Authorization guide for the sandbox environment

Kanta Personal Health Record

10.5.2023

Kela

Kela
Kanta-palvelut
PL 450, 00056 Kela

FPA
Kanta tjänsterma
PB 450, 00056 FPA

Kanta 2020
www.kanta.fi
kanta@kanta.fi

Change History

Version	Change	Author	Date
1.0	First version	Kela Kanta services	08.05.2017
1.1	Updated to match the current rules, clarified the usage of test SSNs and updated the new addresses of the sandbox-servers	Kela Kanta services	17.04.2018
1.2	Technical scopes section updated. Terminology harmonized.	Kela Kanta services	23.10.2018
1.2	Updated state-parameter to token endpoint, updated request examples and urls, added maintenance only -scopes	Kela Kanta services	5.4.2019
1.3	Updated using of state-parameter to token endpoint.	Kela Kanta services	24.9.2019
1.4	Edited document to meet AA-level of WCAG guidelines	Kela Kanta services	10.7.2020
1.5	Changed terms of use to information notice concerning the use of the Kanta PHR	Kela Kanta serivices	1.11.2021
1.6	Added optional language parameter to authorization request.	Kela Kanta serivices	12.11.2021
1.7	Added link to new Kanta PHR Sandbox Management Service.	Kela Kanta services	23.2.2023
1.8	Refined value of redirect_uri and added information about usage of test social security number. Added more examples.	Kela Kanta services	11.4.2023
1.9	Added chapter 1.1	Kela Kanta Services	12.4.2023
2.0	Error corrections	Kela Kanta Services	10.5.2023

Sisällysluettelo

1 Introduction

This document describes OAuth 2.0 profile for My Kanta Personal Health Record (Finnish Kanta PHR) sandbox environment with authorization. You can find more information about Finnish Kanta PHR sandbox environments from kanta.fi-pages.

Kentän koodi muuttunut

This profile will evolve during the project and therefore isn't same as the profile that will be used in the customer test or production environments of Kanta PHR.

All of the endpoints are protected by TLS 1.2. Client certificate isn't required in the sandbox environment.

The sandbox environment is open environment for testing and development. The storage of any data in the sandbox is not guaranteed and data stored there may be lost. As the environment is open and public use of any official identifiers is forbidden. To identify your test person use a test-SSN. The Finnish SSN system works in a way that the individual identifiers in the range of 900-999 are reserved for unofficial testing purposes.

You must not use real social security numbers or names in this test environment!

1.1 Kanta PHR Sandbox Management Service

Kommentoinut [AA1]: Lisätään tähän mitä kälissä pitää tehdä ennen kuin voi tehdä mitään

Before authorization and token requests, the user must create a test client in Kanta PHR Sandbox Management Service. You can log in into the Finnish PHR Sandbox Management Service using this link. All of the clients using sandbox environment are authenticated with http basic authentication.

Kentän koodi muuttunut

1.2 Client registration

The process of using the sandbox environment typically starts with client registration. During registration, the user provides essential information about the client, including its name, redirect URI, and necessary scopes. The user also selects the desired authentication method for the client. Based on the user's choices, the system generates the information necessary for authorization and token requests.

Once the registration process is complete, the client must use the selected authentication method for authorization and token requests. If the client wishes to switch to a different authentication method, the user must update the client settings from the user interface to ensure that the correct method is used.

It's worth noting that the authorization request should only include scopes that have been registered for the client.

2 Authorization flow

The Oauth flow to be used in the sandbox is the Authorization code flow. In the flow first step is request authorization of the user. The environment has a simple demo login page and then with the given code access token requests can be made.

