CMPE 273-Lab 2

Dropbox

By

ADITYA PARMAR

Student ID – 011819964

Introduction:

- To developed Data storage like Dropbox where User can save their data and share with others.
- Use Kafka for fault tolerance and high throughput. Use PassportJS for security and persistent session.

System Design:

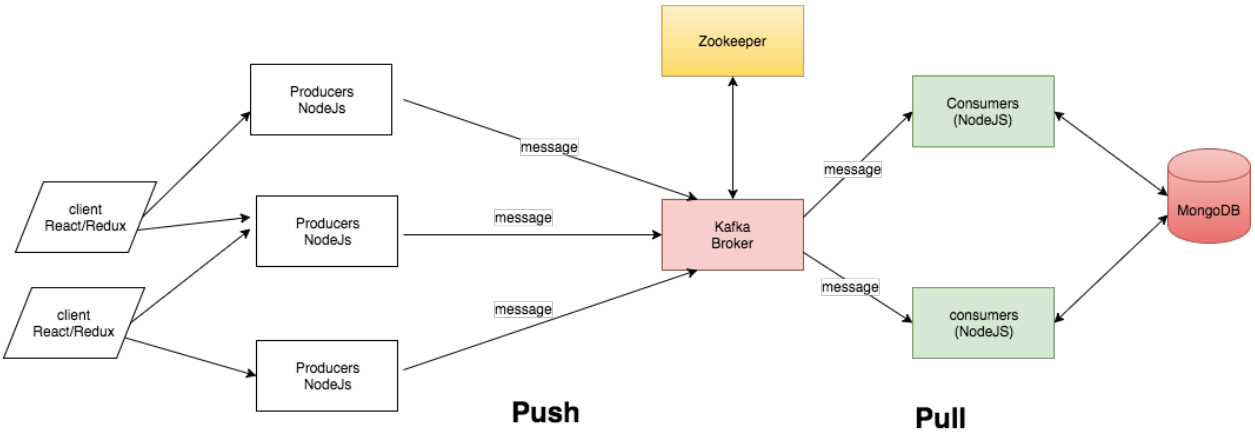
Client –

- Client is developed in React-Redux as JavaScript framework and HTML,CSS and Bootstrap for attractive UI.
- It interacts with Users and HTTP requests to sever as per user's requirements.
- React – Redux wonderfully update components.

Server –

- The "Dropbox" application server handles the requests sent by the browser i.e. Client to the server.
- The server here consists of two components – Kafka and Node.js. The requests sent by the Client is received by Node server. The Functionality of node server is to send those requests to Kafka in form of messaging queue to process the received requests.
- The requests are sent to Kafka in form of topics along with the message. The task of Kafka is to process those requests sent by the Node Backend.
- Kafka server interacts with the Database i.e. Mongo DB in these case in order to process the requests like insertion, updating, delete.
- Now, after Kafka process the request, it becomes the producer to send the response back to Node backend. In these case, Kafka and Node backend becomes the producer and consumer simultaneously.
- After the requests are received by the Node backend, it sends the required response back to Client in JSON.
- This is how the whole architecture is created and implemented wherein Kafka acts as a middleware between the requests and processes it in form of messaging queues.

Architecture Diagram



FrontEnd

BackEnd

Testing

Database

User

Interface

Dropbox

Create an Account

First name

Last name

Email

Password

Create an Account

Sign In

localhost:3000/signin

Dropbox

Create an Account

Ad

nadm a

mnc mnas

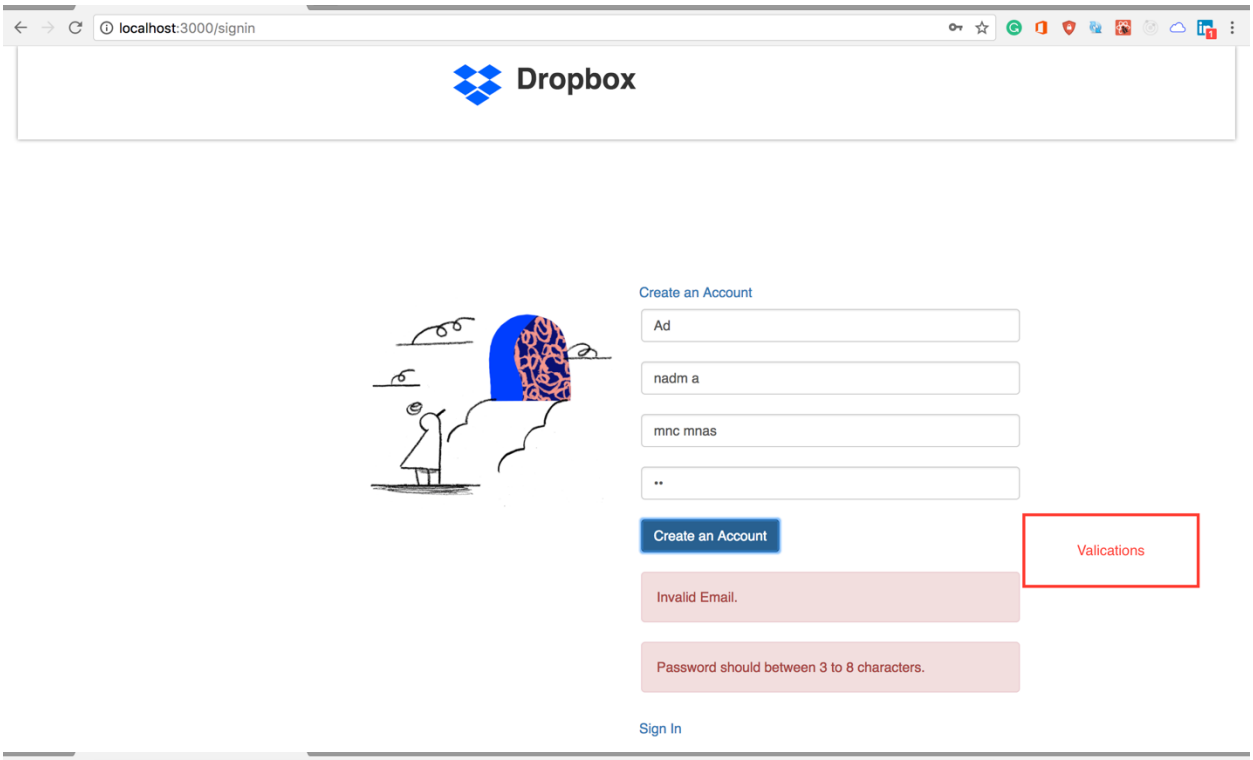
••

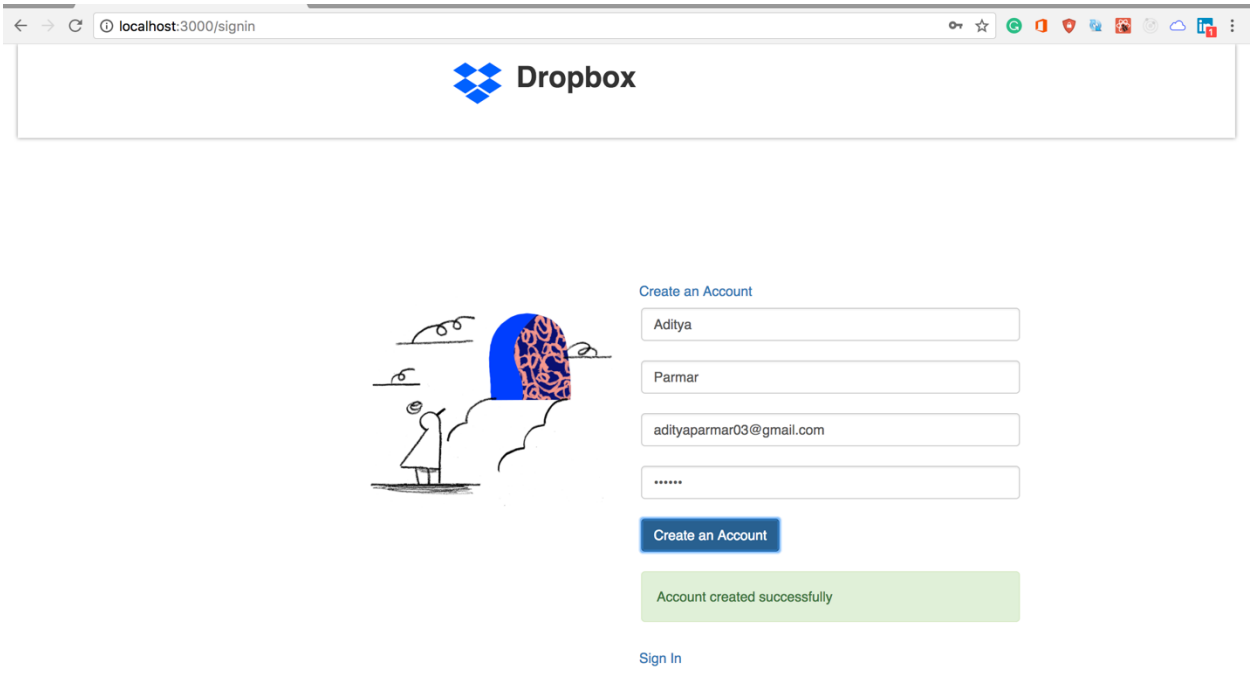
Create an Account

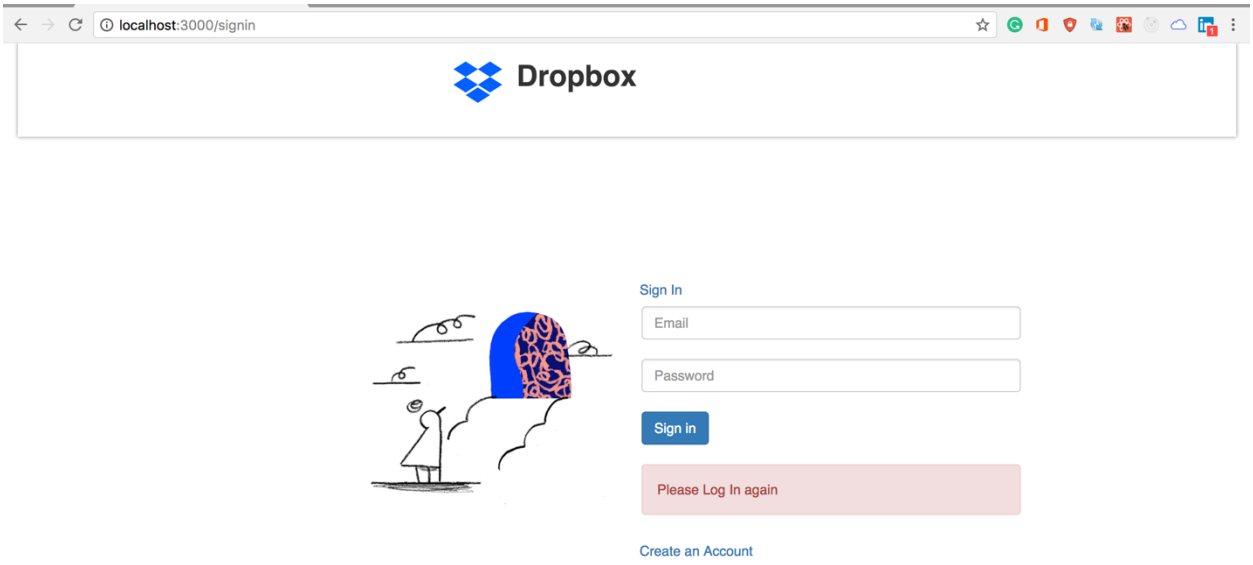
Invalid Email.

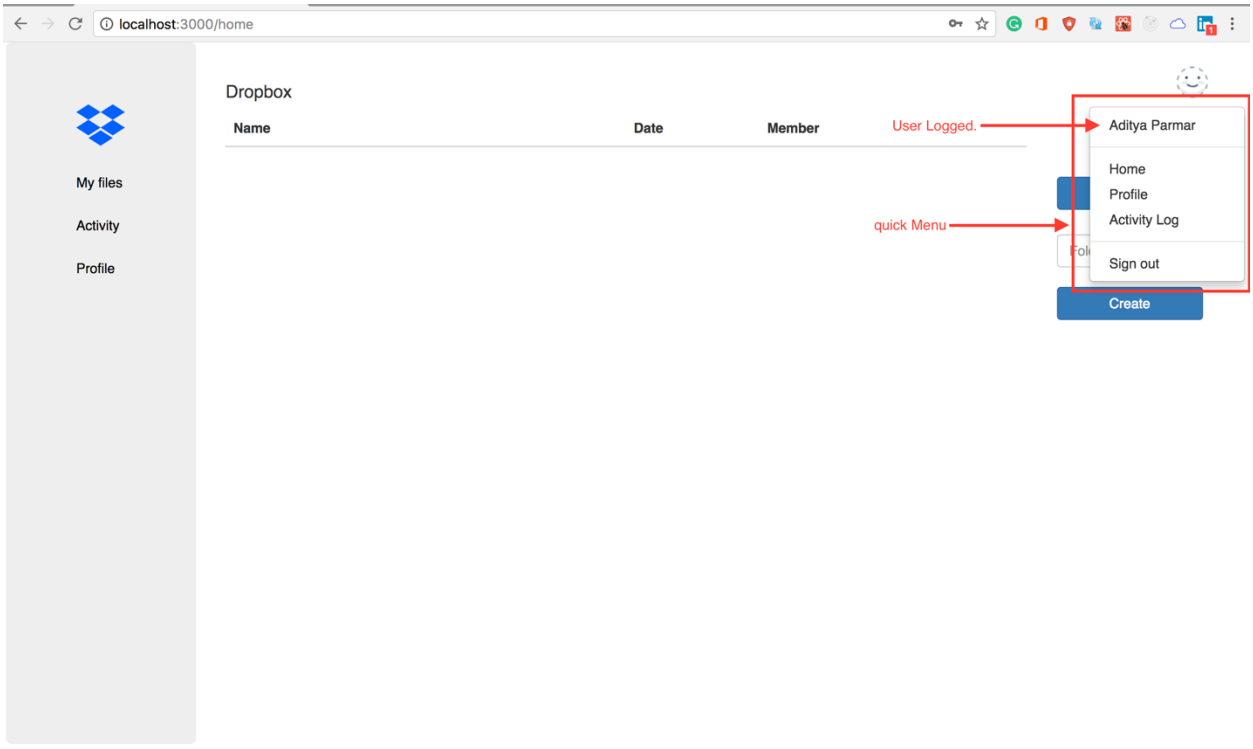
Password should between 3 to 8 characters.

