

Biometrics configuration

The Reference Application has some built-in support for collecting biometrics and searching for existing patients based on scanned biometrics. Currently widgets are available for collecting fingerprints, but the design is intended to be extensible to allow for other types of biometric data collection.

Step 1: Define a BiometricEngine

Because there are many different biometrics services, SDKs, and APIs available, from different vendors, an approach was developed to enable groups to plug in their particular biometric implementation [behind a standard interface called a BiometricEngine](#). The idea is that standard BiometricEngine implementations could be developed that may be shared, or individual implementations could provide their own based on their specific configurations. PIH has created an implementation of this for some of their implementations, [which can be found here](#) as an example (this implementation connects restfully to an independent biometric application that was developed using the Neurotechnology SDK. [That code can be found here](#)).

Step 2: Configure the Registration Core module to enable this BiometricEngine

This is a matter of setting the appropriate global property value. For example:

Biometrics Implementation

Which Biometrics engine implementation to should we connect to? Specify a Spring bean name, or leave blank to disable

Step 3: Configure the Registration App with one or more widgets to collect Biometrics

Currently, there is only a single widget available that is designed to collect one or more fingerprints. Below is an example configuration:

Example configuration

```
{
  "type": "fingerprint",
  "uuid": "uuid-of-patient-identifier-type-for-storing-fingerprint-reference",
  "widget": {
    "providerName": "registrationapp",
    "fragmentId": "field/fingerprint"
    "config": {
      "format": "Whatever code your engine and scanner expect to determine what template format to use (eg.
ISO)"
      "scanUrl": "a url that initiates a scan operation on the fingerprint reader and returns a fingerprint
template"
      "devicesUrl": "a url that returns a list of available fingerprint scanners"
      "fingers": [
        {
          "formFieldName": "a unique field name to use to capture this"
          "label": "Can enter either a text label or a message code here",
          "type": "Whatever code is needed by your BiometricEngine to identify which finger this represents"
        },
        {
          "formFieldName": "leftIndexTemplate",
          "label": "zl.registration.patient.biometrics.fingerprints.leftIndexLabel",
          "type": "LEFT_INDEX_FINGER"
        }
      ]
    }
  }
}
```

Step 4: Extend Patient Search Widget to search for patients by Biometrics

A widget has been added to Registration App that can be injected into the Patient Search widget in Core Apps to integrate searching by biometrics /fingerprints into the main basic search. This widget currently assumed a RESTful interaction with a Biometrics server and is based around the pattern defined in RestBiometricEngine mentioned above, but is not dependent on it.

The widget is provided by the registration app module and is called "biometrics/fingerprintSearch". It would be configured by setting up an Extension similar to the one below (this sample has **not** fully been testing, because in our use case we programmatically set up our extensions. If you try it and have issues please post on Talk and we can debug).

Note that this extension must be injected into the same page as the Core Apps Patient Search. If everything is working successfully, the search widget will use the "devicesUrl" to confirm that a fingerprint scanner is present, and the "scanUrl" to repeatedly poll for a fingerprint during the search process.

```
{
  "type": "include-fragment",
  "uuid": "uuid",
  "extensionPoint": "coreapps.patientSearch.extension",
  "extensionParams": {
    "provider": "registrationapp",
    "fragment": "biometrics/fingerprintSearch",
    "fragmentConfig": {
      "scanUrl": "a url that initiates a scan operation on the fingerprint reader and returns
a fingerprint template"
      "devicesUrl": "a url that returns a list of available fingerprint scanners"
    }
  }
}
```