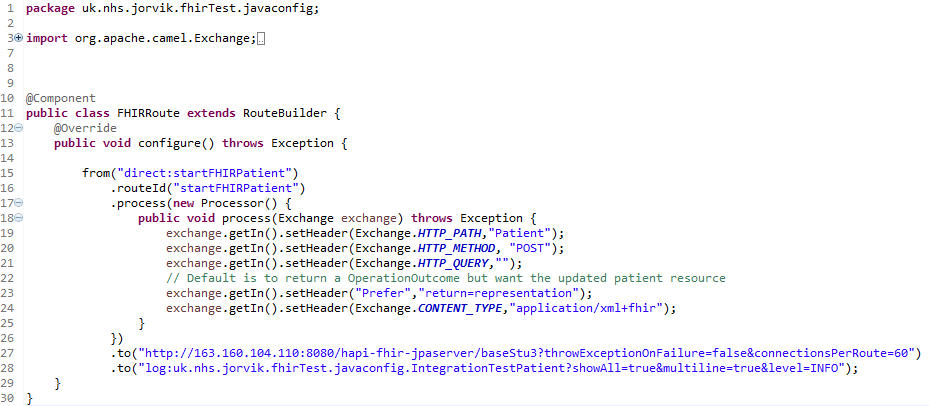
## Creating a Patient Resource and Testing

Creating a Patient resource and then send it to a FHIR Reference server. The resource conforms to InterOpen’s CareConnect Patient profile.

The code is java based (using HAPI FHIR libraries) and can be found @ <https://github.com/KevinMayfield/FHIRTest/tree/master/CareConnectPatient>

Although HAPI can communicate directly with a FHIR server, Camel is a popular lightweight integration tool that can use a wide variety of transports and allows testing to become more advanced. [Such as the HL7v2 version HAPI which is used in Mirth]

The camel route below is configured to send the resource to 163.160.104.110 server on port 8080 (line 27), you will need to change to work with your own FHIR server.



|  |
| --- |
| **Reference Servers**  HAPI FHR has instructions for setting up a Java based reference server (DSTU2 and STU3) <http://hapifhir.io/doc_jpa.html> (see also <https://www.openhealthhub.org/t/howto-build-a-health-database-and-fhir-api-server-in-15-mins-using-open-source/155>)  C# Furore Spark (DSTU2) <https://github.com/furore-fhir/spark>  Intersystems Health Connect (2016) also includes a FHIR Reference Server (DSTU2)  *Note: STU3 will be the current version of FHIR around the 20th March 2017* |

*CareConectPatientTest.java* is the main test class in the project, which initialises the testing framework, sends a test message and checks the response.



*buildCareConnectFHIRPatient* function creates the Patient resource.