aus mehreren *Konversationsschritten* bestehen, in denen XML-Dokumenten zwischen den Beteiligten ausgetauscht werden.

Zwischen dem Zugriff eines ePb-Clients und eines entfernten ePb-Servers besteht ein grundsätzlicher Unterschied:

- Ein ePb-Anwender meldet sich bei seinem ePb-Server an und erhält ein *Session Ticket*, das für einen langen Zeitraum, etwa einen Arbeitstag, verwendet wird. Alle Konversationen mit dem ePb-Server werden durch dieses Session Ticket einem *Session Context* für den ePb-Anwender zugeordnet. Die Session wird durch den ePb-Client beendet.

Dieses Verfahren wurde gewählt, damit die Anmeldung des ePb-Anwenders von seinem ePb-Client, also dem gewohnten Standasamtsprogramm auf den ePb-Server ausgeweitet wird, so dass die Anmeldung am ePb-Client die am ePb-Server automatisch einschließt.

- Ein entfernter ePb-Server erhält bei der Anmeldung bei einem ePb-Server ein *Conversation Ticket*, das ihm ermöglicht, eine (genau eine) Konversation durchzuführen. Das Conversation Ticket wird danach automatisch ungültig.

  Dieses Verfahren erzwingt, dass ein entfernter ePb-Server für jede Konversation erneut den Zugang zum ePb-Server herstellt. Das Konzept der Session kann nicht verwendet werden, weil ja verschiedene entfernte ePb-Anwender gleichzeitig die Dienste des ePb-Servers nutzen können.

Für beide Zugriffsarten gilt allerdings gleichermaßen, dass durch die Infrastruktur des Netzes die Sicherheit der Daten auf den Verbindungen zwischen Rechnern gewährleistet werden muss. Der ePb-Server setzt diese Sicherheit sowohl bei seinen ePb-Clients wie bei der Verbindung mit entfernten ePb-Servern voraus.

Kapitel 2

Übersicht der Schnittstellen

Sprechweise: Ein einzelner Aufruf, also ein Request-Response-Paar des Web-Services, wird *Service* genannt. Eine inhaltlich aufeinander folgende Reihe von Services heißt *Konversation*.

2.1 Status- und Fehlerbehandlung

Beim erfolgreichen Aufruf eines ePb-Service, bei dem keine Nutzdaten als Antwort übergeben werden, erhält der Aufrufer das Element `rtOK`. Welche Nutzdaten bei welchem Service geliefert werden, wird bei der Servicebeschreibung in den späteren Kapiteln aufgeführt.

Bei jedem Aufruf eines ePb-Service kann ein Fehler auftreten. Fehlernachrichten werden nach der Spezifikation von SOAP Fault konstruiert (siehe SOAP Version 1.2 Part 1: 5.4 Messaging Framework).

2.2 Zugangstickets

Diese Kategorie enthält die ePb-Services, mit denen eine Session bzw. eine Konversation begonnen und beendet wird.

`getSessionTicket`
: Dieser Service ist nur für einen lokalen ePb-Anwender erlaubt. Der Anwender identifiziert sich beim ePb-Server und erhält ein Session Ticket, das den Zugriff auf die Services erlaubt, für die der Anwender berechtigt ist.
Antwort: `rtSessionTicket`

`discardSessionTicket`
: Dieser Service beendet die Session und macht das SessionTicket ungültig.
Antwort: `rtOK`

`getConversationTicket`
: Dieser Service ist nur einem externen ePb-Server erlaubt. Er identifiziert sich und erhält ein Conversation Ticket, das den Zugriff auf *eine* Konversation gestattet. Danach verliert das Ticket seine Gültigkeit. Das Ticket gilt nur für ein bestimmtes Standesamt, das auf dem anfragenden ePb-Server geführt wird.
Antwort: `rtConversationTicket`

2.3 Services für das Standesamt

2.3.1 Archivieren

Diese Kategorie umfasst die ePb-Services, die das Archivieren und Fortführen von Personenstandseinträgen ermöglichen. Entfernte ePb-Server können keinen Service in dieser Kategorie aufrufen.

`insertEintrag`
: Archiviert einen neuen, oder einen bereits angelegten papiergeführten Eintrag. Nach der erfolgreichen Archivierung, können Einträge elektronisch geführt werden.
Antwort: `rtOK`

`coEintrag`
: Dieser Service ist der erste, der bei einer Fortführung (Konversation) verwendet werden muss. Ein Eintrag wird aus dem Archiv des ePb-Servers zur Fortführung ausgeliefert und gesperrt. Andere ePb-Anwender können auf den Eintrag nur noch lesend zugreifen.
Antwort: `rtEintrag`

`ciEintrag`
: Ein Eintrag, der mit coEintrag zur Fortführung aus dem Archiv ausgeliefert wurde, wird nun nach der Fortführung wieder ins Archiv gestellt. Nach dem `ciEintrag` wird die Sperre auf den Eintrag aufgehoben.
Dieser Service ist Teil einer Konversation: er muss direkt auf `coEintrag` folgen (und sich natürlich auf denselben Eintrag beziehen).
Anwort: `rtOK`

`unlockEintrag`
: Eine mit `coEintrag` gesetzte Sperre wird aufgehoben, ohne dass die Fortführung durchgeführt wurde.
Auch dieser Service ist nur innerhalb der Konversation möglich, die mit

coEintrag beginnt.
Anwort: rtOK

insertEdListe
: Dieser Service archiviert Eintragsdaten zu einem oder mehreren Einträgen. Es werden nur Eintragsdaten ins Archiv übernommen, die bisher nicht verzeichnet waren. Sollen vorhandene Eintragsdaten verändert werden, muss der Service updateEdListe verwendet werden. Die bei einem Serviceaufruf übermittelten Eintragsdaten beziehen sich immer auf ein Register.
Antwort: rtOK

updateEdListe
: Mit diesem Service können Eintragsdaten zu einem oder mehreren Einträgen auf dem ePb-Server gespeichert werden. Bereits vorhandene Eintragsdaten werden durch diesen Service überschrieben. Die Daten werden nicht überschrieben, wenn ein Eintrag gesperrt ist. Die bei einem Serviceaufruf übermittelten Eintragsdaten beziehen sich immer auf ein Register.
Antwort: rtOK

Verhalten des ePb-Servers bei Sperren: Mit dem Service coEintrag ist eine Sperre verbunden, der Eintrag kann nur von dem Anwender fortgeführt werden, der coEintrag ausgeführt hat. Diese Sperre gilt nur innerhalb der Konversation, sie wird durch ciEintrag oder unlockEintrag aufgehoben. Wird nach coEintrag die Session beendet oder wird die Konversation nicht korrekt zu Ende geführt, führt der ePb-Server automatisch unlockEintrag aus.

2.3.2 Suchen

Die Antwort der Services *getEintrag* und *findEintrag* ist abhängig von der Berechtigug des Benutzers. Sie bestimmt, ob alle verfügbaren Informationen zu einem Eintrag geliefert werden, oder ob nur ein Hinweis auf Vorhandensein eines Eintrags übermittelt wird.

getEintrag
: Punktsuche nach einem Eintrag durch Angabe der EintragsID. Wird der Service durch einen lokalen ePb-Anwender angefordert, wird die Suche eventuell an einen entfernten ePb-Server weitergeleitet, wenn der gesuchte Eintrag nicht im lokalen Archiv enthalten ist. Wenn der Service durch einen entfernten ePb-Server angefordert wird, wird die Suche nur auf dem eigenen, lokalen Archiv ausgeführt. Es wird immer die aktuellste Fortführung eines Eintrags geliefert
Antwort: rtEintrag

findEintrag

Unscharfe Suche nach einem Eintrag durch die Angabe von diversen Suchinformationen. Wird der Service durch einen lokalen ePb-Anwender angefordert, werden die Suchinformationen analysiert und ggfs wird die Suche an einen externen ePb-Server weitergeleitet. Wenn der Service durch einen entfernten ePb-Server angefordert wird, wird die Suche nur auf dem eigenen, lokalen Archiv ausgeführt.
Antwort: `rtSuchergebnis`

`getEintragHistorie`
Dieser Service liefert alle Fortführungen zu einem bestimmten Eintrag, die im ePb-Server archiviert sind. Handelt es sich bei dem gesuchten Eintrag um eine Nacherfassung, dann werden alle Fortführungen bis zu der Nacherfassung geliefert.
Antwort: `rtEintragHistorie`

`getEintragSecInfo`
Zu einem Eintrag werden alle Sicherheitsinformationen geliefert, die zur Überprüfung der Gültigkeit der Signatur des Eintrags benötigt werden.
Antwort: `rtSecInfoListe`

2.3.3 Registerverwaltung

`getNamenverz`
Dieser Service liefert zu einem Zeitraum und einem Register eine Liste der Suchdaten all der Einträge und Fortführungen, deren Beurkundungsdatum in diesen Zeitraum fallen. Dieser Service ist nur für lokale ePb-Anwender erlaubt und bezieht sich nur auf die Daten im lokalen Archiv.
Antwort: `rtNamenverz`

`makeJahresabschluss`
Dieser Service führt einen Jahresabschluss für ein bestimmtes Register eines Standesamtes durch. Danach ist es nicht mehr möglich, neue Einträge in diesem Jahr zu archivieren. Der Jahresabschluss bezieht sich immer auf das Vorjahr relativ vom Ausführungszeitpunkt. Als Rückgabe werden Informationen geliefert zu:

- Anzahl der in diesem Jahr verzeichneten neuen Einträge
- Anzahl der in diesem Jahr gemachten Fortführungen mit Datum der Ersteintragung
- Vergabe von Eintragsnummern: niedrigste/höchste Nummer, Lücken und Zwischennummern

Antwort: `rtJahresabschluss`

`getJahresinfo`
: Dieser Service testet, ob ein Jahresabschluss für ein bestimmtes Register eines Standesamtes durchgeführt werden kann. Der Jahrgang, in dem der Jahresabschluss getestet wird, kann beliebig gewählt werden. Antwort: `rtJahresabschluss`

`getRegisterJahressicherung`
: Liefert alle Informationen eines Jahrgangs eines Registers in einem datenbankneutralem Format. Zusätzlich wird eine Prüfsumme gebildet, mit der die Datenintegrität geprüft werden kann.
Antwort: `rtRegisterJahressicherung`

2.4 Administrieren

Die folgenden Services können nur von Benutzern verwendet werden, die eine Administrationsberechtigung auf dem ePb-Server besitzen.

In der Beschreibung verwenden wir folgende Sprechweise: ein *Usr* ist ein lokaler ePb-Anwender (einschließlich Administratoren ...), ein *Svr* ist ein entfernter ePb-Server. *CA* steht für Certification Authority.

Entfernte ePb-Server kommen in einer doppelten Rolle vor: Sie können Dienste des ePb-Servers verwenden, d.h. sie sind ein Client; sie können aber auch selbst als Server verwendet werden, wenn Anfragen an sie delegiert werden. Die Serverinformationen beinhalten Informationen für beide Situationen. Insbesondere wird für jeden entfernten ePb-Server, der Anfragen beantwortet, verzeichnet, für welche Standesämter und Orte er Einträge verwaltet.

