Apex Library for Box API

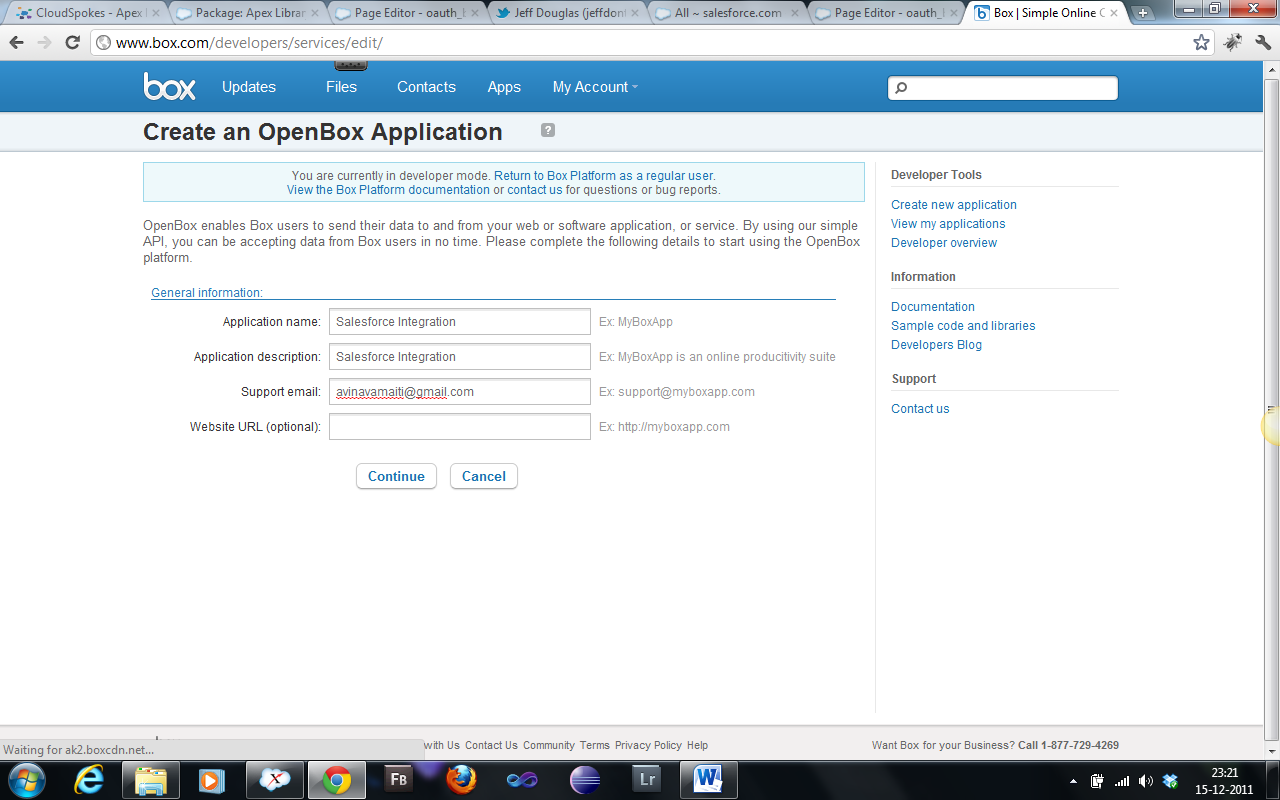
Installation URL: <https://login.salesforce.com/packaging/installPackage.apexp?p0=04t900000009SS0>

