UNIVERSITÄTSmedizin. MAINZ
Institut für Medizinische Biometrie, Epidemiologie und Informatik (IMBEI)

Andreas Borg, M.A.
Dipl.-Inf. Martin Lablans

Obere Zahlbacher Straße 69
55131 Mainz

Fon: +49 (0) 6131 17-8210
Fax: +49 (0) 6131 17-2968
E-Mail: lablans@uni-mainz.de

Mainzelliste – Zugriff durch MDAT-Server im Anwendungsfall „behandlungsfernes Register“

Versionshistorie

Version	Datum	Änderung/Status	Autor
0.1	19.06.2012	Erster Wurf	Andreas Borg, Martin Lablans
0.2	20.06.2012	Shared Secret, Tokengültigkeit bei Fehlschlag, Anlege-Callback-Handling, Beispielsitzung	Martin Lablans
0.3	25.06.2012	Content-Type, Anlege-Callback-Parameter	Andreas Borg
0.4	26.06.2012	Sequenzdiagramm für Beispielsitzung, Strategien zur Gewinnung der Token-ID	Andreas Borg
0.5	26.06.2012	Kommentierung Sequenzdiagramm	Martin Lablans
0.6	27.06.2012	Korrekturen Sequenzdiagramm	Andreas Borg
1.0	27.06.2012	Freigabe an ersten MDAT-Partner	Martin Lablans
1.1	06.08.2012	Änderung bei der Rückgabe: URL in Location-Header, URL und ID in JSON-Objekt	Andreas Borg
1.2	06.02.2013	Anwendungsbeispiel für Session	Andreas Borg
1.3	19.02.2013	Erweiterungen aus SecuTrial-Anbindung	Andreas Borg
1.4	08.05.2013	Überarbeitung Abschnitt „Allgemeines“	Martin Lablans
1.4.1	09.08.2013	Ergänzung „mainzellisteApiKey“ beim Anlegen eines Tokens	Andreas Borg

1. Allgemeines

Dieses Dokument beschreibt den Zugriff auf die Mainzelliste im Anwendungsfall „behandlungsfernes Patientenregister“, in dem Personenidentifikatoren für genau ein patientendatenführendes System (bspw. ein Register, hier „MDAT-Server“ genannt) erzeugt werden und keine Patientendaten nachgeladen werden müssen.

Alle Serverfunktionen werden über eine REST-artige Schnittstelle angesprochen. Grundlegender Ablauf für alle Aktionen: Der MDAT-Server registriert pro Benutzersitzung zunächst eine *Session*. Innerhalb dieser Session kann eine beliebige Anzahl *Tokens* angefordert werden, welches jeweils für eine bestimmte Aktion (z.B. Patienten hinzufügen) einmal gültig sind. Ein Token wird beim Abruf der eigentlichen Funktion zur Identifikation mitgeliefert und ist nach Ausführen der Funktion ungültig.

2. Identifikation, Session- und Tokenhandling

Der MDAT-Server identifiziert sich gegenüber der Mainzelliste mit einem einmal vereinbarten, geteilten Geheimnis (*mainzellisteApiKey*), welches auf dem IDAT-Server (der Mainzelliste) eingetragen wird. Neben diesem Schlüssel muss dem Betreiber der Mainzelliste die Adresse des MDAT-Servers mitgeteilt werden.

Eine Session wird angelegt durch POST-Request auf /sessions[1]; das Geheimnis wird als Parameter *mainzelliste-ApiKey* übergeben. Rückgabe ist die URL der Session im Header *Location* (im Folgenden *session-url* benannt). URL und Session-ID werden außerdem als JSON-Objekt ausgegeben.

Sessions kommen in erster Linie in Anwendungsfällen zum Einsatz, bei denen eine Rückidentifizierung im Browser mittels temporärer Identifikatoren (Temp-IDs) erfolgt. In diesem Fall können beim Abmelden eines Benutzers alle für ihn erzeugten Temp-IDs durch Löschen der Session invalidiert werden. Im hier dargestellten Fall (Mainzelliste dient nur zum Abruf von PIDs) kann für jeden Zugriff eine neue Session erstellt werden, diese verfällt nach Ablauf einer hinreichenden Zeitspanne automatisch.