`getUsr`
: liefert die Stammdaten eines lokalen Benutzers, die im ePb-Server abgelegt sind.
Antwort: `rtUsr`

`getUsrListe`
: liefert eine Liste der Stammdaten aller lokalen Benutzers, die im ePb-Server abgelegt sind.
Antwort: `rtUsrListe`

`insertUsr`
: legt einen neuen Benutzer im System an. Das neue Konto kann erst aktiv geschaltet werden, wenn ein Zertifikat damit verknüpft wurde.
Antwort: `rtOK`

`updateUsr`
: ändert die im ePb-Server hinterlegten Stammdaten eines bestehenden Be-

nutzers.
Antwort: `rtOK`

`setUsrRechte`
ändert die im ePb-Server hinterlegten Benutzerrechte eines bestehenden Benutzers.
Antwort: `rtOK`

`insertUsrCert`
legt ein neues Benutzerzertifikat im System ab. Dabei muss das Zertifikat einem einem bestimmten ePb-Benutzer zugeordnet werden.
Antwort: `rtCertID`

`getUsrCertListe`
liefert eine Liste aller Zertifikate eines Benutzers zurück.
Antwort: `rtUsrCertListe`

`getSvr`
liefert alle Informationen über einen angeschlossenen ePb-Server.
Antwort: `rtSvrInfo`

`getSvrListe`
liefert eine Liste aller Informationen über alle angeschlossenen ePb-Server.
Antwort: `rtSvrInfoListe`

`insertSvr`
Mit dem Service insertSvr wird ein externer ePb-Server mit dem lokalen gekoppelt. Damit wird die Suche und eventuell das Abrufen von Einträgen auf diesem externen Server ermöglicht. Die Kommunikation mit dem externen Server wird erst dann möglich, wenn auch ein Serverzertifikat eingetragen wurde.
Antwort: `rtOK`

`updateSvr`
ändert die lokalen Informationen über den eigenen Server, wie auch über externe Server.
Antwort: `rtOK`

`insertSvrCert`
legt ein neues Serverzertifikat im System an und orndet es einem Server zu. Serverzertifikate dienen der Authentifikation zwischen ePb-Servern und ermöglichen eine sichere, verschlüsselte Kommunikation.
Antwort: `rtCertID`

`getSvrCertListe`
liefert eine Liste aller Serverzertifikate, in binärer Form, zurück, die dem ausgewählten ePb-Server zugeordnet werden.
Antwort: `rtSvrCertListe`

`insertCACert`
: legt ein neues Trustcenterzertifikat im System ab. Es müssen alle Zertifikate der vertrauenswürdigen Trustcenter im System hinterlegt sein. Ansonsten können Benutzer- und Serverzertifikate nicht auf ihre Gültigkeit hin überprüft werden und werden vom System nicht anerkannt.
Antwort: `rtCertID`

`getCACertListe`
: liefert eine Liste aller Trustcenterzertifikate, in binärer Form, die auf dem lokalen ePb-Server hinterlegt sind.
Antwort: `rtCACertListe`

`checkZertStatus`
: überprüft alle lokal abgelegten Zertifikate, deren Gültigkeitsdauer noch nicht abgelaufen ist, auf ihre Gültigkeit.
Antwort: `rtZertStatus`

`updateSecStatus`
: überprüft alle lokalen Einträge auf die Gültigkeit der letzten Zeitstempel und fügt bei Bedarf neue Zeitstempel an. Der Bedarf wird daran festgestellt, dass ein aktuelleres Zertifikat des Trustcenters existiert, das den letzten zeitstempel ausgestellt hat.
Antwort: `rtSecStatus`

`getSecConfListe`
: liefert eine Liste der Sicherheitskonfigurationen des lokalen ePb-Servers seit seiner Inbetriebnahme.
Antwort: `rtSecConfListe`

`saveSecConfListe`
: legt neue Sicherheitseinstellungen für den ePb-Server ab. Bestehende Einstellungen werden nicht gelöscht, sondern deren Gültigkeitszeitraum beendet. Die neuen Einstellungen gelten ab sofort bis neue Einstellungen abgelegt werden.
Antwort: `rtOK`

`getSvrConfListe`
: liefert eine Liste der allgemeinen Konfigurationen des lokalen ePb-Servers seit seiner Inbetriebnahme.
Antwort: `rtSvrConfListe`

`saveSvrConfListe`
: legt neue allgemeine Einstellungen für den ePb-Server ab. Bestehende Einstellungen werden nicht gelöscht, sondern deren Gültigkeitszeitraum beendet. Die neuen Einstellungen gelten ab sofort bis neue Einstellungen abgelegt werden.
Antwort: `rtOK`

2.5 Revision

Die folgenden Services können nur von Benutzern verwendet werden, die eine Revisionsberechtigung auf dem ePb-Server besitzen.

`getRevConfListe`
: liefert eine Liste der Konfigurationen für die Revision des lokalen ePb-Servers seit seiner Inbetriebnahme.
Antwort: `rtRevConfListe`

`saveRevConfListe`
: legt neue Einstellungen für die Revision innerhalb des lokalen ePb-Servers ab. Bestehende Einstellungen werden nicht gelöscht, sondern deren Gültigkeitszeitraum beendet. Die neuen Einstellungen gelten ab sofort bis neue Einstellungen abgelegt werden.
Antwort: `rtOK`

`getUsrRevInfo`
: liefert eine Liste aller Aktionen eines Users ab einem bestimmten Zeitpunkt.
Antwort: `rtRevInfoListe`

`getEintragRevInfo`
: liefert eine Liste aller Aktionen die auf den gewählten Eintrag ab einem gewählten Zeitpunkt angewandt wurden.
Antwort: `rtRevInfoListe`

`getServiceRevInfo`
: liefert eine Liste aller Aufrufe des ausgewählten Services ab einem gewählten Zeitpunkt
Antwort: `rtRevInfoListe`

`getSvrRevInfo`
: liefert eine Liste aller Aktionen, die ein externer ePb-Server auf dem lokalen Server ab einem bestimmten Zeitpunkt ausgeführt hat.
Antwort: `rtRevInfoListe`

`purgeRevInfo`
: löscht die Revisionsinformationen eines gewählten Zeitraums. Zusätzlich können die gelöschten Informationen zurückgeliefert werden um sie der Archivierung zuzuführen.
Antwort: `rtOK` oder `rtRevInfoListe`

Kapitel 3

Struktur der archivierten Informationen

Dieses Kapitel beschreibt die Struktur der Daten, die im ePb-Server zu einem Eintrag archiviert werden.

3.1 Eigenschaften der Schemata

Target Namespace

`http://www.vfst.de/xsta_epb`

Deklarierte Namespaces

Prefix	Namespace
Default	http://www.vfst.de/xsta_epb
xs	http://www.w3.org/2001/XMLSchema

3.2 Datentypen

Im folgenden werden die Strukturen der Datentypen erläutert, die in den Datenstrukturen der Einträge verwendet werden.

3.2.1 tNamen

Dieser Datentyp beschreibt einen Namenskomplex aus Vornamen, Geburtsnamen, Familienname, Art des Zusatznamens, Zusatznamen und akademische Gra-

de.

```
<xs:complexType name="tNamen">
  <xs:sequence>
    <xs:element name="Vornamen" type="xs:string">
    </xs:element>
    <xs:element name="Geburtsname" type="xs:string"
       minOccurs="0">
    </xs:element>
    <xs:element name="Familienname" type="xs:string">
    </xs:element>
    <xs:element name="ArtdesZusatznamens" type="xs:string
       " minOccurs="0">
    </xs:element>
    <xs:element name="Zusatzname" type="xs:string"
       minOccurs="0">
    </xs:element>
    <xs:element name="akademischeGrade" type="xs:string"
       minOccurs="0">
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Bemerkung: Der Inhalt des Elements „ArtdesZusatznamens“ ist abhängig von der jeweiligen Software im Standesamt zur Vorgangsbearbeitung.

3.2.2 tOrt

Dieser Datentyp beschreibt einen Ort aus Ort, Landkreis und das Land.

```
<xs:complexType name="tOrt">
  <xs:sequence>
    <xs:element name="Ort" type="xs:string" minOccurs
       ="0">
    </xs:element>
    <xs:element name="Landkreis" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="Anmerkungen" type="xs:
               string" use="optional">
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Land" type="xs:string" minOccurs
       ="0">
```

```
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

3.2.3 tDatum

Dieser Datentyp beschreibt ein Datum aus Tag, Monat und das Jahr. Hierbei wurde ein eigener Datentyp entworfen, da es bei den Einträgen vorkommen kann, dass die Datumsangaben unvollständig sind.

```
<xs:complexType name="tDatum">
  <xs:sequence>
    <xs:element name="Tag" type="xs:int" minOccurs="0">
    </xs:element>
    <xs:element name="Monat" type="xs:int" minOccurs="0">
    </xs:element>
    <xs:element name="Jahr" type="xs:int" minOccurs="0">
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

3.2.4 tPerson

Dieser Datentyp beschreibt eine Person. Es werden nur die grundlegendsten Daten verwendet, die einer natürlichen Person zugeordnet werden können. tPerson ist der Obertyp zu allen erweiterten Personentypen.

```
<xs:complexType name="tPerson">
  <xs:sequence>
    <xs:element name="Namen" type="tNamen">
    </xs:element>
    <xs:element name="Geburtsdatum" type="tDatum">
    </xs:element>
    <xs:element name="Geburtsort" type="tOrt">
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Geschlecht" use="optional">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="m"/>
        <xs:enumeration value="w"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

3.2.5 tZeit

Dieser Datentyp wird überall dort verwendet, wo ein Zeitpunkt benötigt wird. Alle Elemente sind optional, wodurch ungenaue Zeitangaben möglich sind. Mit dem Element Zusatz kann an Tagen, bei denen die Umstellung Sommerzeit auf Winterzeit erfolgte, bestimmt werden, ob es die angebene Stunde die erste (A) oder die zweite Stunde war (B).

```
<xs:complexType name="tZeit">
  <xs:sequence>
    <xs:element name="Stunde" type="xs:int" minOccurs
       ="0">
    </xs:element>
    <xs:element name="Minute" type="xs:int" minOccurs
       ="0">
    </xs:element>
    <xs:element name="Zusatz" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="A"/>
          <xs:enumeration value="B"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

3.2.6 tRegister

Dieser Datentyp spezifiziert das Register eines Eintrags. Mögliche Werte sind:

- **g** - Geburtsregister
- **e** - Eheregister
- **l** - Lebenspartnerschaftsregister
- **s** - Sterberegister

```
<xs:simpleType name="tRegister">
  <xs:restriction base="xs:string">
    <xs:enumeration value="g"/>
    <xs:enumeration value="e"/>
    <xs:enumeration value="l"/>
    <xs:enumeration value="s"/>
  </xs:restriction>
</xs:simpleType>
```

3.2.7 tEintragsID

Dieser Datentyp spezifiziert einen eindeutigen Eintragsschlüssel aus Standesamtsnummer, Register, Jahr und der Eintragsnummer. Das Konzept setzt voraus, dass eine EintragsID bundesweit eindeutig ist und genau zu einem einzigen Eintrag gehört.

```
<xs:complexType name="tEintragsID">
  <xs:sequence>
    <xs:element name="Standesamtsnummer" type="xs:string
       "/>
    <xs:element name="Register" type="xs:string"/>
    <xs:element name="Jahr" type="xs:gYear"/>
    <xs:element name="Eintragsnummer" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

3.2.8 EintragsFID

Dieses Element spezifiziert einen eindeutigen Eintragsschlüssel zu einer Fortführung.

```
<xs:element name="EintragsFID">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="tEintragsID">
        <xs:sequence>
          <xs:element name="Fortfuehrungsnummer"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

3.3 Datenstruktur der Eintragsdaten

3.3.1 Geburtseintrag

```
<xs:element name="Geburtseintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID"/>
      <xs:element name="Eintragsdatum" type="tDatum"/>
      <xs:element name="Sperrvermerk" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VonDatum" type="tDatum"/>
            <xs:element name="BisDatum" type="tDatum"/>
            <xs:element name="Grund" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Mutter" type="tPerson" minOccurs
          ="0"/>
      <xs:element name="Vater" type="tPerson" minOccurs
          ="0"/>
      <xs:element name="Kind">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tPerson">
              <xs:sequence>
                <xs:element name="Geburtszeit" type="
                    tZeit"/>
                <xs:element name="Totgeburt" type="xs:
                    boolean" minOccurs="0"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

3.3.2 Eheeintrag

```
<xs:element name="Eheeintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID"/>
      <xs:element name="Eintragsdatum" type="tDatum"/>
      <xs:element name="Sperrvermerk" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VonDatum" type="tDatum"/>
            <xs:element name="BisDatum" type="tDatum"/>
            <xs:element name="Grund" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OrtDerEhe" type="tOrt"/>
      <xs:element name="Ehedatum" type="tDatum"/>
      <xs:element name="Mann">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tPerson">
              <xs:sequence>
                <xs:element name="NameNachEhe" type="xs:
                    string"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Frau">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tPerson">
              <xs:sequence>
                <xs:element name="NameNachEhe" type="xs:
                    string"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

3.3.3 Lebenspartnerschaftseintrag

```
<xs:element name="Lebenspartnerschaftseintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID"/>
      <xs:element name="Eintragsdatum" type="tDatum"/>
      <xs:element name="OrtDerBegruendung" type="tOrt"/>
      <xs:element name="Begruendungsdatum" type="tDatum
          "/>
      <xs:element name="Partner1">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tPerson">
              <xs:sequence>
                <xs:element name="NameNachLP" type="xs:
                    string"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Partner2">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tPerson">
              <xs:sequence>
                <xs:element name="NameNachLP" type="xs:
                    string"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

3.3.4 Sterbeeintrag

```
<xs:element name="Sterbeeintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID"/>
      <xs:element name="Eintragsdatum" type="tDatum"/>
      <xs:element name="Verstorbene">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tPerson">
              <xs:sequence>
                <xs:element name="Familienstand" type="xs
                    :string"/>
                <xs:element name="Sterbedaten">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Sterbedatum" type
                          ="tDatum"/>
                      <xs:element name="SterbedatumBis"
                          type="tDatum" minOccurs="0"/>
                      <xs:element name="Sterbezeit" type
                          ="tZeit" minOccurs="0"/>
                      <xs:element name="SterbezeitBis"
                          type="tZeit" minOccurs="0"/>
                      <xs:element name="Sterbeort" type="
                          xs:string" minOccurs="0"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Partner" type="tPerson" minOccurs
          ="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Kapitel 4

Struktur der Schnittstellen

4.1 Datentypen der Schnittstelle

4.1.1 Attribute

Im Folgenden werden Attribute beschrieben, die bei den späteren Elementen und Datentypen verwendet werden.

SigAlgorithmus Enthält den Algorithmus einer Signatur.

