Instructions for Deploying CADET on a local machine

Introduction:

These are instructions for downloading, standing up cadet.

***TODO* Explain CADET**

This guide requiers access to the COE's gitlab. Request access from Ben.

Downloading CADET

Install Tomcat & Maven

1. Tomcat is a tool for Java Servers. Here are instructions for downloading/setting up Tomcat on MAC, Linux, Windows

   ***TODO*: add links for linux & windows**

2. Maven is the tool we use to manage dependencies and build our java applications

Repos to clone

Creating a new directory called `cadet-home` and clone the following repos there:

1. cadet - code for the frontend-client
2. cadet-search-lucene - code for setting up search capabilities
3. docker-ingest - server for ingesting data into CADET's database (or file system)
4. docker-file-access - server for accessing data from CADET's database and for storing annotated data.

Directories for data

In `cadet-home`, create a new directory called `data`. In `data`, make the following three new directories:

1. index
2. input_data
3. storage_data

At this point your structure should look like this:

***TODO*: make a screenshot or figure of the directories**

Standing up CADET

CADET Frontend

1. **Compile and build the front-end client**. From `cadet-home/cadet`, run:

   ```
   mvn clean install
   ```

   If the installation is successful you should see the following printed out:

   ```
   [INFO] ------------------------------------------------------------------------
   [INFO] Reactor Summary:
   [INFO]
   [INFO] CADET (Parent) ..................................... SUCCESS [  6.849 s]
   [INFO] CADET broker library ............................... SUCCESS [ 13.338 s]
   [INFO] CADET command line tools ........................... SUCCESS [ 11.495 s]
   [INFO] CADET user interface ............................... SUCCESS [  9.419 s]
   [INFO] ------------------------------------------------------------------------
   [INFO] BUILD SUCCESS
   [INFO] ------------------------------------------------------------------------
   [INFO] Total time: 41.245 s
   [INFO] Finished at: 2017-09-27T15:55:48-04:00
   [INFO] Final Memory: 33M/357M
   [INFO] ------------------------------------------------------------------------
   ```

2. **Deploy war file to tomcat server**: From `cadet-home/cadet/cadet-ui`, run:

   ```
   mvn tomcat7:redeploy
   ```

   At this point, you should be unable to deploy the war file to the tomcat server because you first need to update your tomcat settings. Follow the instructions in `cadet-home/cadet/cadet-ui/tomcat.md` to update your tomcat settings.

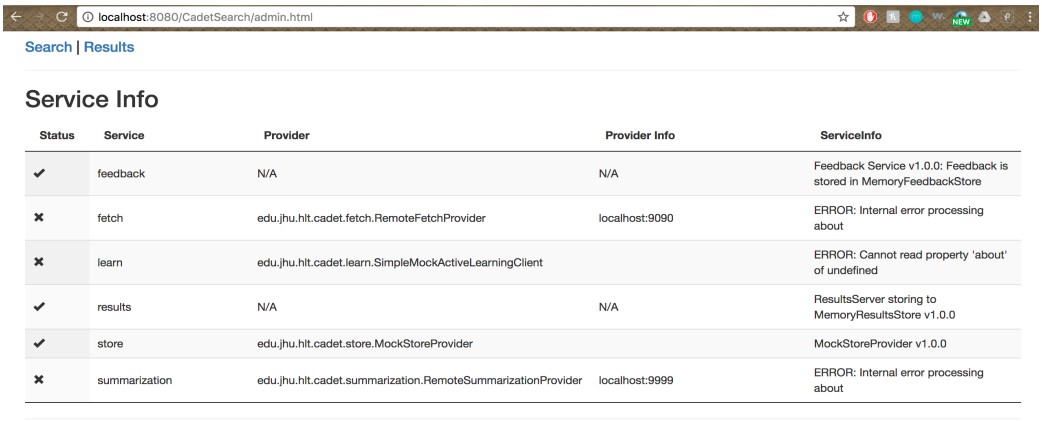
   After updating your tomcat settings and restarting the tomcat server, as described in `tomcat.md`, run this command again:

