**Developers Manual**

**For**

**Disaster Response and Reporting System**

**Table of Contents**

[**System Requirements** 3](#_Toc532218653)

[**Tools and Technologies Used** 3](#_Toc532218654)

[**Installation instruction** 3](#_Toc532218655)

[**Set up the Development Environment** 12](#_Toc532218656)

[**Steps to run application** 13](#_Toc532218657)

[**Application Testing** 14](#_Toc532218658)

[**Deployment** 18](#_Toc532218659)

[**Known Incomplete Functionality** 21](#_Toc532218660)

[**Methodologies** 22](#_Toc532218661)

[**Glossary of errors noticed during application development** 23](#_Toc532218662)

# **System Requirements**

|  |  |
| --- | --- |
| Processor | **Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71GHz** |
| RAM | 4 GB |
| System Type | 32/64-bit Operating System |
| Operating System | Windows 8 or later |
| Privileges | Administrative right required during installation only |
| Display Resolution | 1366 x 768 |

# **Tools and Technologies Used**

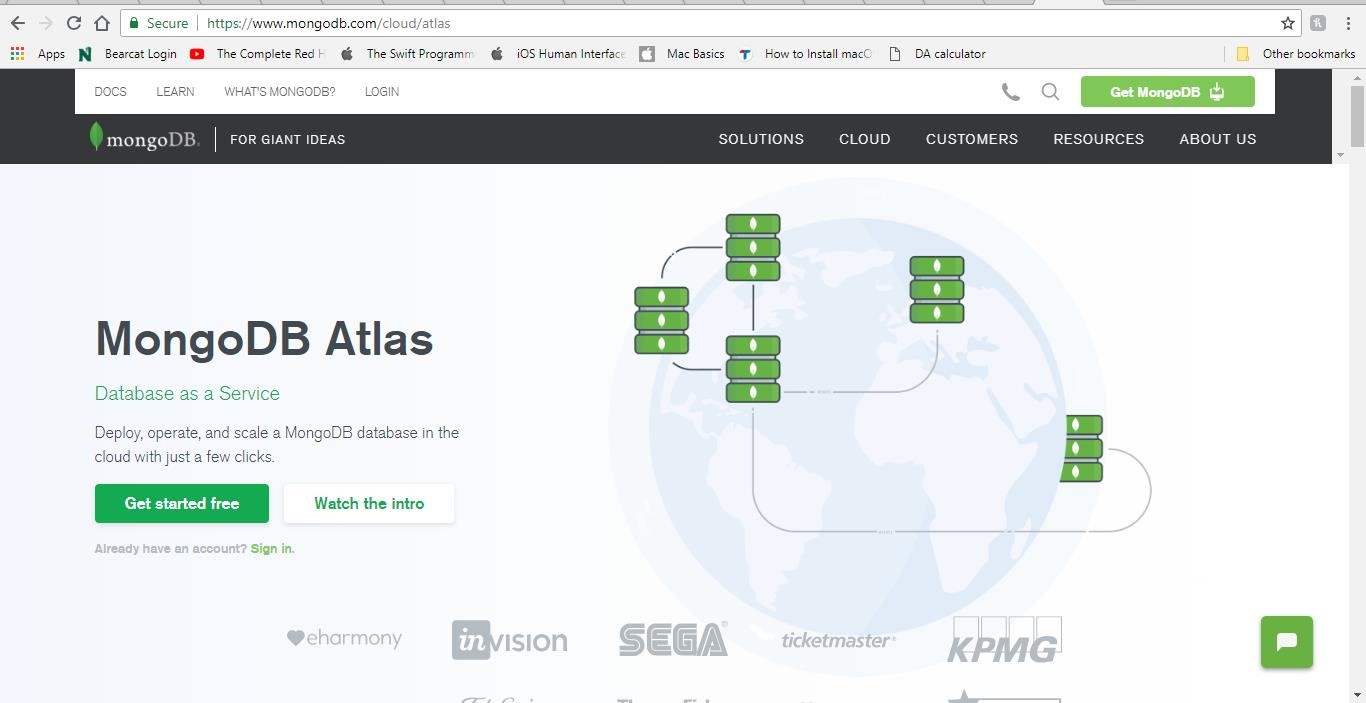
|  |  |
| --- | --- |
| Frontend | Angular v5, Ionic v2 |
| Command Line Interface | Angular CLI v6.0.8, Ionic CLI 4.3.0 |
| Backend | NodeJS v8.12.0 |
| Database | Mongodb Atlas |
| Database GUI | Mongodb Compass |
| Mockup | Adobe XD v13.1.32.8 |
| Server | Heroku |
| Version Control System | GitHub |
| Project Management Applications | Trello, Google Spread sheets |
| Web Browser | Google Chrome, Firefox |
| API Development Environment | Postman |
| Software-Testing Framework | Selenium |
| Code Editor | Visual Studio Code |
| Package Manager | npm |
| npm Libraries | fcm-node, mongoose, multer, nodemailer, passport |