2.1 Authorization code flow

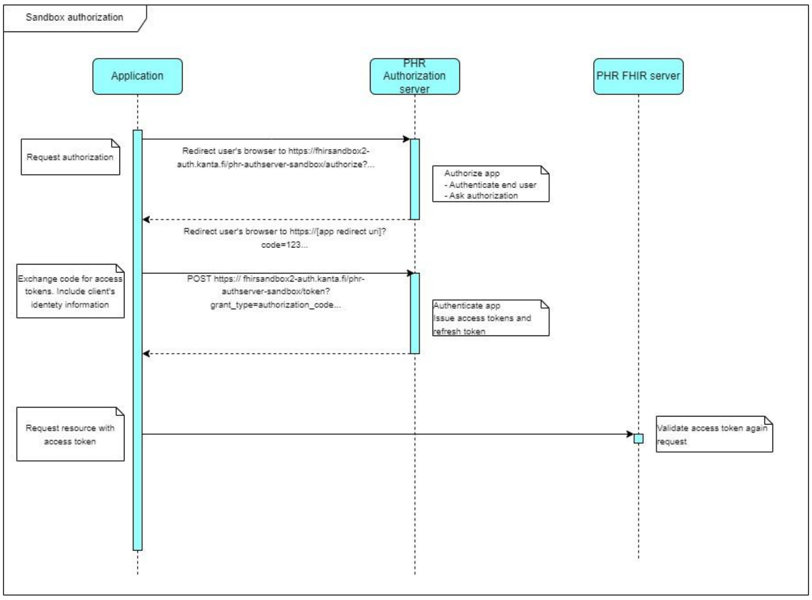

Sandbox authorization
Application
PHR Authorization server
PHR FHIR server
Request authorization
Redirect user's browser to https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/authorize?...
Authorize app
- Authenticate end user
- Ask authorization
Redirect user's browser to https://[app redirect uri]?code=123...
Exchange code for access tokens. Include client's identety information
POST https:// fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/token?grant_type=authorization_code...
Authenticate app
Issue access tokens and refresh token
Request resource with access token
Validate access token again request

Figure 1 PHR Sandbox Authorization flow

The authorization endpoint is used when the client needs authorization from the user to access resources. This may be the first time the client is used or if the client has not been granted a scope that it needs to access a resource. First the client requests and receives a short-lived authorization code which will be then traded for a set of tokens at the token endpoint. Before issuing the code, the authorization server authenticates the user and requests the user to grant access to their PHR account.

3 Authorization server endpoints

3.1 Authorization endpoint

The authorization endpoint is called when the client needs authorization from the user to access resources. This may be the first time the client is used or if the client hasn't been granted a scope that it needs to access a resource.

The endpoint is called over TLS at the URL:
https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/authorize.

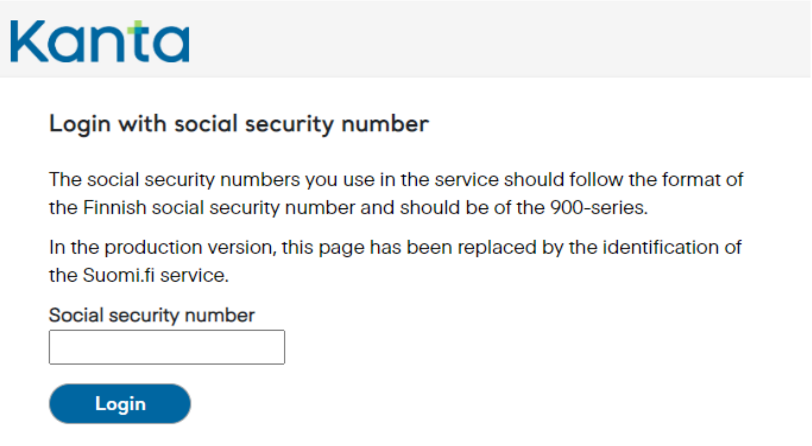
The GET parameters for clients following the authorization code flow are:

Parameter	Value	Required
response_type	Constant: "code"	required
client id	The id client application has been given at registration, for example "Client123"	required
scope	The scopes the client wants to be granted. Scopes are defined in section 4.	required
redirect_uri	The url registered for the client at the registration time	required localhost is not allowed but for example 127.0.0.1 is
state	An opaque value used by the client to maintain state between the request and callback. The authorization server includes this	required

	value when redirecting the user-agent back to the client. The parameter MUST be used for preventing cross-site request forgery or session fixation attacks.	
lg	optional	The language parameter provided by the application redirects the user to identify and authorize. Accepted languages are ‘fi’/ ‘fi-FI’/’fi-SE’ , ‘sv’/’sv-SE’/’sv-FI’ or ‘en’/’en-GB’. If the parameter is unknown, the user is redirected to the English authorization page. If the parameter is missing from the request, the user is redirected to the Finnish authorization page.

