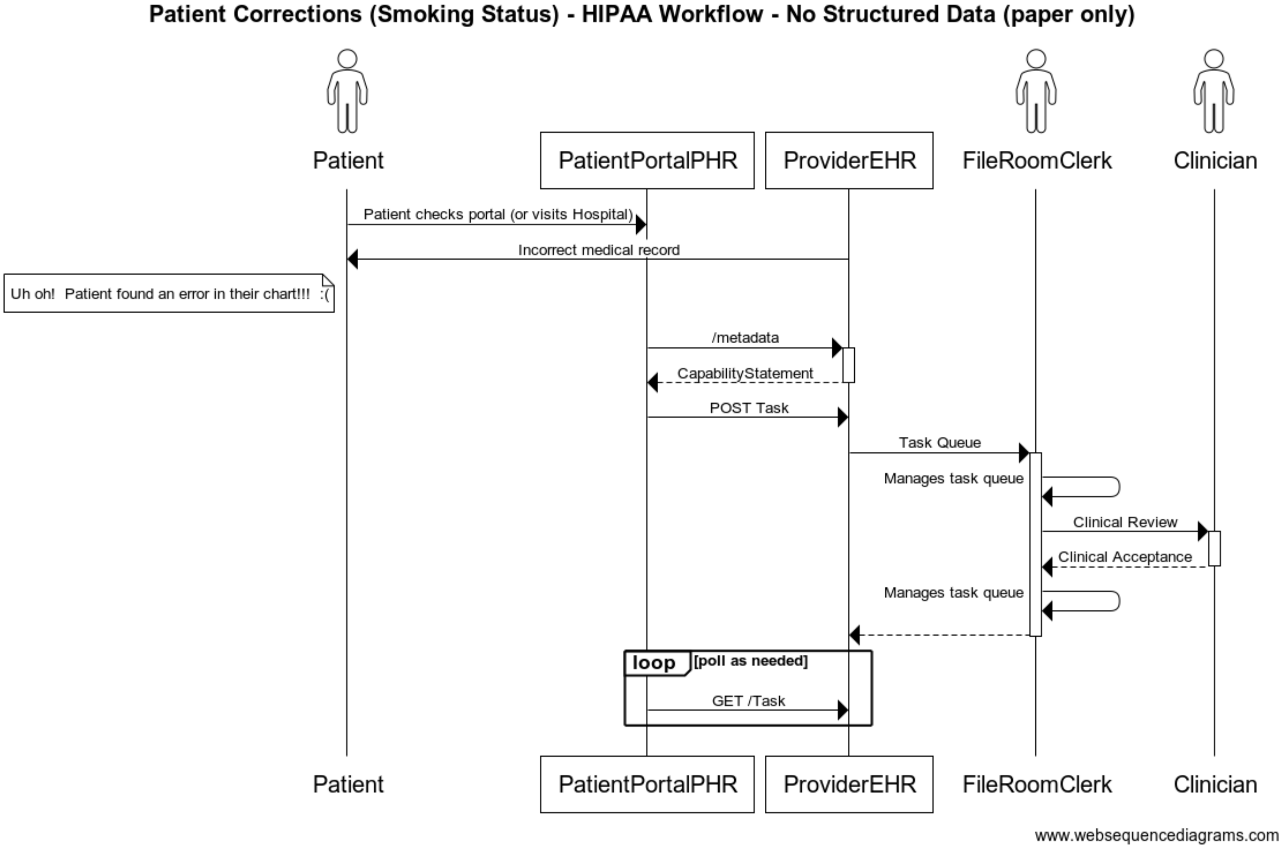
HL7® FHIR
TestScript Template

Scenario 1 – Patient Correction for Incorrect Smoking Status (without Structured Data)

Demonstrating the interactions between a Provider EHR and 3rd party Patient Portal/PHR, when a Patient initiates a correction to their medical records. The Patient Correction process provides a mechanism to request, authorize and testify that a change to the legal medical record occur.

The patient is a non-smoker who visits a hospital, and receives a printed copy of her medical records . As the patient is reviewing her records, she sees that one of them lists her as a smoker. The patient signs into their Patient Portal/PHR, and initiates the PatientCorrection process by writing a letter based on the Morgan Gleason template and submitting it via a Task to the ProviderEHR.



FHIR Interactions

GET	{server}/metadata
GET	{server}/ValueSet/hipaa-corrections-business-statuses
GET	{server}/Task
POST	{server}/Task
GET	{server}/Task/{taskId}
GET	{server}/Task/{taskId}/_history
PUT	{server}/Task/{taskId}

Use Case Interactions

Describe the steps for the use case. Not all steps will have FHIR Interactions. This is meant to be the plain text of the various interactions needed to complete the use case fully.

