



# CDS for Pediatric Antibiotics using SMART on FHIR

Seth Russell

[seth.russell@gmail.com](mailto:seth.russell@gmail.com)

Contributions from:

Jeremy Gleed, Nicholas Link, Sashi Poudel,

Jace Richards



# System Specifications

- Over or misprescribing of pediatric antibiotics brings the need for advanced decision making assistance
- System recommendations based on age, weight, symptoms, lab values
- Limited scope CDS: Acute Otitis Media (AOM) and Strep throat

# Knowledge Source

- Pediatric Care Process Model developed for Intermountain Health Care
- All CPMs: [https://intermountainphysician.org/clinical/Pages/Care-Process-Models-\(CPMs\).aspx](https://intermountainphysician.org/clinical/Pages/Care-Process-Models-(CPMs).aspx)
- CPMs used for this SMART on FHIR application:
  - Acute Otitis Media <https://intermountainhealthcare.org/ext/Dcmnt?ncid=522927223>
  - Streptococcal Pharyngitis <https://intermountainhealthcare.org/ext/Dcmnt?ncid=525953897>



# Data Requirements

- Patient data from Health Services Platform Consortium (HSPC) and Electronic Health Record (EHR)
- Observations and labs from Logical Observation Identifier Names and Codes (LOINC)
- Symptoms, procedures, and diagnoses from Systematized Nomenclature of Medicine - Clinical Terms (SNOMED-CT)



# Workflow Integration

- Directly integrated with outpatient EHR
  - Assumed physician has access to EHR for chart review
- System facilitates decision making
  - Medication recommendations, dosage calculations



# System Architecture

- Built using EmberJS, Bootstrap, JQuery, JavaScript client for FHIR
- No server side component required other than FHIR compliant server and hosting of static web pages



# Application Information

- Hosted at <http://cds-magiclantern.rhcloud.com>
- Source Code available at: <https://github.com/magic-lantern/AntimicrobialCDS>
- More guidance available from <https://healthservices.atlassian.net/wiki/display/HSPC/For+Developers>



# Future Direction

- Improved application logic - rules engine or user configurable setup
- Add patient component to CDS - help patients understand importance of Antibiotics
- More advanced drug dosing, pharmacy configurable
- Send updates back via FHIR
  - Updated observations
  - Prescription sent to CPOE system





For this application, critical observations are age, weight, temperature. Antibiotic dosing is age dependent; weight determines amount of dosage for some infections, and temperature is a key sign of an infection.

```
readWeight: function() {  
    var self = this;  
    self.getObservation('3141-9', function(r){  
        self.patient.weight = r;  
    });  
},
```

**Sample LOINC information:** <http://s.details.loinc.org/LOINC/3141-9.html?sections=Comprehensive>

Use jQuery to asynchronously call `patientContext.api.search` (`fhirclient.patient.api.search`). I'm looking for specific values – dates, value of observation, unit of value, etc. Each FHIR resource returns different information, so depending on what you are getting back you'll need to do something different.

```
getObservation: function(code, callback, count = 1) {
  var ret = {value: 'No Observation'};

  Ember.$.when(this.patientContext.api.search({
    'type': "Observation",
    'query': {
      'code': code,
      '_sort:desc': 'date'},
    'count': count}))
    .done(function(observations) {
      console.log('observations: ', observations);
      if (!Ember.isNone(observations.data.entry)) {
        observations.data.entry.forEach(function(obs) {
          if (obs.resource.hasOwnProperty('effectiveDateTime') &&
              obs.resource.hasOwnProperty('valueQuantity') &&
              obs.resource.valueQuantity.hasOwnProperty('value') &&
              obs.resource.valueQuantity.hasOwnProperty('unit')) {
            ret.value = obs.resource.valueQuantity.value;
            ret.date = obs.resource.effectiveDateTime;
            ret.unit = obs.resource.valueQuantity.unit;
          }
          else {
            console.log("fhir-client - expected properties missing for code ", code);
          }
        });
      }
      if (typeof callback === 'function') {
        callback(ret);
      }
    });
},
```