The app MUST use an unpredictable value for the state parameter with at least 128 bits of entropy. The app MUST validate the value of the state parameter upon return to the redirect URL and MUST ensure that the state value is securely tied to the user’s current session (e.g., by relating the state value to a session identifier issued by the app). The app SHOULD limit the grants, scope, and period of time requested to the minimum necessary.

Example call to the authorize endpoint is https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/authorize?response_type=code&client_id=982585da-08db-48ea-b535-d1ea6da2a5cf&state=state_83ae3469-8bd4-42ac-ba25-8507ed88dd5f&scope=patient/Observation.write+patient/Observation.read&redirect_uri=https://127.0.0.1&lg=fi



Kanta

Login with social security number

The social security numbers you use in the service should follow the format of the Finnish social security number and should be of the 900-series.

In the production version, this page has been replaced by the identification of the Suomi.fi service.

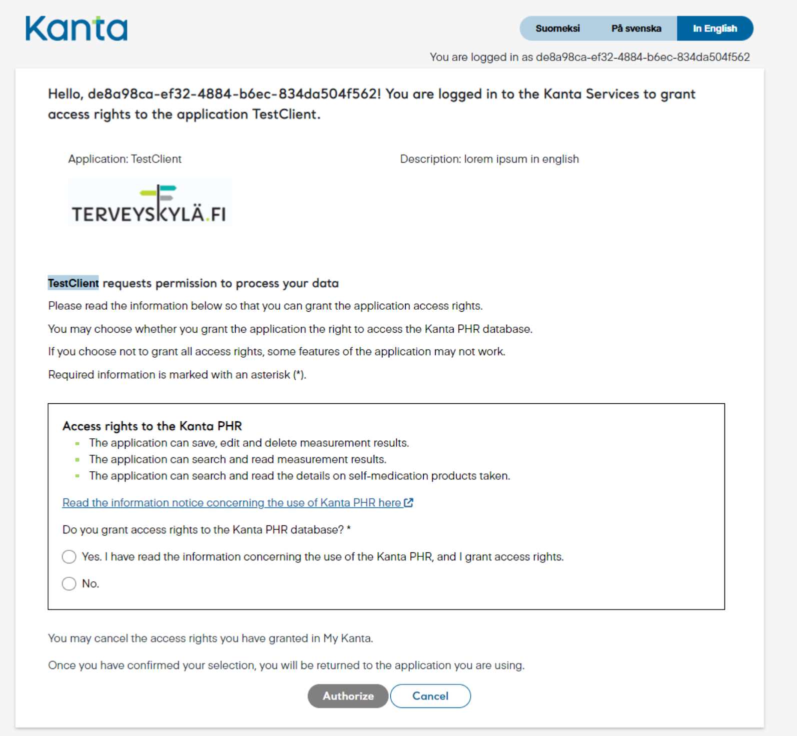
Social security number

Figure 2 PHR Sandbox Authorization UI Login page

In the sandbox environment user authentication is replaced by mock service that asks user's test social security number (SSN). The 900-series SSN's are meant for test use, and they meet the requirements for the format of the SSN. Read more about SSN (dvv.fi).It is strictly forbidden to use other than the 900-series SSN's in the PHR Sandbox Authorization service. It is possible to create a test SNN using the tools intended for it, for example on the Lintukoto.net website. Read more about the website (lintukoto.net). The PHR Sandbox Authorization login page is shown in figure 2.