```
<xs:attribute name="SigAlgorithmus" type="xs:string">
</xs:attribute>
```

SigFormat Enthält das Format einer Signatur.

```
<xs:attribute name="SigFormat">
</xs:attribute>
```

CertFormat Enthält das Format eines Zertifikats.

```
<xs:attribute name="CertFormat">
</xs:attribute>
```

SecInfoFormat Enthält das Format einer Sicherheitsinformation.

```
<xs:attribute name="SecInfoFormat" type="xs:string"/>
```

SecInfoAlgorithmus Enthält den Algorithmus einer Sicherheitsinformation.

```
<xs:attribute name="SecInfoAlgorithmus" type="xs:string
   "/>
```

SecInfoTyp Enthält den Typ einer Sicherheitsinformation.

```
<xs:attribute name="SecInfoTyp" type="xs:string">
</xs:attribute>
```

4.1.2 Eintrag

Dieses Element beschreibt die Struktur der Daten, die beim Senden/Empfangen eines Eintrags übermittelt werden. Restriktionen, welche Daten bei einem bestimmten Serviceaufruf übermittelt werden müssen, werden nicht im Schema beschrieben, sondern in der Spezifikation der Serviceschnittstellen. Jede Serviceschnittstelle prüft die für ihre Abarbeitung benötigten Daten eigenständig.

```
<xs:element name="Eintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element ref="Geburtseintrag"/>
        <xs:element ref="Eheeintrag"/>
        <xs:element ref="Lebenspartnerschaftseintrag"/>
        <xs:element ref="Sterbeeintrag"/>
      </xs:choice>
      <xs:element name="Eintragsart" type="xs:string"/>
      <xs:element name="PDF" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
              <xs:attribute name="version" type="xs:
                 string"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Signatur" minOccurs="0">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
              <xs:attribute name="sigAlgorithmus" type="
                 xs:string" use="required"/>
              <xs:attribute name="sigFormat" type="xs:
                 string" use="required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

- **Auswahl zwischen:**
  - **Geburts-, Ehe-, Lebenspartnerschafts- und Sterbeeintrag:** Enthalten buchspezifische Daten.

- **Eintragsart:** Dieses Feld beschreibt die Art der Eintragung. Handelt es sich um einen neu erfassten Eintrag, dann enthält das Feld den Wert „Neuerfassung“ . Handelt es sich bei dem übermittelten Eintrag um eine Nacherfassung, dann enthält das Feld den Wert „Nacherfassung“ . Handelt es sich um eine Fortführung, dann enthält das Feld den Grund der Fortführung.
- **PDF:** Enthält den Eintrag als PDF im Base64-Format.
- **Signatur:** Enthält die Signatur (base64)

4.1.3 tZeitraumDatum

Datentyp für einen Datumsintervall.

```
<xs:complexType name="tZeitraumDatum">
  <xs:sequence>
    <xs:element name="Datum" type="tDatum"/>
    <xs:element name="DatumBis" type="tDatum"/>
  </xs:sequence>
</xs:complexType>
```

4.1.4 tASuchkriterien

Dieser Datentyp beinhaltet alle allgemeinen Kriterien, die bei einer unscharfen Suche im ePb verwendet werden. Bei einer unscharfen Suche im ePb-Server ist es möglich, verschiedene Suchkriterien mit „Jokerzeichen“ zu versehen. Zwei Zeichen stehen zur Verfügung:

- _ ersetzt ein beliebiges Zeichen, die Suche nach „M__er“ findet z.B. „Maier, Mayer, Meyer, ...“ .
- % ersetzt eine beliebig lange Zeichenfolge. Die Suche nach „Ma% “ findet alle Namen, die mit „Ma“ beginnen.

```
<xs:complexType name="tASuchkriterien">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Vornamen" type="xs:string"
       minOccurs="0"/>
    <xs:element name="Eintragszeitraum" type="
       tZeitraumDatum">
    </xs:element>
    <xs:element name="Ereigniszeitraum" type="
       tZeitraumDatum">
    </xs:element>
  </xs:sequence>
```

```
</xs:complexType>
```

- **Name:** Name der gesuchten Person
- **Vornamen:** Vornamen der gesuchten Person
- **Eintragszeitraum:** Zeitraum in dem die Eintragung stattgefunden hat.
- **Ereigniszeitraum:** Zeitraum in dem das Ereignis stattgefunden hat.

4.1.5 tGSuchkriterien

Dieser Datentyp ist eine Erweiterung der allgemeinen Suchkriterien `tASuchkriterien` für das Geburtenregister.

```
<xs:complexType name="tGSuchkriterien">
  <xs:complexContent>
    <xs:extension base="tASuchkriterien">
      <xs:sequence>
        <xs:element ref="Geschlecht" minOccurs="0"/>
        <xs:element name="Totgeburt" type="xs:boolean"
           minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

- **Geschlecht:** Geschlecht der gesuchten Person
- **Totgeburt:** Suche nach einer Totgeburt. Das Element ist vom Typ Boolean und kann die Werte `true` oder `false` enthalten.

4.1.6 tSSuchkriterien

Dieser Datentyp ist eine Erweiterung der allgemeinen Suchkriterien `tASuchkriterien` für das Sterberegister.

```
<xs:complexType name="tSSuchkriterien">
  <xs:complexContent>
    <xs:extension base="tASuchkriterien">
      <xs:sequence>
        <xs:element name="Geburtszeitraum" type="
           tZeitraumDatum" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Das zusätzliche Element **Geburtszeitraum** enthält einen Zeitraum in dem nach dem Geburtsdatum der gesuchten Person gesucht wird.

4.1.7 Geschlecht

```
<xs:element name="Geschlecht">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="m"/>
      <xs:enumeration value="w"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Element vom Typ String, das nur die Werte „m“ oder „w“ annehmen kann.

4.1.8 tUsr

Dieser Datentyp enthält die Grundinformationen zu einem ePb-Server-Benutzerkonto.

```
<xs:complexType name="tUsr">
  <xs:sequence>
    <xs:element name="Standesamtsnummer" type="xs:string
       "/>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Vorname" type="xs:string"/>
    <xs:element name="Gueltigkeit" type="tZeitraumDatum
       "/>
    <xs:element name="aktiv" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

- **Standesamtsnummer:** bundesweit einheitliche Standesamtsnummer
- **id:** eindeutige Benutzerkennung in einem Standesamt
- **Name:** Name des Benutzers
- **Vorname:** Vorname des Benutzers
- **Gueltigkeit:** Gültigkeitszeitraum des Benutzerkontos
- **aktiv:** Ist das Benutzerkonto aktiv oder inaktiv?

4.2 Ablauf der Konversationen

Serviceaufrufe eines ePb-Clients (z.B. Software zur Vorgangsbearbeitung) bei einem ePb-Server werden nach dem Request-/Responseverfahren durchgeführt. Hierbei werden XML-Nachrichten ausgetauscht, die nach der SOAP-Spezifikation (SOAP Version 1.2) aufgebaut sind. Im Folgenden das Grundgerüst einer SOAP-Nachricht:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Envelope xmlns="http://schemas.xmlsoaporg.org/soap/
   envelope/">
  <Header>
    <!-- Service-Informationen -->
  </Header>
  <Body>
    <!-- Service-Nachricht -->
  </Body>
</Envelope>
```

Die SOAP-Nachricht bildet einen Rahmen (Container), für die zu übertragenden Informationen. Die zwei wichtigsten Bereiche in diesem Container sind der Header- und Body-Bereich. Im Header der SOAP-Nachricht werden zusätzliche Konversationsinformationen übertragen. Diese bestehen bei einem Request in der Regel aus dem Session Ticket und dem vom ePb-Client gewünschten Servicenamen. Im Body-Bereich wird die gewünschte Servicenachricht eingebettet. Bei der Antwort (Response) des ePb-Servers enthält der Body die Antwort des angeforderten Services.

Nachfolgend wird der generelle Ablauf bei einer Konversation beschrieben:
ePb-Client

- ePb-Client erzeugt eine Service-Nachricht, nach der für diesen Service festgelegten Struktur
- ePb-Client erzeugt eine SOAP-Nachricht
- Der Header-Bereich der SOAP-Nachricht wird mit den entsprechenden Konversations-Informationen befüllt
- Im Body-Bereich der SOAP-Nachricht wird die Service-Nachricht eingebettet
- Die SOAP-Nachricht wird an den ePb-Server übermittelt (Request)

ePb-Server

- ePb-Server erzeugt eine Serviceantwort-Nachricht, nach der für dieser Serviceantwort festgelegten Struktur

- ePb-Server erzeugt eine SOAP-Nachricht
- Im Header-Bereich der SOAP-Nachricht wird die Serviceantwort-Name hineingeschrieben
- Im Body-Bereich der SOAP-Nachricht wird die Serviceantwort-Nachricht eingebettet
- Die SOAP-Nachricht wird an den ePb-Client zurück gesendet (Response).

4.2.1 Zugangsberechtigung erlangen

Um die Services eines ePb-Servers zu verwenden, wird ein Session Ticket benötigt. Wie ein ePb-Client dieses Ticket bekommt, soll im Folgenden erläutert werden.

Beispiel: Session Ticket anfordern:

Möchte ein ePb-Client ein Session Ticket von einem ePb-Server bekommen, so muss er die Service-Nachricht *getSessionTicket* an den ePb-Server senden. Die Service-Nachricht enthält für die Authentifizierung des Benutzers seine eindeutige BenutzerID. Nachfolgend die SOAP-Nachricht, die an den ePb-Server gesendet wird:

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoaporg.org/soap/
   envelope/">
  <Header>
    <ServiceName>getSessionTicket</ServiceName>
  </Header>
  <Body>
    <getSessionTicket>
      <Standesamtsnummer>0815</Standesamtsnummer>
      <BenutzerId>47111704</BenutzerId>
    </getSessionTicket>
  </Body>
</Envelope>
```

Ist der Benutzer dem ePb-Server bekannt und das Benutzerkonto aktiv, dann erhält der ePb-Client die folgende Serviceantwort vom ePb-Server:

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoaporg.org/soap/
   envelope/">
  <Header>
    <ServiceName>rtSessionTicket</ServiceName>
```

```
  </Header>
  <Body>
    <rtSessionTicket>
      <SessionTicket>A8TZJ765ATZS654G5</SessionTicket>
    </rtSessionTicket>
  </Body>
</Envelope>
```

Die Serviceantwort enthält ein verschlüsseltes Session Ticket, welches für die weitere Kommunikation mit dem ePb-Server berechtigt. Das Ticket wurde mit dem öffentlichen Schlüssel des Benutzers verschlüsselt. Der öffentliche Schlüssel wurde aus dem Zertifikat entnommen, das mit der vom ePb-Client übermittelten BenutzerID verknüpft ist. Nur mit dem privaten Schlüssel des Benutzers kann das Session Ticket entschlüsselt werden.

4.2.2 Einen Service verwenden

Nachdem nun ein gültiges Session Ticket vorhanden ist, können weitere Services bei dem ePb-Server genutzt werden. Im Folgenden wird davon ausgegangen, dass der ePb-Benutzer die nötige Berechtigung besitzt, Einträge aus dem Geburtsregister einzusehen.

Beispiel: Service getEintrag:

Um einen bestimmten Eintrag aus dem Geburtenregister zu holen, wird der Service *getEintrag* verwendet. Die Service-Nachricht wird mit den benötigten Informationen versehen und in den Body-Bereich der SOAP-Nachricht eingebettet. Im Header-Bereich werden das Session Ticket und der Servicename eingetragen. Die an den ePb-Server gesendete SOAP-Nachricht sieht folgendermaßen aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoaporg.org/soap/
   envelope/">
  <Header>
    <SessionTicket>3H27UZT32423JJKU</SessionTicket>
    <ServiceName>getEintrag</ServiceName>
  </Header>
  <Body>
    <getEintrag>
      <EintragsID>
        <Standesamtsnummer>4711</Standesamtsnummer>
        <Register>g</Register>
        <Jahr>2001</Jahr>
        <Eintragsnummer>5011</Eintragsnummer>
      </EintragsID>
```

```
    </getEintrag>
  </Body>
</Envelope>
```

Wurde der Service erfolgreich vom ePb-Server verarbeitet, erhält der ePb-Client die folgende SOAP-Nachricht zurück:

```
<Envelope xmlns="http://schemas.xmlsoaporg.org/soap/
   envelope/">
  <Header>
    <ServiceName>rtEintrag</ServiceName>
  </Header>
  <Body>
    <rtEintrag>
      <Eintragsdaten>
        <Geburtseintrag>
          <!-- Eintragsdaten für diesen Eintrag -->
        </Geburtseintrag>
      </Eintragsdaten>
      <PDF version="1.4">8ue3i34jl23o4j...</PDF>
    </rtEintrag>
  </Body>
</Envelope>
```

Im Body-Bereich werden die angeforderten Informationen geliefert. Sie bestehen aus den Eintragsdaten (verkürzt dargestellt) und dem dazugehörigen Dokument als PDF im Base64-Format (rfc1521 5.2).