   ```
   mvn tomcat7:redeploy
   ```

   If that is successful you should see the following printed out:

```
[INFO] tomcatManager status code:200, ReasonPhrase:
[INFO] OK - Deployed application at context path [/CadetSearch]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 24.480 s
[INFO] Finished at: 2017-09-27T16:13:18-04:00
[INFO] Final Memory: 18M/200M
[INFO] ------------------------------------------------------------------------
```

Additionally, go to localhost:8080/CadetSearch/admin.html where you should see the following admin page that will give us the status & information about the different services we will now set up:



Setting up Micro-Services

Fetch and Store

TODO - briefly describe role of fetch and store

Requirements

Before starting the services, make sure you have the most up-to-date version of concrete-python. Otherwise, there will be issues later in these instructions.[1]

Starting the services

In `cadet-home/docker-file-access/scripts` run the following command:

```
./launch --path =../../data/storage_data
```

If the fetch and store services are now up and running, the following should be printed in terminal:

```
➜ ./launch --path=../../data/storage_data
INFO:root:Starting file-based FetchCommunicationService on port 9090 for dir ../../data/storage_data...
INFO:root:Starting file-based StoreCommunicationService on port 9091 for dir ../../data/storage_data...
```

Typing Ctrl+C or closing the terminal window will turn off the fetch and store services

Confirming Fetch service in the Admin UI

We will also confirm that the service is up by checking the admin page. Please refresh localhost:8080/CadetSearch/admin.html where you should now see the error resolved for the fetch service. The ServiceInfo should now specify that the fetch service is using file_fetch_serverv1.0.0 and give a brief description about it.

Status	Service	Provider	Provider Info	ServiceInfo
✔	feedback	N/A	N/A	Feedback Service v1.0.0: Feedback is stored in MemoryFeedbackStore
✔	fetch	edu.jhu.hlt.cadet.fetch.RemoteFetchProvider	localhost:9090	file_fetch_server v1.0.0: File-backed implementation of FetchCommunicationService service.
✖	learn	edu.jhu.hlt.cadet.learn.SimpleMockActiveLearningClient		ERROR: Cannot read property 'about' of undefined

TODO: highlight the difference in the picture

On your own: try stopping the fetch and store services in terminal, and then refresh the admin page to see the difference in the UI when the services are down and up

Ingest

Now that fetch and store are up and running, we need to ingest data into our database [2].

Getting Data

Download this tar file and store it in `cadet-home/data/input_data`

***TODO* Figure out which data and host that data on the nlp.jhu.edu/cadet or hltcoe.github.io/cadet site**

Ingesting Data

From `cadet-home/docker-ingest/scripts`, run the following command:

```
./communications --host localhost --port 9091 < ../../data/input_data/data.tar
```

To check that the ingester worked, peek into `cadet-home/data/storage_data` by running `ls` on that file. There should be 4 gz files there.

TODO: change the number based on the data we use in this tutorial

Search

The search micro-service repo can be found here and contains detailed instructions. For our purposes, you can just follow these instructions:

Building the search service

From `cadet-home/cadet-search-lucene`, run the following:

```
mvn clean package
```

If you do not have access to the HLTCOE's maven server, you will need to manually install all the jars. One quick solution is to just do the following for these dependencies in `cadet-home/cadet-search-lucene/pom.xml`

1. concrete-services-4.14.1: remove "-SNAPSHOT" in the pom
2. concrete-lucene-4.14.1: remove "-SNAPSHOT" in the pom

Now running the command to clean and package cadet-search-lucene should work as desired.
***TODO* figure out best way to get the dependencies that are currently being worked on**

Running the search service

To start the search service run:

```
./start.sh --fh localhost --fp 9090 -d ../data/index/ -r -b -p 8888
```

---

1. Ask Baekchun Kim about that one ;)↩
2. Adam P.: we are using a file system to mimic a database - gotta love best NLP practices↩