Ein Token wird angelegt durch POST auf {session-url}/tokens. Auch hier ist das Geheimnis im Header *mainzelliste-ApiKey* anzugeben. Als Parameter wird ein JSON-Objekt der Form {type : *Typ*, data : {}} erwartet. Derzeit findet als *Typ* des Tokens nur *addPatient* Verwendung. In *data* werden weitere Parameter übermittelt; derzeit kann dort mit dem Namen *callback* eine Adresse angegeben werden, die aufgerufen wird, wenn mit dem Token eine Abfrage (d.h. Anlegen eines Patienten) durchgeführt wurde. Dies erlaubt es zum Beispiel, auf dem MDAT-Server den generierten PID automatisch mit anderen Daten zu verknüpfen (s.u.). Die Rückgabe des Aufrufs ist wieder eine URL im Header Location, und zwar im Format {session-url}/tokens/{tokenId}, sowie Token-ID und Pfad als JSON-Objekt. Für die weitere Verwendung des Tokens ist die *tokenId* aus dem Pfad zu ermitteln – entweder durch Extraktion aus dem Pfad oder aus dem zurückgegebenen Objekt (Attribut „tokenId“).

2.1 Callbackaufruf

Die Callbackadresse wird durch die Mainzelliste als POST-Request (Content-Type: application/json) mit den folgenden Parametern aufgerufen:

- *tokenId*: Token, mit dem der Patient angelegt wurde. Erlaubt die Zuordnung des gelieferten PIDs zu den Patientendaten auf dem MDAT.
- *id*: JSON-Objekt, welches den zurückgegebenen PID repräsentiert. Ein Array mit den Komponenten:
  - *idString* (String): Der PID.
  - *type* (String): Art der ID. Ist für zukünftige Erweiterungen (Verwalten multipler IDs) vorgesehen.
  - *tentative* (Boolean): Ein Wert „true“ gibt an, dass der PID vorläufig ist, also eventuell ein Duplikat eines vorher bestehenden Patienten darstellt (siehe 5)
- Encoding der Daten ist UTF-8

[1] alle Pfade relativ zur Basis-URL der Mainzelliste.

Wichtig: der Callback muss eine vollständige URL inklusive Protokoll sein (https://...)

2.2 Redirect

Über den Parameter *redirect* kann eine URL angegeben werden, auf die der Browser nach erfolgreichem Abfragen einer ID verwiesen wird (Status Code 303 + URL im Header *Location*). Damit kann zum Beispiel auf ein Formular zur Eingabe von medizinischen Daten auf dem MDAT-Server weitergeleitet werden.

2.3 ID-Typ

Der Parameter *idtypes* gibt in Form eines Arrays die ID-Typen an, die zurückgegeben werden sollen. Damit können spezifische IDs, zum Beispiel für Teilprojekte oder als Labor-IDs, erzeugt werden. Die Namen der verfügbaren ID-Typen und Details ihrer Generierung sind zwischen Auftraggeber und Betreiber der Mainzelliste abzustimmen.

2.4 Vordefinierte Felder

- Es ist möglich, Felder eines Patientendatensatzes (IDAT) serverseitig auszufüllen. Daten, für die das sinnvoll ist, sind zum Beispiel:Name des eintragenden Benutzers
- Klinik / Institution des eintragenden Benutzers
- Aufnahmedatum

Diese werden als Schlüssel-Wert-Paare in Form eines JSON-Objekts codiert und dieses als Element *fields* in das Objekt *data* eingefügt. Die Namen und Datentypen der vom MDAT-Server ausfüllbaren Felder sind zwischen Auftraggeber und Betreiber abzustimmen und werden in der Konfiguration angegeben.

3. Anlegen bzw. Abrufen eines PIDs

Im Folgenden werden das Anlegen eines neuen sowie das Abrufen eines existierenden PIDs einheitlich als „Anlegen eines Patienten“ bezeichnet. Zum Anlegen eines Patienten wird ein Token vom Typ *addPatient* benötigt, das auf die oben beschriebene Weise geholt wird (s.o.). Der Benutzer muss nun (über automatisch generierten Link, Button o.ä.) auf dem IDAT-Server das Formular /html/createPatient aufrufen; dabei muss die Id des Tokens als URL-Parameter „tokenId“ übergeben werden. Nach erfolgreichem Request wird die bei der Erstellung des Tokens angegebene Callbackadresse aufgerufen.