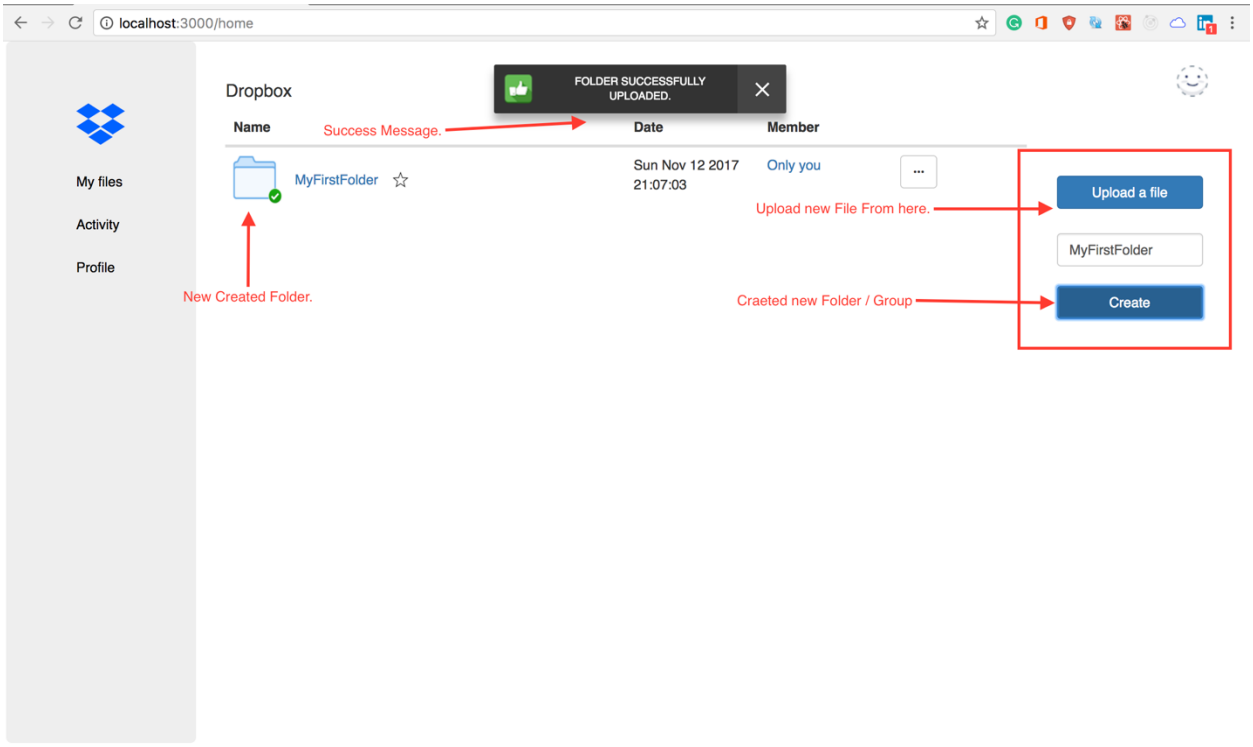
Sign In

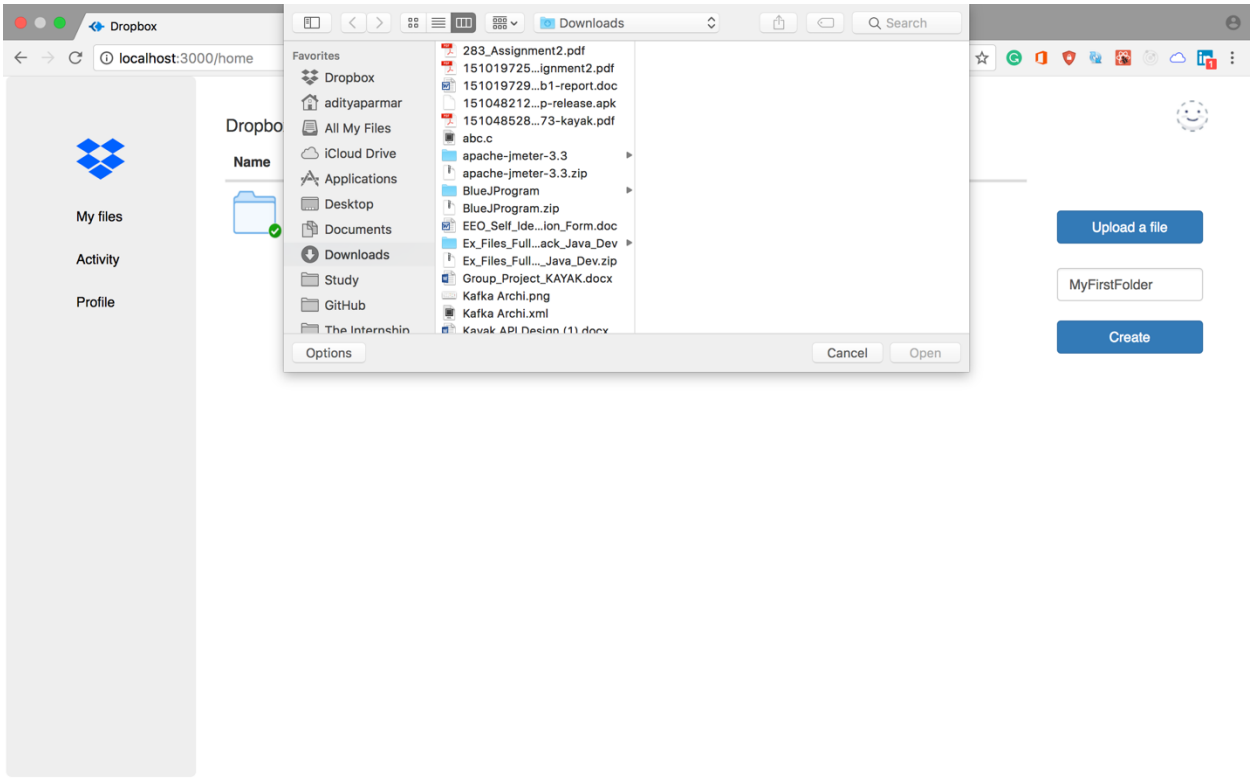

localhost:3000/signin
Dropbox
Create an Account
Ad
nadm a
mnc mnas
••
Create an Account
Valications
Invalid Email.
Password should between 3 to 8 characters.
Sign In

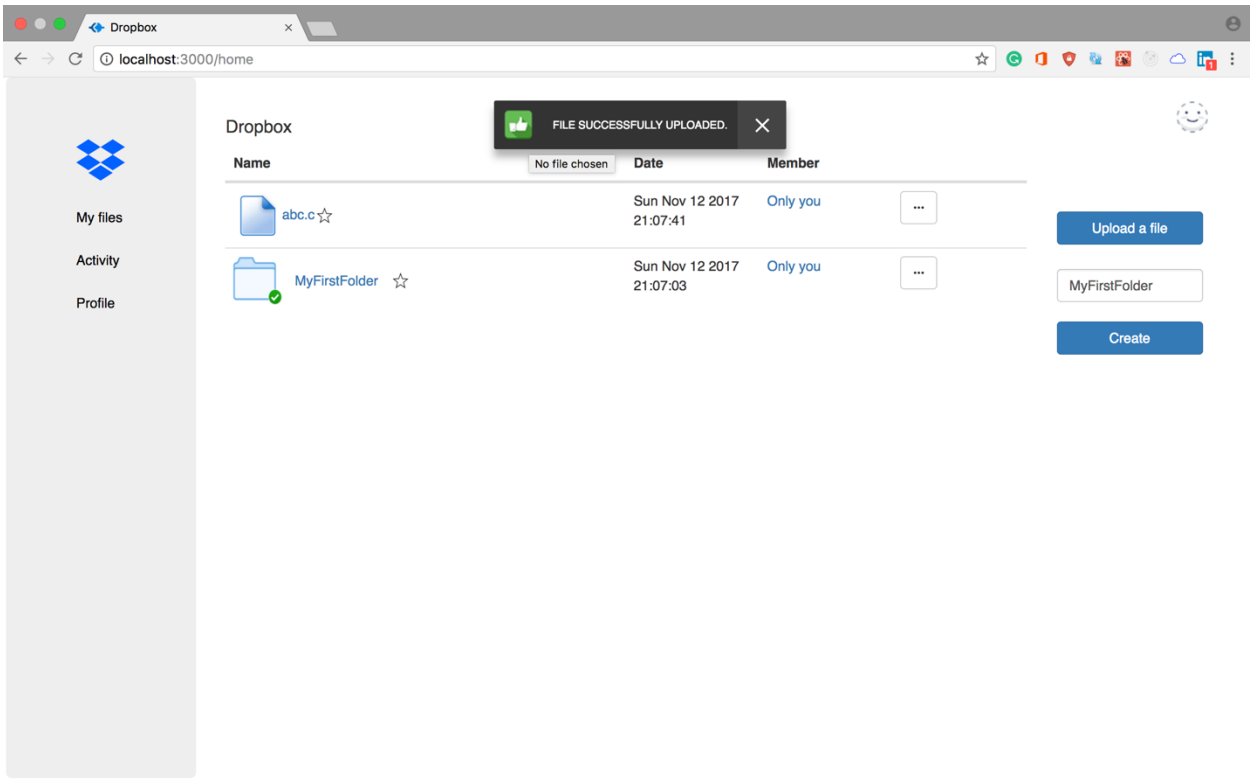

localhost:3000/signin
Dropbox
Create an Account
Aditya
Parmar
adityaparmar03@gmail.com
••••••
Create an Account
Account created successfully
Sign In

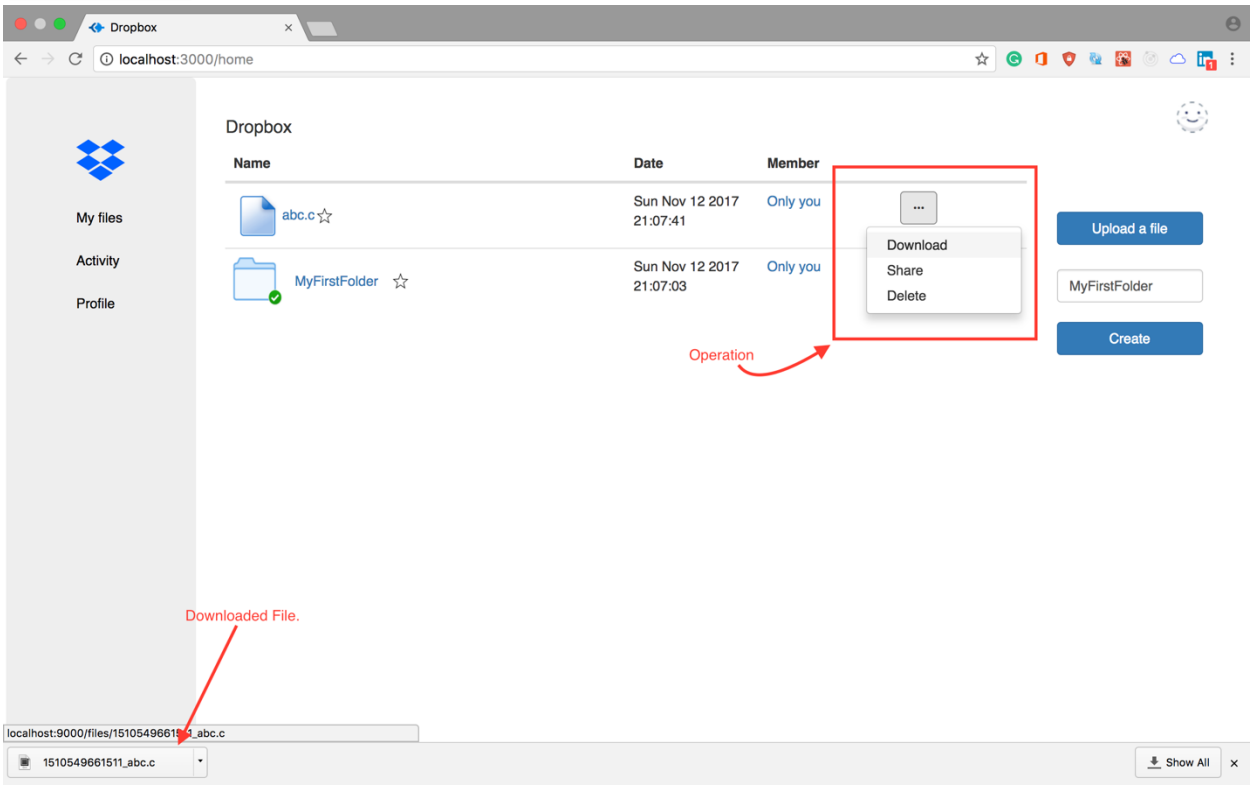

localhost:3000/signin
Dropbox
Sign In
Email
Password
Sign in
Please Log In again
Create an Account