# Setup:

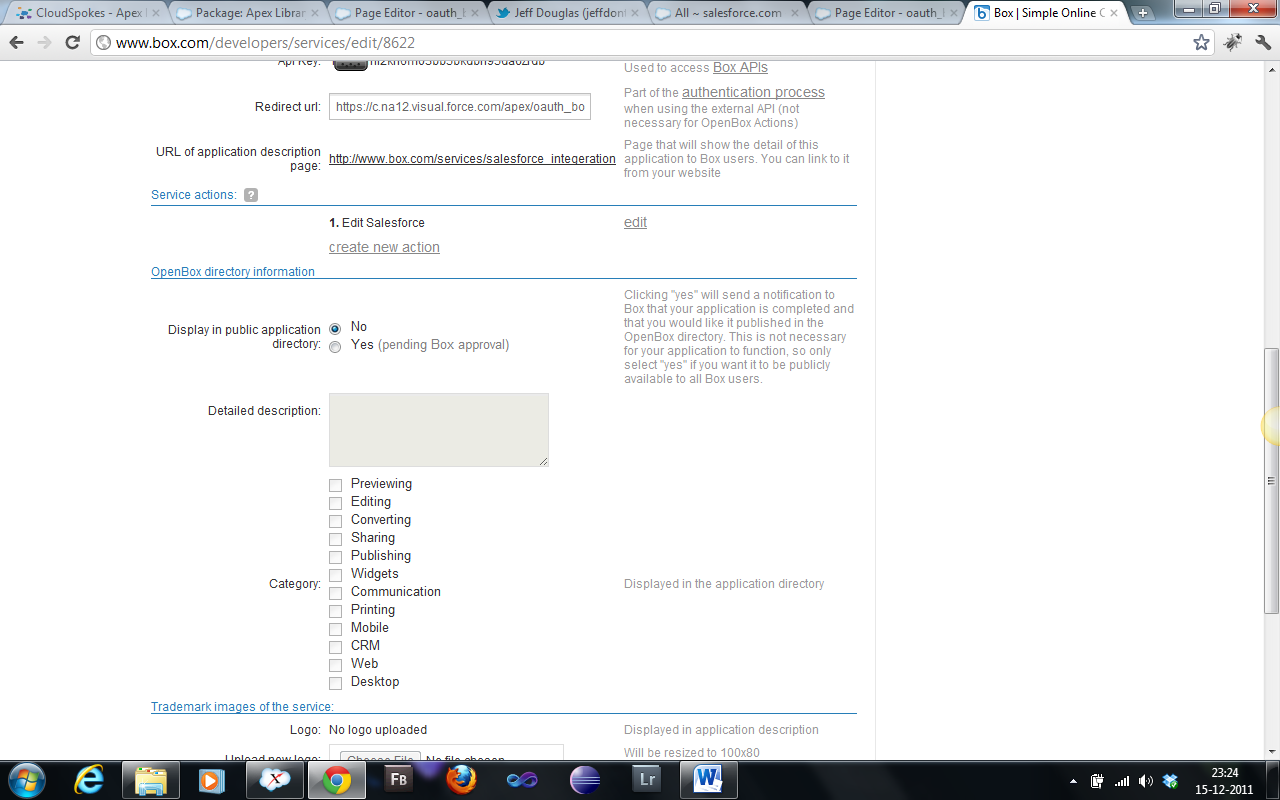
* Install the unmanaged package
* **Make sure API key from box.com is entered in the custom setting “BoxConnection”. Put the Name of the record as “Box.com” and enter the API key in Api\_Key\_\_c field.**

# Obtaining API key from box.com

* **Go to** <http://www.box.com/developers/services> and login using your username and password.
* From developer tools select **“Create new Applicaiton”** and accept the agreement.

