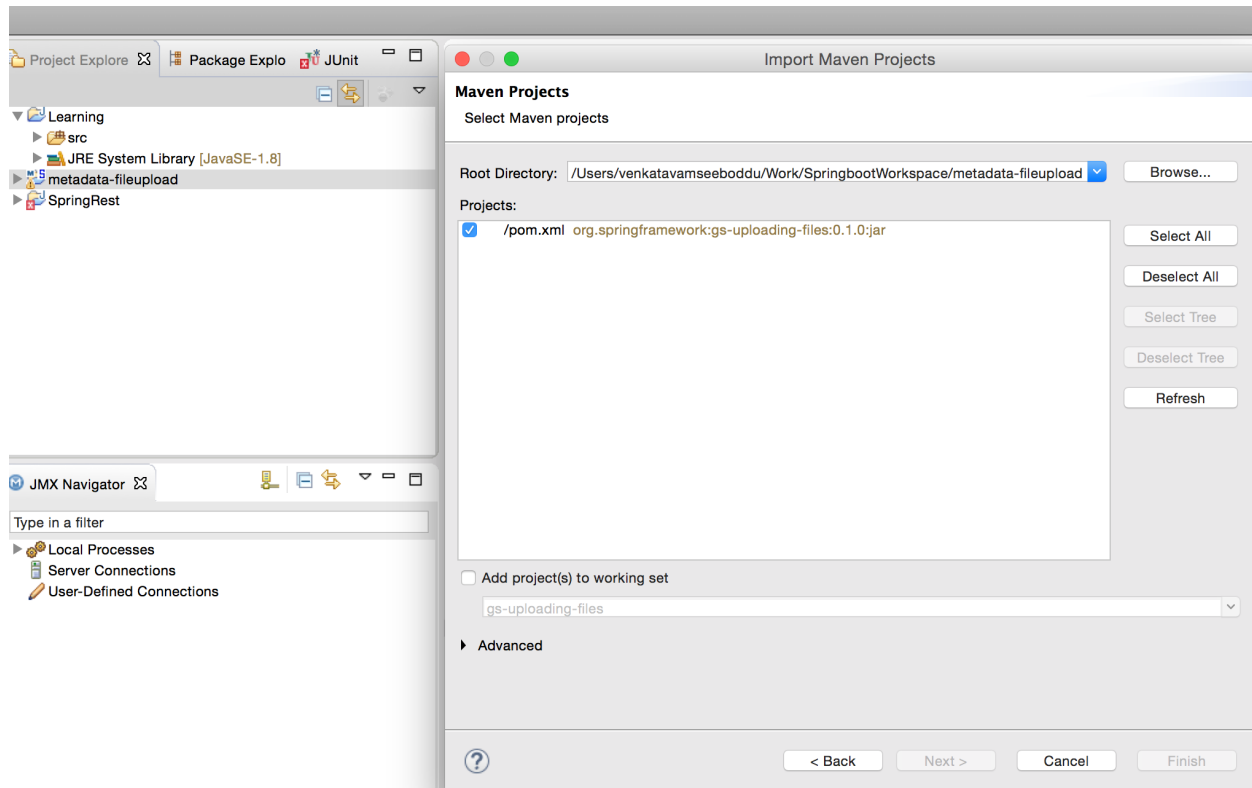


1. Download and extract the zip file from Git
2. Import the project as existing Maven project



3. In the terminal, navigate to the project workspace, run: **mvn package**, then **mvn spring-boot:run** to bring up the server and container

```
metadata-fileupload — bash — 204x53
bash
Venkatas-MacBook-Pro:~ venkatavamseeboddu$ cd /Users/venkatavamseeboddu/Work/SpringbootWorkspace/metadata-fileupload
Venkatas-MacBook-Pro:metadata-fileupload venkatavamseeboddu$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building gs-uploading-files 0.1.0
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ gs-uploading-files ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ gs-uploading-files ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 13 source files to /Users/venkatavamseeboddu/Work/SpringbootWorkspace/metadata-fileupload/target/classes
[WARNING] /Users/venkatavamseeboddu/Work/SpringbootWorkspace/metadata-fileupload/src/main/java/fileupload/metadata/FileMetadataStorageService.java: /Users/venkatavamseeboddu/Work/SpringbootWorkspace/metad
ata-fileupload/src/main/java/fileupload/metadata/FileMetadataStorageService.java uses unchecked or unsafe operations.
[WARNING] /Users/venkatavamseeboddu/Work/SpringbootWorkspace/metadata-fileupload/src/main/java/fileupload/metadata/FileMetadataStorageService.java: Recompile with -Xlint:unchecked for details.
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ gs-uploading-files ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/venkatavamseeboddu/Work/SpringbootWorkspace/metadata-fileupload/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ gs-uploading-files ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.18.1:test (default-test) @ gs-uploading-files ---
[INFO]
[INFO] --- maven-jar-plugin:2.6:jar (default-jar) @ gs-uploading-files ---
[INFO] Building jar: /Users/venkatavamseeboddu/Work/SpringbootWorkspace/metadata-fileupload/target/gs-uploading-files-0.1.0.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:1.5.3.RELEASE:repackage (default) @ gs-uploading-files ---
