|  |  |
| --- | --- |
| {  "meta": {  "href": "http://ehrscape.code-4-health.org/rest/v1/composition/a31957ec-9f7c-4c7b-aeb9-555ff4067d95::ripple\_osi.ehrscape.c4h::1"  },  "format": "STRUCTURED",  "templateId": "LCR Problem Encounter.v0",  "composition": {  "problem\_diagnosis\_record": {  "\_uid": [  "a31957ec-9f7c-4c7b-aeb9-555ff4067d95::ripple\_osi.ehrscape.c4h::1"  ], | condition.setId("a31957ec-9f7c-4c7b-aeb9-555ff4067d95::ripple\_osi.ehrscape.c4h::1"); |
| "context": [  {  "start\_time": [  "2015-10-19T12:49:10.000Z"  ],  "setting": [  {  "|code": "238",  "|value": "other care",  "|terminology": "openehr"  }  ]  }  ], | Encounter encounter = **new** Encounter();  encounter.setId("#enc");    Period period = **new** Period();  Date startDate;  **try** {  startDate = fmt.parse("2015-10-19 12:49:10");  period.setStart(startDate);  encounter.setPeriod(period);    } **catch** (ParseException e) {  // **TODO** Auto-generated catch block  }  encounter.setClass\_(EncounterClass.***OTHER***);    condition.getContained().add(encounter);  condition.addIdentifier()  .setSystem(GOVFictitiousCouncil.***URI\_openEHR\_PROBLEM\_DIAGNOSIS***)  .setValue("64ed73f2-50e2-44e4-9ea5-a2d383be6b97");    CodeableConcept code = **new** CodeableConcept();  code.addCoding()  .setSystem(NHSUKConstants.***URI\_SNOMED***)  .setCode("22298006") // Note swapped for real code  .setDisplay("myocardial infarction");  condition.setCode(code); |
| "problem diagnosis": [  {  "\_uid": [  "64ed73f2-50e2-44e4-9ea5-a2d383be6b97"  ],  "problem": [  {  "|code": "1234567",  "|value": "myocardial infarction",  "|terminology": "snomed-ct"  }  ], |
| "date\_time\_of\_onset": [  "2015-10-19T12:32:17.708Z"  ], | DateTimeType onset=**new** DateTimeType();  SimpleDateFormat fmt = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  **try** {  Date onsetDate;  onsetDate = fmt.parse("2015-10-19 12:32:17");  onset.setValue(onsetDate);  condition.setOnset(onset);  }  **catch** (ParseException e1) {  // **TODO** Auto-generated catch block  } |
| "diagnostic\_certainty": [  {  "|code": "at0076",  "|value": "Confirmed",  "|terminology": "local"  }  ], | condition.setVerificationStatus(ConditionVerificationStatus.CONFIRMED); |
| "comment": [  "A comment"  ] |  |
| }  ],  "composer": [  {  "|name": "Dr Tony Shannon"  }  ]  }  }, | Reference drRef = **new** Reference();  drRef.setReference("Practitioner/#dr");  condition.setAsserter(patRef);    Practitioner dr = **new** Practitioner();  dr.setId("#dr");  dr.getName()  .addFamily("Shannon")  .addGiven("Tony")  .addPrefix("Dr")  .setText("Dr Tony Shannon");  condition.getContained().add(dr); |
| "deleted": false,  "lastVersion": true  } | Reference patRef = **new** Reference();  patRef.setReference("Patient/64ed73f2-50e2-44e4-9ea5-a2d383be6b97");  condition.setSubject(patRef); |
|  |  |