Kentän koodi muuttunut

Kentän koodi muuttunut



You are logged in as de8a98ca-ef32-4884-b6ec-834da504f562

Hello, de8a98ca-ef32-4884-b6ec-834da504f562! You are logged in to the Kanta Services to grant access rights to the application TestClient.

Application: TestClient

Description: lorem ipsum in english

TestClient requests permission to process your data

Please read the information below so that you can grant the application access rights.

You may choose whether you grant the application the right to access the Kanta PHR database.

If you choose not to grant all access rights, some features of the application may not work.

Required information is marked with an asterisk (*).

Access rights to the Kanta PHR

- The application can save, edit and delete measurement results.
- The application can search and read measurement results.
- The application can search and read the details on self-medication products taken.

Read the information notice concerning the use of Kanta PHR here

Do you grant access rights to the Kanta PHR database? *

○ Yes. I have read the information concerning the use of the Kanta PHR, and I grant access rights.

○ No.

You may cancel the access rights you have granted in My Kanta.

Once you have confirmed your selection, you will be returned to the application you are using.

Figure 3 PHR Application Approval page

After logging in the application approval page will be shown. This is shown in the figure 3. The page includes information about the application and the scopes that the application is requesting.

After the user is authenticated PHR authorization server will create a pseudonym for the user which is used as patient id in PHR. After successful authorization user's browser will be redirected to the preregistered redirect URI with authorization code as a get parameter, for example:
https://app/after-auth?code=z24EGT&state=b5de575e-ac1b-4c00-b1f1-844d1b9bdb03

When the authorization is cancelled, the user will be redirected to the address specified in the client registration. The URI of this address will include an error message and a description. For example, if the authorization was denied by the user, the URI may look like this:

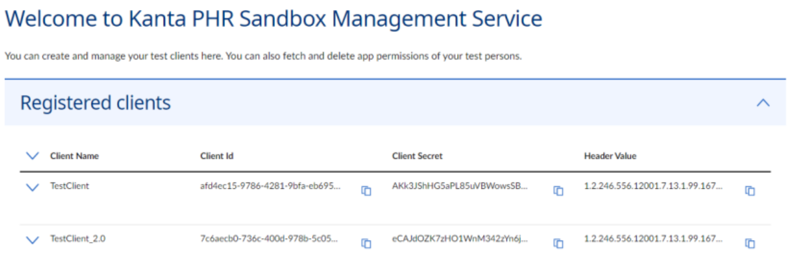
https://app/after_auth?error=access_denied&error_description=User%20denied%20access&state=b5de575e-ac1b-4c00-b1f1-844d1b9bdb03.

3.2 Token endpoint

After the client receives an authorization code from the authorization endpoint the client presents the authorization code along with its own credentials to the authorization server's token endpoint to obtain an access token.

Another reason to call the token endpoint is that the original access token has expired. In this scenario, the client application calls the resource server with a refresh token that was obtained along with an access token.

The token endpoint is called over TLS at the URL:
https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/token



Welcome to Kanta PHR Sandbox Management Service

You can create and manage your test clients here. You can also fetch and delete app permissions of your test persons.

Registered clients

Client Name	Client Id	Client Secret	Header Value
TestClient	afd4ec15-9786-4281-9bfa-eb695...	AKk3JShHG5aPL85uVBWowsSB...	1.2.246.556.12001.7.13.1.99.167...
TestClient_2.0	7c6aecb0-736c-400d-978b-5c05...	eCAJdOZK7zHO1WnM342zYn6j...	1.2.246.556.12001.7.13.1.99.167...

Figure 4 Client informations are presented in Kanta PHR Sandbox Management Service where the user can register clients for test use.

Kanta PHR Sandbox Management service utilizes two different authentication methods: basic authentication and a solution that simulates SSL authentication. Basic authentication uses client Id and client secret. A solution imitating SSL authentication uses a special header. These authentication details can be accessed through the user interface and are generated for each registered application. Figure 4 shows two registered clients and their information.

3.2.1 Authetication with basic auth

Token request examples are presented in the next two chapters. For basic authorization, the HTTP header Authorization is used, which contains the client ID and client secret in a base64-encoded format. This information can be found in the Kanta PHR Sandbox Management Service after the user has registered the client.