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.907 s
[INFO] Finished at: 2017-07-21T14:49:32-04:00
[INFO] Final Memory: 33M/387M
[INFO]
Venkatas-MacBook-Pro:metadata-fileupload venkatavamseeboddu$ mvn spring-boot:run
```

4. Upload the files as below with URL: **<http://localhost:8080/upload>**

Note: Metadata **must** contain Key with name : **id** which is considered as file id; the corresponding value is taken for File ID value. If another file is uploaded with same Id value, it will be overwritten in the cache.

Environment: No Environment

Method: POST URL: http://localhost:8080/upload

Authorization: Headers: **Body** Pre-request Script: Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
file	Choose Files manifest.xml	
id	value1	
id2	value2	
id3	value3	
New key	Value	Description

Body Cookies Headers (3) Tests Status: 200 OK Time: 24 ms

Pretty Raw Preview Text

1 You successfully uploaded manifest.xml

Environment: No Environment

Method: POST URL: http://localhost:8080/upload

Authorization: Headers: **Body** Pre-request Script: Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description
file	File Choose Files PointsToPonder.rtf	
id	key1	
id2	key2	
id3	key3	
New key	Value	Description

Body Cookies Headers (3) Tests Status: 200 OK Time: 37 ms

Pretty Raw Preview Text

1 You successfully uploaded PointsToPonder.rtf

5. Search file details/metadata by FileID **<http://localhost:8080/list/key1>**

The screenshot shows a REST client interface with the following details:

- URL bar: `http://localhost:8080/` (selected), `http://localhost:8080/upload`, `http://localhost:8080/list`, `http://localhost:8080/files/r`, `http://localhost:8080/fileID:`, and a plus sign.
- Method: `GET`
- Path: `http://localhost:8080/list/key1`
- Params: Empty
- Buttons: `Send` (blue), `Save` (grey)
- Authorization: `No Auth`
- Body tab: Selected, showing a JSON response in Pretty view:

```
1 {
2   "id1": "key1",
3   "id2": "key2",
4   "id3": "key3"
5 }
```
- Status: `200 OK`, Time: `14 ms`

