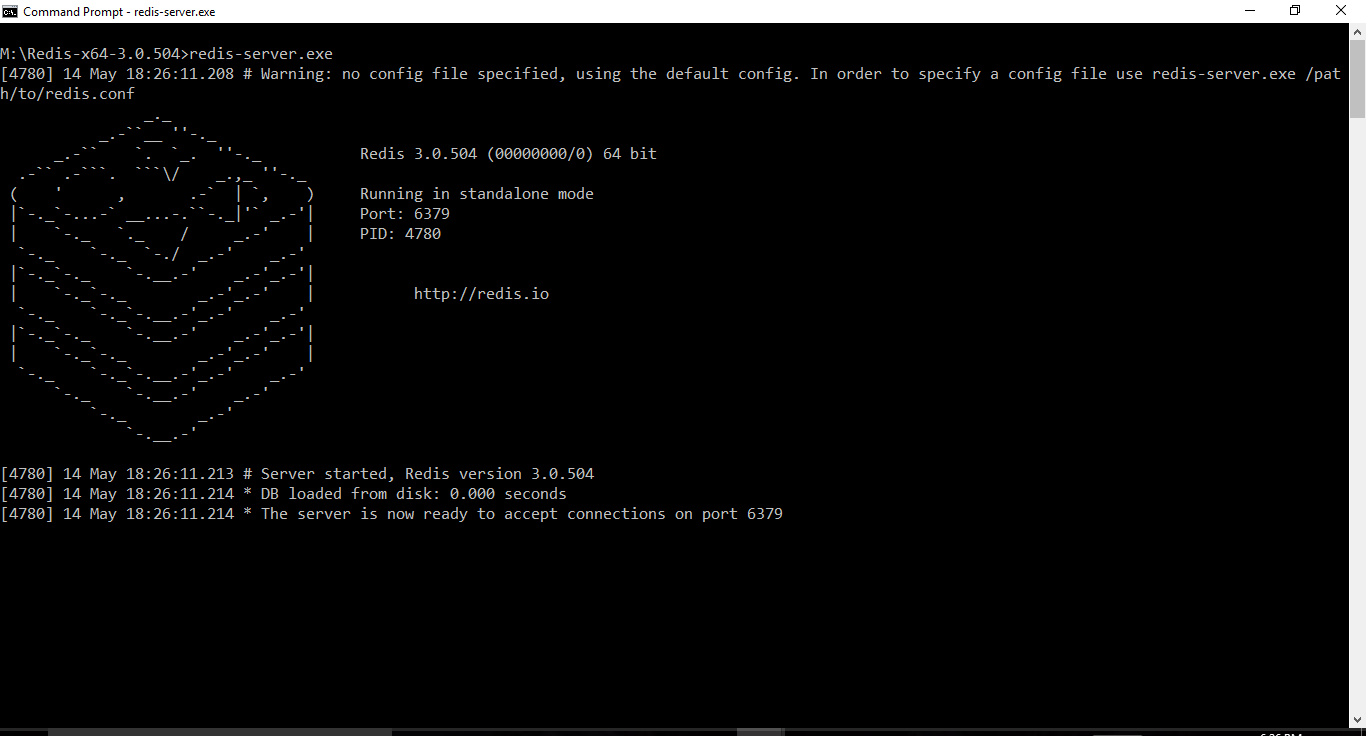
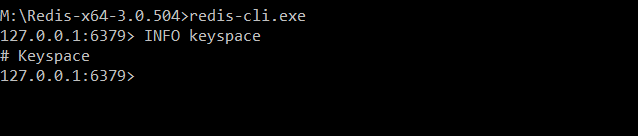
# Assignment 11: Redis Cache and Static Image Access

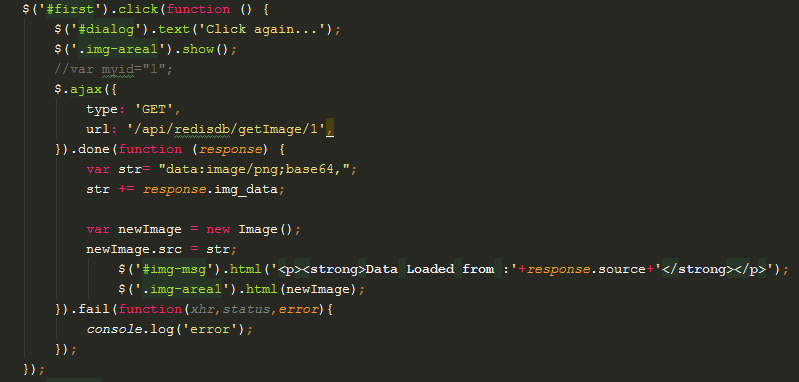
1. Unzip the “280Assignment11\_Vishwanath” folder and place it in the current working directory.
2. The zip contains “Assignment11\_Redis” Folder and Readme file.
3. The application is built using Node.js and Express module.
4. The application will run on <http://localhost:3000> address
5. **PreRequisites** –
   1. Navigate to the “Assignment11\_Redis” folder in command prompt and run the following commands
      1. npm install
      2. bower install
   2. The following commands above will install all the necessary packages
   3. Install Redis-x64-3.0.504.zip for Windows or appropriate package for your OS and unzip the folder.
   4. The folder should contain “redis-server.exe”. Go to the command line, navigate to the directory and type redis-server.exe and press Enter/Return. The server should be running before you proceed with Step 6.



