**(I) Discover the URLs**:

<https://fhir-ehr-code.cerner.com/r4/ec2458f2-1e24-41c8-b71b-0e701af7583d/.well-known/smart-configuration>

{

    "authorization\_endpoint": "https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/protocols/oauth2/profiles/smart-v1/personas/provider/authorize",

    "token\_endpoint": "https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/hosts/api.cernermillennium.com/protocols/oauth2/profiles/smart-v1/token",

    "revocation\_endpoint": "https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/protocols/oauth2/profiles/smart-v1/token/revoke",

    "token\_endpoint\_auth\_methods\_supported": [

        "client\_secret\_basic",

        "private\_key\_jwt"

    ],

    "token\_endpoint\_auth\_signing\_alg\_values\_supported": [

        "RS384",

        "ES384"

    ],

    "jwks\_uri": "https://authorization.cerner.com/jwk",

    "grant\_types\_supported": [

        "authorization\_code",

        "client\_credentials"

    ],

    "scopes\_supported": [

        "launch",

        "profile",

        "fhirUser",

        "openid",

        "online\_access",

        "offline\_access",

        "user/Account.read",

        "patient/Account.read",

        "system/Account.read",

        "user/AllergyIntolerance.read",

        "patient/AllergyIntolerance.read",

        "system/AllergyIntolerance.read",

        "user/AllergyIntolerance.write",

        "patient/AllergyIntolerance.write",

        "system/AllergyIntolerance.write",

        "user/Appointment.read",

        "patient/Appointment.read",

        "system/Appointment.read",

        "user/Appointment.write",

        "patient/Appointment.write",

        "system/Appointment.write",

        "user/Basic.write",

        "patient/Basic.write",

        "system/Basic.write",

        "user/Binary.read",

        "patient/Binary.read",

        "system/Binary.read",

        "user/CarePlan.read",

        "patient/CarePlan.read",

        "system/CarePlan.read",

        "user/CareTeam.read",

        "patient/CareTeam.read",

        "system/CareTeam.read",

        "user/ChargeItem.read",

        "patient/ChargeItem.read",

        "system/ChargeItem.read",

        "user/ChargeItem.write",

        "patient/ChargeItem.write",

        "system/ChargeItem.write",

        "user/Communication.read",

        "patient/Communication.read",

        "system/Communication.read",

        "user/Communication.write",

        "patient/Communication.write",

        "system/Communication.write",

        "user/Condition.read",

        "patient/Condition.read",

        "system/Condition.read",

        "user/Condition.write",

        "patient/Condition.write",

        "system/Condition.write",

        "user/Consent.read",

        "patient/Consent.read",

        "system/Consent.read",

        "user/Coverage.read",

        "patient/Coverage.read",

        "system/Coverage.read",

        "user/Coverage.write",

        "patient/Coverage.write",

        "system/Coverage.write",

        "user/Device.read",

        "patient/Device.read",

        "system/Device.read",

        "user/DiagnosticReport.read",

        "patient/DiagnosticReport.read",

        "system/DiagnosticReport.read",

        "user/DocumentReference.read",

        "patient/DocumentReference.read",

        "system/DocumentReference.read",

        "user/DocumentReference.write",

        "patient/DocumentReference.write",

        "system/DocumentReference.write",

        "user/Encounter.read",

        "patient/Encounter.read",

        "system/Encounter.read",

        "user/Encounter.write",

        "patient/Encounter.write",

        "system/Encounter.write",

        "user/FamilyMemberHistory.read",

        "patient/FamilyMemberHistory.read",

        "system/FamilyMemberHistory.read",

        "user/FamilyMemberHistory.write",

        "patient/FamilyMemberHistory.write",

        "system/FamilyMemberHistory.write",

        "user/Goal.read",

        "patient/Goal.read",

        "system/Goal.read",

        "user/Immunization.read",

        "patient/Immunization.read",

        "system/Immunization.read",

        "user/Immunization.write",

        "patient/Immunization.write",

        "system/Immunization.write",

        "user/InsurancePlan.read",

        "patient/InsurancePlan.read",

        "system/InsurancePlan.read",

        "user/Location.read",

        "system/Location.read",

        "user/MedicationAdministration.read",

        "patient/MedicationAdministration.read",

        "system/MedicationAdministration.read",

        "user/MedicationDispense.read",

        "patient/MedicationDispense.read",

        "system/MedicationDispense.read",

        "user/MedicationRequest.read",

        "patient/MedicationRequest.read",

        "system/MedicationRequest.read",