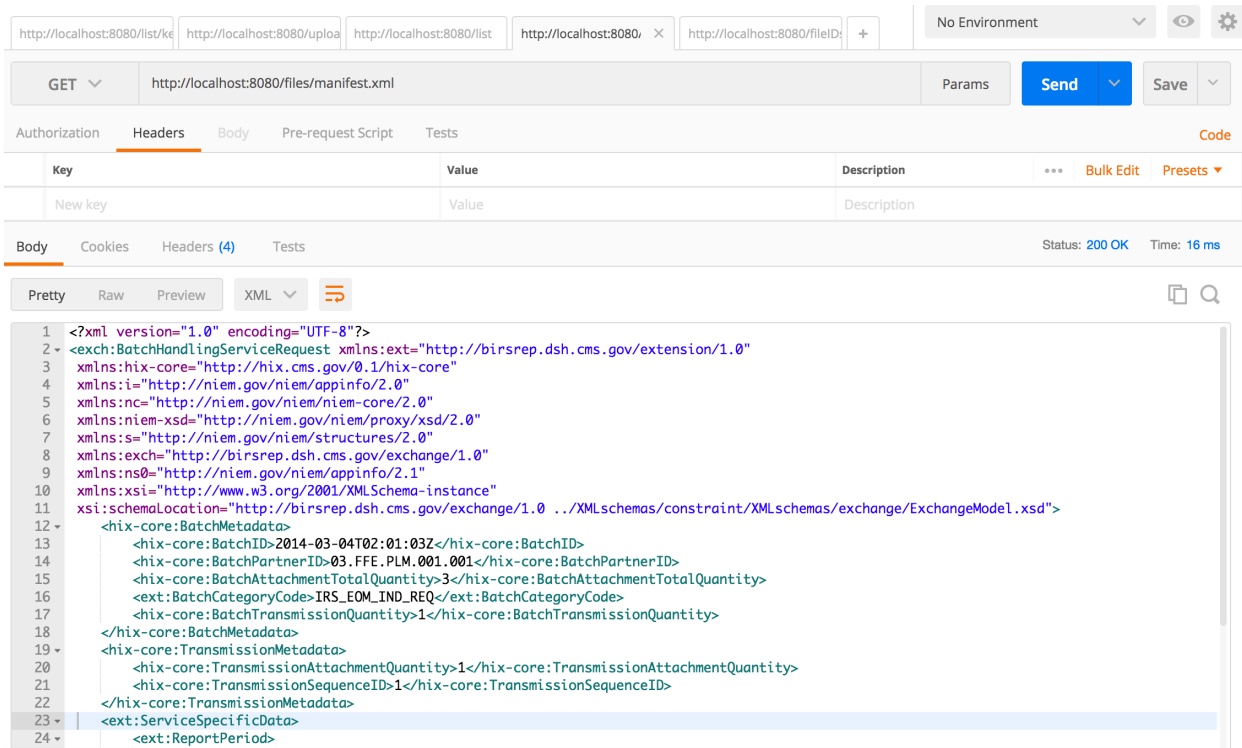
6. Get the list of all file metadata **<http://localhost:8080/list>**

The screenshot shows a REST client interface with the following details:

- URL bar: `http://localhost:8080/list/ke` (selected), `http://localhost:8080/upload`, `http://localhost:8080/` (with a close button), `http://localhost:8080/files/r`, `http://localhost:8080/fileID:`, and a plus sign.
- Method: `GET`
- Path: `http://localhost:8080/list`
- Params: Empty
- Buttons: `Send` (blue), `Save` (grey)
- Authorization: `No Auth`
- Body tab: Selected, showing a JSON response in Pretty view:

```
1 [
2   {
3     "id1": "key1",
4     "id2": "key2",
5     "id3": "key3"
6   },
7   {
8     "id1": "value1",
9     "id2": "value2",
10    "id3": "value3"
11  }
12 ]
```
- Status: `200 OK`, Time: `15 ms`

