



HL7 FHIR

HL7
International

FHIR Intermediate Course

MODULE 4: SMART ON FHIR & CDS HOOKS

Assignments

Course Overview

Module I: Implementation Guides

Most relevant FHIR Implementation Guides: Argonaut & IPS

Argonaut Development and Roadmap

Argonaut Data Query IG: Scope, Use Cases

Argonaut Provider Directory IG: Scope, Use Cases

IPS FHIR IG: Scope, Use Cases

Module II: FHIR Clients

General Guidelines for FHIR Clients

FHIR Clients in JavaScript / C# / Java [1 - Elective]

Module III: FHIR Facades

Why use FHIR Server Facade: your system on FHIR

Specific FHIR Servers (FHIR Facade)

Facade Use Case / Scenarios

Facade Architecture / Patterns

Where to put the FHIR Facade

System Integration / Integration Engine / Bus / Messaging

Facade in Java / Node.JS [1 - Elective]

Module IV: FHIR Applications

Smart-On-FHIR

CDS-Hooks

Integration with Smart-On-FHIR / CDS-Hooks [1 - Elective]

Assignments for Module 4

Assignments For Module 4	3
Goals and Options	4
1. Assignment 01 : Create a new launch.html	5
2. Assignment 02: Load Patient Prescriptions	6
3. Assignment 03: Fix the card	8

Goals and Options

In this week you will improve some parts of the smart app in the “Build your Own” section and review some code, in the “Code Review” section.

Assignments

01	Create a new launch.html	20 points
02	Load Patient prescriptions	60 points
03	Fix the card	20 points

Maximum grade is 100 points

You need to obtain at least 60 points in order to complete this unit.

1. Assignment 01: Create a new launch.html



BUILD YOUR OWN

You need to create a correct new module for the main app.

The application must be able to launch a SMART-ON-FHIR application.

Context Info: current user and patient info

Who will be using it: the practitioner

Launch Type: the app will be launched from the EHR

FHIR Resources Needed (read and search only): demographics, vital signs, prescriptions, immunizations, problems and procedures

Your task: create a new **launch.html** file that includes the correct SMART-ON-FHIR scope

Note: You can use either of the two specifications (V1 or V2)

Submit the new launch.html code.

2. Assignment 02: Load Patient Prescriptions



BUILD YOUR OWN


According to the US Core specification, <https://hl7.org/fhir/us/core/StructureDefinition-us-core-medicationrequest.html>, all mandatory elements must be shown in the prescription table, except the prescriber.

This is what is displayed now:


EHR

Sara Angulo

Sign out



ARMSTRONG Alberto

 20 years

Medications




Medications

Vital Signs

Immunizations

Conditions

Procedures

STATUS	MEDICATION	DATE	CHECK PRICE
active	12 HR Oxycodone Hydrochloride 80 MG Extended Release Oral Tablet [Oxycontin]	2023-03-16	
stopped	Penicillin V Potassium 250 MG Oral Tablet	2016-03-28	
stopped	Naproxen sodium 220 MG Oral Tablet	2020-07-14	

Add the necessary code to the function createMedicationGrid (bundle, patientId, userid) and the index.html to show these required fields.

HINT: Only one element is missing and you must include it before the STATUS column.

Note: The original function is the next page of this document, just in case you cannot find it in the JavaScript project.

Submit the createMedicationGrid(bundle, patientId, userid) updated.

This is the original function, just in case you cannot find it

```
function createMedicationGrid(bundle, patientId, userid) {
  b = bundle;
  const total = bundle.length;
  const tableData = bundle.map(v => {
    console.log(v);
    return (
      `<tr>

      <td class="text-center">${v.resource.status}</td>
      <td>${v.resource.medicationCodeableConcept.text}</td>
      <td class="text-center">${formatdate(v.resource.authoredOn)}</td>
      <td class="text-center"><i title="Check Price" class="fa fa-check-square fa-lg text-
success pointer checkprice
      data-cdssURL="${cdssURL}" data-fhirserver="${fhirserver}"
      data-userid="${userid}" data-patientId="${patientId}"
      data-medicationRequestId="${v.resource.id}"></i></td>

      </tr>`
    );
  }).join('');
  const tableBody = document.querySelector("#body");
  tableBody.innerHTML = tableData;
}
```

3. Assignment 03: Fix the card



CODE REVIEW

The CDS Service has a bug that could not be detected by our team and it fails in the applications that consume the service.

Your task: Find the bug and fix it.

Submit the code fixed.

```
if (price) {  
    card.put("summary", " Cost: $500.41. Save $426.47 with a generic medication.");  
    card.put("detail", "Last price update: Two years ago");  
    card.put("indicator", "info");  
    JSONArray sources = new JSONArray();  
    JSONObject source = new JSONObject();  
    source.put("label", " CMS Public Use Files");  
    sources.put(source);  
    card.put("sources", sources);  
    JSONArray suggestions = new JSONArray();  
  
    JSONObject sugg = new JSONObject();  
    sugg.put("label", "Check price update");  
    sugg.put("uuid", UUID.randomUUID());  
    suggestions.put(sugg);  
  
    cardsArray.put(card);  
    break;  
}
```