1) *Patient receives copies of paper medical records. Finds mistake.*
2) *Patient signs into PatientPortal/PHR*
3) *PatientPortal constructs a narrative request, similar to the Morgan Gleason example.*
4) *PatientPortal creates a Task resource, and sends the narrative request to the ProviderEHR.*
5) *Internal workflow at the ProviderEHR progresses the Task request through business states until it is rejected or accepted.*
6) *The PatientPortal/PHR periodically polls the ProviderEHR to determine updates to the Task.*
7) *When the Task is accepted and complete, Patient accesses either the ProviderEHR or PatientPortal/PHR, and views the corrected record.*

Preconditions

List out any pre-conditions that are needed for the use case. This may include having a certain patient or kind of patient or provider loaded to the test system, etc.

- Patient A is registered in the Provider EHR
- Patient A has a medical history including, but not limited to, a Condition with a smoking related SMOMED code.
- Provider EHR supports Task resource.
- Provider EHR supports SMART on FHIR authentication
- Systems are using FHIR R4

Test Assertions

Use cases may include one or more FHIR interactions. For each FHIR interaction in your Use Case, determine which assertions (meaning, "this must or may be true") should be tested within that interaction.

Test 0 – Authentication (Optional)

PatientPortal/PHR issues a request to the Provider EHR for Patient health history

Request Asserts

- *Validate the request access to the FHIR API using OAuth2.0*

Response Asserts

- *Validate response code must be one of the following:*
  - *200 (subscriber found)*
  - *404 (subscriber not found)*
  - *500 (subscriber not unique)*

Test 01 – Capability Statement

PatientPortal/PHR should retrieve the EMR's meta/capabilityStatement/ConformanceStatement to determine the version of FHIR and the resources that are writeable.

Request Asserts (Peer-to-Peer)

- *Validate Capability Statement retrieve request is correct*

Response Asserts

- *HTTP GET response code is 200 (OK)*
- *HTTP response body is a FHIR CapabilityStatement*

Test 02 - Create Task

PatientPortal should be able to post a new Task to the Provider EHR

Request Asserts (Peer-to-Peer)

- *Validate can post new Task*

Response Asserts

- *HTTP POST response code is 201 (Created)*
- *HTTP response body is a FHIR Task with a server-assigned ID {taskId}*

Test 03 - Read Task

PatientPortal should be able to read and fetch updates regarding the Task it created.

Request Asserts (Peer-to-Peer)

- *Validate can read new created Task*

Response Asserts

- *HTTP GET response code is 200 (OK)*
- *HTTP response body is a FHIR Task with a server-assigned ID {taskId}*

Test 03 - Update Task

PatientPortal should be able to update the Task.

Request Asserts (Peer-to-Peer)

- *Validate can update Task*

Response Asserts

- *HTTP PUT response code is 202 (Accepted)*
- *HTTP response body is a FHIR Task with a server-assigned ID {taskId}*

Test 03 - View Task History

PatientPortal should be able to view the Task history and how it's changed over time.

Request Asserts (Peer-to-Peer)