3.2.2 Simulating SLL authentication

Instead of relying on an SSL certificate, the sandbox environment uses a special authentication header to verify the identity of the client. This header is called PhrDemoAuth and it is mandatory for the client to include the 'Header value' provided in the user interface. It is important to note that this header is only used in the sandbox environment; in the actual production environment of PHR, a different header is used.

3.2.3 Basic authorization example

```
POST https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/token
HTTP/1.1 Content-Type: application/x-www-form-urlencoded
Authorization: dGVzdF9jbGllbnQ6cXdlcnR5MTIzNDU2
grant_type=authorization_code&
code=SJ2DlI&
redirect_uri=https%3A%2F%2F127.0.0.1&
state=sdgfoewew2335twes&
client_id=test_client
```

3.2.4 PhrDemoAuth authorization example

```
POST https://fhirsandbox2-auth.te.kanta.fi/phr-authserver-sandbox/token
HTTP/1.1 Accept-Encoding: gzip,deflate
Content-Type: application/x-www-form-urlencoded
PhrDemoAuth: 1.2.246.556.12003.7.12.3
Host: fhirsandbox2-auth.te.kanta.fi
Connection: Keep-Alive
grant_type=authorization_code&
code=1KbPP5yTjcxUrEBaPWDjYZTMYc3oDr6jJRedlKE7R4_ORBYwIuDzTNWhmh
7Y8vMwoKYE6QUHhWEmqbvbbuJ0wFXYyMhrsoOpYK81wqCc3OS-
U3blc8QwOBTWK6591coU&
redirect_uri=https://127.0.0.1&
client_id=test_client2
```

The client MUST use the HTTP "POST" method when making access token requests.

Parameter	Value	Required
Authorization	HTTP Basic Authentication. HTTP header Authorization containing client id and client secret MUST be in base64-encoded format separated by a colon, for example “Authorization Client123:ClientSecret”	required
grant_type	Fixed value: “authorization_code” if parameter “code” is used, or fixed value “refresh_token” if parameter “refresh_token” is used.	required
code	Value of code, for example “z24EGT”, as obtained from the response when calling authorization endpoint. Only in use if grant_type=authorization_code.	required if grant_type=authorization_code
refresh_token	Value of refresh token as obtained from a former call to the token endpoint.	required if grant_type=refresh_token
redirect_uri	One of the registered redirect urls for the client at the registration time	required localhost is not allowed but for example 127.0.0.1 is

Parameter	Value	Required
client_id	he id client application has been given at registration, for example “Client123”	required
http-header “Accept: application/json”		optional
state	An opaque value used by the client to maintain state between the request and callback. If state-parameter is provided with the request, PHR authorization server will return the exact value to the client.	optional

The authorization server will return a JSON structure that includes an access token or a message indicating that the authorization request has been denied. The JSON structure includes the following parameters:

Parameter	Value
access_token	The access token issued by the authorization server
token_type	Fixed value: “Bearer”
refresh_token	Token that can be used to obtain a new access token, using the same or a subset of the original requested scopes. A refresh token is generated only if it has been requested in the authorization endpoint call by including the scope “offline”.
expires_in	Lifetime in seconds of the access token, after which the token is not accepted by the resource server.
scope	Scopes that the client has been granted. Note that this can be different from the scopes requested by the client.
sub	Patient pseudonym which MUST be used in all search requests.
state	If the token endpoint call included a state-parameter its value will be returned also here. Can be used to check that the token endpoint response has not been changed by a hostile third party.