1. In a new command prompt navigate to the “/Assignment11\_Redis /bin folder and run the following command to run the application
   1. node www (Note: Please do not modify this file)
2. Open a new CMD prompt, navigate to Redis directory and type the following commands. You can see no keys are added.



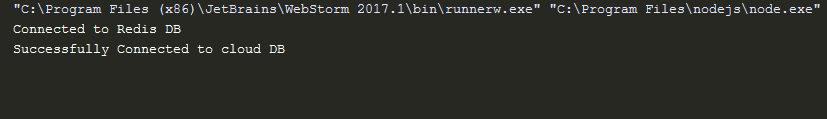
1. The below code sends a request to the getImage API with image ID and populates the response in HTML. The code for calling Redis DB API is as follows



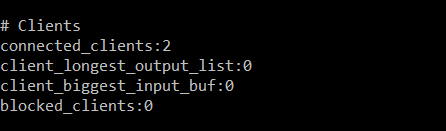
Get Image API checks first Redis Cache for the image. If not present, then queries Mongo DB.

router.get('/getImage/:id',function (*req*,*res*,*next*) {  
 *console*.log("/getImage");  
 var image\_id = *req*.params.id;  
  
 redisclient.exists(image\_id,function (*err*,*reply*) {  
 if (*reply* === 1) {  
 redisclient.get(image\_id, function (*err*, *resp*) {  
 *res*.json({"source": "Redis","img\_data":*resp*});  
 });  
 }  
 else{  
 imgHandle.findOne({ img\_id:image\_id}, function (*err*, *document*){  
 if(*err*){  
 *console*.log(*err*);  
 throw *err*;  
 }  
 var base64 = (*document*.img.data.toString('base64'));  
 //console.log(typeof (image\_id)+" "+typeof (base64));  
 redisclient.set(image\_id,base64);  
 *res*.json({"source": "Mongo","img\_data":base64});  
 });  
 }  
 });  
});

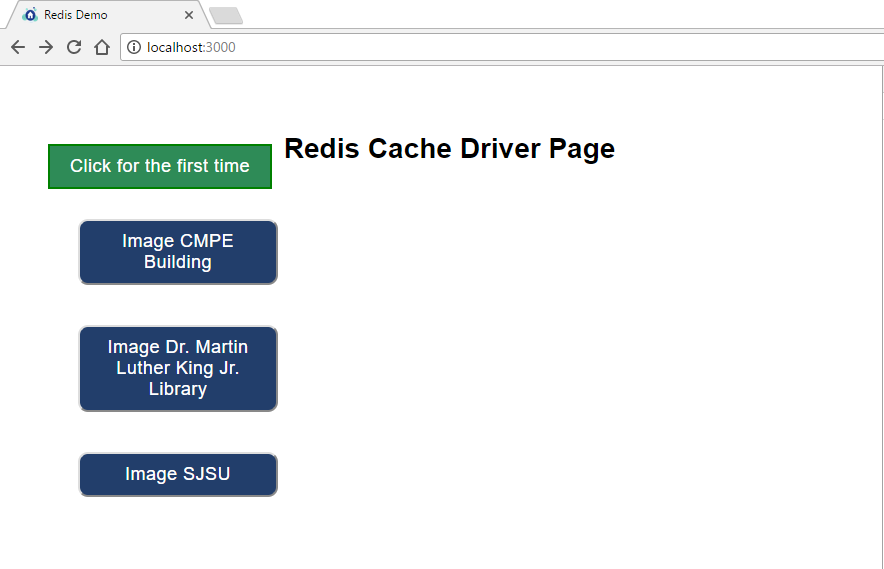
1. On running the application, the Redis Server and Mongo DB is connected.

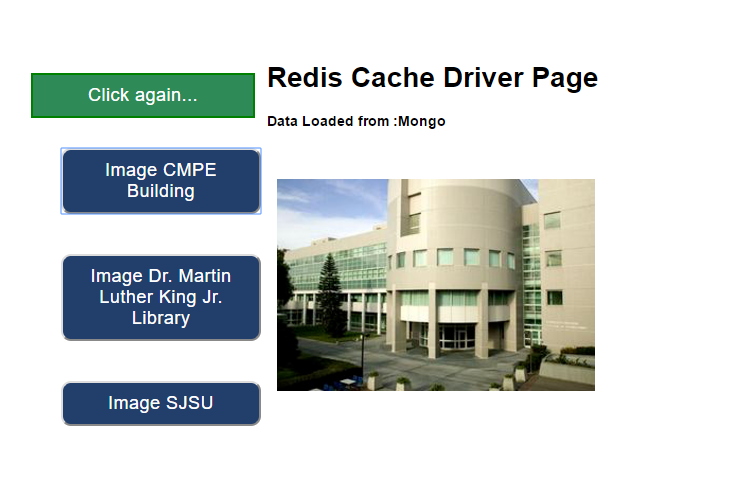


Once connected you can type INFO in redis-cli.exe prompt to see number of connected clients.