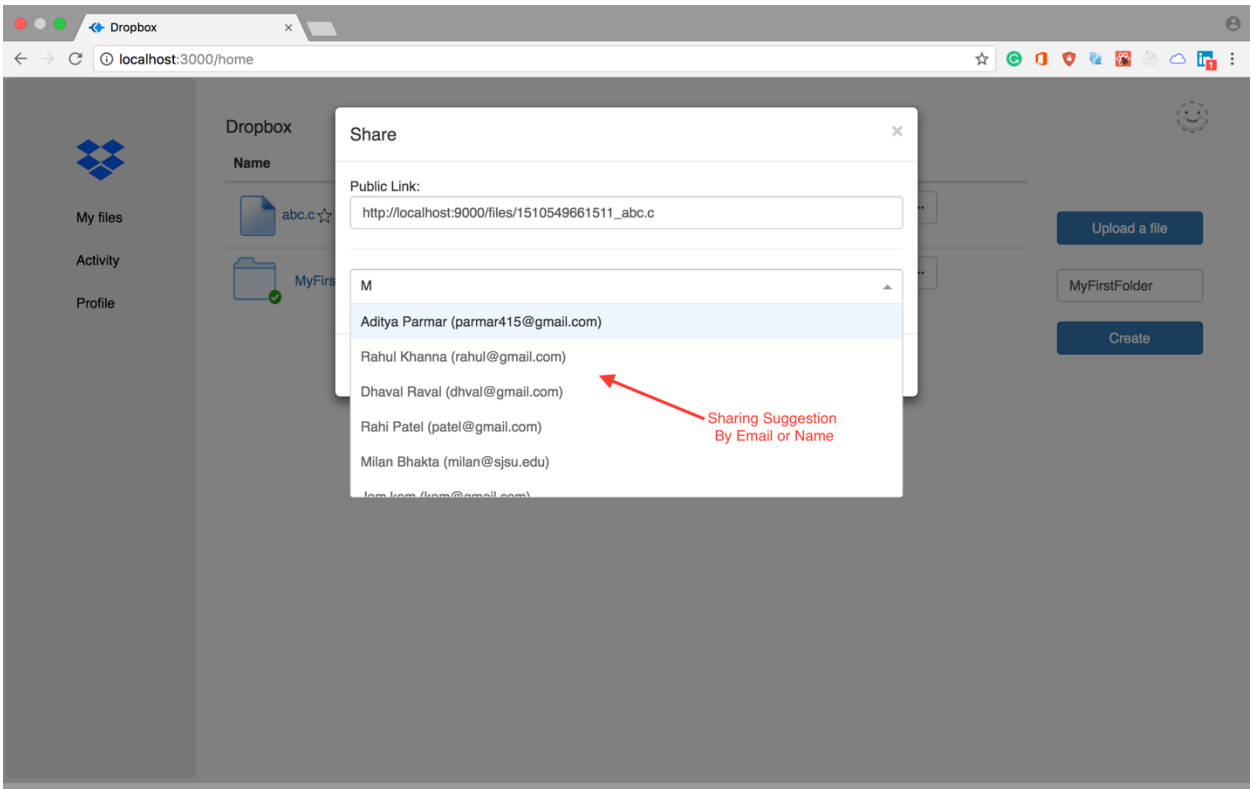

localhost:3000/home
Dropbox
My files
Activity
Profile
Name
Date
Member
User Logged.
quick Menu
Aditya Parmar
Home
Profile
Activity Log
Sign out
Create

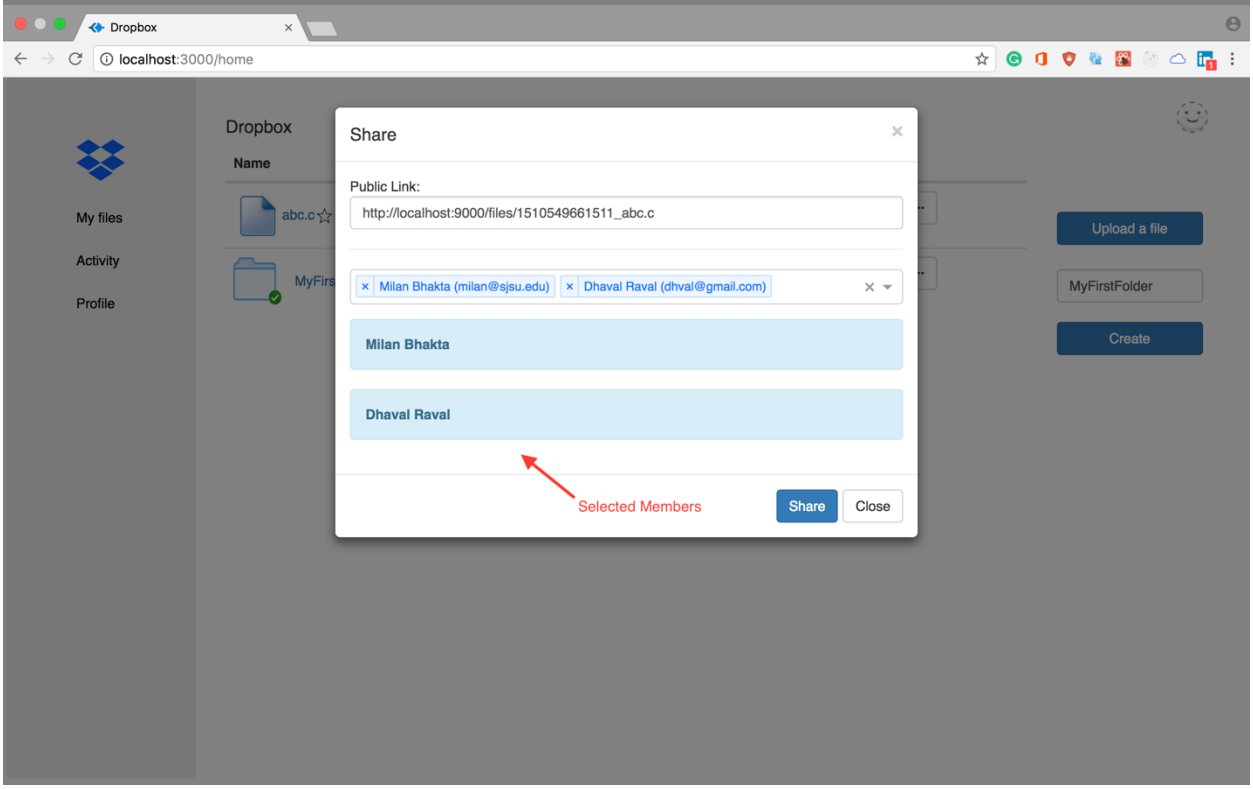

localhost:3000/home
Dropbox
FOLDER SUCCESSFULLY UPLOADED.
Name
Success Message.
Date
Member
My files
Activity
Profile
MyFirstFolder
Sun Nov 12 2017 21:07:03
Only you
Upload new File From here.
Upload a file
MyFirstFolder
Create
New Created Folder.
Craeted new Folder / Group

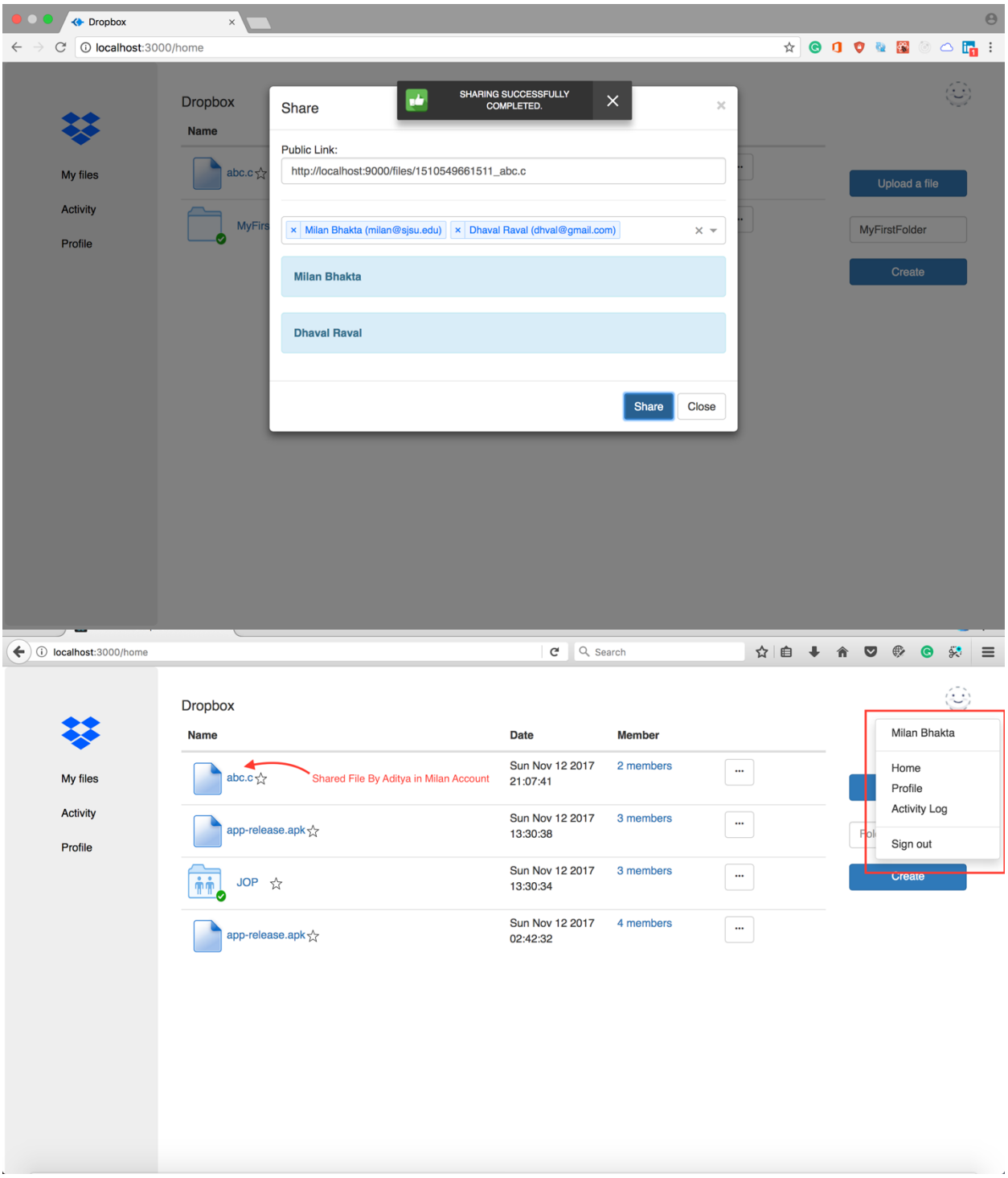

Dropbox
localhost:3000/home
Downloads
Search
Favorites
Dropbox
adityaparmar
All My Files
iCloud Drive
Applications
Desktop
Documents
Downloads
Study
GitHub
283_Assignment2.pdf
151019725...ignment2.pdf
151019729...b1-report.doc
151048212...p-release.apk
151048528...73-kayak.pdf
abc.c
apache-jmeter-3.3
apache-jmeter-3.3.zip
BlueJProgram
BlueJProgram.zip
EEO_Self_Ide...ion_Form.doc
Ex_Files_Full...ack_Java_Dev
Ex_Files_Full..._Java_Dev.zip
Group_Project_KAYAK.docx
Kafka Archi.png
Kafka Archi.xml
Options
Cancel
Open
Name
My files
Activity
Profile
Upload a file
MyFirstFolder
Create

