REST operations with HTTP

2. REST Exercise: using HTTP commands for CRUD

This exercise aims to allow you to familiarize yourself with how FHIR works with REST to do CRUD operations.

To carry out these exercises, you need to have a tool that monitors your HTTP traffic, like Postman or Fiddler.

Exercises

1. Try getting a single patient:
   - GET a patient with id 'example' (on vonk.furore.com)
   - Look at the response headers and validate that they are correct
   - Try to get the same patient in another format (JSON or XML)
     - Using the _format parameter
     - Using an Accept header

2. Store the patient XML/JSON somewhere on your machine and edit it to match your own fictional situation.

3. Try creating a new patient on the FHIR server with this data. Note the id of the newly created resource.

4. Try updating and deleting the patient.

5. After deleting the patient, what response do you get when you try to get the patient again?

6. What happens when you update the patient again after the deletion?

To see which information needs to be filled with your request, look at the details of the REST operations on (http://www.hl7.org/fhir/http.html).

Solution

The solution shown below identifies the correct HTTP requests.

The fhir endpoint you can use for these exercises is:
`http://vonk.furore.com`

Exercise 2.1. Getting a patient

Use the following command to get the patient:

```
GET [fhir endpoint]/Patient/example
```

The status code of the response should be 200 if this patient exists on the chosen server.
The response headers should show the content type, the content location with the version specific url, an ETag to show the version of the resource instance and the last modified date.

To request the patient in JSON format, use

```
GET [fhir endpoint]/Patient/example?_format=json
```

or add an accept header to the request headers:

```
Accept: application/json+fhir
```

For XML, use the same syntax with '`xml`' instead of '`json`'.

Exercise 2.3. Creating a new patient

Use the following command to create a new patient:

```
POST [fhir endpoint]/Patient
```

Note that you need to supply the request with a content type, by adding this to the request headers:

```
Content-Type: application/fhir+xml
```

The body of the request should contain the (valid) XML data. For JSON content, use '`application/fhir+json`' as content type and submit valid JSON data.
The response will be 201 Created on success.

Exercise 2.4. Updating and deleting the patient

Check the response of the previous exercise to find the id of the patient you had created. Again change some data in the patient's XML. Then use the following command to update that patient:

```
PUT [fhir endpoint]/Patient/[id]
```

Again you will need to supply the content type (see exercise 2.3) and fill the body of the request with valid data. This resource data should contain the same id value.
The response will be 200 OK on success.

Delete the resource with this command:

```
DELETE [fhir endpoint]/Patient/[id]
```

The response will be 200 OK or 204 No Content on success.

Exercise 2.5. Getting the deleted patient

Use the following command to get the deleted patient:

```
GET [fhir endpoint]/Patient/[id]
```

If the server is version aware, the response you will get is 410 Gone. Some servers will also send back an OperationOutcome in the body of the response.

Exercise 2.6. Updating the deleted patient

When you update the deleted patient (see exercise 2.4 for the update command), a new version of that patient resource will be created on the server. You can retrieve a specific version of the resource using this command:

```
GET [fhir endpoint]/Patient/[id]/_history/[versionid]
```