4. Beispiel-Sitzung

In diesem Beispiel lausche unter *https://mainzelliste.de/bsp/* eine Mainzelliste mit MDAT-Geheimnis *Sesam*. In Abbildung 1 ist die Sitzung als Sequenzdiagramm dargestellt.

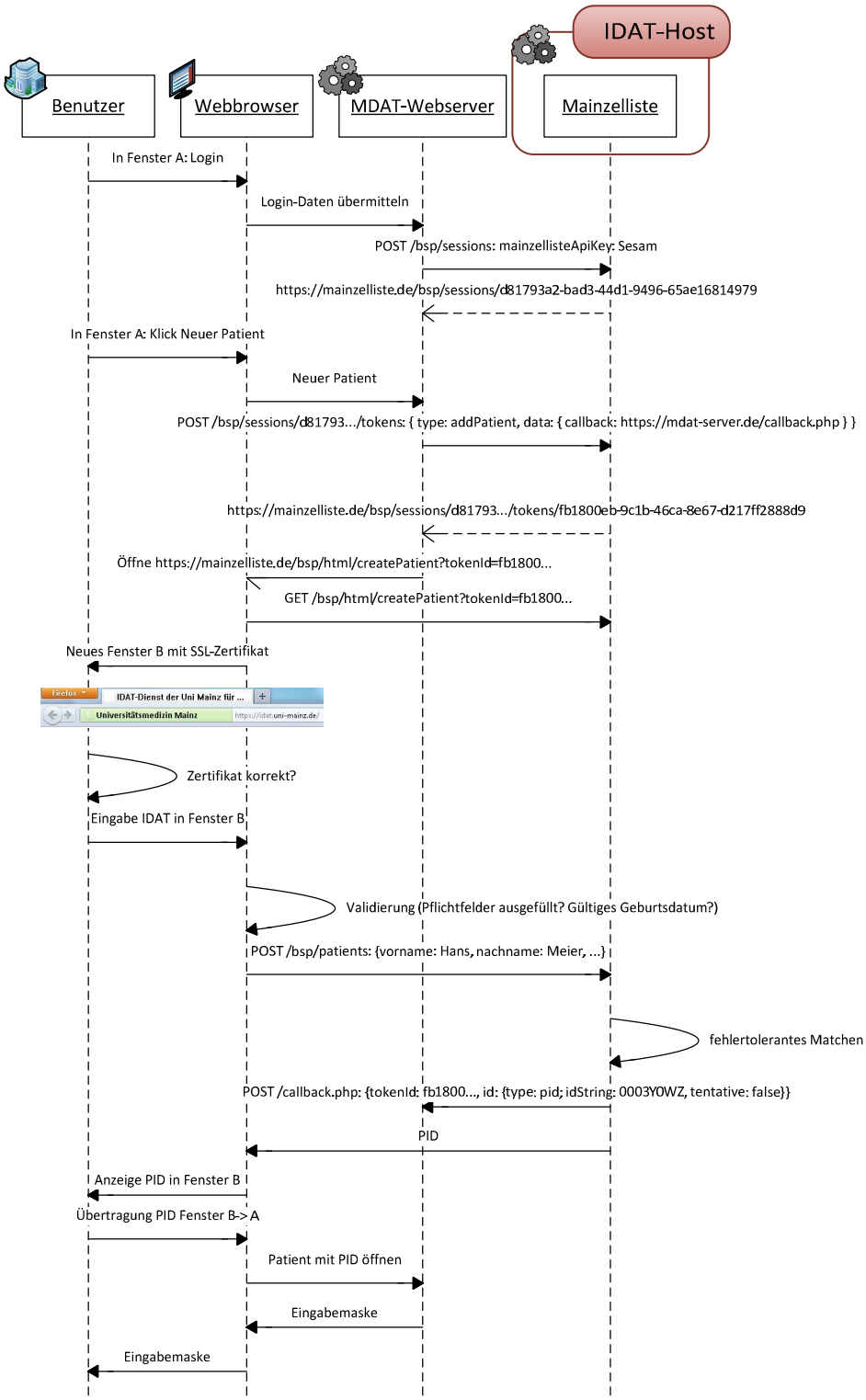

Abbildung 1 - Sequenzdiagramm für die hier skizzierte Beispielsitzung.