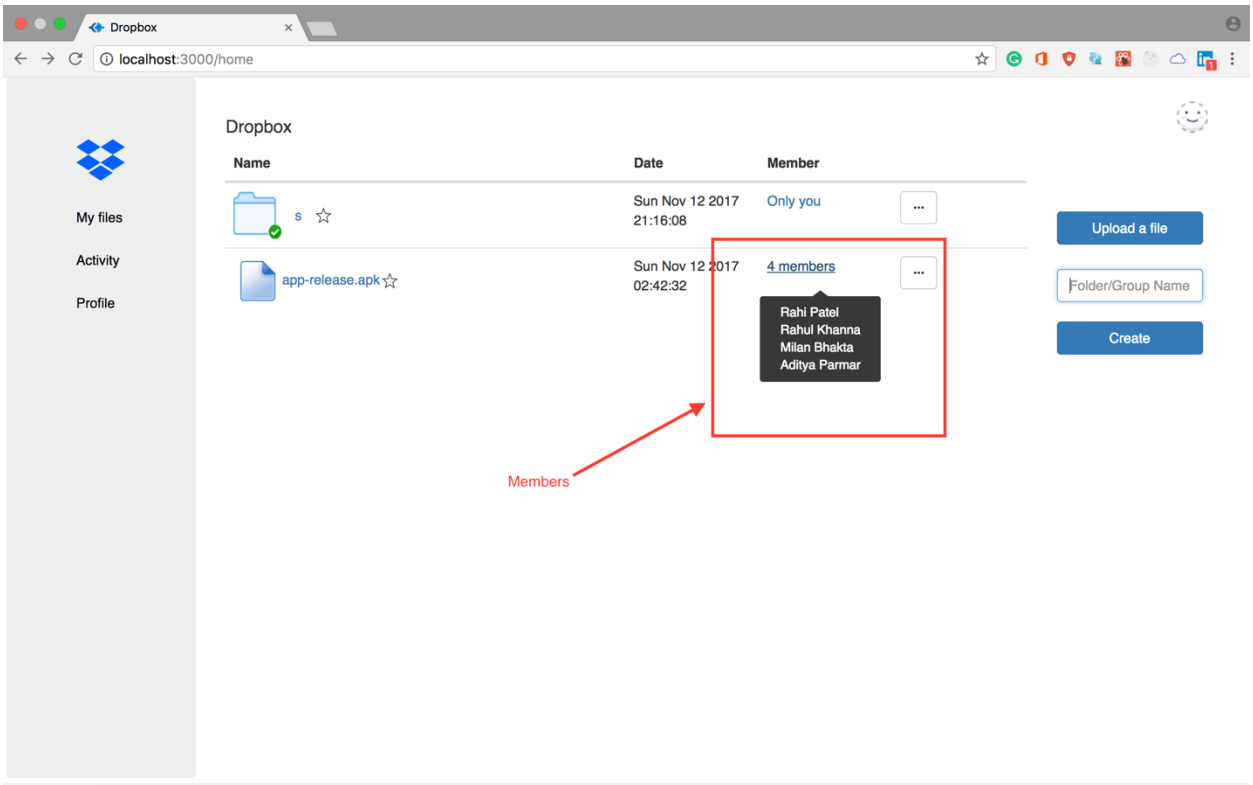

Dropbox
localhost:3000/home
Dropbox
FILE SUCCESSFULLY UPLOADED.
No file chosen
My files
Activity
Profile
Name
Date
Member
abc.c
Sun Nov 12 2017 21:07:41
Only you
MyFirstFolder
Sun Nov 12 2017 21:07:03
Only you
Upload a file
MyFirstFolder
Create

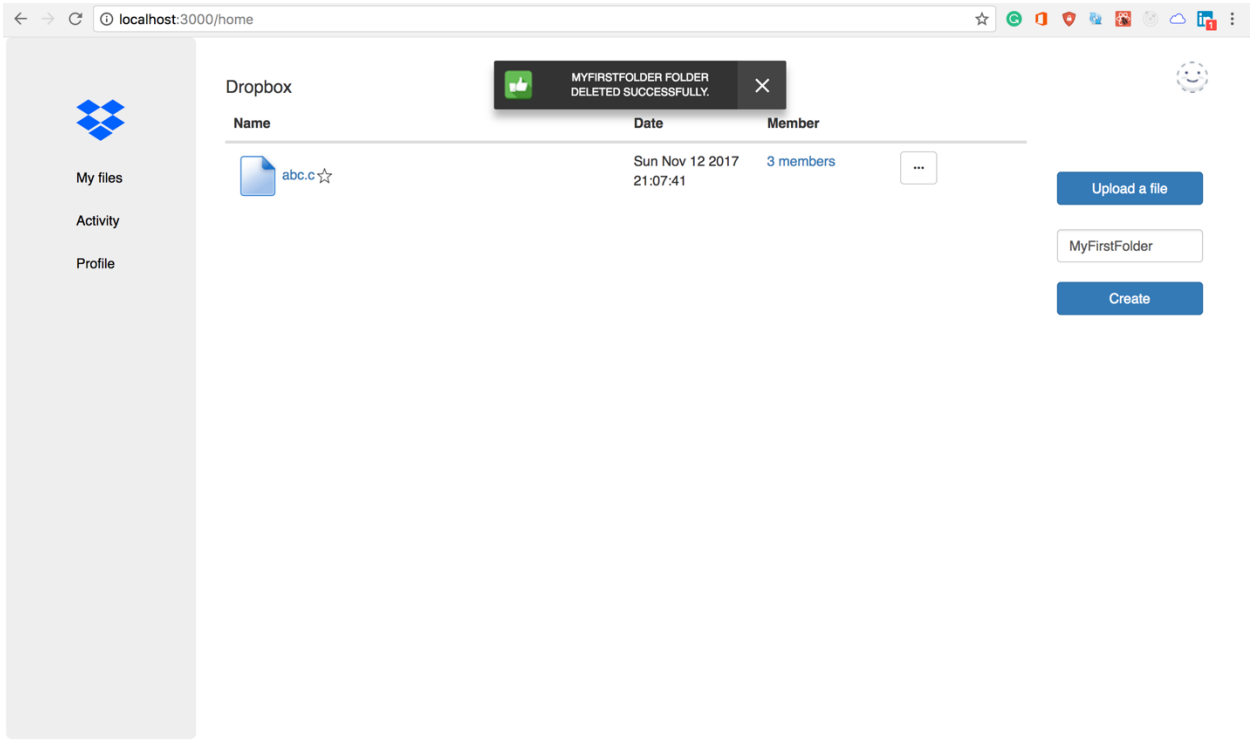

Dropbox
localhost:3000/home
Dropbox
My files
Activity
Profile
Name
Date
Member
abc.c
Sun Nov 12 2017 21:07:41
Only you
MyFirstFolder
Sun Nov 12 2017 21:07:03
Only you
Download
Share
Delete
Upload a file
MyFirstFolder
Create
Operation
Downloaded File.
1510549661511_abc.c
Show All

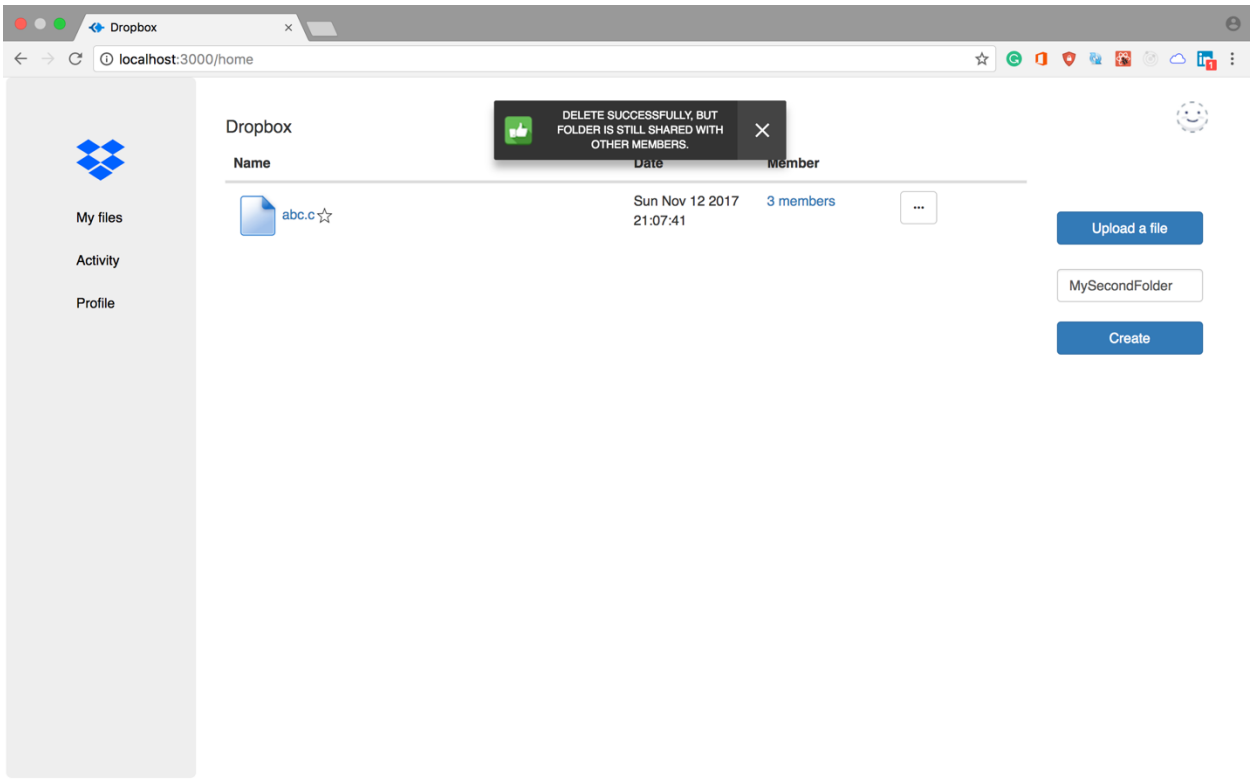

Dropbox
localhost:3000/home
Dropbox
Name
My files
Activity
Profile
abc.c
MyFirs
Upload a file
MyFirstFolder
Create
Share
Public Link:
http://localhost:9000/files/1510549661511_abc.c
M
Aditya Parmar (parmar415@gmail.com)
Rahul Khanna (rahul@gmail.com)
Dhaval Raval (dhval@gmail.com)
Rahi Patel (patel@gmail.com)
Milan Bhakta (milan@sjsu.edu)
Sharing Suggestion
By Email or Name

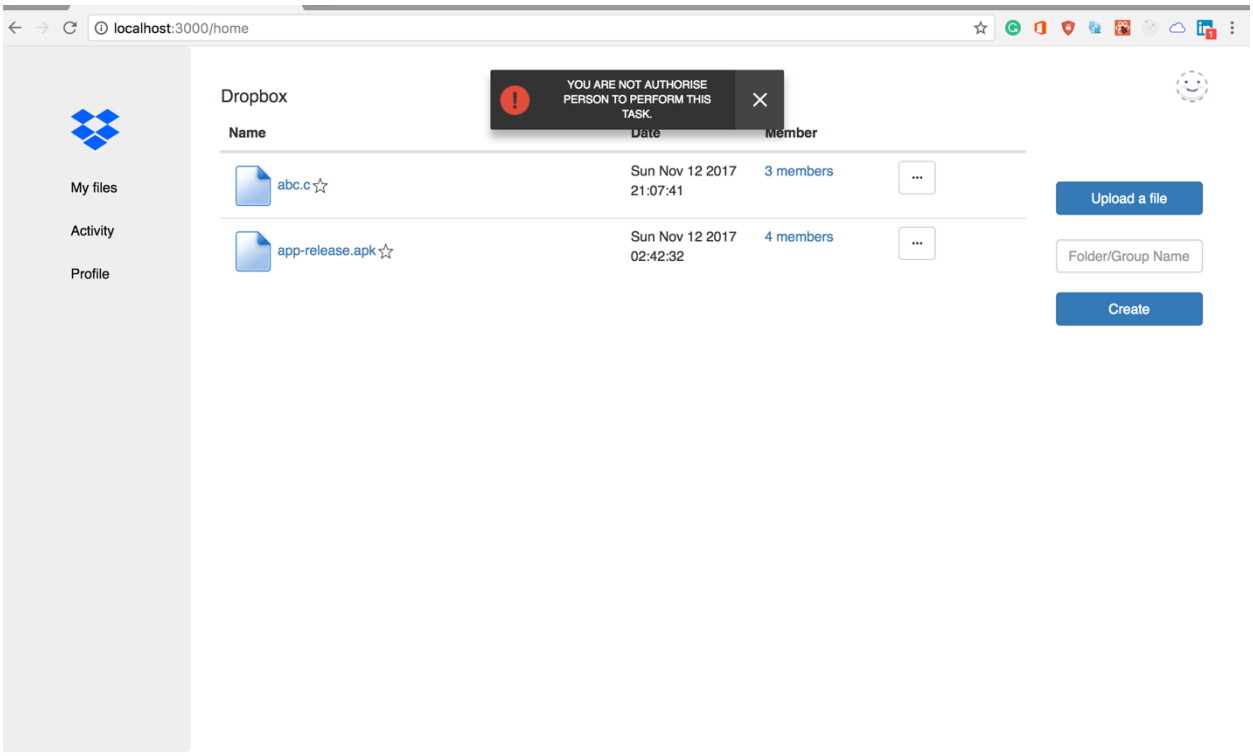

Dropbox
localhost:3000/home
Dropbox
Name
My files
Activity
Profile
abc.c
MyFirs
Upload a file
MyFirstFolder
Create
Share
Public Link:
http://localhost:9000/files/1510549661511_abc.c
Milan Bhakta (milan@sjsu.edu)
Dhaval Raval (dhval@gmail.com)
Milan Bhakta
Dhaval Raval
Selected Members
Share
Close