- *Validate can query /Task/_history*

Response Asserts

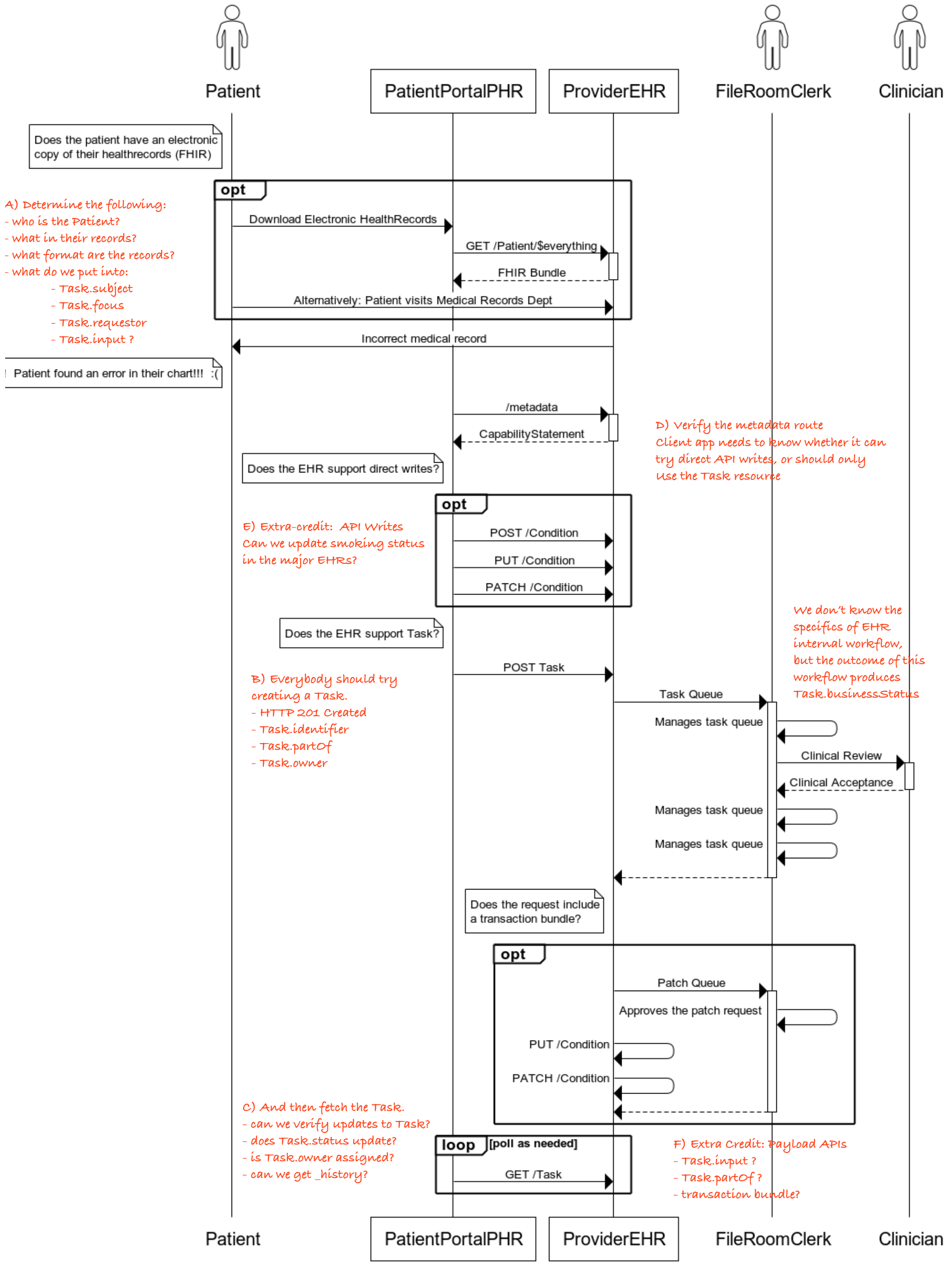
- *HTTP PUT response code is 200 (OK)*
- *HTTP response body is a FHIR Bundle of Task resources*

Scenario 2 – Patient Correction for Incorrect Smoking Status (with Structured Data)

Demonstrating the interactions between a Provider EHR and 3rd party Patient Portal/PHR, when a Patient initiates a correction to their medical records. The Patient Correction process provides a mechanism to request, authorize and testify that a change to the legal medical record occur.

The patient is a non-smoker who uses a PatientPortal/PHR. The patient signs into their Patient Portal/PHR, and downloads their personal health data from ProviderEHR via FHIR. As the patient is reviewing her records, she sees that one of them lists her as a smoker. She selects the incorrect record, and then initiates the PatientCorrection process.

If both the Provider EHR and PatientPortal/PHR are on release 4.0 (FHIR R4), the PatientPortal system can query for a Capability Statement to determine which resources support write access, and whether the EHR supports the Task. The PatientPortal/PHR constructs the appropriate API calls for the information that can be written directly to, and creates a Task for that data which can't.



A) Determine the following:
- who is the Patient?
- what in their records?
- what format are the records?
- what do we put into:
    - Task.subject
    - Task.focus
    - Task.requestor
    - Task.input ?

D) Verify the metadata route
Client app needs to know whether it can try direct API writes, or should only Use the Task resource

E) Extra-credit: API Writes
Can we update smoking status in the major EHRs?

We don't know the specifics of EHR internal workflow, but the outcome of this workflow produces Task.businessStatus

B) Everybody should try creating a Task.
- HTTP 201 Created
- Task.identifier
- Task.partOf
- Task.owner

C) And then fetch the Task.
- can we verify updates to Task?
- does Task.status update?
- is Task.owner assigned?
- can we get _history?

F) Extra Credit: Payload APIs
- Task.input ?
- Task.partOf ?
- transaction bundle?

FHIR Interactions

- GET {server}/metadata
- GET {server}/ValueSet/us-core-observation-ccdasmokingstatus
- GET {server}/ValueSet/hipaa-corrections-business-statuses

- GET {server}/Patient
- GET {server}/Patient/$everything

- GET {server}/Task
- POST {server}/Task
- GET {server}/Task/{taskId}
- GET {server}/Task/{taskId}/_history
- PUT {server}/Task/{taskId}

- GET {server}/Condition
- GET {server}/Condition?code={snomedCode}
- GET {server}/Condition?code=449868002,428041000124106,8517006,266919005,77176002,266927001, 428071000124103,428061000124105
- POST {server}/Condition
- GET {server}/Condition/{conditionId}
- PUT {server}/Condition/{conditionId}

- POST {server}/DocumentReference
- GET {server}/DocumentReference/{documentReferenceId}

Use Case Interactions

Describe the steps for the use case. Not all steps will have FHIR Interactions. This is meant to be the plain text of the various interactions needed to complete the use case fully.

