Back and Front end project

Guillaume Becan, Johann Bourcier, David Bromberg

4 mai 2016

1 PROJECT

The main aim of this project is to build a file sharing and storing service by leveraging on existing cloud storage services. More precisely, you must develop a `REST` based service that enables to store seamlessly files on at least both `DropBox` and `GoogleDrive` cloud storage See Figure 1.

This project is based on two different parts :

➠ A `REST` service based on either the `Jersey` or `RestEasy` Java based framework that provide a set of accessible resources to manage cloud storage based files. Each call to your own exposed `REST` APIs will be mapped to APIs provided by `DropBox` and `GoogleDrive` as illustrated in Figure 1.

➠ A client side that enables to explore files stored *via* your own API. It is advised to develop the client in `HTML, CSS & JS`. However, you are free to implement it in the programming language you want. However, the following technologies are recommended : `Angular JS V1 or V2, WebSocket, Material, Bootstrap`.

Your application must at least implement the underlying list of features :

➠ A user must be able to authentify once, whatever the underlying cloud storage space. The `REST` service must do authentification for the behalf of the user.

Base revision 11762bc, Tue Apr 14 11:20:53 2015 +0200, David Bromberg.

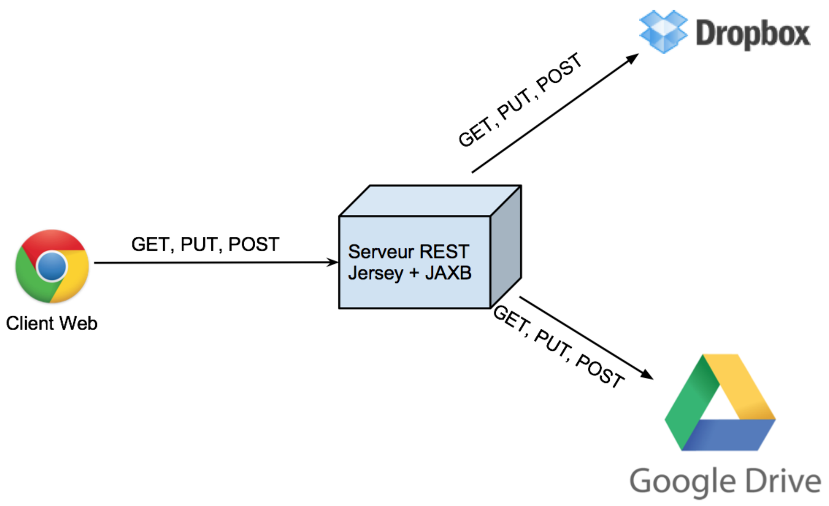

FIGURE 1 – Project overview

- The client side must provide a unique file explorer unifying file coming from both `DropBox` and `GoogleDrive`.
- If two folders share a similar name on either `DropBox` and `GoogleDrive`, they are merged.
- The user should move, rename, remove add files whatever the underlying cloud storage.
- The user should see in an intuitive manner details of a file (size, rights, shares).
- The user should share any files to an other user.
- The user should be able to show the available space, updated dynamically.
- You must use adequately persistency at the service side to cache data get from the different cloud storage services.
- Your application must show the file hierarchy of the different drives.
- The user should be notified dynamically about any modifications ofthe file hierarchy.
- The user must be able to upload files whatever the underlying storage.
- The user should have a dashboard showing all interesting information about its storage accounts.

Advanced Features. Each project must compete and must have advanced features. More features there are, better will be the mark.

A list of non exhaustive possible features :

- Taking in charge other cloud service storage such as `Microsoft OneDrive`, `Amazon S3`, ...
- Having different strategy of merging when folders from different cloud storage have the same name.
- Deploying your `REST` service to `Heroku` to ease its deployment and test. A `maven` plugin can be used for this purpose.
- Managing file revisions
- Enabling to crypt files
- Notifying files changes
- Enabling notifications between online users about shared files modifications.
- Any other ideas will be welcomed ... :-)

Important notes. **The project must be done without using the provided Java SDK from neither Dropbox and Google.** It means that you must use only core API provided by `HTTP`. For instance, for `DropBox` https://www.dropbox.com/developers/core/docs, and for `GoogleDrive` https://developers.google.com/drive/v2/reference/. The Java code must managed by the `Maven` tool project management. Further, a `GitHub` account must be used to manage versioning of the project.

Organization. The project should be done in a group of two students only.

Requirements. The Java code part must be managed by maven. Further, you must provide an archive of your source code cleaned (without compiled classes). The archive will include also a report that describes :

- The structure of your project.
- Your Rest API.
- frameworks and library used.
- Short user manual.
- Features implemented and not implemented.
- Advanced features implemented.
- Difficulties encountered.
- ...