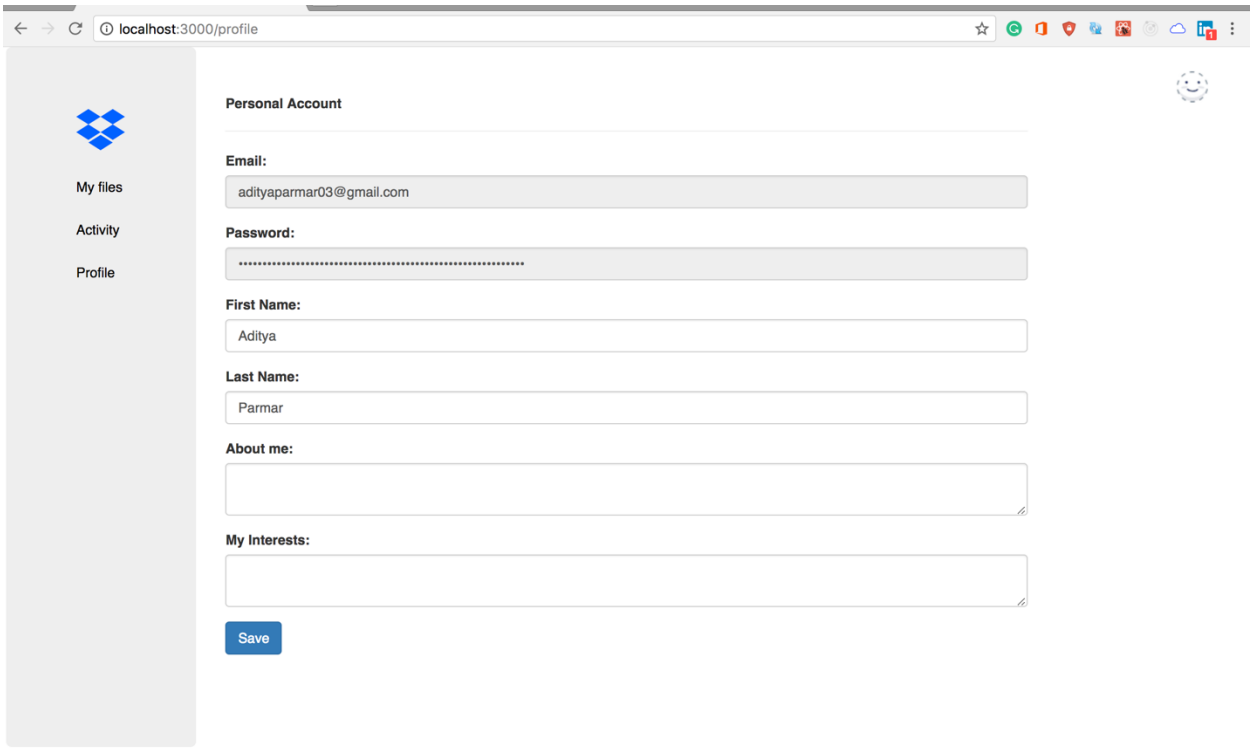

Dropbox
localhost:3000/home
Dropbox
Name
My files
Activity
Profile
abc.c
MyFirs
Share
SHARING SUCCESSFULLY COMPLETED.
Public Link:
http://localhost:9000/files/1510549661511_abc.c
Milan Bhakta (milan@sjsu.edu)
Dhaval Raval (dhval@gmail.com)
Milan Bhakta
Dhaval Raval
Share
Close
Upload a file
MyFirstFolder
Create
localhost:3000/home
Search
Dropbox
Name
Date
Member
My files
Activity
Profile
abc.c
Shared File By Aditya in Milan Account
Sun Nov 12 2017 21:07:41
2 members
app-release.apk
Sun Nov 12 2017 13:30:38
3 members
JOP
Sun Nov 12 2017 13:30:34
3 members
app-release.apk
Sun Nov 12 2017 02:42:32
4 members
Milan Bhakta
Home
Profile
Activity Log
Sign out
Create

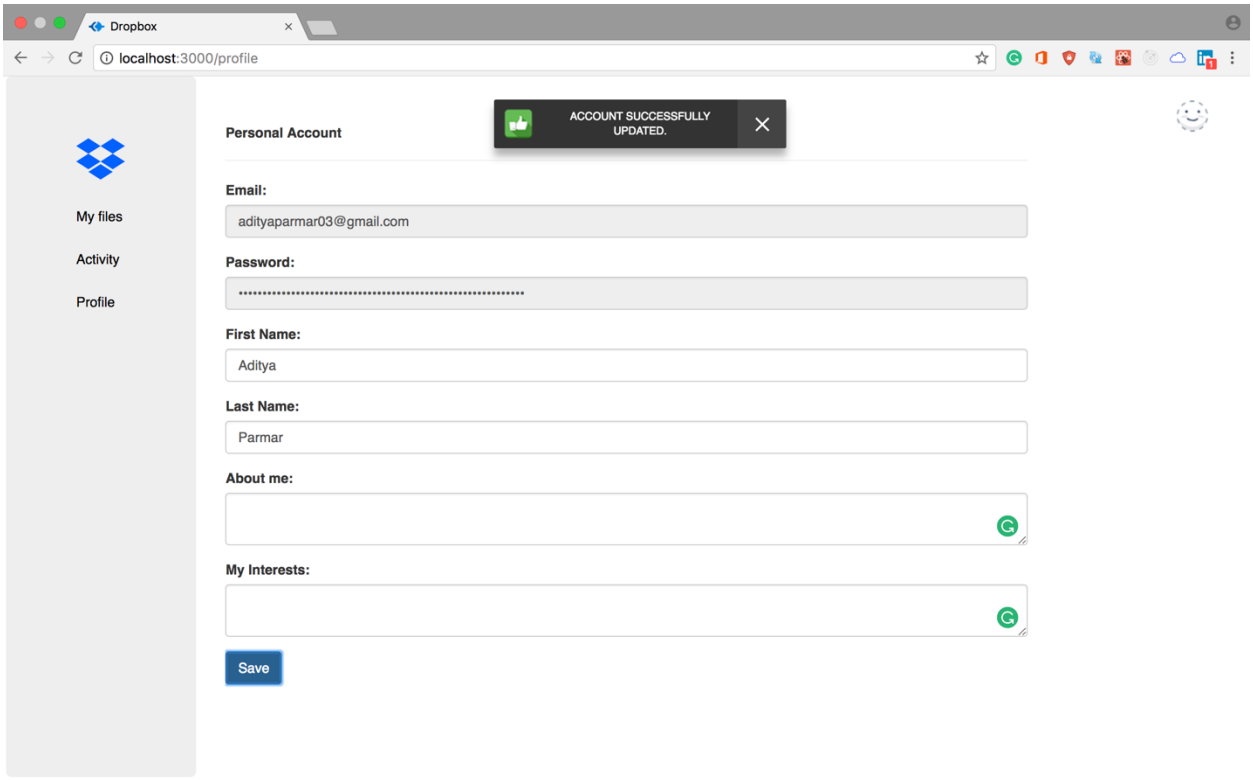

Dropbox
localhost:3000/home
Dropbox
My files
Activity
Profile
Name
Date
Member
s
Sun Nov 12 2017
21:16:08
Only you
app-release.apk
Sun Nov 12 2017
02:42:32
4 members
Rahi Patel
Rahul Khanna
Milan Bhakta
Aditya Parmar
Members
Upload a file
Folder/Group Name
Create

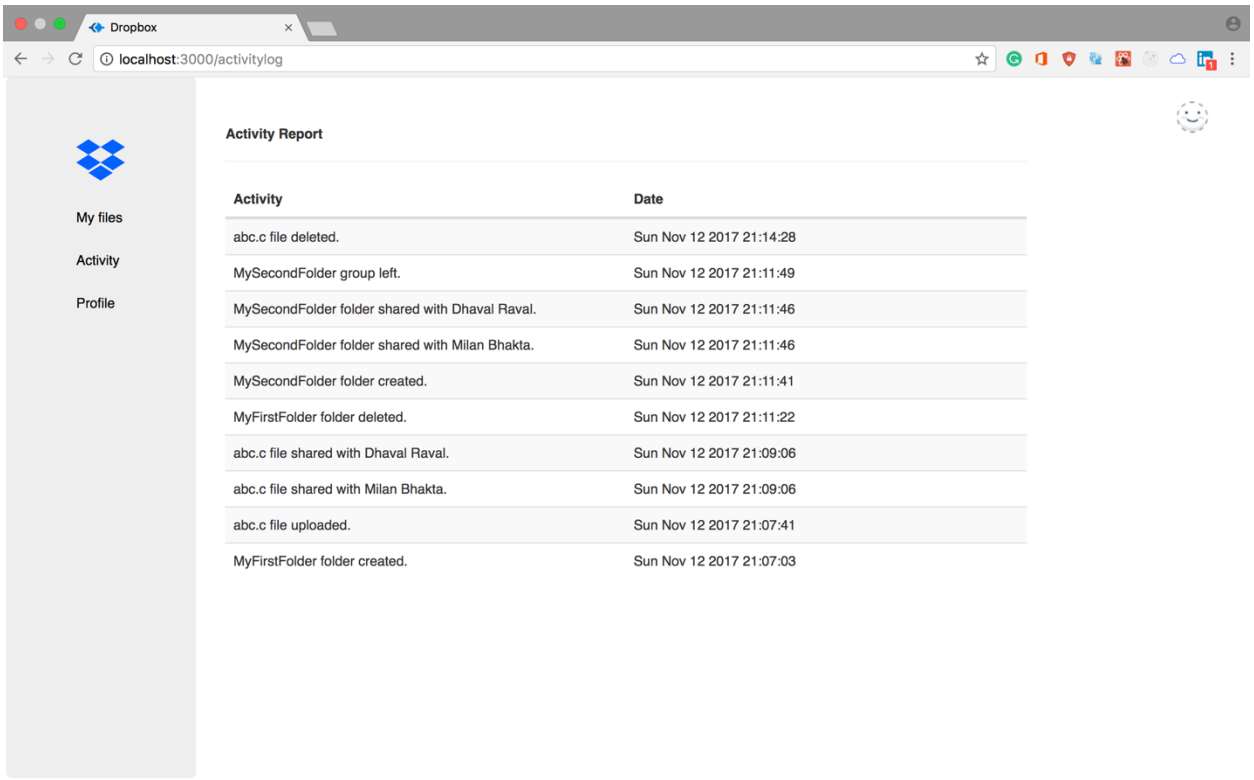

localhost:3000/home
Dropbox
MYFIRSTFOLDER FOLDER
DELETED SUCCESSFULLY.
My files
Activity
Profile
Name
Date
Member
abc.c
Sun Nov 12 2017
21:07:41
3 members
Upload a file
MyFirstFolder
Create


Dropbox
localhost:3000/home
DELETE SUCCESSFULLY, BUT FOLDER IS STILL SHARED WITH OTHER MEMBERS.
Dropbox
My files
Activity
Profile
Name
Date
Member
abc.c
Sun Nov 12 2017 21:07:41
3 members
Upload a file
MySecondFolder
Create


localhost:3000/home
YOU ARE NOT AUTHORISE PERSON TO PERFORM THIS TASK.
Dropbox
My files
Activity
Profile
Name
Date
Member
abc.c
Sun Nov 12 2017 21:07:41
3 members
app-release.apk
Sun Nov 12 2017 02:42:32
4 members
Upload a file
Folder/Group Name
Create


localhost:3000/profile
My files
Activity
Profile
Personal Account
Email:
adityaparmar03@gmail.com
Password:
First Name:
Aditya
Last Name:
Parmar
About me:
My Interests:
Save

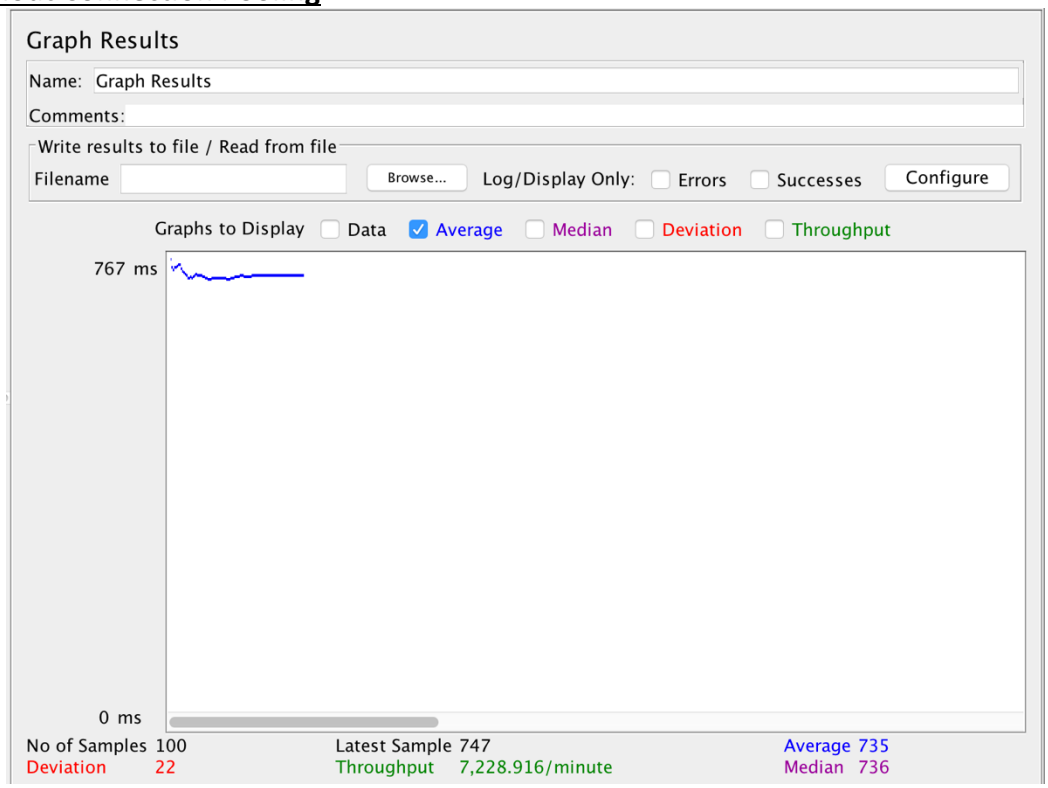

Dropbox
localhost:3000/profile
My files
Activity
Profile
Personal Account
ACCOUNT SUCCESSFULLY UPDATED.
Email:
adityaparmar03@gmail.com
Password:
First Name:
Aditya
Last Name:
Parmar
About me:
My Interests:
Save

