

# Authorization guide for the sandbox environment

My Kanta Pages PHR

5.4.2019 Kela



## 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



## Contents

1	Introduction .....	4
2	Authorization flow .....	4
2.1	Authorization code flow .....	4
3	Authorization server endpoints .....	5
3.1	Authorization endpoint .....	5
3.2	Token endpoint .....	6
4	Supported scopes in the authenticating sandbox .....	8
4.1	User scopes for data access .....	8
4.2	Non user-specific scopes for administration purposes .....	9
4.3	Technical scopes .....	10
5	Removing authorization .....	10
6	Request to FHIR server .....	11



# 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: <https://www.kanta.fi/en/system-developers/sandbox-environments>.

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 clients using sandbox environment are authenticated with http basic authentication.

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

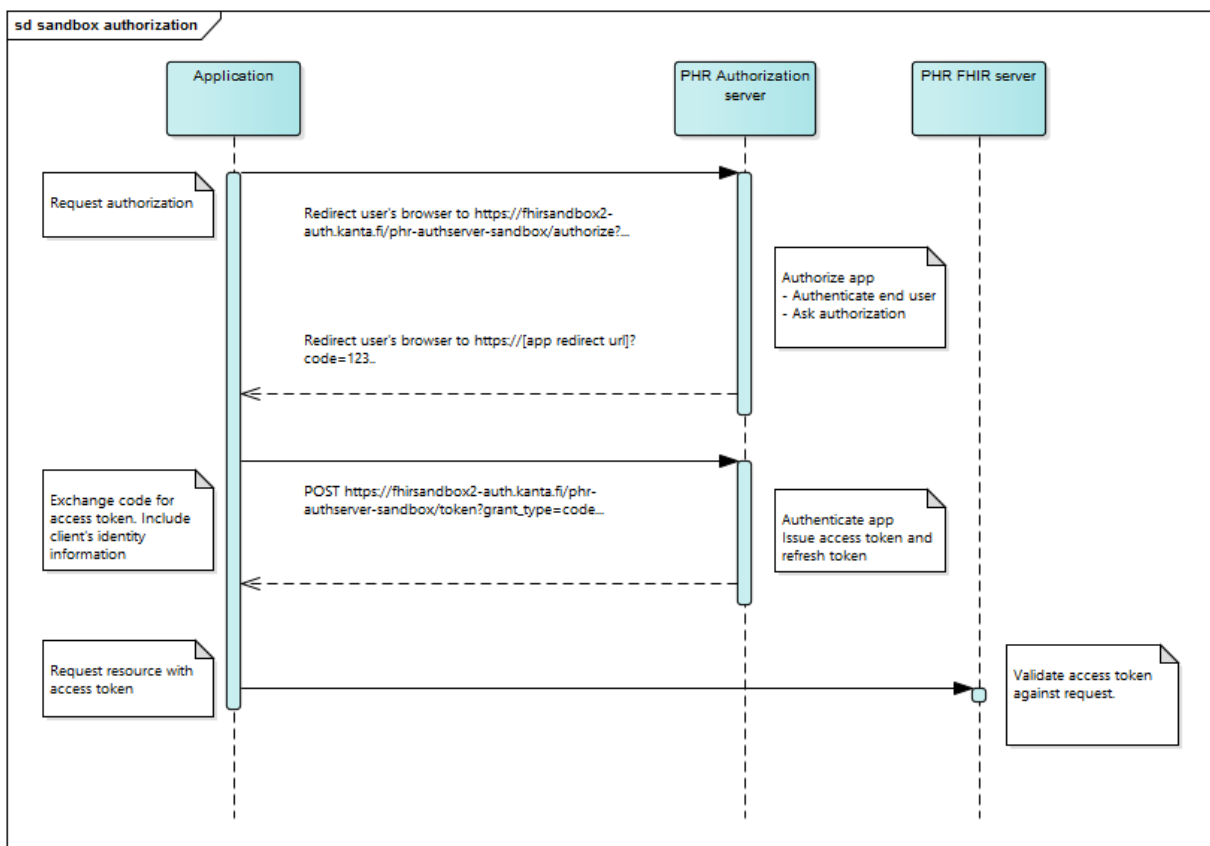
The sandbox environment is open environment for testing and development. The storage of any data in the sandbox is not guaranteed and data stored there may be lost. As the environment is open and public use of any official identifiers is forbidden. To identify your test person use a test-SSN. The Finnish SSN1 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!**