4.3 Fehlermeldungen

4.3.1 Aufbau einer SOAP Fault Message

Im folgenden wird eine beispielhafte SOAP Fault Nachricht aufgeführt.

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-
   envelope" xmlns:m="http://www.vfst.de/xsta_epb" xmlns
   :xml="http://www.w3.org/XML/1998/namespace">
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:Sender</env:Value>
        <env:Subcode>
          <env:Value>m:genDatenUnvollstaendig</env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
```

```
        <env:Text xml:lang="de-DE">Unvollstaendige Daten
           fuer diesen Service</env:Text>
      </env:Reason>
      <env:Detail>
        <m:Service>getEintrag</m:Service>
        <m:Element>Jahr</m:Element>
      </env:Detail>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

Das Beispiel zeigt eine komplette SOAP-Nachricht, wobei der wichtigste Teil im SOAP-Body untergebracht ist. Diese Fehlernachricht wurde vom ePb-Server an den ePb-Client gesendet. Der Fehler ist aufgetreten, da der ePb-Client unvollständige Daten beim Aufruf des Service `getEintrag` geliefert hat.

Eine SOAP-Fehlernachricht beginnt mit dem Element `Fault`, welches mindestens die Elemente `Code` und `Reason` enthält. Als weiteres optionales Element wird hier `Detail` verwendet. Das `Code`-Element enthält zwei Unterelemente. Das Element `Value` enthält einen von der SOAP-Spezifikation vorgegebenen Hauptfehlercode (siehe SOAP Fault Codes). In dem obigen Beispiel wird der Wert `Sender` übermittelt. Er gibt an, dass ohne eine Änderung der SOAP-Nachricht vom ePb-Client der Service auf dem ePb-Server nicht ausgeführt werden kann. Welche Änderungen das sind, geht aus den nachfolgenden Informationen `Subcode` und `Detail` hervor. Der `Subcode` ist ein eindeutiger Fehleruntercode, der den Fehler genauer spezifiziert. Welche `Subcodes` es gibt, wird in den nachfolgenden Kapiteln beschrieben.
Das Element `Reason` enthält eine kurze Fehlerbeschreibung. Diese Beschreibung kann sprach- und landesspezifisch angegeben werden.
Im Element `Detail` erfolgt eine genauere Beschreibung des aufgetretenen Fehlers. Im Beispiel werden die Elemente aufgeführt, die der Service benötigt, aber nicht vom ePb-Client geliefert wurden.

Weitere Informationen über den Aufbau von SOAP Fault Nachrichten findet man hier: SOAP Version 1.2 Part 1: Messaging Framework

4.3.2 Allgemeine Fehler

Dieser Abschnitt beschreibt Fehler, die bei den meisten Services auftreten können.

genKeineBerechtigung

Ein ePb-Client, oder ePb-Server besitzt nicht die Berechtigung, einen bestimmten Service aufzurufen.
Reason:
Serviceberechtigung fehlt
Detail:
Im Detail-Teil befindet sich der Service, der aufgerufen wurde.

genTicketInvalid

Das bei einem Serviceaufruf verwendete Session- oder Konversationsticket ist ungültig.
Reason:
Ticket ist ungueltig
Detail:
Im Detail-Teil befindet sich das Ticket, das mit dem Serviceaufruf gesendet wurde.

genInternerServerfehler

Bei der bearbeitung eines Services auf dem ePb-Server ist ein Fehler aufgetreten. Ein Fehler dieser Art verhindert die weitere bearbeitung des Services.
Reason:
Interner Serverfehler
Detail:
Hier steht im Detail, um welchen Fehler es sich genau handelt. Dies könnte z.B. sein: Datenbank nicht erreichbar, Speicherfehler, ePb-Backend nicht erreichbar, externer Server nicht erreichbar

genDatenUnvollstaendig

Beim Aufruf eines Service ist eine Service-Nachricht übermittelt worden, die nicht alle Daten enthält, die für die Abarbeitung des Service benötigt werden.
Reason:
Unvollständige Daten für diesen Service
Detail:
Hier steht im Detail, um welchen Service es sich handelt und welche Daten fehlen.

genEintragInkonsistent

Bei der Auslieferung eines Eintrags wird dieser auf Konsistenz geprüft. Damit soll sichergestellt werden, dass der Eintrag so ausgeliefert wird, wie er archiviert wurde. Tritt bei dieser Prüfung ein Fehler auf, dann bekommt der ePb-Client diese Fehlermeldung
Reason:
Eintrag ist Inkonsistent
Detail:
In diesem Bereich wird die Eintragsnummer angegeben.

genStandesamtsnummerInvalid

Bei einigen Services wird die Standesamtsnummer übermittelt, damit der Service korrekt ausgeführt werden kann. Ist die übermittelte Nummer dem ePb-Server unbekannt, dann wird dieser Fehler zurück gesendet.
Reason:
Eintrag ist Inkonsistent
Detail:
In diesem Bereich wird die Eintragsnummer angegeben.

genEintragGesperrt

Bei Services, die Modifikationen bei einem Eintrag vornehmen, muss der Eintrag zuvor gesperrt werden. Ist der Eintrag bereits gesperrt, dann wird dieser Fehler zurückgesendet.
Reason:
Eintrag ist bereits gesperrt
Detail:
In diesem Bereich wird die Eintragsnummer und der Benutzer angegeben, der das Dokument bereits gesperrt hat.

genEintragInvalid

Ist kein Eintrag zu einer gewählten Eintragsnummer gespeichert, wird dieser Fehler zurückgeliefert.
Reason:
Eintrag unbekannt

Detail:
In diesem Bereich wird die gewünschte Eintragsnummer angegeben.

genBenutzerInvalid

Bei einigen Services wird die BenutzerID übermittelt, damit der Service korrekt ausgeführt werden kann. Ist die übermittelte ID dem ePb-Server unbekannt, dann wird dieser Fehler zurück gesendet.
Reason:
Benutzer ist unbekannt
Detail:
In diesem Bereich wird die unbekannte BenutzerID angegeben.

genServiceInvalid

Der vom ePb-Client angeforderte Service ist dem ePb-Server nicht bekannt.
Reason:
Service ist unbekannt
Detail:
In diesem Bereich wird der unbekannte Servicename angegeben.

genSchemavalidierungInvalid

Die im Header und im Body angegebenen Daten werden mit einem festgelegten XML-Schema validiert. Tritt hierbei ein Fehler auf, dann wird dieser Fehler zurück gegeben.
Reason:
Die Schemavalidierung ist fehlgeschlagen
Detail:
Hier wird der Bereich (Header, Body) angegeben, wo die Validierung fehlgeschlagen ist. Des Weiteren befindet sich hier der Servicename, des aufgerufenen Service.

genBadTimescape

Der angegebene Zeitraum ist ungültig
Reason:

Zeitraum ungültig
Detail:
Hier wird der gelieferte Zeitraum angegeben.

4.3.3 Fehler Zugangs-API

Dieser Abschnitt beschreibt Fehler, die bei den Services der Zugangs-API auftreten können.

zugAnmeldungFehlgeschlagen

Sind die gesendeten Authentifizierungsdaten dem ePb-Server bekannt, aber es kann aus irgend einem Grund keine Anmeldung erfolgen, dann wird diese Fehlermeldung geliefert. Gründe hierfür können sein:

- Zertifikat (Benutzer, Server) ist abgelaufen
- Benutzer- oder Serverkonto ist deaktiviert

Reason:
Die Anmeldung ist fehlgeschlagen
Detail:
Hier wird der Grund angegeben, warum die Authentifizierung fehlgeschlagen ist.

4.3.4 Fehler Archivierungs-API

Dieser Abschnitt beschreibt Fehler, die bei den Services der Archivierungs-API auftreten können.

arcArchivierungInvalid

Tritt bei der Archivierung eines Eintrags ein Fehler auf, dann wird diese Fehlermeldung geliefert. Gründe hierfür können sein:

- EintragsID-Konflikt
- Die Signatur ist ungültig
- Der Jahrgang ist bereits abgeschlossen

Reason:
Die Archivierung ist fehlgeschlagen

Detail:
Hier wird der Grund angegeben, warum die Archivierung fehlgeschlagen ist.

4.3.5 Fehler Registerverwaltungs-API

Dieser Abschnitt beschreibt Fehler, die bei den Services der Registerverwaltungs-API auftreten können.

regJahresabschlussInvalid

Tritt bei der Erstellung eines Jahresabschlusses ein Fehler auf, dann wird diese Fehlermeldung geliefert.
Reason:
Jahresabschluss fehlgeschlagen
Detail:
Hier wird der Grund angegeben, warum der Jahresabschluss fehlgeschlagen ist.

regJahrgangInvalid

Wird bei der Registerjahrgangssicherung ein Jahrgang angegeben, der nicht auf dem ePb-Server geführt wird, dann wird diese Fehlermeldung geliefert.
Reason:
Jahrgang nicht vorhanden
Detail:
Hier wird der übermittelte Jahrgang geliefert.

4.3.6 Fehler Administrations-API

Dieser Abschnitt beschreibt Fehler, die bei den Services der Administrations-API auftreten können.

admBenutzerIDVergeben

Wird ein neuer Benutzer angelegt, muss die zu vergebene Benutzer ID angegeben werden. Ist diese ID schon für einen anderen Benutzer vergeben, wird dieser Fehler ausgelöst.
Reason:
BenutzerID ist schon vergeben
Detail:
Hier wird der Benutzer aufgeführt, an den die gewünschte ID vergeben wurde.

admRechtInvalid

Soll einem Benutzer ein Recht zugeordnet werden, dass dem Server unbekannt ist, wird dieser Fehler zurückgeliefert.
Reason:
Gewähltes Recht ist unbekannt.
Detail:
Hier wird das gewünschte Recht zurückgeliefert.

admCertInvalid

Ein übermitteltes Zertifikat kann nicht auf seine Gültigkeit überprüft werden. Gründe können sein:

- Das digitale Zertifikat des Zertifikats ist ungültig.
- Zertifikat ist abgelaufen
- Das Zertifikat wurde nachträglich für ungültig erklärt. (per OCSP-Abfrage ermittelt)
- Das Zertifikat des ausstellenden Trustcenters dieses Zertifikats ist nicht im System eingetragen.

Reason:
Zertifikat nicht anerkannt.
Detail:
Hier wird der Grund der Ablehnung angegeben.

admServerInvalid

Die übermittelte ServerID ist dem System nicht bekannt.
Reason:
ServerID unbekannt.
Detail:
Hier wird die nicht erkannte ServerID zurückgeliefert.

admConfImpossible

Die gewünschte Konfiguration ist nicht möglich. Wahrscheinlich enthält sie wiedersprüchliche Angaben.
Reason:
Konfiguration nicht möglich.

Detail:
Hier wird der Grund der Ablehnung angegeben.

Kapitel 5

Services der Schnittstellen

5.1 Zugang

5.1.1 getSessionTicket

Dieser Service ist nur für einen lokalen ePb-Anwender erlaubt. Der Anwender identifiziert sich beim ePb-Server und erhält ein Session Ticket, das den Zugriff auf die Services erlaubt, für die der Anwender berechtigt ist.