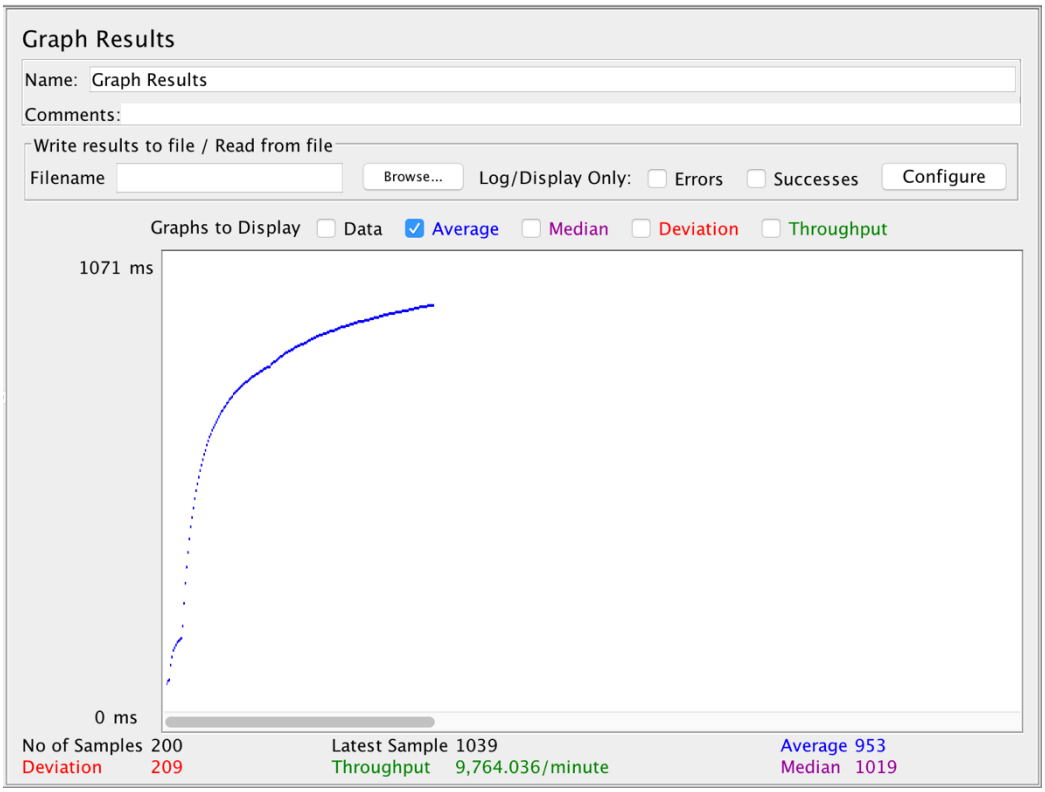

Dropbox
localhost:3000/activitylog
My files
Activity
Profile
Activity Report
Activity
Date
abc.c file deleted.
Sun Nov 12 2017 21:14:28
MySecondFolder group left.
Sun Nov 12 2017 21:11:49
MySecondFolder folder shared with Dhaval Raval.
Sun Nov 12 2017 21:11:46
MySecondFolder folder shared with Milan Bhakta.
Sun Nov 12 2017 21:11:46
MySecondFolder folder created.
Sun Nov 12 2017 21:11:41
MyFirstFolder folder deleted.
Sun Nov 12 2017 21:11:22
abc.c file shared with Dhaval Raval.
Sun Nov 12 2017 21:09:06
abc.c file shared with Milan Bhakta.
Sun Nov 12 2017 21:09:06
abc.c file uploaded.
Sun Nov 12 2017 21:07:41
MyFirstFolder folder created.
Sun Nov 12 2017 21:07:03