## 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



<sup>1</sup> <http://vrk.fi/en/personal-identity-code1>  
Kela, Kanta-palvelut, PL 450, 00056 Kela

## 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
state	An opaque value used by the client to maintain state between the request and callback. The authorization server includes this 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.	required

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=Client123&redirect\\_uri=http%3A%2F%2Flocalhost&state=b5de575e-ac1b-4c00-b1f1-844d1b9bdb03](https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/authorize?response_type=code&client_id=Client123&redirect_uri=http%3A%2F%2Flocalhost&state=b5de575e-ac1b-4c00-b1f1-844d1b9bdb03)

In the sandbox environment user authentication is replaced by mock service that asks user's social security number and name. The first time the user authenticates both name and social security number must be filled in. Next time only social security number is required.

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

## Kirjaudu käyttäjätunnuksella

After logging in the application approval page will be shown if a new set of scopes or a new application is asking for the approval.

Sovellus **Matin Testiclientti** pyytää käyttöoikeuksia Omakannan Omatietovarantoon ja sinne talletettuihin tietoihisi.

#### Lisätietoja sovelluksesta:

- Ylläpitäjä:  
[matti@testiclient](mailto:matti@testiclient)

#### Käyttöoikeudet, joita sovellus pyytää:

- mittaustulosten tietojen haku ja luku

Kun annat sovellukselle mainitut käyttöoikeudet, hyväksyt samalla myös **Omakannan Omatietovarannon käyttöehdot**.

**HUOM!** Jos et hyväksy Omakannan Omatietovarannon käyttöehtoja, et voi antaa sovellukselle sen pyytämiä käyttöoikeuksia. Jos haluat poistaa aiemmin hyväksymiäsi muiden sovellusten käyttöoikeuksia tai aiemmin Omatietovarantoon tallennettuja tietojasi, sinun tulee tehdä se itse **Omakanta-palvelussa**. Ennen sovelluksen käyttöoikeuksien poistamista voit poistaa tallennettuja tietojasi myös sovelluksen kautta, mikäli sovellus tukee tietojen poistamista Omatietovarannosta.

Hyväksytkö Omakannan Omatietovarannon käyttöehdot sekä yllä mainitut käyttöoikeudet sovellukselle **Matin Testiclientti**?

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>

After the user is authenticated PHR authorization server will create a pseudonym for the user which is used as patient id in PHR.

### 3.2 Token endpoint

After 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 access token.

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

Example call would be as follows:

```
POST https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/token
HTTP/1.1 Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code&
code=SJ2D1I&
redirect_uri=http%3A%2F%2Flocalhost&
state=sdgfoewew2335twes&
client_id=clientid123
```

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

Parameter	Value	Required
Authorization	HTTP Basic Authentication. HTTP field Authorization containing client id and client secret in non-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=code.	required if grant_type=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
client_id	The 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. The client <b>MUST</b> validate the state value for any request sent that the value in response matches the value sent in the request.	required







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.read	Reading the medication statement-resource
patient/MedicationStatement.write	Creating, updating and deleting medication statement
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

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, contained 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 of 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 needs 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 administration purposes

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.

### 4.3 Technical scopes

There are in addition to the user scopes that provide access to protected resources on the server some more technical scopes. Relevant one is the “offline\_access” scope.