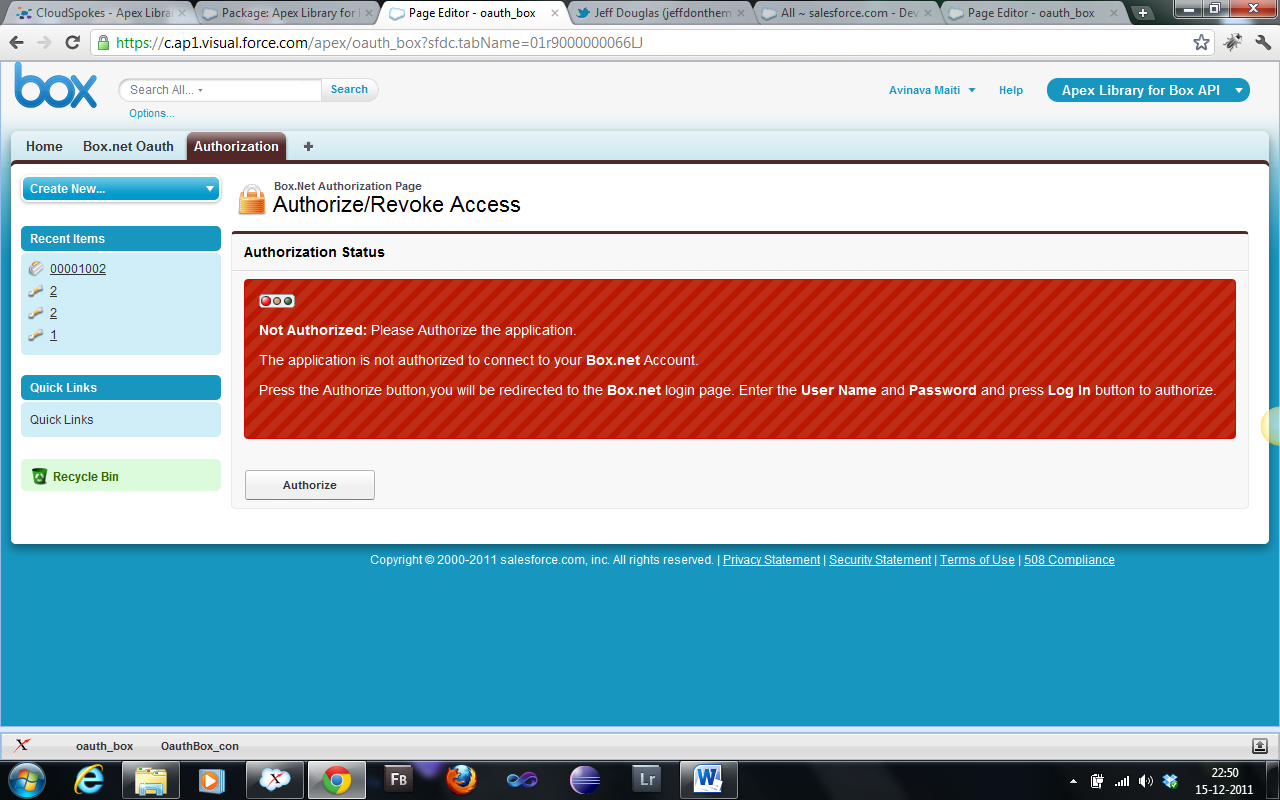


* Upon clicking continue on the very next page you can see that your api key is generated.
* Now click edit application and look for “**Redirect url**” and fill up the url of the authorization page which will be something like **“https://c.ap1.visual.force.com/apex/oauth\_box”**