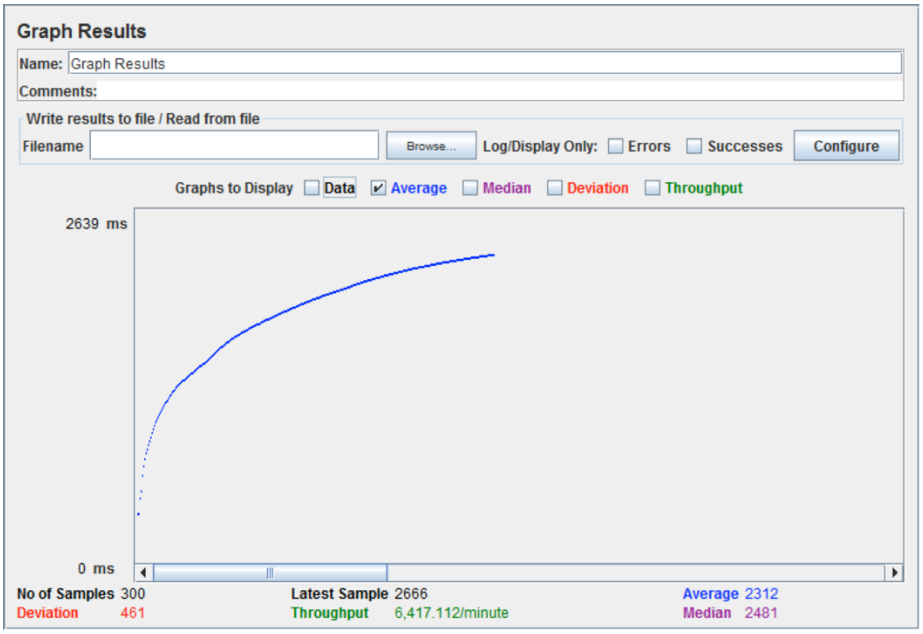
TESTING

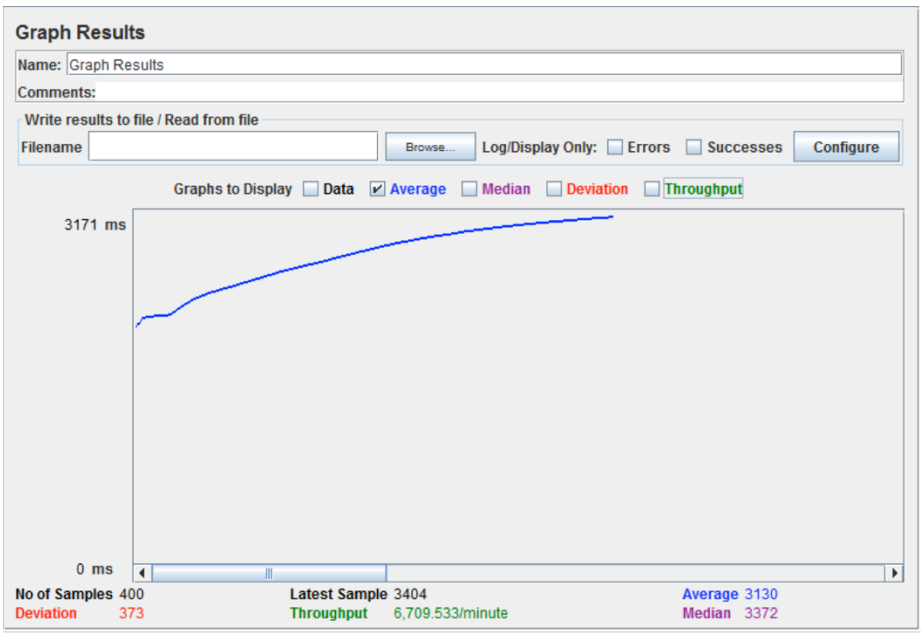
JMeter

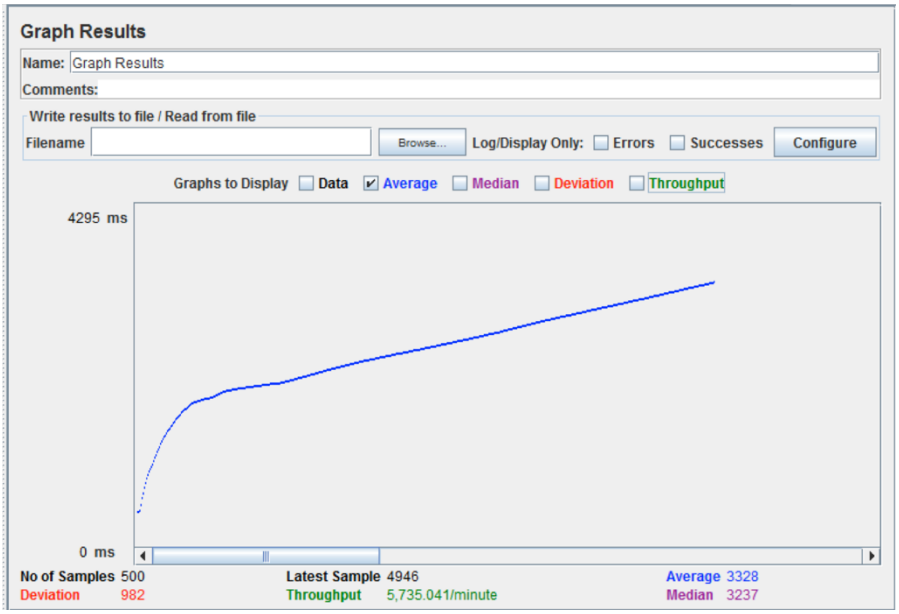
A . Without Connection Pooling



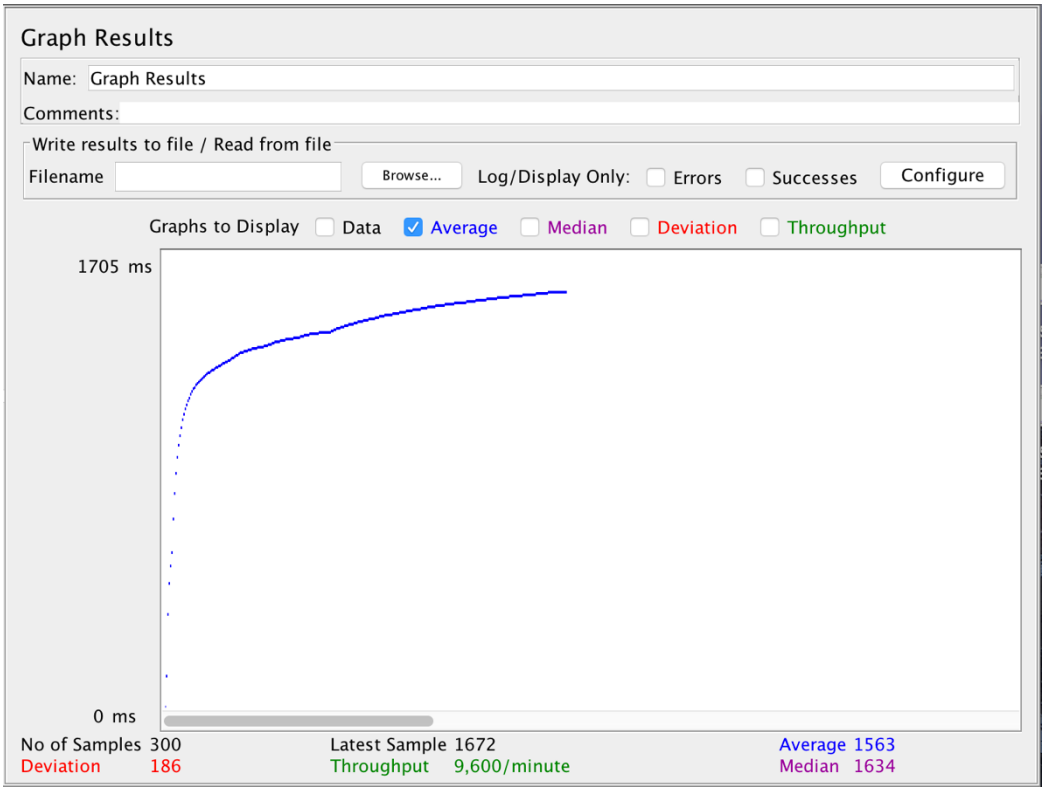


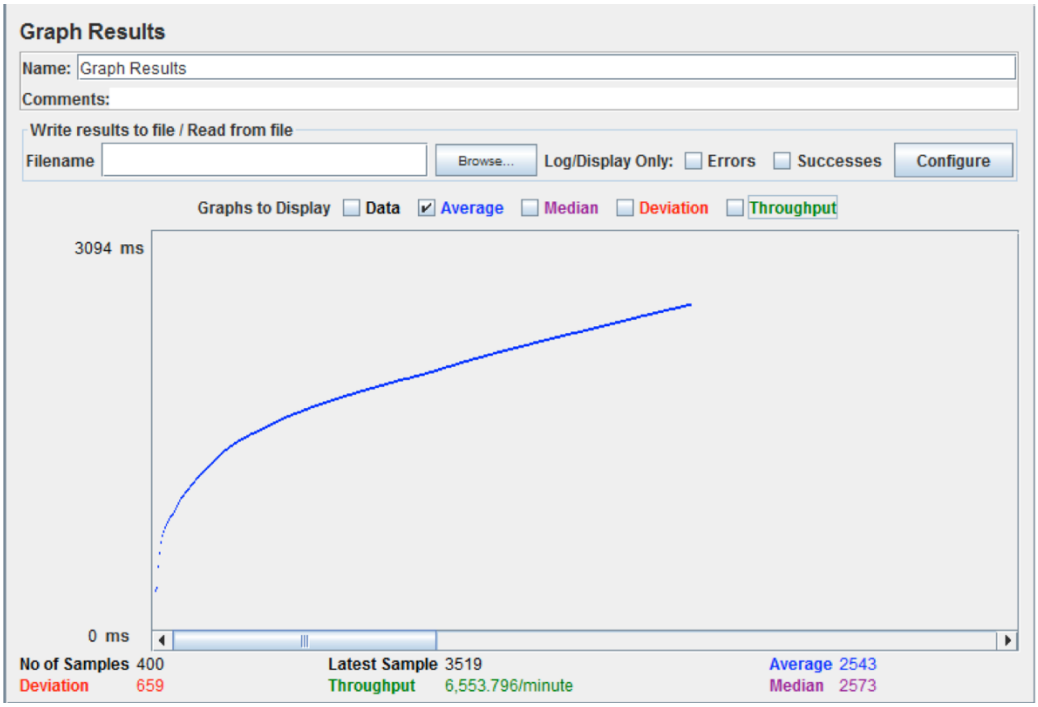

Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
2639 ms
0 ms
No of Samples 300
Latest Sample 2666
Average 2312
Deviation 461
Throughput 6,417.112/minute
Median 2481

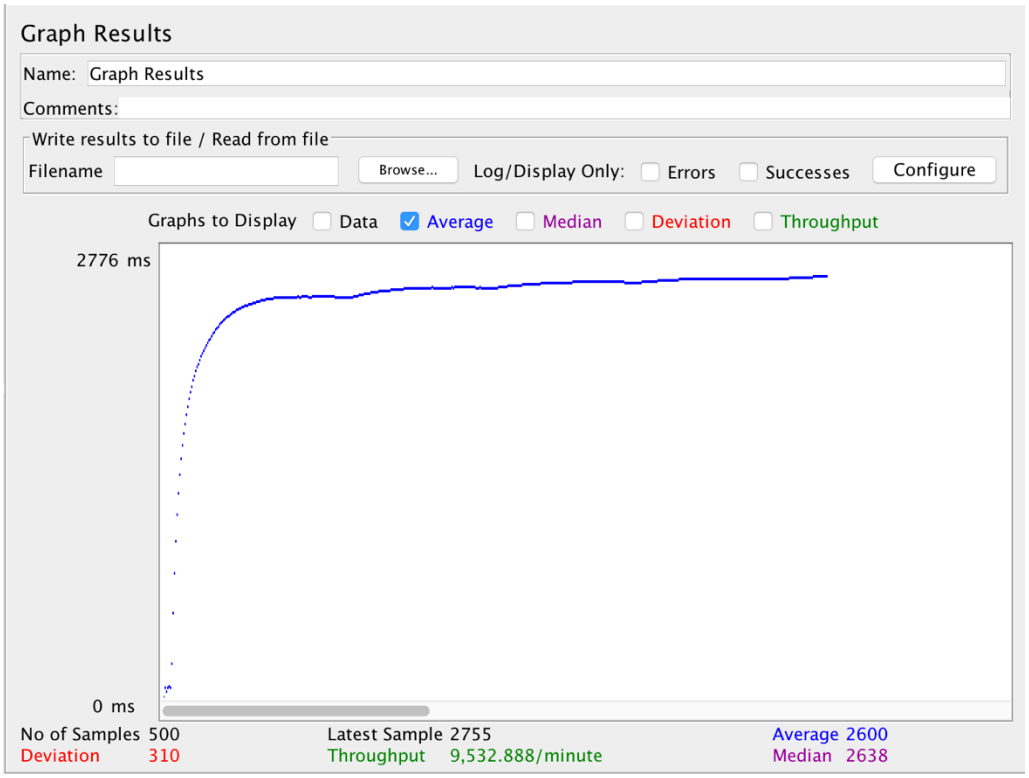

Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
3171 ms
0 ms
No of Samples 400
Latest Sample 3404
Average 3130
Deviation 373
Throughput 6,709.533/minute
Median 3372


Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
4295 ms
0 ms
No of Samples 500
Latest Sample 4946
Average 3328
Deviation 982
Throughput 5,735.041/minute
Median 3237

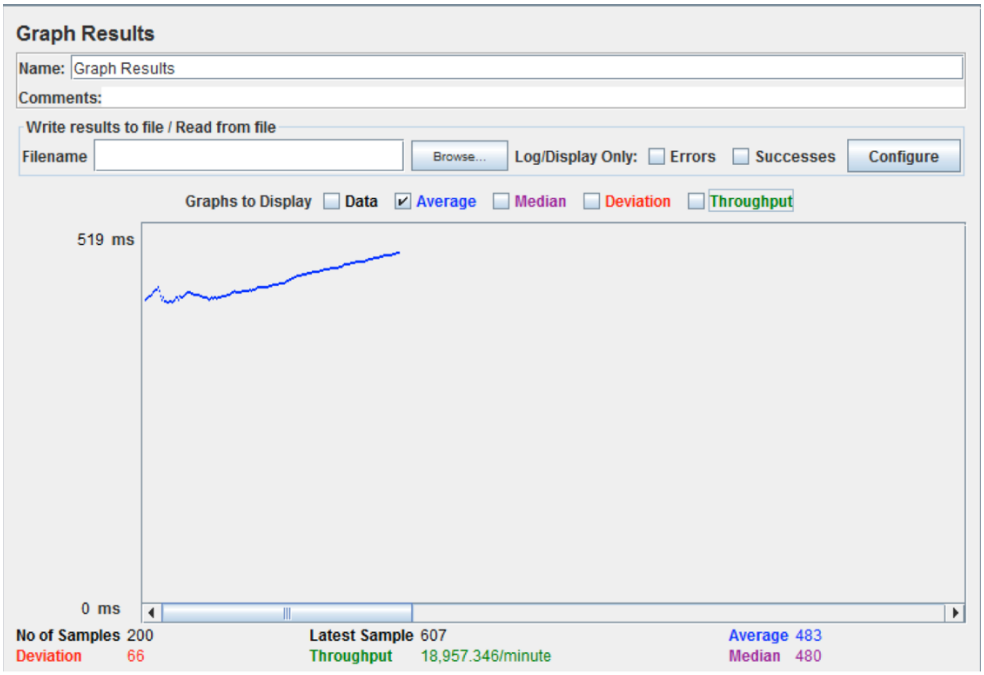
B. With own implemented Connection Pooling

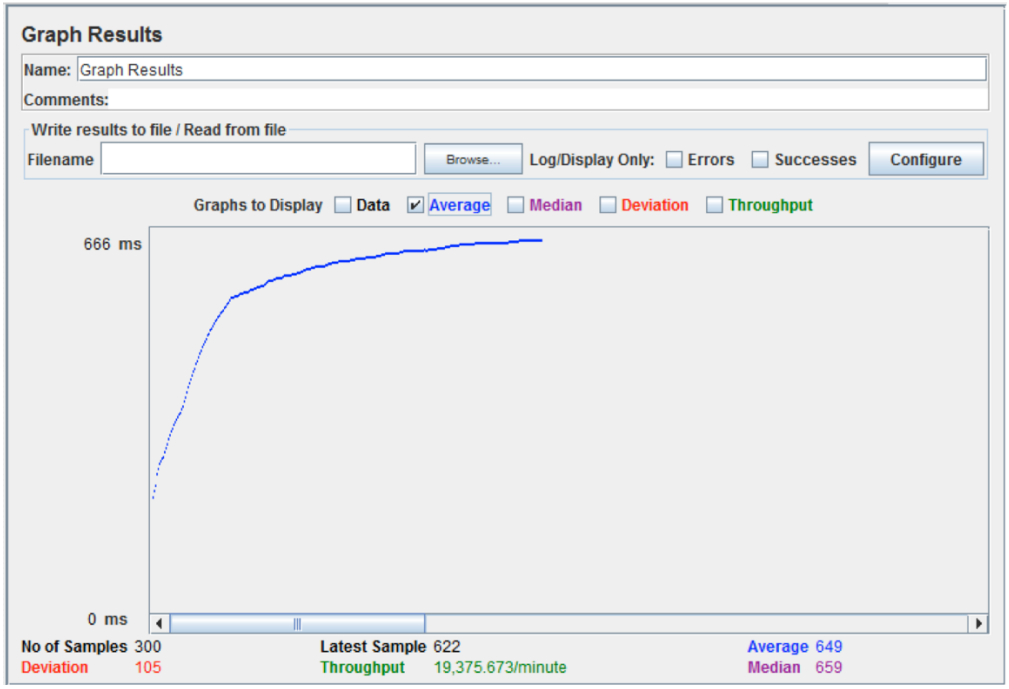

Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
1705 ms
0 ms
No of Samples 300
Latest Sample 1672
Average 1563
Deviation 186
Throughput 9,600/minute
Median 1634

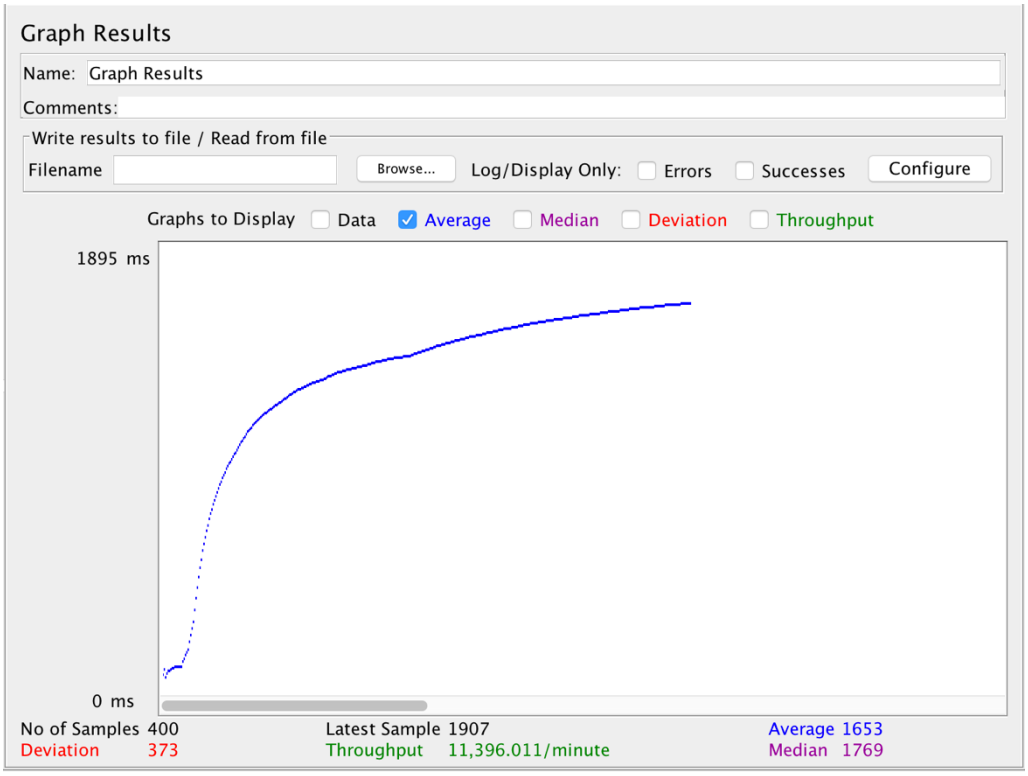

Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
3094 ms
0 ms
No of Samples 400
Latest Sample 3519
Average 2543
Deviation 659
Throughput 6,553.796/minute
Median 2573