        "user/MedicationRequest.write",

        "patient/MedicationRequest.write",

        "system/MedicationRequest.write",

        "user/NutritionOrder.read",

        "patient/NutritionOrder.read",

        "system/NutritionOrder.read",

        "user/Observation.read",

        "patient/Observation.read",

        "system/Observation.read",

        "user/Observation.write",

        "patient/Observation.write",

        "system/Observation.write",

        "user/Organization.read",

        "system/Organization.read",

        "user/Organization.write",

        "system/Organization.write",

        "user/Patient.read",

        "patient/Patient.read",

        "system/Patient.read",

        "user/Patient.write",

        "patient/Patient.write",

        "system/Patient.write",

        "user/Person.read",

        "patient/Person.read",

        "system/Person.read",

        "user/Practitioner.read",

        "system/Practitioner.read",

        "user/Practitioner.write",

        "system/Practitioner.write",

        "user/Procedure.read",

        "patient/Procedure.read",

        "system/Procedure.read",

        "user/Procedure.write",

        "patient/Procedure.write",

        "system/Procedure.write",

        "user/Provenance.read",

        "patient/Provenance.read",

        "system/Provenance.read",

        "user/Provenance.write",

        "patient/Provenance.write",

        "system/Provenance.write",

        "user/Questionnaire.read",

        "patient/Questionnaire.read",

        "system/Questionnaire.read",

        "user/QuestionnaireResponse.read",

        "patient/QuestionnaireResponse.read",

        "system/QuestionnaireResponse.read",

        "user/QuestionnaireResponse.write",

        "patient/QuestionnaireResponse.write",

        "system/QuestionnaireResponse.write",

        "user/RelatedPerson.read",

        "patient/RelatedPerson.read",

        "system/RelatedPerson.read",

        "user/RelatedPerson.write",

        "patient/RelatedPerson.write",

        "system/RelatedPerson.write",

        "user/Schedule.read",

        "patient/Schedule.read",

        "system/Schedule.read",

        "user/ServiceRequest.read",

        "patient/ServiceRequest.read",

        "system/ServiceRequest.read",

        "user/Slot.read",

        "patient/Slot.read",

        "system/Slot.read",

        "user/Slot.write",

        "patient/Slot.write",

        "system/Slot.write",

        "system/FinancialTransaction.write"

    ],

    "response\_types\_supported": [

        "code"

    ],

    "management\_endpoint": "https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/personas/provider/my-authorizations",

    "introspection\_endpoint": "https://authorization.cerner.com/tokeninfo",

    "capabilities": [

        "authorize-post",

        "launch-ehr",

        "launch-standalone",

        "client-public",

        "client-confidential-asymmetric",

        "client-confidential-symmetric",

        "sso-openid-connect",

        "context-banner",

        "context-style",

        "context-ehr-patient",

        "context-ehr-encounter",

        "permission-patient",

        "permission-user",

        "permission-offline",

        "permission-online",

        "permission-v1",

        "health-cards"

    ]

}

(II) **Steps to Build a SMART on FHIR App with Standalone Launch**

1. **Authorization Endpoint**:
   * The authorization URL where users will be directed to log in and approve access to their FHIR data.
   * **Authorization Endpoint**:

https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/protocols/oauth2/profiles/smart-v1/personas/provider/authorize

1. **Token Endpoint**:
   * After you receive the authorization code, you will exchange it for an access token by sending a request to this URL.
   * **Token Endpoint**:

https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/hosts/api.cernermillennium.com/protocols/oauth2/profiles/smart-v1/token

1. **Scopes Supported**:
   * Your app can request specific scopes to define what data it will access. For a standalone launch, you’ll often need:
     + launch, profile, fhirUser, openid, patient/\*.read
   * Example scopes:

scope=launch profile fhirUser openid patient/\*.read

1. **Grant Type**:
   * The supported grant type for the authorization code flow is authorization\_code.
2. **JWKS URI**:
   * The URL for retrieving the JSON Web Key Set (JWKS), which is used for verifying tokens.
   * **JWKS URI**:

https://authorization.cerner.com/jwk

1. **Construct the Authorization URL**:
   * To begin the authorization process, your app should redirect the user to the authorization endpoint. Here's an example URL with query parameters for a standalone launch:

https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/protocols/oauth2/profiles/smart-v1/personas/provider/authorize