Request:

```
<xs:element name="getSessionTicket">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
      <xs:element name="UsrId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der ePb-Client sendet für die Authentifizierung die bundesweit eindeutige Nummer des Standesamtes (Standesamtsnummer) und die Benutzer-Identifikation (UsrId) des ePb-Anwenders. Die UsrId muss in einem Standesamt eindeutig sein.

Response:

```
<xs:element name="rtSessionTicket" type="xs:string"/>
```

Als Rückgabe erhält der ePb-Client ein Session Ticket. Dies ist mit dem öffentlichen Schlüssel des ePb-Anwenders verschlüsselt worden. Der ePb-Client kann nun mit dem privaten Schlüssel des ePb-Anwenders das Session Ticket entschlüsseln und verwenden.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genStandesamtsnummerInvalid, genBenutzerInvalid

Spezielle Fehler:
zugAnmeldungFehlgeschlagen

offene Punkte:

Ticket soll an eine IP gebunden werden

5.1.2 discardSessionTicket

Dieser Service beendet die Session und macht das SessionTicket ungültig.

Request:

```
<xs:element name="discardSessionTicket" type="xs:string
   "/>
```

Der ePb-Client sendet hierzu das Session Ticket, welches ungültig gemacht werden soll.

Response:

```
<xs:element name="rtOK"/>
```

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genTicketInvalid

5.1.3 getConversationTicket

Dieser Service ist nur einem externen ePb-Server erlaubt. Er identifiziert sich und erhält ein Conversation Ticket, das den Zugriff auf *eine* Konversation gestattet. Danach verliert das Ticket seine Gültigkeit. Das Ticket gilt nur für ein bestimmtes Standesamt, das auf dem anfragenden ePb-Server geführt wird.

Request:

```
<xs:element name="getConversationTicket">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

In der Anfragenachricht wird die Standesamtsnummer übermittelt, für das Standesamt, das eine Konversation auf dem externen ePb-Server ausführen möchte.

Response:

```
<xs:element name="rtConversationTicket" type="xs:string
   "/>
```

Als Rückgabe erhält der ePb-Server ein Conversation Ticket. Dieses Ticket wurde mit dem öffentlichen Schlüssel des ePb-Servers verschlüsselt. Wurde das Ticket mit dem privaten Schlüssel des ePb-Servers entschlüsselt, kann eine Konversation auf dem externen ePb-Server durchgeführt werden.

Fehlerfall:

Die folgenden Fehlermeldungen werden an einen ePb-Server gesendet und nicht an einen ePb-Client. Der ePb-Server gibt diese Fehlermeldungen in Statuscodes weiter an den ePb-Client, wie es bei der unscharfen Suche (findEintrag) der Fall ist. Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig
Spezielle Fehler:
zugAnmeldungFehlgeschlagen (keine Berechtigung für das Standesamt)

5.2 Archivierung

5.2.1 insertEintrag

Archiviert einen neuen, oder einen bereits angelegten papiergeführten Eintrag. Nach der erfolgreichen Archivierung, können Einträge elektronisch geführt werden.

Request:

```
<xs:element name="insertEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Eintrag"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Bei der Anfrage übermittelt der ePb-Client das Element Eintrag (siehe 4.1.2 Eintrag). Die folgenden Daten müssen in dem Element Eintrag enthalten sein:

- Eintragsdaten
- Das PDF mit der Version (version).
- Die Signatur mit dem Signatur-Algorithmus(sigAlgorithmus) und -Format(sigFormat)

Response:

```
<xs:element name="rtOK"/>
```

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid
Spezielle Fehler:
arcArchivierungInvalid

5.2.2 Fortführung

coEintrag

Dieser Service ist der erste, der bei einer Fortführung (Konversation) verwendet werden muss. Ein Eintrag wird aus dem Archiv des ePb-Servers zur Fortführung ausgeliefert und gesperrt. Andere ePb-Anwender können auf den Eintrag nur noch lesend zugreifen.

Request:

```
<xs:element name="coEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der ePb-Client sendet die EintragsFID an den ePb-Server. Handelt es sich um einen Ersteintrag, dann ist die Fortführungsnummer = 0.

Response:

```
<xs:element name="rtEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Eintrag"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der angeforderte Eintrag wird ausgeliefert. Die Antwort enthält das Element Eintrag (siehe 4.1.2 Eintrag). In diesem sind die folgenden Daten enthalten:

- Eintragsdaten
- Das PDF mit einer Versionsangabe

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig,

genKeineBerechtigung, genTicketInvalid, genEintragGesperrt, genEintragInkonsistent, genEintragInvalid

ciEintrag

Ein Eintrag, der mit coEintrag zur Fortführung aus dem Archiv ausgeliefert wurde, wird nun nach der Fortführung wieder ins Archiv gestellt. Nach dem `ciEintrag` wird die Sperre auf den Eintrag aufgehoben.
Dieser Service ist Teil einer Konversation: er muss direkt auf `coEintrag` folgen (und sich natürlich auf denselben Eintrag beziehen).

Request:

```
<xs:element name="ciEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Eintrag"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der ePb-Server bekommt das Element Eintrag (siehe 4.1.2 Eintrag) übermittelt.
In diesem müssen die folgenden Daten enthalten sein.

- Eintragsdaten
- Fortführungsgrund
- Das PDF mit der Version (version).
- Die Signatur mit dem Signatur-Algorithmus(sigAlgorithmus) und -Format(sigFormat)

Response:

```
<xs:element name="rtOK"/>
```

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid
Spezielle Fehler:

arcArchivierungInvalid

unlockEintrag

Eine mit coEintrag gesetzte Sperre wird aufgehoben, ohne dass die Fortführung durchgeführt wurde.
Auch dieser Service ist nur innerhalb der Konversation möglich, die mit coEintrag beginnt.

Request:

```
<xs:element name="unlockEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EintragsFID" type="tEintragsFID
          "/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Es wird die EintragsID des zu entsperrenden Eintrags übermittelt.

Response:

```
<xs:element name="rtOK"/>
```

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genEintragInvalid

5.2.3 insertEdListe

Dieser Service archiviert Eintragsdaten zu einem oder mehreren Einträgen. Es werden nur Eintragsdaten ins Archiv übernommen, die bisher nicht verzeichnet waren. Sollen vorhandene Eintragsdaten verändert werden, muss der Service `updateEdListe` verwendet werden. Die bei einem Serviceaufruf übermittelten Eintragsdaten beziehen sich immer auf ein Register.

Request:

```
<xs:element name="insertEdListe">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="Geburtseintrag" maxOccurs="
         unbounded"/>
      <xs:element ref="Eheeintrag" maxOccurs="unbounded
         "/>
      <xs:element ref="Lebenspartnerschaftseintrag"
         maxOccurs="unbounded"/>
      <xs:element ref="Sterbeeintrag" maxOccurs="
         unbounded"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Beim Aufruf wird eine Liste von Eintragsdaten an den ePb-Server übermittelt. Die Eintragsdaten (Listenelemente) gehören hierbei alle zu ein und dem selben Register.

Response:

```
<xs:element name="rtEdInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID" minOccurs="0"
         maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Das Resultat enthält eine Liste mit EintragsFIDs, bei denen bereits Eintragsdaten auf dem ePb-Server vorliegen. Diese Eintragsdaten werden mit diesem Service nicht überschrieben. Ist die gelieferte Liste leer, dann wurden alle übermittelten Eintragsdaten gespeichert.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid

5.2.4 updateEdListe

Mit diesem Service können Eintragsdaten zu einem oder mehreren Einträgen auf dem ePb-Server gespeichert werden. Bereits vorhandene Eintragsdaten werden durch diesen Service überschrieben. Die Daten werden nicht überschrieben, wenn ein Eintrag gesperrt ist. Die bei einem Serviceaufruf übermittelten Eintragsdaten beziehen sich immer auf ein Register.

Request:

```
<xs:element name="updateEdListe">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="Geburtseintrag" maxOccurs="
         unbounded"/>
      <xs:element ref="Eheeintrag" maxOccurs="unbounded
         "/>
      <xs:element ref="Lebenspartnerschaftseintrag"
         maxOccurs="unbounded"/>
      <xs:element ref="Sterbeeintrag" maxOccurs="
         unbounded"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Beim Aufruf wird eine Liste von Eintragsdaten an den ePb-Server übermittelt. Die Eintragsdaten (Listenelemente) gehören hierbei alle zu ein und dem selben Register.

Response:

```
<xs:element name="rtEdInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID" minOccurs="0"
         maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Das Resultat enthält eine Liste mit EintragsFIDs, bei denen Zurzeit eine Sperrung vorliegt. Diese Eintragsdatendaten wurden nicht überschrieben. Ist die gelieferte Liste leer, dann wurden alle übermittelten Eintragsdaten gespeichert.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid

5.3 Suche

5.3.1 getEintrag

Punktsuche nach einem Eintrag durch Angabe der EintragsID. Wird der Service durch einen lokalen ePb-Anwender angefordert, wird die Suche eventuell an einen entfernten ePb-Server weitergeleitet, wenn der gesuchte Eintrag nicht im lokalen Archiv enthalten ist. Wenn der Service durch einen entfernten ePb-Server angefordert wird, wird die Suche nur auf dem eigenen, lokalen Archiv ausgeführt. Es wird immer die aktuellste Fortführung eines Eintrags geliefert

Request:

```
<xs:element name="getEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EintragsID" type="tEintragsID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Bei der Anfrage wird die EintragsID des gesuchten Eintrags übermittelt.

Response:

```
<xs:element name="rtEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Eintrag"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der ePb-Server liefert das Element Eintrag zurück. Die folgenden Informationen sind in diesem Element enthalten:

- Eintragsdaten
- Fortführungsgrund
- Das PDF mit der Version

Sind zu einem Eintrag nur die Eintragsdaten vorhanden, dann wird natürlich kein PDF geliefert.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genEintragInkonsistent, genEintragInvalid

5.3.2 findEintrag

Unscharfe Suche nach einem Eintrag durch die Angabe von diversen Suchinformationen. Wird der Service durch einen lokalen ePb-Anwender angefordert, werden die Suchinformationen analysiert und ggfs wird die Suche an einen externen ePb-Server weitergeleitet. Wenn der Service durch einen entfernten ePb-Server angefordert wird, wird die Suche nur auf dem eigenen, lokalen Archiv ausgeführt.

Request:

```
<xs:element name="findEintrag">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ortName" type="xs:string"
          minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Standesamtsnummer" type="xs:
          string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:choice>
        <xs:element name="g" type="tGSuchkriterien"/>
        <xs:element name="h" type="tASuchkriterien"/>
        <xs:element name="l" type="tASuchkriterien"/>
        <xs:element name="s" type="tSSuchkriterien"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Bei der Anfrage werden verschiedene Suchattribute angegeben. Diese unterscheiden sich teilweise bei den einzelnen Registern. Eine Suchanfrage kann immer nur auf ein Register gemacht werden. Die Suchkriterien der einzelnen Register enthalten alle allgemeine Suchkriterien, die in Kapitel 4.1.1 beschrieben werden. Des Weiteren kann im Geburtenregister nach einer Totgeburt und dem Geschlecht der Person gesucht werden (`tGSuchkriterien`). Im Sterberegister kann zusätzlich nach dem Geburtszeitpunkt der verstorbenen Person gesucht werden (`tSSuchkriterien`).

Response:

```
<xs:element name="rtSuchergebnis">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Eintrag" minOccurs="0" maxOccurs
          ="unbounded">
        <xs:complexType>
```

```
          <xs:sequence>
            <xs:element ref="EintragsFID"/>
            <xs:choice>
              <xs:element name="g" type="tGSuchkriterien
                 "/>
              <xs:element name="e" type="tASuchkriterien
                 "/>
              <xs:element name="l" type="tASuchkriterien
                 "/>
              <xs:element name="s" type="tSSuchkriterien
                 "/>
            </xs:choice>
            <xs:element name="PDFverfuegbar" type="xs:
               boolean"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Standesamt" maxOccurs="unbounded
         ">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Standesamtsnummer" type="xs
               :string"/>
            <xs:element name="Status" type="xs:int"
               minOccurs="0">
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Das Ergebnis besteht aus einer geordneten Liste, die als erstes die zutreffenden Einträge enthält und als zweites Informationen über die Standesämter, die an der Suche beteiligt waren. Das Listenelement Eintrag enthält die Suchdaten + EintragsID und Information darüber, ob ein PDF zu diesem Eintrag vorhanden ist. Das Listenelement Standesamt enthält die Standesamtsnummer und einen Statuscode. Der Statuscode informiert darüber, inwieweit das Standesamt an der Suche beteiligt war und welche Einsichtsberechtigung gewährt wird.
Nachfolgend die möglcihen Statuscodes:

- 1 - OK mit Einsichtsberechtigung in den PDF-Eintrag
- 2 - OK ohne Einsichtsberechtigung in den PDF-Eintrag
- 3 - nicht erreichbar

- 4 - keine Berechtigung
- 5 - kein ePb-Server

Bei Statuscode 1 und 2 muss folgendes beachtet werden. War die Suche erfolgreich in einem Standesamt, aber der gewünschte Eintrag wurde nicht gefunden, dann kann es sein, dass der Eintrag evtl. noch auf Papier geführt wird.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid

5.3.3 getEintragHistorie

Dieser Service liefert alle Fortführungen zu einem bestimmten Eintrag, die im ePb-Server archiviert sind. Handelt es sich bei dem gesuchten Eintrag um eine Nacherfassung, dann werden alle Fortführungen bis zu der Nacherfassung geliefert.

Request:

```
<xs:element name="getEintragHistorie">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EintragsID" type="tEintragsID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

In der Serviceanfrage wird die EintragsID geliefert.