1) *Patient signs into PatientPortal/PHR*
2) *Patient provides credentials for a Provider EHR and authorizes the retrieval of health data by PatientPortal/PHR*
3) *PatientPortal/PHR queries Provider EHR for patient data associated with Patient*
4) *Provider EHR responds with the Patient record*
5) *Patient recognizes an error in one of the records, selects the record, and begins the Patient Correction process.*
6) *PatientPortal then queries the Provider EHR Capability Statement to determine which resources support write access, and whether the EHR supports the Task.*
7) *If the ProviderEHR supports writable APIs, the corrected patient information is sent directly back to the Provider EMR via FHIR API endpoint.*
8) *If the Provider EHR supports the Task resource, a Task is created that includes the DocumentReference and a transaction Bundle containing any writeable FHIR APIs, and the data to be corrected.*
9) *The Provider EHR will either then accept the changes via API, or initiate a Task workflow.*
10) *The PatientPortal polls the writable endpoints to determine if/when the data is corrected and updated*
11) *If available, the PatientPortal polls the Task endpoint and the Task workflow proceeds back and forth between the systems, until the Task is resolved and returned an Accepted or Rejected state to the PatientPortal/PHR*

Preconditions

List out any pre-conditions that are needed for the use case. This may include having a certain patient or kind of patient or provider loaded to the test system, etc.

- Patient A is registered in the Provider EHR
- Patient A has a medical history including, but not limited to, a Condition with a smoking related SMOMED code.
- Provider EHR supports writeable API endpoints
- Provider EHR supports SMART on FHIR authentication
- Systems are using FHIR R4

Test Assertions

Use cases may include one or more FHIR interactions. For each FHIR interaction in your Use Case, determine which assertions (meaning, "this must or may be true") should be tested within that interaction.

Test 0 – Authentication (Optional)

PatientPortal/PHR issues a request to the Provider EHR for Patient health history

Request Asserts

- *Validate the request access to the FHIR API using OAuth2.0*

Response Asserts

- *Validate response code must be one of the following:*
  - *200 (subscriber found)*
  - *404 (subscriber not found)*
  - *500 (subscriber not unique)*

Test 01 – Capability Statement

PatientPortal/PHR should retrieve the EMR's meta/capabilityStatement/ConformanceStatement to determine the version of FHIR and the resources that are writeable.

Request Asserts (Peer-to-Peer)

- *Validate Capability Statement retrieve request is correct*

Response Asserts

- *HTTP GET response code is 200 (OK)*
- *HTTP response body is a FHIR CapabilityStatement*

Test 02 - Create Task

PatientPortal should be able to post a new Task to the Provider EHR

Request Asserts (Peer-to-Peer)

- *Validate can post new Task*

Response Asserts

- *HTTP POST response code is 201 (Created)*
- *HTTP response body is a FHIR Task with a server-assigned ID {taskId}*

Test 03 - Read Task

PatientPortal should be able to read and fetch updates regarding the Task it created.

Request Asserts (Peer-to-Peer)

- *Validate can read new created Task*

<u>Response Asserts</u>

- *HTTP GET response code is 200 (OK)*
- *HTTP response body is a FHIR Task with a server-assigned ID {taskId}*

Test 03 - Update Task

PatientPortal should be able to update the Task.

<u>Request Asserts</u> (Peer-to-Peer)

- *Validate can update Task*

<u>Response Asserts</u>

- *HTTP PUT response code is 202 (Accepted)*
- *HTTP response body is a FHIR Task with a server-assigned ID {taskId}*

Test 04 - Create Condition

PatientPortal should be able to post a new Condition to the Provider EHR

<u>Request Asserts</u> (Peer-to-Peer)

- *Validate can post new Condition*

<u>Response Asserts</u>

- *HTTP POST response code is 201 (Created)*
- *HTTP response body is a FHIR Condition with a server-assigned ID {taskId}*

Test 05 - Read Condition

PatientPortal should be able to read and fetch updates regarding the Condition it created.

<u>Request Asserts</u> (Peer-to-Peer)

- *Validate can read new created Condition*

<u>Response Asserts</u>

- *HTTP GET response code is 200 (OK)*
- *HTTP response body is a FHIR Condition with a server-assigned ID {taskId}*

Test 06 - Update Condition

PatientPortal should be able to update the Condition.

<u>Request Asserts</u> (Peer-to-Peer)

- *Validate can update Condition*

<u>Response Asserts</u>

- *HTTP PUT response code is 202 (Accepted)*
- *HTTP response body is a FHIR Condition with a server-assigned ID {taskId}*