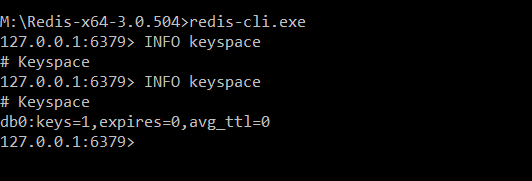


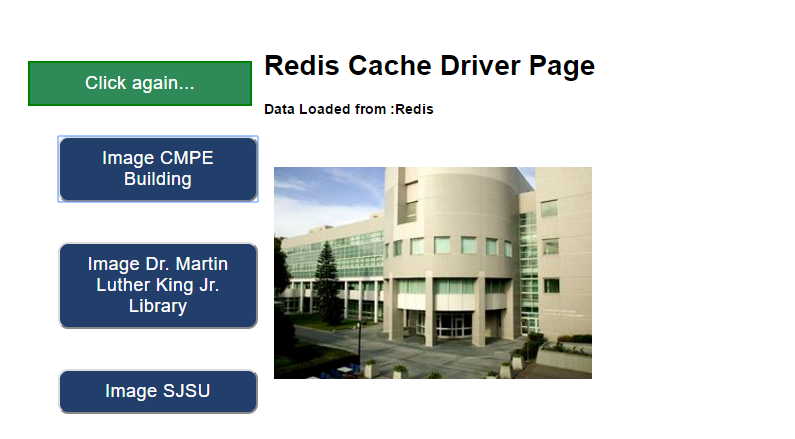
1. When you open <http://localhost:3000> , you can see only buttons.



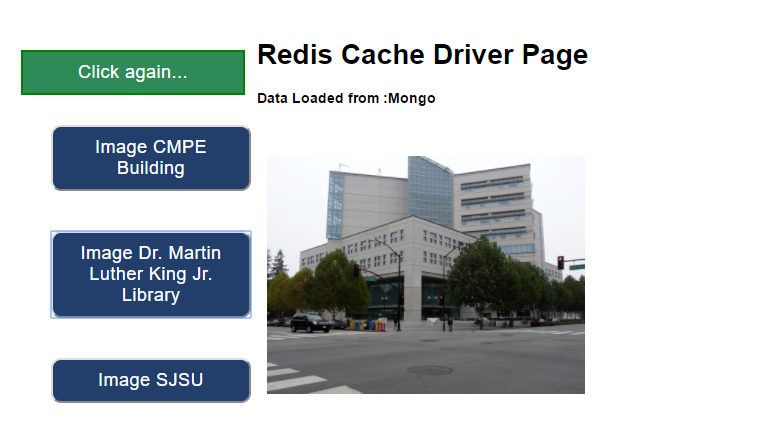
1. When you click on one of the buttons the data first loads from Mongo DB as the key is not present in Redis Cache. 

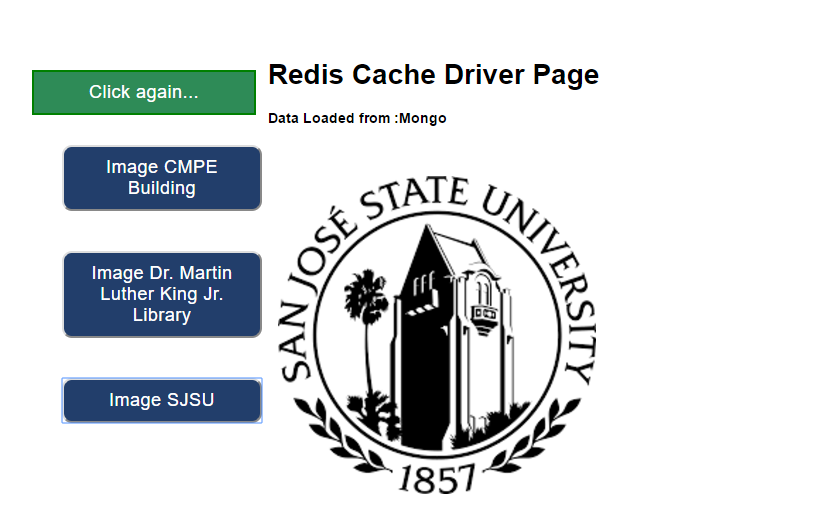
The button click will also store a key in Redis cache.



When you click on the button again, the image is loaded from Redis cache as follows.

1. Similarly, for other buttons the image is first loaded from Mongo DB and then after one more click it is loaded from Redis.





The keys are added as well to the Redis cache.