Response:

```
<xs:element name="rtEintragHistorie">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Eintrag" minOccurs="0" maxOccurs="
          unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Als Antwort erhält der ePb-Client eine Liste von Eintrag-Elementen. Die einzelnen Elemente enthalten alle Informationen aus dem Element Eintrag, bis auf die Signatur. Diese kann bei Bedarf mit dem Service `getEintragSecInfo` angefordert werden.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genEintragInkonsistent, genEintragInvalid

5.3.4 getEintragSecInfo

Zu einem Eintrag werden alle Sicherheitsinformationen geliefert, die zur Überprüfung der Gültigkeit der Signatur des Eintrags benötigt werden.

Request:

```
<xs:element name="getEintragSecInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Serviceanfrage enthält die EintragsFID des gewünschten Eintrags.

Response:

```
<xs:element name="rtSecInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CertBin" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
              <xs:attribute name="id" type="xs:int" use="
                 required"/>
              <xs:attribute name="wid" type="xs:int"/>
              <xs:attribute name="CertFormat" use="
                 required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="SecInfo" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Datum" type="xs:date"/>
            <xs:element name="SecInfoBin" type="xs:
               base64Binary"/>
          </xs:sequence>
          <xs:attribute name="SecInfoTyp" use="required
             "/>
          <xs:attribute name="SecInfoFormat" use="
             required"/>
```

```
          <xs:attribute name="SecInfoAlgorithmus" use="
             required"/>
          <xs:attribute name="CertId" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Als Antwort erhält der ePb-Client eine Liste von Zertifikaten (`CertBin`) und Sicherheitsinformationen. Die Zertifikate werden im Base64-Format übermittelt. Das Attribute CertFormat gibt auskunft darüber, welches Format das Zertifikat besitzt. Die `id` ist eine eindeutige Zertifikats-ID, die vom ePb-Server für dieses Zertifikat vergeben wurde. Das Attribut `wid` enthält die Zertifikats-ID des Stammzertifikates. Die Stammzertifikate muss sich auch in der Zertifikatsliste befinden.
Des Weiteren folgen die Sicherheitsinformationen (`SecInfo`) zu einem Eintrag. Sicherheitsinformationen zu einem Eintrag können sein:

- Erstunterschrift des Eintrags bei der Neuerfassung oder Nacherfassung
- OCSP-Abfrage bei der Zertifizierungsstelle
- Zeitstempel von der Zertifizierungsstelle

Als Elemente enthält eine Sicherheitinformation das Erstellungsdatum und die Sicherheitsinfos im Base64-Format. Weiterhin werden die folgenden Attribute angegeben:

- `SecInfoTyp` Enthält den Typ der Sicherheitsinformation.
  - „eu“ - Erstunterschrift
  - „nu“ - Nacherfassungsunterschrift
  - „ocsp“ - OCSP-Abfrage
  - „zs“ - Zeitstempel
- `SecInfoFormat` Format der Sicherheitsinformation
- `SecInfoAlgorithmus` Verwendeter Verschlüsselungsalgorithmus
- `CertId` Zertifikats-ID, des Zertifikates, mit dem die Sicherheitsinformation geprüft werden kann. Das Zertifikat befindet sich in der Liste der übermittelten Zertifikate (`CertBin`).

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genEintragInvalid

5.4 Registerverwaltung

5.4.1 getNamenverz

Dieser Service liefert zu einem Zeitraum und einem Register eine Liste der Suchdaten all der Einträge und Fortführungen, deren Beurkundungsdatum in diesen Zeitraum fallen. Dieser Service ist nur für lokale ePb-Anwender erlaubt und bezieht sich nur auf die Daten im lokalen Archiv.

Request:

```
<xs:element name="getNamenverz">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Register" type="tRegister"/>
      <xs:element name="Zeitraum" type="tZeitraumDatum"/>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Bei der Anfrage werden die Standesamtsnummer, das Register un der Zeitraum übermittelt. Der Zeitraum bezieht sich auf alle Beurkundungen, die in dieser Zeit stattgefunden haben.

Response:

```
<xs:element name="rtNamenverz">
  <xs:complexType>
    <xs:choice>
      <xs:element name="g" type="tGSuchkriterien"
          minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="e" type="tASuchkriterien"
          minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="l" type="tASuchkriterien"
          minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="s" type="tSSuchkriterien"
          minOccurs="0" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

In der Serviceantwort sind alle Eintrags-Suchdaten zu einem Register enthalten, die in dem angegebenen Zeitraum gefunden wurden.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genStandesamtsnummerInvalid, genBadTimescape

5.4.2 makeJahresabschluss

Dieser Service führt einen Jahresabschluss für ein bestimmtes Register durch. Danach ist es nicht mehr möglich, neue Einträge in diesem Jahr zu archivieren. Als Rückgabe werden Informationen geliefert zu:

- Anzahl der in diesem Jahr verzeichneten neuen Einträge
- Vergabe von Eintragsnummern: niedrigste/höchste Nummer, Lücken und Zwischennummern
- Anzahl der in diesem Jahr gemachten Fortführungen mit Datum der Ersteintragung und dem Fortführungsgrund

Request:

```
<xs:element name="makeJahresabschluss">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
      <xs:element name="Register" type="tRegister"/>
      <xs:element name="Jahrgang" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

In der Anfragenachricht wird die Standesamtsnummer, das Register und der Jahrgang übermittelt.

Response:

```
<xs:element name="rtJahresabschluss">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Neueintraege">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="EintragsFID" minOccurs="0"
                maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="Anzahl" type="xs:int" use="
              required"/>
          <xs:attribute name="kleinsteNummer" type="xs:
              int" use="required"/>
          <xs:attribute name="groessteNummer" type="xs:
              int" use="required"/>
```

```
        </xs:complexType>
      </xs:element>
      <xs:element name="Zwischennummern" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Zwischennummer" type="xs:
               string" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Lueckennummern" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Lueckennummer" type="xs:
               string" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Fortfuehrungen">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Fortfuehrung" minOccurs
               ="0" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="EintragsFID"/>
                  <xs:element name="Ersteintragsdatum"
                     type="tDatum"/>
                  <xs:element name="Fortfuehrungsgrund"
                     type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="Anzahl" type="xs:int" use="
             required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Serviceantwort enthält als erstes eine Liste (`Neueintraege`) von Suchdaten der Neueinträgen, die in dem vergangenen Jahr gemacht wurden. Als Attribute werden die Anzahl, die kleinste und groesste Eintragsnummer der Neueintraege angegeben. Weiterhin folgen Listen über Zwischennummern und Lückennummern in diesem Jahrgang. Als nächstes folgt eine Liste (Fortfuehrungen) von Suchdaten der Fortführungen, die in dem vergangenen Jahr gemacht wurden.

Hierbei wird zu jeder Fortführung das Datum der Ersteintragung mitgeliefert. Die Anzahl der Fortführungen werden im Attribut `Anzahl` angegeben.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genStandesamtsnummerInvalid
Spezielle Fehler:
regJahresabschlussInvalid

5.4.3 getJahresinfo

Dieser Service testet, ob ein Jahresabschluss für ein bestimmtes Register eines Standesamtes durchgeführt werden kann. Der Jahrgang, in dem der Jahresabschluss getestet wird, kann beliebig gewählt werden.

Request:

```
<xs:element name="getJahresinfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
      <xs:element name="Register" type="tRegister"/>
      <xs:element name="Jahrgang" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

In der Anfragenachricht wird die Standesamtsnummer, das Register und der Jahrgang übermittelt.

Response:

Die Antwort dieses Services ist analog zu `makeJahresabschluss`.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genStandesamtsnummerInvalid
Spezielle Fehler:
regJahresabschlussInvalid

5.4.4 getRegisterJahressicherung

Liefert alle Informationen eines Jahrgangs eines Registers in einem datenbankneutralem Format. Zusätzlich wird eine Prüfsumme gebildet, mit der die Datenintegrität geprüft werden kann.

Request:

```
<xs:element name="getRegisterJahressicherung">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
      <xs:element name="Register" type="tRegister"/>
      <xs:element name="Jahrgang" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

In der Anfragenachricht wird die Standesamtsnummer, das Register und der Jahrgang übermittelt.

Response:

```
<xs:element name="rtRegisterJahressicherung">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Nutzinformationen" type="xs:
          base64Binary"/>
      <xs:element name="Pruefsumme" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwortnachricht enthält die zwei Hauptelemente `Nutzinformationen` und `Pruefsumme`. Die Nutzinformationen bestehen aus Daten im Base64-Format. Damit die Integrität der Nutzinformationen geprüft werden kann, wird eine Prüfsumme mitgesendet. Diese wurde über die Daten im Base64-Format gebildet. Werden die Base64-Daten decodiert, erhält man die Nutzinformation in XML-Notation, nach folgendem Schema:

```
<xs:element name="Nutzinformationen">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Eintragsinformation" maxOccurs="
          unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="Eintrag"/>
            <xs:element name="SecInfo" maxOccurs="
                unbounded">
              <xs:complexType>
```

```
                <xs:sequence>
                  <xs:element name="Datum" type="xs:date
                     "/>
                  <xs:element name="SecInfoBin" type="xs:
                     base64Binary"/>
                </xs:sequence>
                <xs:attribute name="SecInfoTyp" use="
                   required"/>
                <xs:attribute name="SecInfoFormat" use="
                   required"/>
                <xs:attribute name="SecInfoAlgorithmus"
                   use="required"/>
                <xs:attribute name="CertId" use="required
                   "/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Zertifikat" maxOccurs="unbounded
         ">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
              <xs:attribute name="id" type="xs:int" use="
                 required"/>
              <xs:attribute name="wid" type="xs:int"/>
              <xs:attribute name="CertFormat" use="
                 required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Nutzinformationen enthalten Daten zu Neueinträgen, Fortführungen (`Eintragsinformationen`) und einer Liste von Zertifikaten. Das Element `Eintragsinformationen` enthält alle Daten zu einem Eintrag, die auf dem ePb-Server vorhanden sind und alle Sicherheitsinformationen (`SecInfo`), die zu diesem Eintrag erstellt wurden. Zertifikate zum prüfen der Sicherheitsinformationen befinden sich in der Zertifikatsliste.

Fehlerfall:

Allgemeine Fehler:
genInternerServerfehler, genSchemavalidierungInvalid, genDatenUnvollstaendig, genKeineBerechtigung, genTicketInvalid, genStandesamtsnummerInvalid
Spezielle Fehler:
regJahrgangInvalid

5.5 Administration

5.5.1 getUsr

Der Service getUsr liefert die Stammdaten eines lokalen Benutzers, die im ePb-Server abgelegt sind.

Request:

```
<xs:element name="getUsr">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
      <xs:element name="UsrId" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Beim Aufruf werden die ID des Benutzers (userID) und die Standesamtsnummer des Benutzers übergeben. Dabei können nur die Daten der lokalen Benutzer abgerufen werden. Die Standesamtsnummer muß angegeben werden, da ein ePb-Server mandentenfähig ist. Ein einzelner Server kann die Benutzer und Daten mehrerer Standesämter beherbergen.

Response:

```
<xs:element name="rtUsr">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="usrData" type="tUsr"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort (rtUsr) besteht im Erfolgsfall aus einem Element userData vom Typ tUsr. Dieses Element enthält den Namen und Vornamen des Benutzers, die Gültigkeitsspanne des Benutzerkontos, die zugehörige Standesamtsnummer und einen Hinweis darauf, ob das Benutzerkonto im Moment aktiv oder inaktiv ist. Für nähere Informationen zum Typ tUsr siehe bei dessen Beschreibung im Kapitel Struktur der Schnittstellen.

Fehlerfall:

Wenn der Benutzer anhand seiner ID und der Standesamtsnummer nicht ermittelt werden konnte oder wenn eine Standesamtsnummer angegeben wurde, für die der ePb-Server nicht zuständig ist, wird eine entsprechende Fehlermeldung zurückgegeben. Ebenso, wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung, genBenutzerInvalid, genStandesamtsnummerInvalid

5.5.2 getUsrListe

Der Service getUsrListe liefert eine Liste der Stammdaten aller lokalen Benutzers, die im ePb-Server abgelegt sind.

Request:

```
<xs:element name="getUsrListe"/>
```

Der Service erfordert keine Parameter.

Response:

```
<xs:element name="rtUsrListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Usr" type="tUsr" minOccurs="0"
          maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort (rtUsrListe) besteht im Erfolgsfall aus einer Sequenz von Elementen des Typs tUsr. Diese Elemente enthalten den Namen und Vornamen des Benutzers, die Gültigkeitsspanne des Benutzerkontos, die zugehörige Standesamtsnummer und einen Hinweis darauf, ob das Benutzerkonto im Moment aktiv oder inaktiv ist. Für nähere Informationen zum Typ tUsr siehe bei dessen Beschreibung im Kapitel Struktur der Schnittstellen. Die Liste enthält alle Benutzer derjenigen Standesämter für die der aufrufende Administrator über Zugriffsrechte verfügt.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung, genBenutzerInvalid

5.5.3 insertUsr

Mit dem Service insertUsr kann ein neuer Benutzer im System angelegt werden. Das neue Konto kann erst aktiv geschaltet werden, wenn ein Zertifikat damit verknüpft wurde.