4.1 Session anlegen

- Aufruf: POST https://mainzelliste.de/bsp/sessions
- Anfrage-Header:
  - `mainzellisteApiKey: Sesam`
- Antwort:

```
Status Code: 201 Created
Content-Type: application/json
Location: https://mainzelliste.de/bsp/sessions/d81793a2-bad3-44d1-9496-
65ae16814979
Date: Tue, 19 Jun 2012 14:42:27 GMT
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=34219BDC1792AD310F0F5E2AF4F13D35; Path=/bsp; HttpOnly
Transfer-Encoding: chunked

{"sessionId":"d81793a2-bad3-44d1-9496-
65ae16814979","uri":"https:\/\/mainzelliste.de\/bsp\/sessions\/d81793a2-bad3-
44d1-9496-65ae16814979"}
```

4.2 Token anfordern

Es sei https://mdat-server.de/callback.php die gewünschte Callbackadresse.

- Aufruf: POST https://mainzelliste.de/bsp/sessions/d81793a2-bad3-44d1-9496-65ae16814979/tokens
- Anfrage-Header:
  - `Content-Type: application/json`
  - `mainzellisteApiKey: Sesam`
- Parameter (als JSON-Array)
  - `type: addPatient`
  - `data: [Array]`
    - `callback: { https://mdat-server.de/callback.php }`

  Anfrage-Body im Klartext also:

  ```
  {"type":"addPatient","data":{"callback":"https://mdat-
  server.de/callback.php"}}
  ```
- Antwort:

```
Status Code: 201 Created
Content-Type: application/json
Location: https://mainzelliste.de/bsp/sessions/d81793a2-bad3-44d1-9496-
65ae16814979/tokens/fb1800eb-9c1b-46ca-8e67-d217ff2888d9
Date: Tue, 19 Jun 2012 14:53:39 GMT
Server: Apache-Coyote/1.1
Transfer-Encoding: chunked

{"tokenId":"fb1800eb-9c1b-46ca-8e67-d217ff2888d9","uri":"https:
:\/\/mainzelliste.de\/bsp\/sessions\/d81793a2-bad3-44d1-9496-
65ae16814979\/tokens\/fb1800eb-9c1b-46ca-8e67-d217ff2888d9"}
```

4.3 Patient anlegen

Dazu wird der Nutzer veranlasst folgenden Link zu öffnen:

https://mainzelliste.de/bsp/html/createPatient?tokenId=fb1800eb-9c1b-46ca-8e67-d217ff2888d9

Bei erfolgreichem Anlegen bzw. Ausgeben eines PID (hier *000GR0W0*) erfolgt dann folgender Callback:

- POST https://mdat-server.de/callback.php
- Header:
  - `Content-Type: application/x-www-form-urlencoded`
- Parameter (als JSONJSON-Array):
  - `tokenId: fb1800eb-9c1b-46ca-8e67-d217ff2888d9`

- id: `{"idString":"000GR0W0","type":"pid","tentative":true}`

Anfrage-Body im Klartext also:

```
{"tokenId":"fb1800eb-9c1b-46ca-8e67-d217ff2888d9",
"id":{"idString":"000GR0W0","type":"pid","tentative":true}}
```

5. Behandlung von unsicheren Fällen

Wenn nicht zweifelsfrei festgestellt werden kann, ob die eingegebenen Daten zu einem vorhandenen Patienten gehören oder einen neuen Eintrag darstellen, wird der Benutzer darüber benachrichtigt und erhält die Möglichkeit, die eingegebenen Daten zu korrigieren oder zu bestätigen. Falls die Daten korrigiert werden, erfolgt eine neue Anfrage. Falls die Daten bestätigt werden, wird ein neuer PID generiert und ausgeliefert. Für diesen PID wird das Attribut *tentative* auf *true* gesetzt (vgl. Rückgabe des PID als JSON-Objekt weiter oben). Mit diesem vorläufigen PID kann ein neuer Patient angelegt und sofort „genutzt“ werden. Es bleibt aber die Möglichkeit, dass dieser neu angelegte Patient später mit einem existierenden Patienten zusammengeführt wird. In diesem Fall ist es – je nach Implementierung auf MDAT-Seite – möglich, dass der vorläufige PID nach der Zusammenführung ungültig wird.