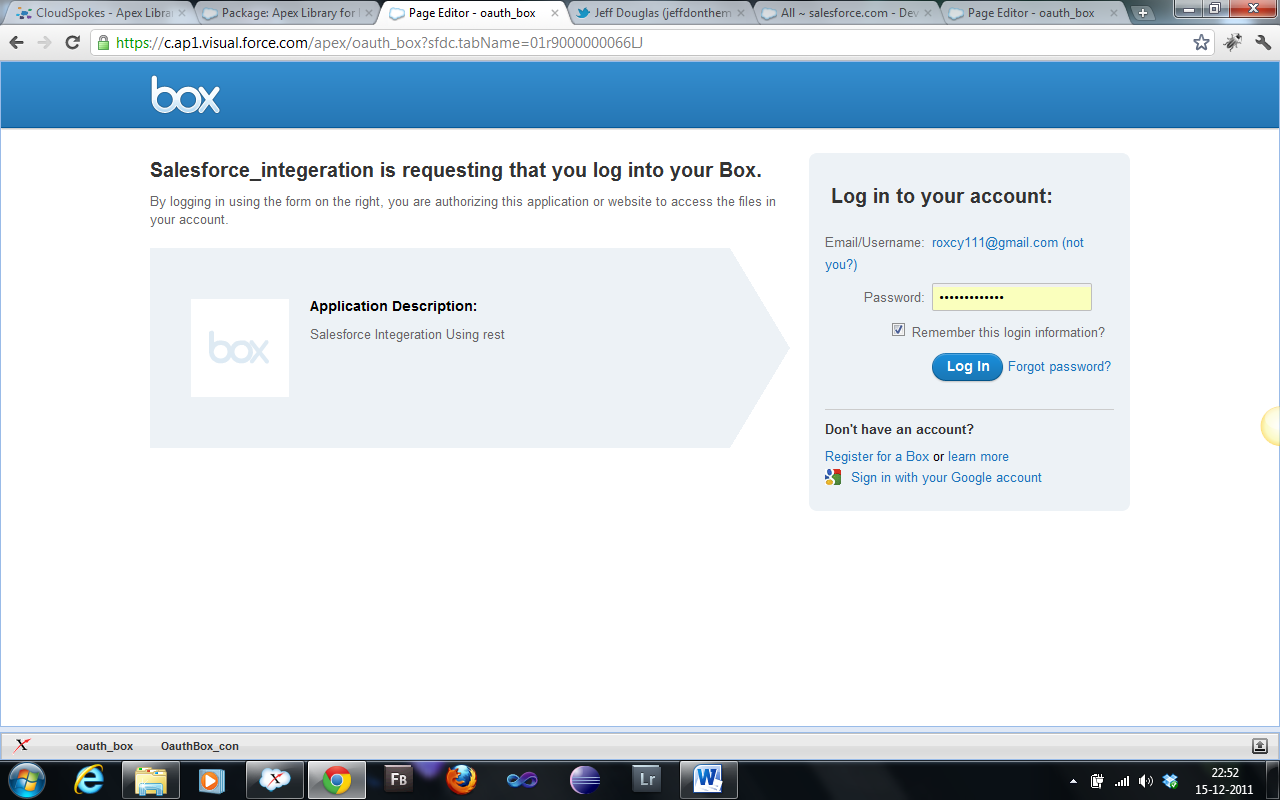


# Running the Application

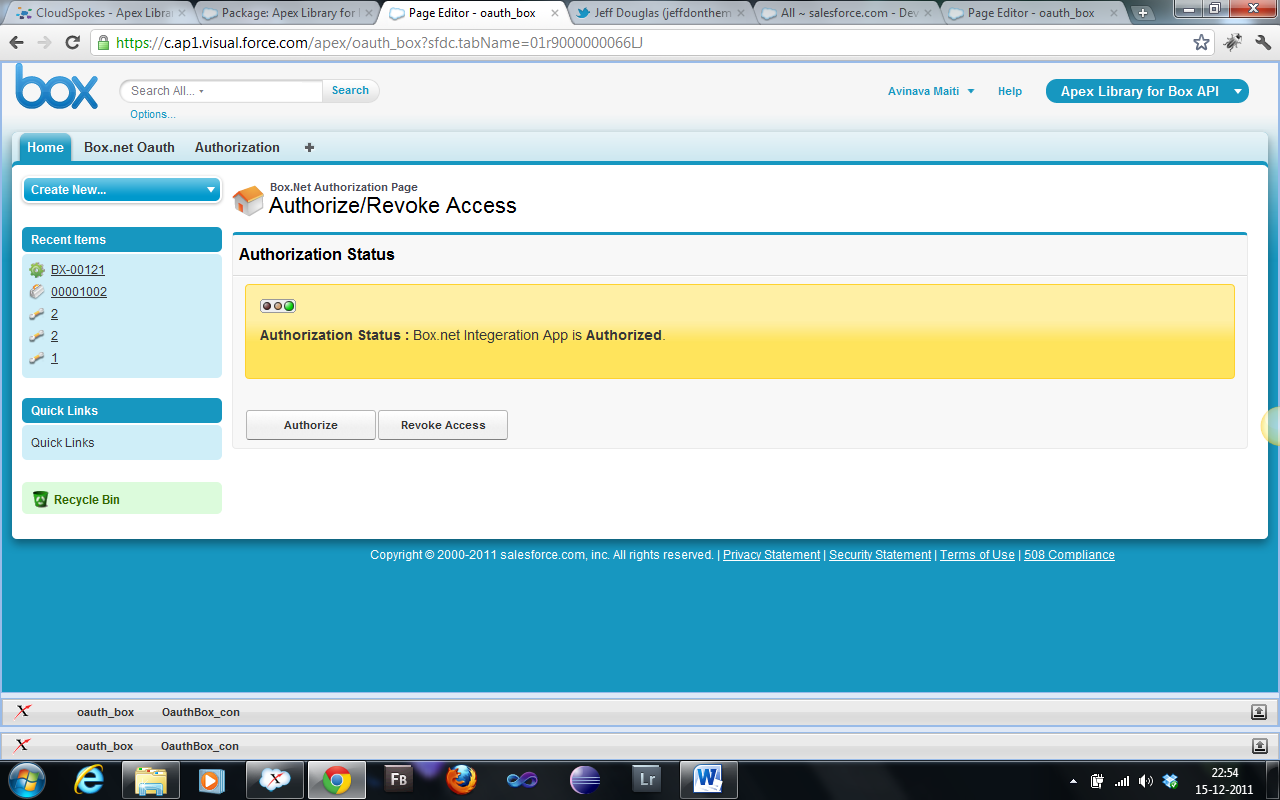
* For this challenge I have created a force.com App namely “Apex Library for Box API”, select the app and you are good to go!
* The landing page of the application will show you the current status of your connection with Box.net. Initially it will say that it is unauthorized.



* To authorize the application click on the authorize button you will be taken to the box.net login page.



* And press the login button and you will be redirect back to Salesforce



* Now you ready to make calls to box.com. you can try uploading doc to the box.net by running the following command

BoxToolkit box = new BoxToolkit();

box.uploadDocumentById(‘<documentId>’,’0’,’upload’);

Running this code in the system log will upload the document with provided Salesforce document id to the box.net folder whose id is ‘0’ (i.e. root folder) and the third parameter states that it is a upload action.

Similarly you can download a document by running the following code. **Currently the document that is uploaded from Salesforce will only be readable if they are downloaded using API calls from box.net to salesforce, due to limitation of Salesforce. Salesforce doesn’t support binary encoded string whereas box.net doesn’t accept base64 encoded data.**

BoxToolkit box = new BoxToolkit();

Document doc=Box. downloadAsDocument(‘<box.net file id>’);

Insert doc;

Running this code will download the specified file from box.net. Since I have inserted the doc by default the file can be found in the user’s folder. Similarly you can try with other calls. All the methods by default return XML that is returned from box.net.

# Some other examples:

**Getting the folder tree:**

* **To get the folder tree of a account from say folder “0”**

BoxToolkit box = new BoxToolkit();

Box. getFolderTree(‘0’);

* **To get the full folder tree of a account including files from say folder “0”**

BoxToolkit box = new BoxToolkit();

Box. getFullFolderTree(‘0’);

**Explore the BoxToolkit class it is duly commented what each method and parameters does**