Request:

```
<xs:element name="insertUsr">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="UsrData" type="tUsr"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Stammdaten des neuen Benutzers werden über das Element userData des Typs tUsr (siehe Kapitel Struktur der Schnittstellen) übergeben.

Response:

```
<xs:element name="rtOK"/>
```

Nach der erfolgreichen Registrierung des neuen Benutzers wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Bei Kollisionen mit bereits vorhandenen Benutzern (mit gleicher userID), nicht vorhandener Standesamtsnummer oder wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admBenutzerIDVergeben

5.5.4 updateUsr

Mit dem Service updatetUsr können die im ePb-Server hinterlegten Stammdaten eines bestehenden Benutzers geändert werden.

Request:

```
<xs:element name="updateUsr">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="UsrData" type="tUsr"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Stammdaten des neuen Benutzers werden über das Element userData des Typs tUsr (siehe Kapitel Struktur der Schnittstellen) übergeben.

Response:

```
<xs:element name="rtOK"/>
```

Nach der erfolgreichen Änderung der Daten wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Existiert der mit der userID spezifizierte Benutzer nicht oder ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung, genBenutzerInvalidBenutzer

5.5.5 setUsrRechte

Mit dem Service updatetUsr können die im ePb-Server hinterlegten Benutzerrechte eines bestehenden Benutzers geändert werden.

Request:

```
<xs:element name="setUsrRechte">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
      <xs:element name="UsrId"/>
      <xs:element name="Recht" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Mod">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="add"/>
                  <xs:enumeration value="sub"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Bereich" type="xs:string">
            </xs:element>
            <xs:element name="ServerID" type="xs:string">
            </xs:element>
            <xs:element name="Level" type="xs:int"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Mit der Angabe von Standesamtsnummer und usrID wird der zu modifizierende Benutzer ausgewählt. Die Zuordnung oder Aberkennung von Rechten geschieht durch eine Liste von Elementen mit dem Namen Recht. Das dort enthaltene Element Mod enthält entweder add um ein Recht hinzuzufügen oder sub um ein Recht zu löschen. Das Element Bereich enthält das Ziel einer Transaktion, für die ein Recht vergeben oder entzogen wird. (Beispiele: Geburtenregister, Adminbereich, Revision). Das Element ServerID enthält die Servernummer des Servers, für den das Recht erteilt werden soll. Die Festlegung des Rechts erfolgt in Stufen über das Element Level. Die Bedeutung dieses Elements:

- 0: keine Berechtigung

- 1: Suchen von Einträgen
- 2: Lesen von Einträgen
- 3: Verfassen / Fortführen von Einträgen
- 4: Revision erlaubt (schließt vorige Rechte aus.)
- 5: Administrieren und Revision erlaubt (schließt vorige Rechte aus.)

Response:

```
<xs:element name="rtOK"/>
```

Nach der erfolgreichen Änderung der Benutzerrechte wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Existiert der mit der userID spezifizierte Benutzer nicht, wurde ein unbekanntes Recht ausgewählt oder ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung, genBenutzerInvalid,
- Spezielle Fehler: admRechtInvalid, admServerInvalid

5.5.6 insertUsrCert

Mit dem Service insertUsrCert kann ein neues Benutzerzertifikat im System abgelegt werden. Dabei muss das Zertifikat einem einem bestimmten ePb-Benutzer zugeordnet werden.

Request:

```
<xs:element name="insertUsrCert">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CertBin">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
              <xs:attribute name="CertFormat" use="
                 required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="UsrId" type="xs:int"/>
      <xs:element name="Standesamtsnummer" type="xs:
         string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erwartet im Element certBin das Zertifikat kodiert als base64. Das Speicherformat wird über das Attribut CertFormat übergeben. Die Zuordnung zu einem Benutzerkonto erfolgt mit der Angabe der usrID und der Standesamtsnummer.

Response:

```
<xs:element name="rtCertID" type="xs:string"/>
```

Nach der erfolgreichen Speicherung des neuen Zertifikats wird ein Element mit der vergebenen ID zurückgeliefert.

Fehlerfall:

Das Zertifikat wird anhand der bei der Konfiguration des ePb-Servers festgelegten Bestimmungen für Zertifikate vor der Abspeicherung auf seine Gültigkeit

geprüft. Dies schließt ein, dass das Zertifikat des ausstellenden Trustcenters im System bekannt ist. Kann die Gültigkeit nicht bewiesen werden, wird eine Fehlermeldung zurückgeliefert. Eine Fehlermeldung wird auch dann ausgegeben, wenn eine dem System unbekannte Benutzernummer übergeben wurde. Ebenso wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung, genBenutzerInvalidm
- Spezielle Fehler: admCertInvalid

5.5.7 getUsrCertListe

Der Service getUsrCertListe liefert eine Liste aller Zertifikate eines Benutzers zurück.

Request:

```
<xs:element name="getUsrCertListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="UsrId" type="xs:int"/>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erfordert die Angabe der usrID und der Standesamtsnummer.

Response:

```
<xs:element name="rtUsrCertListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Cert" minOccurs="0" maxOccurs="
          unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="CertBin">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:base64Binary">
                    <xs:attribute name="CertFormat" use="
                        required"/>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="tsCertId" type="xs:int"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste aller Zertifikate des Benutzers in Binärform mit den IDs deren Trustcenterzertifikate und dem Namen des Inhabers. Zusätzlich wird über das Attribut CertFormat das Speicherformat des Zertifikats übermittelt.

Fehlerfall:

Existiert die Kombination aus usrID und Standesamtsnummer nicht oder ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung, genBenutzerInvalid

5.5.8 getSvr

Der Service getSvr liefert alle Informationen über einen angeschlossenen ePb-Server.

Request:

```
<xs:element name="getSvr">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SvrId" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erfordert die Übergabe der SvrID des gewünschten Servers.

Response:

```
<xs:element name="rtSvrInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="svrID" type="xs:string"/>
      <xs:element name="Serverbeschreibung" type="xs:
         string"/>
      <xs:element name="Standesamt" minOccurs="0"
         maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Standesamtsnummer" type="xs
               :string"/>
            <xs:element name="Standesamtsname" type="xs:
               string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Zurückgeliefert wird die svrID und die Beschreibung des Servers. Zusätzlich eine Liste aller Standesämter, für die dieser Server zuständig ist. Dabei wird die Standesamtsnummer und der Name des Standesamtes geliefert.

Fehlerfall:

Ist die SvrID unbekannt oder ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admServerInvalid

5.5.9 getSvrListe

Der Service getSvr liefert eine Liste aller Informationen über alle angeschlossenen ePb-Server.

Request:

```
<xs:element name="getSvrListe"/>
```

Der Service erfordert keine Parameter.

Response:

```
<xs:element name="rtSvrInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="rtSvrInfo" minOccurs="0" maxOccurs
         ="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Zurückgeliefert wird eine Liste von Severinfos, die jeweils genau wie die Antwort von getSvr aufgebaut sind (s.o.).

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung

5.5.10 insertSvr

Mit dem Service insertSvr wird ein externer ePb-Server mit dem lokalen gekoppelt. Damit wird die Suche und eventuell das Abrufen von Einträgen auf diesem externen Server ermöglicht. Die Kommunikation mit dem externen Server wird erst dann möglich, wenn auch ein Serverzertifikat eingetragen wurde.

Request:

```
<xs:element name="insertSvr">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="svrID" type="xs:string"/>
      <xs:element name="Standesamtsnummer" type="xs:
          string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erwartet die SvrID und eine Liste der Standesamtsnummers, für die dieser Server zuständig ist.

Response:

```
<xs:element name="rtOK"/>
```

Wurde der Server erfolgreich eingetragen, wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert. Bei der Anbindung können verschiedene Fehler auftreten. Von Kommunikationsproblemen bis zu Unstimmigkeiten bei den Zuständigkeiten.

- Allgemeine Fehler: genInternerSeverfehler, genBeutzerInvalid, genKeineBerechtigung,
- Spezielle Fehler: admConfImpossible

5.5.11 updateSvr

Mit dem Service updateSvr können die lokalen Informationen über den eigenen Server, wie auch über externe Server geändert werden.

Request:

```
<xs:element name="updateSvr">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="svrID" type="xs:string"/>
      <xs:element name="Standesamtsnummer" type="xs:
          string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erwartet die SvrID und eine Lister der Standesamtsnumern, für die dieser Server zuständig ist.

Response:

```
<xs:element name="rtOK"/>
```

Wurden die Informationen erfolgreich eingetragen, wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admServerInvalid

5.5.12 insertSvrCert

Mit dem Service insertSvrCert kann ein neues Serverzertifikat im System abgelegt und einem Server zugeordnet werden. Serverzertifikate dienen der Authentifikation zwischen ePb-Servern und ermöglichen eine sichere, verschlüsselte Kommunikation.

Request:

```
<xs:element name="insertSvrCert">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CertBin">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
              <xs:attribute name="CertFormat" use="
                 required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="SvrId" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erwartet im Element certBin das Zertifikat kodiert als base64. Das Speicherformat ist im Attribut certFormat enthalten. Mit svrID muss die ID des Servers angegeben werden, dem das Zertifikat zugeordnet werden soll.

Response:

```
<xs:element name="rtCertID" type="xs:string"/>
```

Nach der erfolgreichen Speicherung des neuen Zertifikats wird ein Element mit der vergebenen ID zurückgeliefert.

Fehlerfall:

Das Zertifikat wird anhand der bei der Konfiguration des ePb-Servers festgelegten Bestimmungen für Zertifikate vor der Abspeicherung auf seine Gültigkeit geprüft. Dies schließt ein, dass das Zertifikat des ausstellenden Trustcenters

im System bekannt ist. Kann die Gültigkeit nicht bewiesen werden, wird eine Fehlermeldung zurückgeliefert. Eine fehlermeldung wird auch dann ausgegeben, wenn die angegebene svrID nicht bekannt ist. Ebenso wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admServerInvalid, admCertInvalid

5.5.13 getSvrCertListe

Der Service getSvrCertListe liefert eine Liste aller Serverzertifikate, in binärer Form, zurück, die dem ausgewählten ePb-Server zugeordnet werden.

Request:

```
<xs:element name="getSvrCertListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="svrID" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erfordert die als Parameter die svrID.

Response:

```
<xs:element name="rtSvrCertListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Cert" minOccurs="0" maxOccurs="
          unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="CertBin">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:base64Binary">
                    <xs:attribute name="CertFormat" use="
                        required"/>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="tsCertId" type="xs:int"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste aller lokalen Serverzertifikaten mit den IDs deren Trustcenterzertifikate, die dem angeforderten Server zugeordnet werden. Über

das Attribut certFormat wird das Speicherformat des Zertifikats übergeben.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert. Ebenso, wenn die angeforderte SvrID dem System nicht bekannt ist.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admServerInvalid

5.5.14 insertCACert

Mit dem Service insertCACert kann ein neues Trustcenterzertifikat im System abgelegt werden. Es müssen alle Zertifikate der vertrauenswürdigen Trustcenter im System hinterlegt sein. Ansonsten können Benutzer- und Serverzertifikate nicht auf ihre Gültigkeit hin überprüft werden und werden vom System nicht anerkannt.

