

# Box Summit/Hackathon

## Notes from the Organizers

We are looking forward to seeing you next week. Spending a little time prepping ahead of the hackathon can pay dividends so we ask that before you arrive you take some time to complete some basics listed below:

### Get Acquainted with your Fellow Attendees:

1. Join the Summit Slack ([https://join.slack.com/t/box-community-summit/shared\\_invite/enQtMzgwMzIxODE3NDkwLTU0MDA5ZjBhM2Y4MTU1YzZjYTawZGRhOThhMDE5NzZINWY3MjkkN2I1YjlxYjZmMjYwZjA0ZDBhNmY5YTgzNmM](https://join.slack.com/t/box-community-summit/shared_invite/enQtMzgwMzIxODE3NDkwLTU0MDA5ZjBhM2Y4MTU1YzZjYTawZGRhOThhMDE5NzZINWY3MjkkN2I1YjlxYjZmMjYwZjA0ZDBhNmY5YTgzNmM)) and the #Hackathon channel
2. Take the pre-Hackathon survey [https://docs.google.com/forms/d/e/1FAIpQLSdaBON0JPGJKLC5B22voXOfOTNq7iaN5Va8lhN3x3cxAWF5MQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdaBON0JPGJKLC5B22voXOfOTNq7iaN5Va8lhN3x3cxAWF5MQ/viewform?usp=sf_link). Please complete by EOB on Friday. Results will be posted in a working folder and to the #Hackathon Slack channel.

### Prepare to Code:

1. Sign up for a Box Developer account (<https://account.box.com/signup/n/developer#sudh3>) using an email address not associated with your current school Box account.
2. If you don't already have one, sign up for a github account (<https://github.com/>)
3. Take a look at the Box Developer Quickstart Guides (<https://developer.box.com/docs/quickstart-guides>)
4. Get familiar with the the Box API documentation (<https://developer.box.com/docs>)
5. Take a look at existing community code (<https://github.com/box-community>)

It's also important to know that there are some general guidelines that we ask everyone participating to follow throughout the hackathon:

- Be kind, listen to others ideas and treat others with respect.
- During the hackathon we ask that you do not develop directly on your Enterprise Box instance, use a developer instance as a best practice.
- Please do not develop against University data. You never know what might happen to the data that you are developing on, including deleting or unintentionally sharing.
- Share your work. Building on the work of others is great. We ask that you share your work with the community in GitHub - giving back helps the community move forward.
- Have fun!

## Our Submission

General overview.

1. User logs into Brown Dog application with OAuth, using their box credentials
2. We create a FILE\_UPLOADED web hook from the Brown Dog app to one of the user's folders. The action URL of this web hook points back to an endpoint of the Brown Dog App.
3. User logs out and closes brown dog
4. User uploads a file to their monitored folder
5. Brown Dog is notified with file ID and JWT
6. App pushes a message onto Rabbit queue
  - a. Updated Rabbit Message payload includes the repository type (Box)
  - b. The Box API endpoint to call including the file id
  - c. Credentials to authenticate the call to the box api
7. Extractor picks up message and uses the Box connector to obtain the file, call process\_message and upload metadata

## Implementation

For simplicity, I've decided to use a modified ClusterMan app since it is a modern Play application that already has OAuth support. My code can be found in the [bdbox](#) branch.

I've set up a VM at SDSC called <https://www.bdbox.ndslabs.org> - I need to have a real https server for Box to allow the url as a web hook target.

Need to find a couple of compelling extractors. I'm assuming open-cv since I'm working on them right now

## Enable PKCS Encryption in your Java:

For Mac OS this is easy. <https://stackoverflow.com/questions/37741142/how-to-install-unlimited-strength-jce-for-java-8-in-os-x>

```
cd $JAVA_HOME/jre/lib/security/  
sudo vi java.security
```

insert

```
crypto.policy=unlimited
```

## Tasks

- ☐ Figure out how to use the JWT auth to impersonate a user in the extractor using the [as-user header](#)
- ☐ Add RabbitMQ integration to ClusterMan for posting messages to an extractor's queue. (We currently have a management API Client, but I don't think we can use that to submit messages to a queue)
- ☐ Add a box connector to pycrowder2 and build some extractor containers. Figure out a new rabbitMQ message that will provide all of the info for the BoxConnector