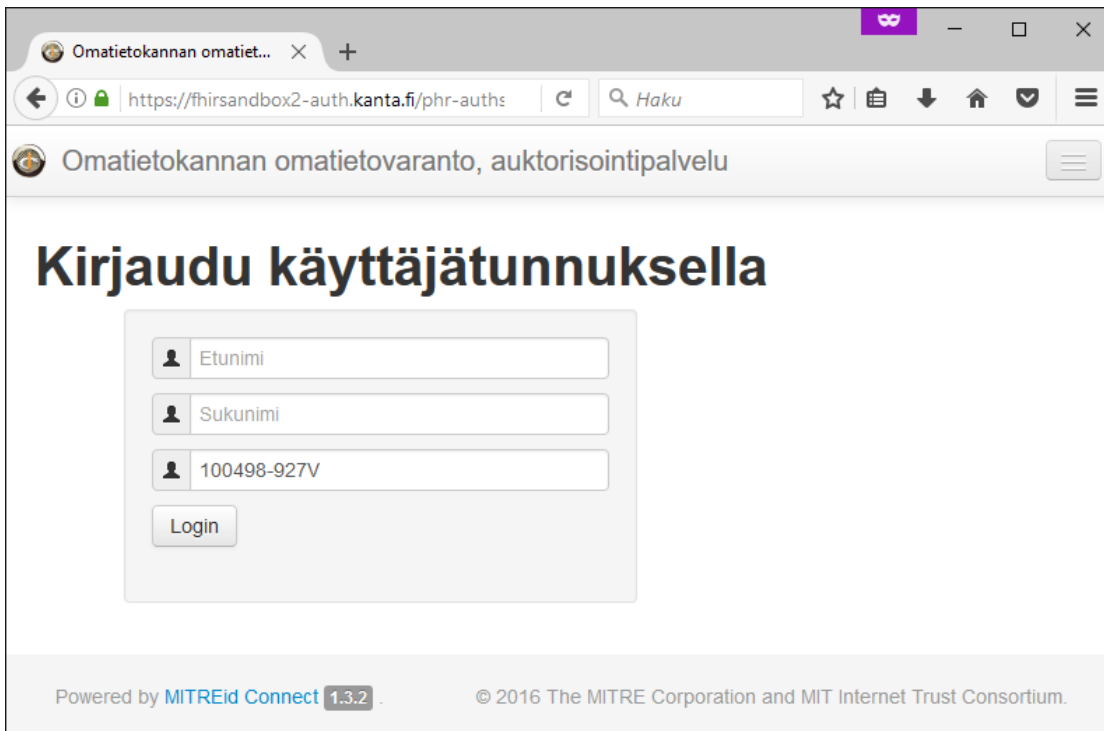
The “offline\_access” is defined in Open ID Core specification and enables the client to request new access token after expiration using the refresh token granted at authorization time.

Do leave the registration token scope selected, as that is required for editing the client.

## 5 Removing authorization

Log in into the UI using url <https://fhirsandbox2-auth.kanta.fi/phr-authserver-sandbox/login>.

The login requires using the test-SSN to approve the client:



Omatietokannan omatiet... X +

https://fhirsandbox2-auth.kanta.fi/phr-auths Haku

Omatietokannan omatietovaranto, auktorisointipalvelu

# Kirjaudu käyttäjätunnuksella

Etunimi

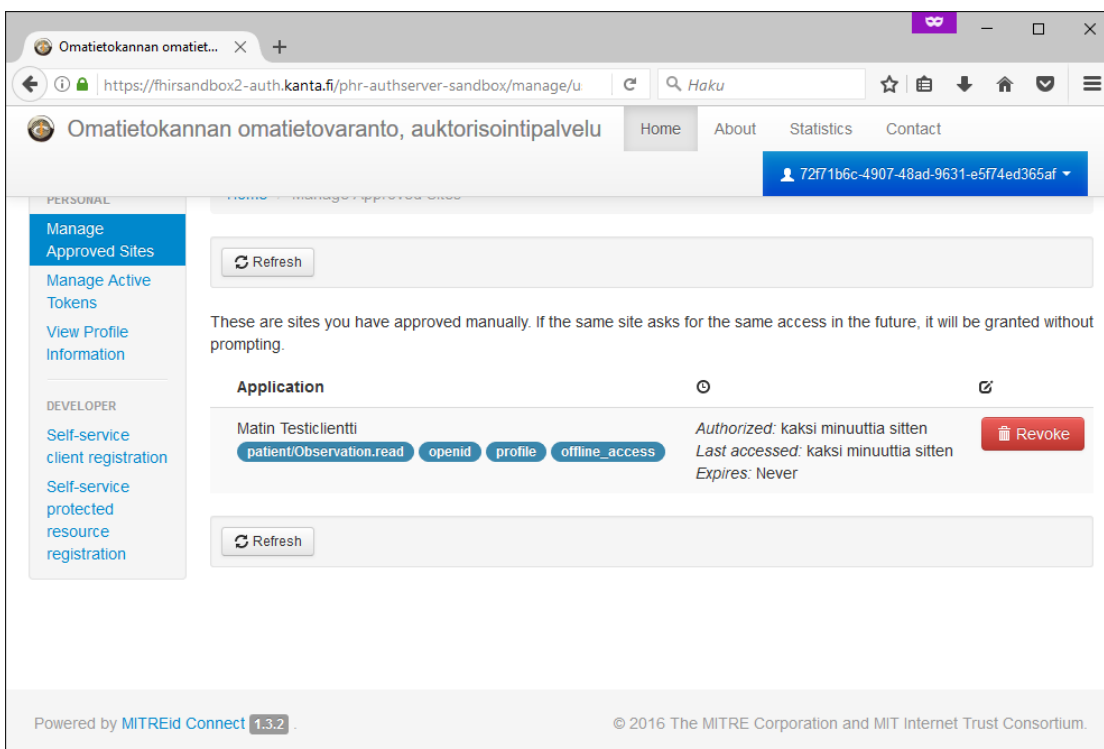
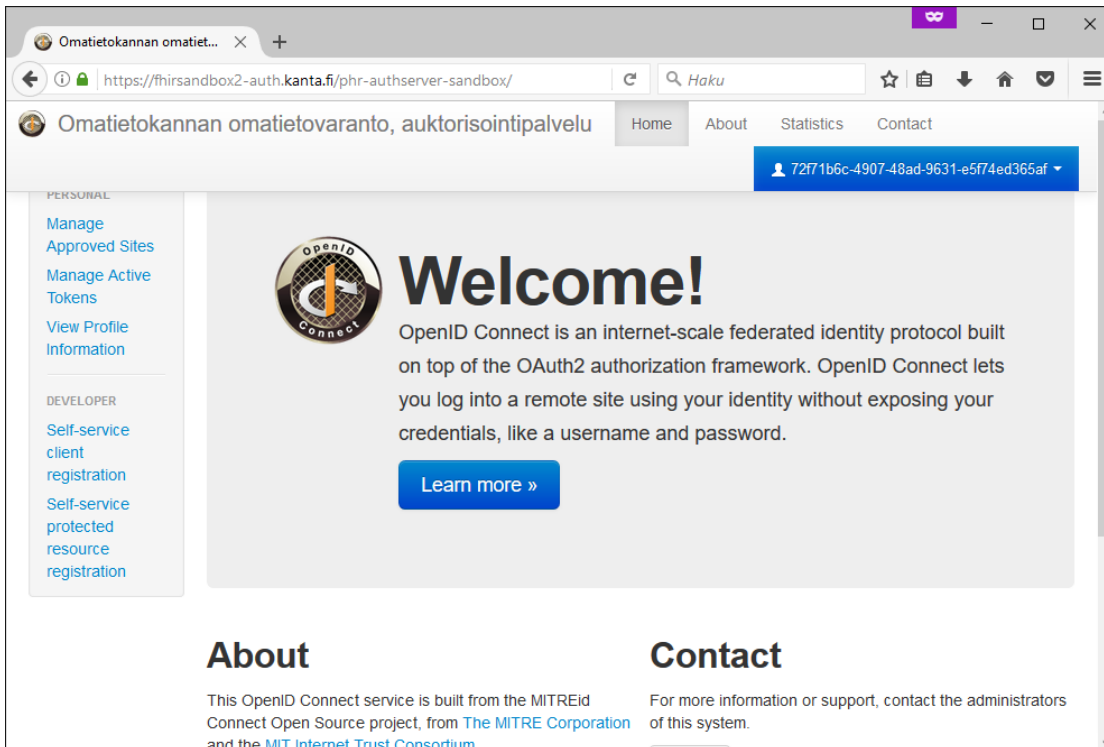
Sukunimi

100498-927V

Login

Powered by MITREid Connect 1.3.2 © 2016 The MITRE Corporation and MIT Internet Trust Consortium.

Once logged in the user UI opens with options to manage token and sites:



The “Manage approved sites” –action shows all clients the user has authorized and “Revoke”–action on that page will remove the authorization.

## 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.



```
{
  "resourceType": "Observation",
  "meta": {
    "profile": [
      "http://phr.kanta.fi/StructureDefinition/fiphr-bodyheight-stu3"
    ],
    "security": [
      {
        "system": "http://hl7.org/fhir/v3/Confidentiality",
        "code": "U",
        "display": "unrestricted"
      }
    ]
  },
  "language": "fi",
  ...
}
```

With the complete resource in the body.