Request:

```
<xs:element name="insertCACert">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CertData">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="tssUrl" type="xs:string"/>
            <xs:element name="ocspUrl" type="xs:string"/>
            <xs:element name="CertBin" type="xs:
               base64Binary"/>
            <xs:element name="wCertId" type="xs:int"
               minOccurs="0">
            </xs:element>
          </xs:sequence>
          <xs:attribute name="certFormat"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Beim Aufruf werden die Adressen für den Zeitstempeldienst, die OCSP-Abfragen und in certBin das Zertifikat kodiert als base64 übergeben. Das optionale Element nimmt die ID eines eventuell abgelegten Wurzeltrustcenterzertifikat auf. Mit dem Attribut certFormat wird das Speicherformat angegeben, damit der ePb-Server das Zertifikat auswerten kann.

Response:

```
<xs:element name="rtCertID" type="xs:string"/>
```

Nach der erfolgreichen Speicherung des neuen Zertifikats wird ein Element mit der vergebenen ID zurückgeliefert.

Fehlerfall:

Das Zertifikat wird anhand der bei der Konfiguration des ePb-Servers festgelegten Bestimmungen für Zertifikate vor der Abspeicherung auf seine Gültigkeit geprüft. Kann die Gültigkeit nicht bewiesen werden, wird eine Fehlermeldung zurückgeliefert. Ebenso wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admCertInvalid

5.5.15 getCACertListe

Der Service getCACertListe liefert eine Liste aller Trustcenterzertifikate, in binärer Form, die auf dem lokalen ePb-Server hinterlegt sind.

Request:

```
<xs:element name="getCACertListe"/>
```

Der Service erfordert keine Parameter.

Response:

```
<xs:element name="rtCACertListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ca" minOccurs="0" maxOccurs="
          unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="id" type="xs:int"/>
            <xs:element name="tssUrl" type="xs:string"/>
            <xs:element name="ocspUrl" type="xs:string"/>
            <xs:element name="CertBin" type="xs:
                base64Binary"/>
            <xs:element name="wCertId" type="xs:int"
                minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="certFormat"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste aller lokalen Trustcenterzertifikate mit den IDs deren Wurzeltrustcenterzertifikate und den Adressen für Zeitstempel- und OCSP-Dienste. Das Attribut certFormat enthält das Speicherformat des Zertifikats.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung

5.5.16 checkZertStatus

Der Service checkSecStatus überprüft alle lokal abgelegten Zertifikate, deren Gültigkeitsdauer noch nicht abgelaufen ist, auf ihre Gültigkeit.

Request:

```
<xs:element name="checkZertStatus"/>
```

Der Service erfordert keine Parameter.

Response:

```
<xs:element name="rtZertStatus">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="serverZertID" type="xs:string"
          minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="userZertID" type="xs:string"
          minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="trustcenterZertID" type="xs:
          string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste aller Zertifikat-IDs, die nicht mehr gültig sind. Aufgeteilt auf die Elemente serverZertID, userZertID und trustcenterZertID.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung

5.5.17 updateSecStatus

Der Service checkSecStatus überprüft alle lokalen Einträge auf die Gültigkeit der letzten Zeitstempel und fügt bei Bedarf neue Zeitstempel an. Der Bedarf wird daran festgestellt, dass ein aktuelleres Zertifikat des Trustcenters existiert, das den letzten zeitstempel ausgestellt hat.

Request:

```
<xs:element name="updateSecStatus"/>
```

Der Service erfordert keine Parameter

Response:

```
<xs:element name="rtSecStatus">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="EintragsFID" minOccurs="0"
         maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste aller Eintragsnummern, deren Einträge mit einem neuen Zeitstempel versehen wurden.

Fehlerfall:

Konnte der letzte Zeitstempel nicht verifiziert werden, wird ein Fehler zurückgeliefert. Diese Einträge sind ungültig geworden. Konnte das Trustcenter nicht erreicht werden, wird ein Fehler zurückgeliefert. Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genEintragInkonsistent, genKeineBerechtigung

5.5.18 getSecConfListe

Der Service getSecConfListe liefert eine Liste der Sicherheitskonfigurationen des lokalen ePb-Servers seit seiner Inbetriebnahme.

Request:

```
<xs:element name="getSecConfListe"/>
```

Der Service erfordert keine Parameter

Response:

```
<xs:element name="rtSecConfListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="confInfo" minOccurs="0" maxOccurs
          ="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="secConfNr" type="xs:int"/>
            <xs:element name="gueltigkeit" type="
                tZeitraumDatum"/>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="wert" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste mit jeweils der ID einer Option, dem Gültigkeitszeitraum und einem Name-Wert-Paar.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung

5.5.19 saveSecConfListe

Mit dem Service saveSecConfListe können neue Sicherheitseinstellungen für den ePb-Server abgelegt werden. Bestehende Einstellungen werden nicht gelöscht, sondern deren Gültigkeitszeitraum beendet. Die neuen Einstellungen gelten ab sofort bis neue Einstellungen abgelegt werden.

Request:

```
<xs:element name="saveSecConfListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="secConf" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="wert" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erthält eine Liste von Name-Wert-Paaren.

Response:

```
<xs:element name="rtOK"/>
```

Sind die neuen Optionen abgelegt wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admConfImpossible

5.5.20 getSvrConfListe

Der Service getSvrConfListe liefert eine Liste der allgemeinen Konfigurationen des lokalen ePb-Servers seit seiner Inbetriebnahme.

Request:

```
<xs:element name="getSvrConfListe"/>
```

Der Service erfordert keine Parameter

Response:

```
<xs:element name="rtSvrConfListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="confInfo" minOccurs="0" maxOccurs
          ="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="svrConfNr" type="xs:int"/>
            <xs:element name="gueltigkeit" type="
                tZeitraumDatum"/>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="wert" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste mit jeweils der ID einer Option, dem Gültigkeitszeitraum und einem Name-Wert-Paar.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admServerInvalid

5.5.21 saveSvrConfListe

Mit dem Service saveSvrConfListe können neue allgemeine Einstellungen für den ePb-Server abgelegt werden. Bestehende Einstellungen werden nicht gelöscht, sondern deren Gültigkeitszeitraum beendet. Die neuen Einstellungen gelten ab sofort bis neue Einstellungen abgelegt werden.

Request:

```
<xs:element name="saveSvrConfListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="svrConf" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="wert" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erthält eine Liste von Name-Wert-Paaren.

Response:

```
<xs:element name="rtOK"/>
```

Sind die neuen Optionen abgelegt wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: genServerInvalid

5.6 Revision

5.6.1 getRevConfListe

Der Service getRevConfListe liefert eine Liste der Konfigurationen für die Revision des lokalen ePb-Servers seit seiner Inbetriebnahme.

Request:

```
<xs:element name="getRevConfListe"/>
```

Der Service erfordert keine Parameter

Response:

```
<xs:element name="rtRevConfListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="confInfo" minOccurs="0" maxOccurs
          ="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="revConfNr" type="xs:int"/>
            <xs:element name="gueltigkeit" type="
                tZeitraumDatum"/>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="wert" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Die Antwort enthält eine Liste mit jeweils der ID einer Option, dem Gültigkeitszeitraum und einem Name-Wert-Paar.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung

5.6.2 saveRevConfListe

Mit dem Service saveRevConfListe können neue Einstellungen für die Revision innerhalb des lokalen ePb-Servers abgelegt werden. Bestehende Einstellungen werden nicht gelöscht, sondern deren Gültigkeitszeitraum beendet. Die neuen Einstellungen gelten ab sofort bis neue Einstellungen abgelegt werden.

Request:

```
<xs:element name="saveRevConfListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="revConf" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="wert" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Der Service erthält eine Liste von Name-Wert-Paaren.

Response:

```
<xs:element name="rtOK"/>
```

Sind die neuen Optionen abgelegt wird ein Element rtOK zurückgeliefert.

Fehlerfall:

Ruft ein Benutzer diesen Service auf, der über unzureichende Zugriffsrechte verfügt, wird eine Fehlermeldung zurückgeliefert.

- Allgemeine Fehler: genKeineBerechtigung

5.6.3 getUsrRevInfo

Der Service getUsrRevInfo liefert eine Liste aller Aktionen eines Users ab einem bestimmten Zeitpunkt.

Request:

```
<xs:element name="getUsrRevInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="usrID" type="xs:string"/>
      <xs:element name="Standesamtsnummer" type="xs:
          string"/>
      <xs:element name="Datum" type="tDatum"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Dem Service wird die userID und die Standesamtsnummer übergeben. Das Element Datum bestimmt das Datum, ab dem die Aktionen ausgegeben werden sollen.

Response:

```
<xs:element name="rtRevInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="revInfo" minOccurs="0" maxOccurs="
          unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Zurück kommt eine Liste mit Paaren von dem Aktionszeitpunkt und einer Beschreibung der Aktion.

Fehlerfall:

Existiert die Kombination aus Standesamtsnummer und userID nicht auf dem lokalen ePb-Server wird eine Fehlermeldung ausgegeben. Ebenso, wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung, genBenutzerInvalid

5.6.4 getEintragRevInfo

Der Service getEintragRevInfo liefert eine Liste aller Aktionen die auf den gewählten Eintrag ab einem gewählten Zeitpunkt angewandt wurden.

Request:

```
<xs:element name="getEintragRevInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EintragID" type="xs:string"/>
      <xs:element name="Datum" type="tDatum"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Dem Service wird die Eintragsnummer übergeben. Das Element Datum bestimmt das Datum, ab dem die Aktionen ausgegeben werden sollen.

Response:

```
<xs:element name="rtRevInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="revInfo" minOccurs="0" maxOccurs="
          unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Zurück kommt eine Liste mit Paaren von dem Aktionszeitpunkt und einer Beschreibung der Aktion.

Fehlerfall:

Existiert die Eintragsnummer nicht auf dem lokalen ePb-Server wird eine Fehlermeldung ausgegeben. Ebenso, wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung,genEintragInvalid

5.6.5 getServiceRevInfo

Der Service getServiceRevInfo liefert eine Liste aller Aufrufe des ausgewählten Services ab einem gewählten Zeitpunkt.

Request:

```
<xs:element name="getServiceRevInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Service" type="xs:string"/>
      <xs:element name="Datum" type="tDatum"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Dem Service wird der Name des zu untersuchenden Services im Klartext übergeben. Das Element Datum bestimmt das Datum, ab dem die Aktionen ausgegeben werden sollen.

Response:

```
<xs:element name="rtRevInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="revInfo" minOccurs="0" maxOccurs="
          unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Zurück kommt eine Liste mit Paaren von dem Aufrufzeitpunkt und einer Beschreibung des Aufrufs (inlusive dem Benutzer).

Fehlerfall:

Existiert der Service nicht, wird eine Fehlermeldung ausgegeben. Ebenso, wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung, genServiceInvalid

5.6.6 getSvrRevInfo

Der Service getSvrRevInfo liefert eine Liste aller Aktionen, die ein externer ePb-Server auf dem lokalen Server ab einem bestimmten Zeitpunkt ausgeführt hat.

Request:

```
<xs:element name="getSvrRevInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="svrID" type="xs:string"/>
      <xs:element name="Datum" type="tDatum"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Dem Service wird die Servernummer übergeben.Das Element Datum bestimmt das Datum, ab dem die Aktionen ausgegeben werden sollen.

Response:

```
<xs:element name="rtRevInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="revInfo" minOccurs="0" maxOccurs="
          unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Zurück kommt eine Liste mit Paaren von dem Aktionszeitpunkt und einer Beschreibung der Aktion.

Fehlerfall:

Existiert die Servernummer nicht auf dem lokalen ePb-Server wird eine Fehlermeldung ausgegeben. Ebenso, wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte verfügt.

- Allgemeine Fehler: genKeineBerechtigung
- Spezielle Fehler: admServerInvalid

5.6.7 purgeRevInfo

Der Service purgeRevInfo löscht die Revisionsinformationen eines gewählten Zeitraums. Zusätzlich können die gelöschten Informationen zurückgeliefert werden um sie der Archivierung zuzuführen.

Request:

```
<xs:element name="purgeRevInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Zeitraum" type="tZeitraumDatum"/>
      <xs:element name="Archiv" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Dem Service wird der Zeitraum übergeben, in der die Revisionsdaten zu löschen sind übergeben. Mit dem Element archiv wird ausgewählt, ob die gelöschten Informationen zurückgeliefert werden sollen.

Response:

```
<xs:element name="rtOK"/>
```

Soll nur gelöscht werden, wird ein Element rtOK zurückgeliefert.

```
<xs:element name="rtRevInfoListe">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="revInfo" minOccurs="0" maxOccurs="
         unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Wurde das Element archiv übergeben, wird eine Liste der gelöschten Informationen als Liste mit Paaren von dem Aktionszeitpunkt und einer Beschreibung der Aktion zurückgeliefert.

Fehlerfall:

Ist der Zeitraum unstimmig wird eine Fehlermeldung ausgegeben. Ebenso, wenn ein Benutzer diesen Service aufruft, der über unzureichende Zugriffsrechte ver-

fügt.

- Allgemeine Fehler: genKeineBerechtigung, genBadTimescape