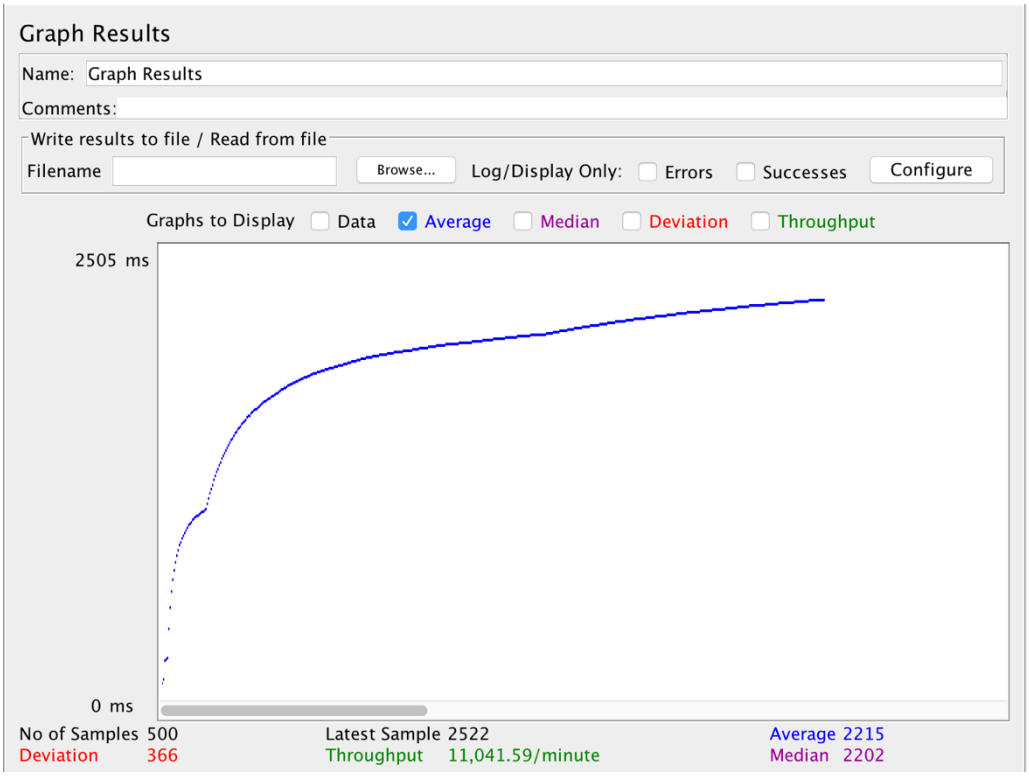

Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Graphs to Display
Data
Average
Median
Deviation
Throughput
2776 ms
0 ms
No of Samples 500
Latest Sample 2755
Average 2600
Deviation 310
Throughput 9,532.888/minute
Median 2638

C. With Connection Pooling




Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
666 ms
0 ms
No of Samples 300
Latest Sample 622
Average 649
Deviation 105
Throughput 19,375.673/minute
Median 659


Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Graphs to Display Data Average Median Deviation Throughput
1895 ms
0 ms
No of Samples 400
Latest Sample 1907
Average 1653
Deviation 373
Throughput 11,396.011/minute
Median 1769

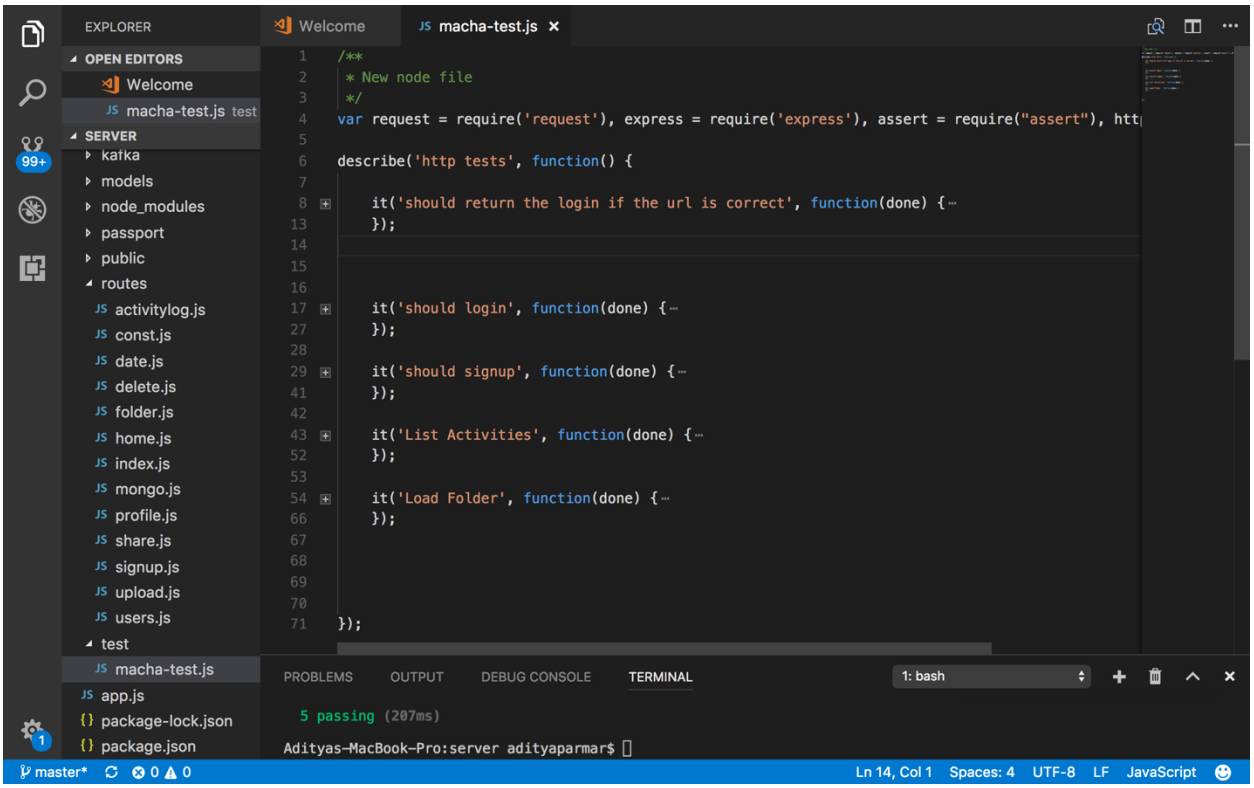


Conclusion

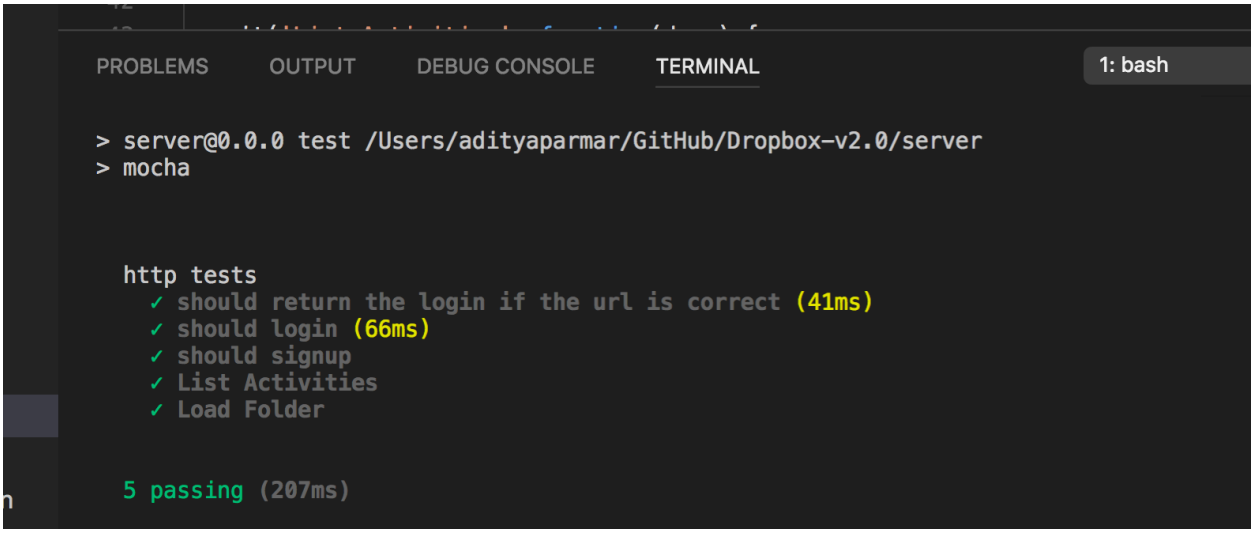
If we compare all three case, we can see that it is beneficial to use connection pooling. If we compare the Average time of 500 concurrent users, one with in-built connection pooling has **1.6x times** better average response time and my own implemented connection pooling has **1.2x times** better average response time. We must use connection pooling for better performance and availability.

TESTING

Mocha



OUTPUT



5 randomly selected API.

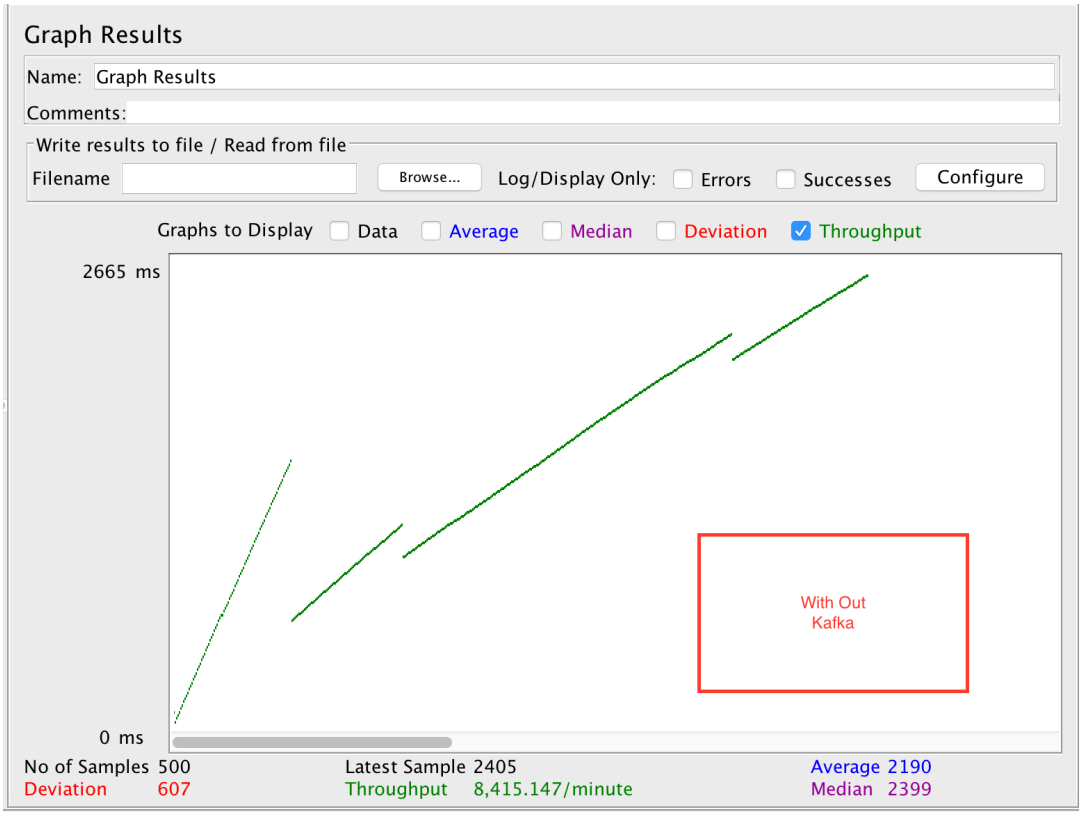
Questions

1. **Compare passport authentication process with the authentication process used in Lab1.**

- Passport is an authentication middleware for Node-Express framework by Jared Henson. Passport adds an additional layer of indirection allowing the separation of authentication logic from the rest of the application keeping the code clean and also offering inbuilt support for various other authentication strategies like Facebook Login, Google Login, etc. In this application, I have used Passport's 'Local Strategy' to implement user authentication mechanism. Passport itself doesn't encrypt the password being stored in the databases but allows to developer to implement any encryption algorithm or policy to implement encryption have use bcrpyt to salt and hash the user's password before saving them to mongo dB.
- Passport does encrypt its sessions and save to database which provide persistent session.
- In Lab1 I had used JWT token, But the problem is it affect me with persistent session. Every time I must decrypt it to establish session. It is very complex logic. But, in passport it is all handle by framework as I discuss above. It provides better performance, User Experience and security than my lab 1 authentication.

Blowfish for encryption. Both are good encryption algorithms but Blowfish does have the added advantage of showing more resilience to rainbow and dictionary attacks.

2. **Compare performance with and without Kafka. Explain in detail the reason for difference in performance.**



- You can clearly see this without Kafka Throughput is 8,415/minute and with Kafka it is 21802/minute. You can see that it clearly more than twice better performance.

- Kafka is able to provide such availability due to being able to delegate the requests to different consumers via topic whereas in case of normal database access, increased accesses result in bottleneck situation when number of such accesses increase.

- Kafka has better throughput, built-in partitioning, replication, and fault-tolerance which makes it a good solution for large scale message processing applications. These is why Kafka is a good solution for large scale message processing applications

3. If given an option to implement MySQL and MongoDB both in your application, specify which data of the applications will you store in MongoDB and MySQL respectively

- For applications that require higher availability and tolerance, I would go for MongoDB and for the ones that need more consistency; I would use a traditional relational database like MySQL. I will use MySQL database for storing sensitive data and transaction data as it is more secure than MongoDB. I will use MongoDB for high amount of data which can be reused again in the application.

- For example, I choose mysql for authentication data which should be secure. For payment module, I use mysql.
- I used Mongodb for saving file data, sharing data and activity data which I need more available for high performance.
- I would use MongoDB for parts that need more reads as with MongoDB I can de-normalize several tables into just one or two collections which may potentially minimize my queries to just one as hard drive space is cheaper than CPU/servers.