3.2.5 An example response in token request workflow:

```
{
"access_token":"eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiI2YTliNTk4ZC0yZ
mU0LTRhNWQtYjE3Yi05N2U4ZDUzOTYwZmIiLCJhenAiOiJUSU1JU1NMIiwiaXNzIjoiaHR0
cHM6XC9cL2ZoaXJzYW5kYm94Mi1hdXRoLmthbnRhLmZpXC9waHItYXV0aHNlcnZlci1zY
W5kYm94XC8iLCJleHAiOjE1MjMzNjA5MzMsImlhdCI6MTUyMzM1NzMzMywianRpIjoiZGZj
MmViZDItNWNmNi00ZWFjLWFjNWYtM2JiNGJlMjBlNjdjIn0.mJWRJDWdaS6_1F_AqCR32X
vnRrXYrVwvfcHtxsCVxsHbbCXtBL4pfUq5uhdvTGifjHMKNPeJUdTWYZoYXYdeF7GVJd37w
zW2kjcnKuN5Ijo9Rvkxf_ofaJ8-
JrPka1mznUEwGWx4kM3Un_o2LjAEQH3B49c6OXtL8HLK9v3wPnY",
"token_type":"Bearer",
"refresh_token":"eyJhbGciOiJub25lIn0.eyJqdGkiOiI1ODQ4Nzc5Ny05YjZiLTRjMGEtOWIwZS
1jNzA2YTE0NTFlODQifQ.",
"expires_in":3599,
"scope":"patient/Patient.read patient/QuestionnaireResponse.read offline_access
patient/Observation.read patient/Observation.write",
"sub":"6a9b598d-2fe4-4a5d-b17b-97e8d53960fb",
“state”:”sdgfoewew2335twes”
}
```

4 Supported scopes in the authenticating sandbox

Scopes supported by Kanta PHR can be divided into the scopes that grant access to specific FHIR resources stored on the resource server and scopes that allow applications to obtain other information and keep the authorization active.

4.1 User scopes for data access

Scopes than can be granted to access resources on the resource server are defined similarly to SMART on FHIR scopes. Requesting empty scope list grants all scopes registered to the client. Scopes are in form of patient/:resourceType.(read|write)

Scope	Contents
patient/Observation.read	Reading patient observations, like heart rate
patient/Observation.write	Creating, updating and deleting observation
patient/Patient.read	Reading the patient-resource
patient/MedicationStatement.write	Creating, updating and deleting the medication statement-resource
patient/MedicationStatement.read	Reading the medication statement-resource
patient/MedicationAdministration.read	Reading the medication administration-resource
patient/MedicationAdministration.write	Creating, updating and deleting medication administration
patient/QuestionnaireResponse.read	Reading the questionnaire response-resource
patient/QuestionnaireResponse.write	Creating, updating and deleting questionnaire response
patient/CarePlan.read	Reading the care plan-resource
patient/CarePlan.write	Creating, updating and deleting care plan

Kommentoinut [AA2]: write puuttuu

To read a resource you need to have the patient/Resource.read scope. For writing, updating and deleting the resource patient/Resource.write scope is needed. A scope is needed only for the main resource type, containing resources that are inline in the resource to be read or written follow the scope of the resource that they are part of. Referenced resources are subject to the scope of their respective type.

All requested scopes that can be authorized by the user needs to be chosen for the client application when the client application is registered. You can request authorization only for scopes chosen for the client application. All scopes that are included in the access token need to be authorized by the user – the user may choose only to accept all scopes or not accept at all.

4.2 Non user-specific scopes for admininstration

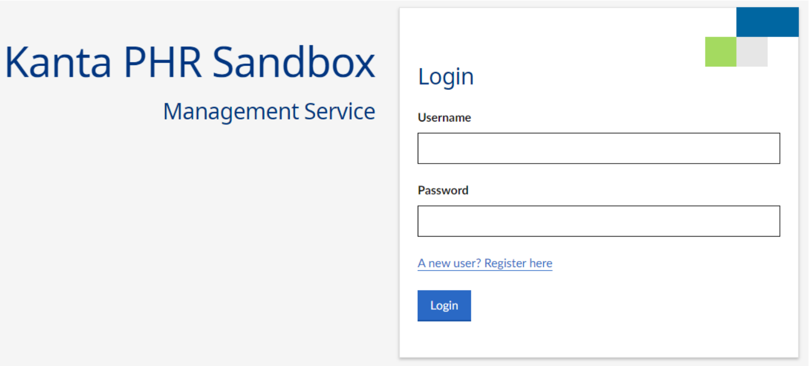
Scope	Contents
ConformanceStatement.read	Reading the conformance statement
StructureDefinition.read	Reading different structure definitions
StructureDefinition.write	Creating, updating and deleting structure definitions
ValueSet.read	Reading different value sets
ValueSet.write	Creating, updating and deleting different value sets
CodeSystem.read	Reading different code systems
CodeSystem.write	Creating, updating and deleting different code systems
Questionnaire.read	Reading the questionnaire-resource
Questionnaire.write	Creating, updating and deleting questionnaire