7. Stream the file by file name. <http://localhost:8080/files/manifest.xml>

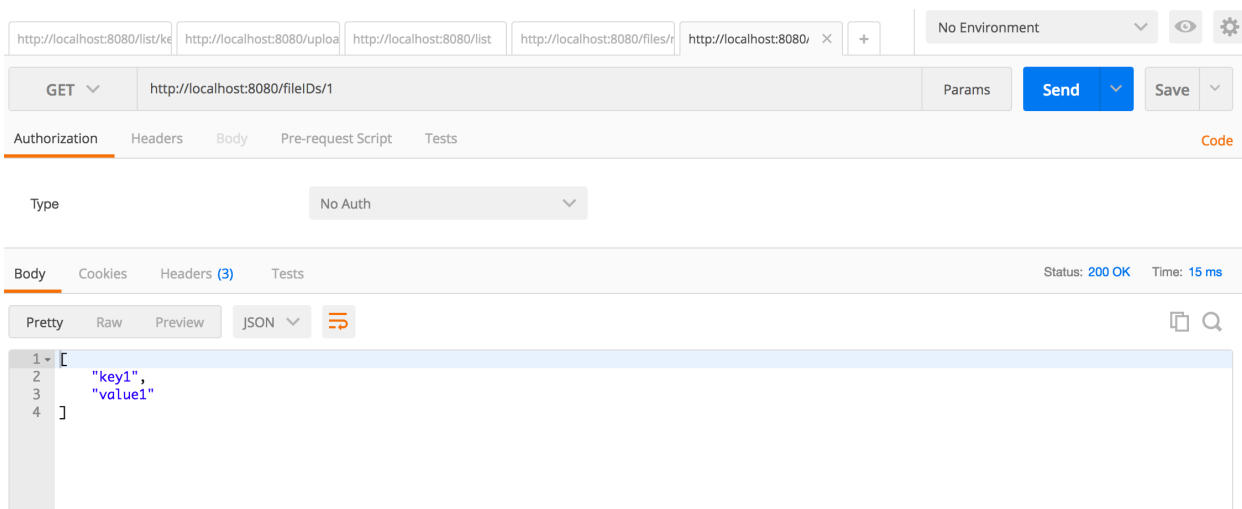


The screenshot shows a web browser interface with the URL `http://localhost:8080/files/manifest.xml` in the address bar. The browser displays the XML response in a 'Pretty' format. The XML is a `BatchHandlingServiceRequest` with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<exch:BatchHandlingServiceRequest xmlns:ext="http://birsrep.dsh.cms.gov/extension/1.0"
  xmlns:hix-core="http://hix.cms.gov/0.1/hix-core"
  xmlns:i="http://niem.gov/niem/appinfo/2.0"
  xmlns:nc="http://niem.gov/niem/niem-core/2.0"
  xmlns:niem-xsd="http://niem.gov/niem/proxy/xsd/2.0"
  xmlns:s="http://niem.gov/niem/structures/2.0"
  xmlns:exch="http://birsrep.dsh.cms.gov/exchange/1.0"
  xmlns:ns0="http://niem.gov/niem/appinfo/2.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://birsrep.dsh.cms.gov/exchange/1.0 ../XMLschemas/constraint/XMLschemas/exchange/ExchangeModel.xsd">
  <hix-core:BatchMetadata>
    <hix-core:BatchID>2014-03-04T02:01:03Z</hix-core:BatchID>
    <hix-core:BatchPartnerID>03.FFE.PLM.001.001</hix-core:BatchPartnerID>
    <hix-core:BatchAttachmentTotalQuantity>3</hix-core:BatchAttachmentTotalQuantity>
    <ext:BatchCategoryCode>IRS_EOM_IND_REQ</ext:BatchCategoryCode>
    <hix-core:BatchTransmissionQuantity>1</hix-core:BatchTransmissionQuantity>
  </hix-core:BatchMetadata>
  <hix-core:TransmissionMetadata>
    <hix-core:TransmissionAttachmentQuantity>1</hix-core:TransmissionAttachmentQuantity>
    <hix-core:TransmissionSequenceID>1</hix-core:TransmissionSequenceID>
  </hix-core:TransmissionMetadata>
  <ext:ServiceSpecificData>
    <ext:ReportPeriod>
```

8. Match the file IDs by the criterion of wild card search.

<http://localhost:8080/fileIDs/1>



The screenshot shows a web browser interface with the URL `http://localhost:8080/fileIDs/1` in the address bar. The browser displays the JSON response in a 'Pretty' format. The JSON is an array containing a single object with the following structure:

```
[
  {
    "key1": "value1"
  }
]
```