?response\_type=code

&client\_id=YOUR\_CLIENT\_ID

&redirect\_uri=YOUR\_REDIRECT\_URI

&scope=launch profile fhirUser openid patient/\*.read

&state=RANDOM\_STATE\_STRING

&aud=https://fhir-ehr-code.cerner.com/r4/ec2458f2-1e24-41c8-b71b-0e701af7583d

1. Replace:
   * YOUR\_CLIENT\_ID with the client ID you received during app registration.
   * YOUR\_REDIRECT\_URI with the redirect URI you registered for your app.
   * RANDOM\_STATE\_STRING with a unique string to maintain state between the request and callback.
   * aud with the FHIR server's base URL.
2. **Exchange the Authorization Code for an Access Token**: After the user authenticates and authorizes your app, they will be redirected back to your redirect\_uri with an authorization code. You need to exchange this code for an access token by making a POST request to the token endpoint.

**Token Request**:

POST https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/hosts/api.cernermillennium.com/protocols/oauth2/profiles/smart-v1/token

Content-Type: application/x-www-form-urlencoded

grant\_type=authorization\_code

&code=AUTHORIZATION\_CODE

&redirect\_uri=YOUR\_REDIRECT\_URI

&client\_id=YOUR\_CLIENT\_ID

&client\_secret=YOUR\_CLIENT\_SECRET

Replace:

* + AUTHORIZATION\_CODE with the code received from the authorization response.
  + YOUR\_CLIENT\_SECRET with your client secret if your app is confidential.

1. **Access FHIR Data**: Once you have the access token, include it in the Authorization header to make requests to the FHIR server. For example:

**FHIR API Request**:

GET https://fhir-ehr-code.cerner.com/r4/ec2458f2-1e24-41c8-b71b-0e701af7583d/Patient/12345

Authorization: Bearer YOUR\_ACCESS\_TOKEN

1. **Revoke Tokens (Optional)**: If you want to revoke a token, use the revocation endpoint:
   * **Revocation Endpoint**:

https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/protocols/oauth2/profiles/smart-v1/token/revoke

1. **Token Introspection (Optional)**: To introspect tokens and verify their validity, use the introspection endpoint:
   * **Introspection Endpoint**:

https://authorization.cerner.com/tokeninfo

**(III) Example Code (JavaScript with Node.js/Express)**

Here’s a simplified example to integrate the Cerner sandbox with a SMART on FHIR standalone launch using Node.js:

const express = require('express');

const axios = require('axios');

const querystring = require('querystring');

const app = express();

const clientId = 'YOUR\_CLIENT\_ID';

const clientSecret = 'YOUR\_CLIENT\_SECRET';

const redirectUri = 'YOUR\_REDIRECT\_URI';

const fhirBaseUrl = 'https://fhir-ehr-code.cerner.com/r4/ec2458f2-1e24-41c8-b71b-0e701af7583d';

app.get('/launch', (req, res) => {

const authUrl = `https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/protocols/oauth2/profiles/smart-v1/personas/provider/authorize?response\_type=code&client\_id=${clientId}&redirect\_uri=${encodeURIComponent(redirectUri)}&scope=launch profile fhirUser openid patient/Patient.read&state=RANDOM\_STATE\_STRING&aud=${encodeURIComponent(fhirBaseUrl)}`;

res.redirect(authUrl);

});

app.get('/callback', async (req, res) => {

const code = req.query.code;

const tokenResponse = await axios.post('https://authorization.cerner.com/tenants/ec2458f2-1e24-41c8-b71b-0e701af7583d/hosts/api.cernermillennium.com/protocols/oauth2/profiles/smart-v1/token', querystring.stringify({

grant\_type: 'authorization\_code',

code: code,

redirect\_uri: redirectUri,

client\_id: clientId,

client\_secret: clientSecret

}), {

headers: { 'Content-Type': 'application/x-www-form-urlencoded' }

});

const accessToken = tokenResponse.data.access\_token;

const patientData = await axios.get(`${fhirBaseUrl}/Patient/12345`, {

headers: { Authorization: `Bearer ${accessToken}` }

});

res.json(patientData.data);

});

app.listen(3000, () => {

console.log('App is running on port 3000');

});

**Conclusion:**

* Use the authorization\_endpoint to redirect users for login.
* Exchange the authorization code at the token\_endpoint for an access token.
* Make FHIR API calls with the access\_token and fetch patient data.