These scopes are intended for internal PHR maintenance use only.

5 Removing authorization

You can log in into the Finnish PHR Sandbox Management Service using this link.

The login requires using username and password. A new user must register on the service and after that the user can log in. The login page is shown in figure 4.



Kanta PHR Sandbox

Management Service

Login

Username

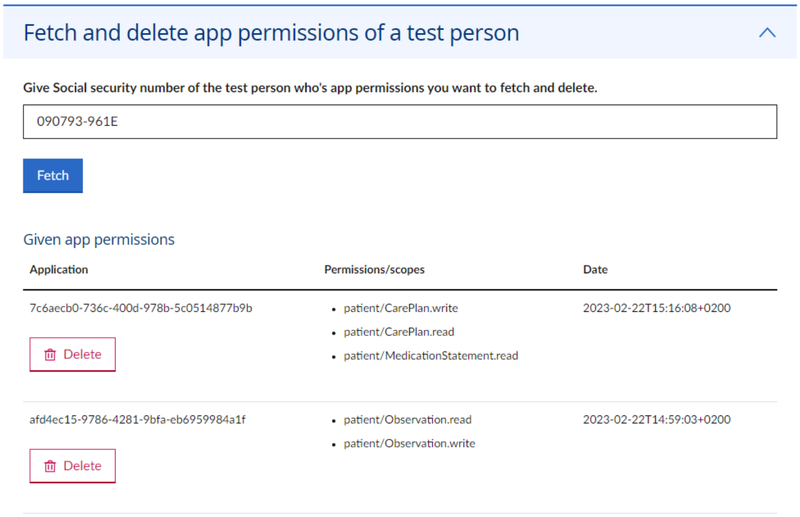
Password

A new user? Register here

Login

Figure 5 PHR Sandbox Management Service Login page

In the PHR Sandbox Management Service users can create new test clients, edit previously created test clients or manage application permissions of a the test user. The welcome page shows registered clients for the user. This can be seen in figure 5.



Fetch and delete app permissions of a test person

Give Social security number of the test person who's app permissions you want to fetch and delete.

090793-961E

Given app permissions

Application	Permissions/scopes	Date
7c6aecb0-736c-400d-978b-5c0514877b9b Delete	patient/CarePlan.write patient/CarePlan.read patient/MedicationStatement.read	2023-02-22T15:16:08+0200
afd4ec15-9786-4281-9bfa-eb6959984a1f Delete	patient/Observation.read patient/Observation.write	2023-02-22T14:59:03+0200

Figure 6 PHR Sandbox Management Service app permissions page

In the Fetch and delete app permissions of a test persons section the user can manage permits for a test user. After the authorizations have been fetched, the user can remove authorizations one by one. This can be seen in figure 7.

6 Request to FHIR server

With a valid access token, the app can access PHR data by issuing a FHIR API call to the FHIR endpoint on the resource server. The request includes an **Authorization** header that presents the **access_token** as a “Bearer” token. If the client is web-client the request to FHIR-server is sent as shown below:

6.1 Example with basic authentication

```
POST https://fhirsandbox2.kanta.fi/phr-resourceserver/baseStu3/Observation
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/json
Authorization: Bearer
eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiI2YTliNTk4ZC0yZmU0LTRh
NWQtYjE3Yi05N2U4ZDUzOTYwZmIiLCJhenAiOiJUSU1JU1NMIiwiaXNzIjoiaHR0c
HM6XC9cL2ZoaXJzYW5kYm94Mi1hdXRoLmthbnRhLmZpXC9waHItYXV0aHNlcnZl
ci1zYW5kYm94XC8iLCJleHAiOjE1MjMzNjA5MzMsImlhdCI6MTUyMzM1NzMzMywi
anRpIjoiZGZjMmViZDItNWNmNi00ZWFjLWFjNWYtM2JiNGJlMjBlNjdjIn0.mJWRJD
WdaS6_1F_AqCR32XvnRrXYrVwvfcHtxsCVxsHbbCXtBL4pfUq5uhdvTGifjHMKNPe
JUdTWYZoYXYdeF7GVJd37wzW2kjcnKuN5Ijo9Rvkxf_ofaJ8-
JrPka1mznUEwGWx4kM3Un_o2LjAEQH3B49c6OXtL8HLK9v3wPnY

Content-Length: 1494
Host: fhirsandbox2.kanta.fi
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
{
 "resourceType":"Observation",
 "meta":{
    "profile":[
        http://phr.kanta.fi/StructureDefinition/fiphr-bodyheight-stu3
 ],
 "language": "fi",
 …
 }
```

With the complete resource in the body.

6.2 Example with PhrDemoAuth authentication:

```
POST <https://fhirsandbox2.te.kanta.fi/phr-resourceserver/baseStu3/Observation
HTTP/1.1Accept-Encoding>: gzip,deflate
Content-Type: application/json

Authorization: Bearer
eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiIyYzk3N2ZlYS0zNWQ4LTR
jNDQtOWU2Yy0yZDU3NmNiMzA3YjQiLCJhdWQiOiJUSU1JU1NMIiwibmJmIjoxNjg
xMjE3OTk3LCJzY29wZSI6WyJwYXRpZW50L1F1ZXN0aW9ubmFpcmVSZXNwb25
zZS5yZWFkIiwib3BlbmlkIiwib2ZmbGluZV9hY2Nlc3MiLCJwYXRpZW50L09ic2Vydm
F0aW9uLnJlYWQiLCJwYXRpZW50L09ic2VydmF0aW9uLndyaXRlIl0sImlzcyI6Imh0
dHBzOi8vZmhpcnNhbmRib3gyLWF1dGgudGUua2FudGEuZmkvcGhyLWF1dGhzZ
XJ2ZXItc2FuZGJveC8iLCJleHAiOjE2ODEyMjE1OTcsImlhdCI6MTY4MTIxNzk5Nywi
anRpIjoiOGIzYzgzMGMtZGFmZS00MmI1LTk1ZDItMjJlYTRhNjQ0MDRlIn0.YYI0dG
AQoSTAbFVb4GNiXI3uw2Or2TmD3qRym9Mur0Z605ArFmWovkTbSe8YWw1BMB
PQvLFh27PPBNpmABwzPUStP4AZHj4vWZVoiORRvk_iFow7-
qygwKKGU5oOxLbnBlM__ztewYtPByTcQgvklJzmq7UciFh4F1ZrkmERNCezP8JtjD
cpuGyiszSFwWR0ebUoZgpwiP2mBLDE12HTRw49Q-
n8qWh3WgxueBKl2hh_kTbgAdsk5ece8EWs4yjZngZT-
x7wEtO5OfqoV2WMtIyTYnaws5YY2dWUOqjUEWNF3A4a23-
tI83l6niqhcV0bzc6sDtFxMYEJ29bklYpJw

PhrDemoAuth: 1.2.246.556.12003.7.12.3
Content-Length: 1496
Host: fhirsandbox2.te.kanta.fi
Connection: Keep-Alive
User-Agent: Apache-Http
Client/4.1.1 (java 1.5)

{
 "resourceType":"Observation",
 "meta":{
    "profile":[
       http://phr.kanta.fi/StructureDefinition/fiphr-bodyheight-stu3
 ],
 "language": "fi",
…
}
```

With the complete resource in the body.