# **Installation instruction**

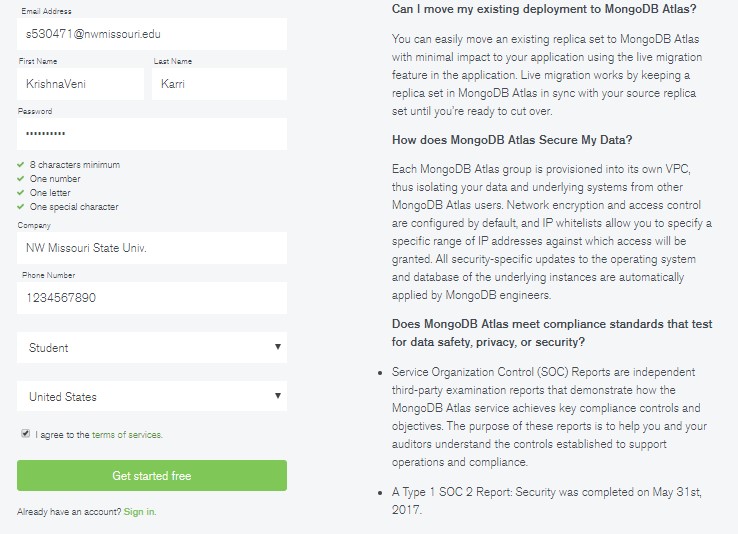
Setting up Windows to run MEAN Stack (MongoDB, ExpressJS, AngularJS, and NodeJS) applications is fairly simple and only requires a couple of things to be installed - namely MongoDB and NodeJS.

**Mongo Atlas Installation**

1. Go to : <https://www.mongodb.com/cloud/atlas>

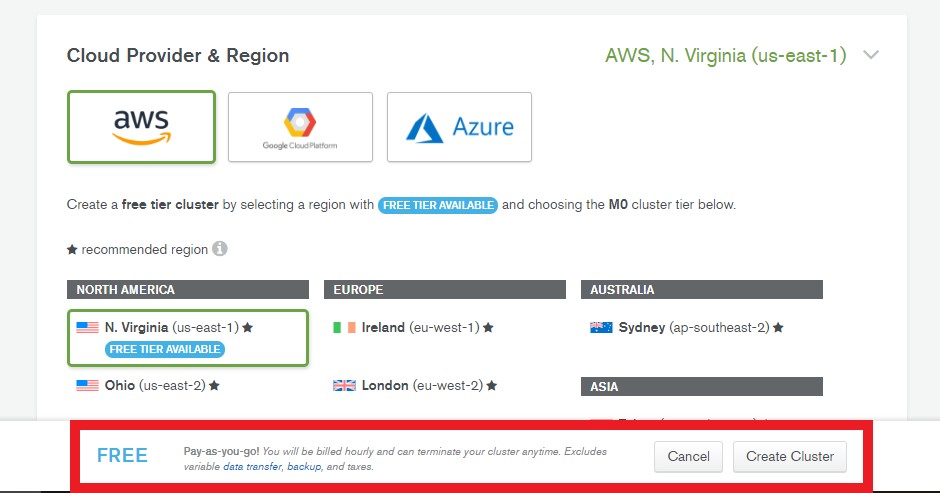


1. Click on ***get started free*** button
2. Enter the details and complete the registration

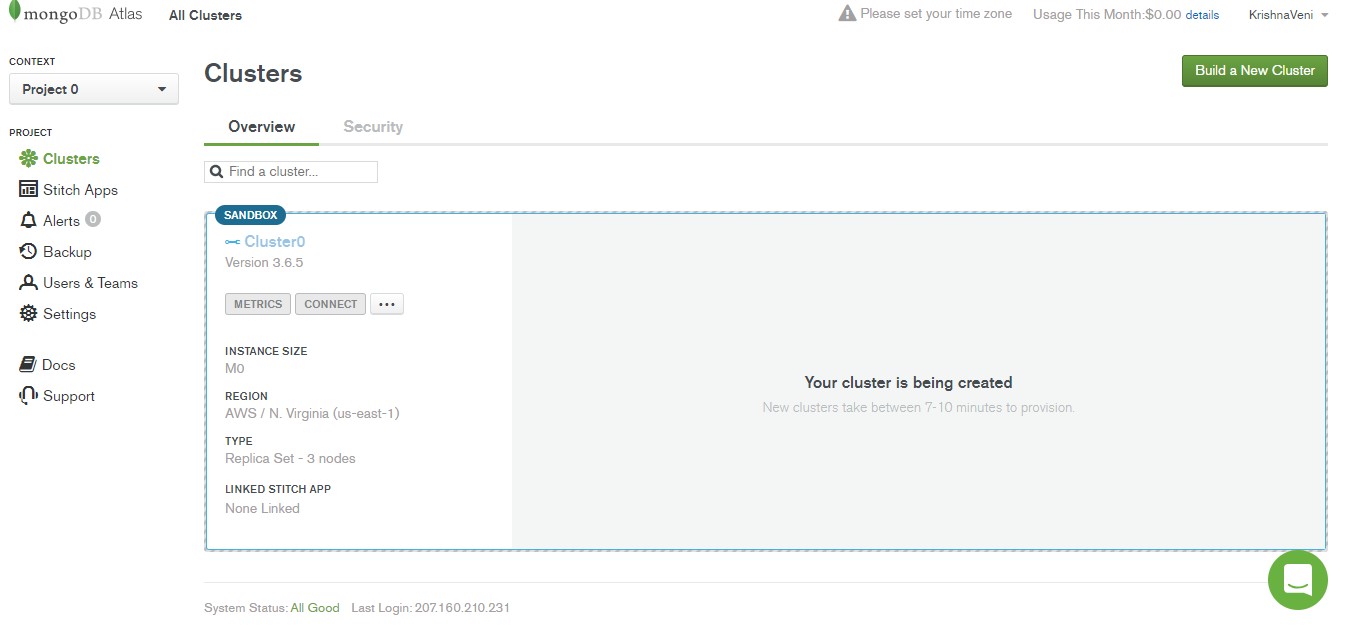


1. Click on create cluster button which is on the rite bottom as shown in the below figure HIGHLITED IN RED

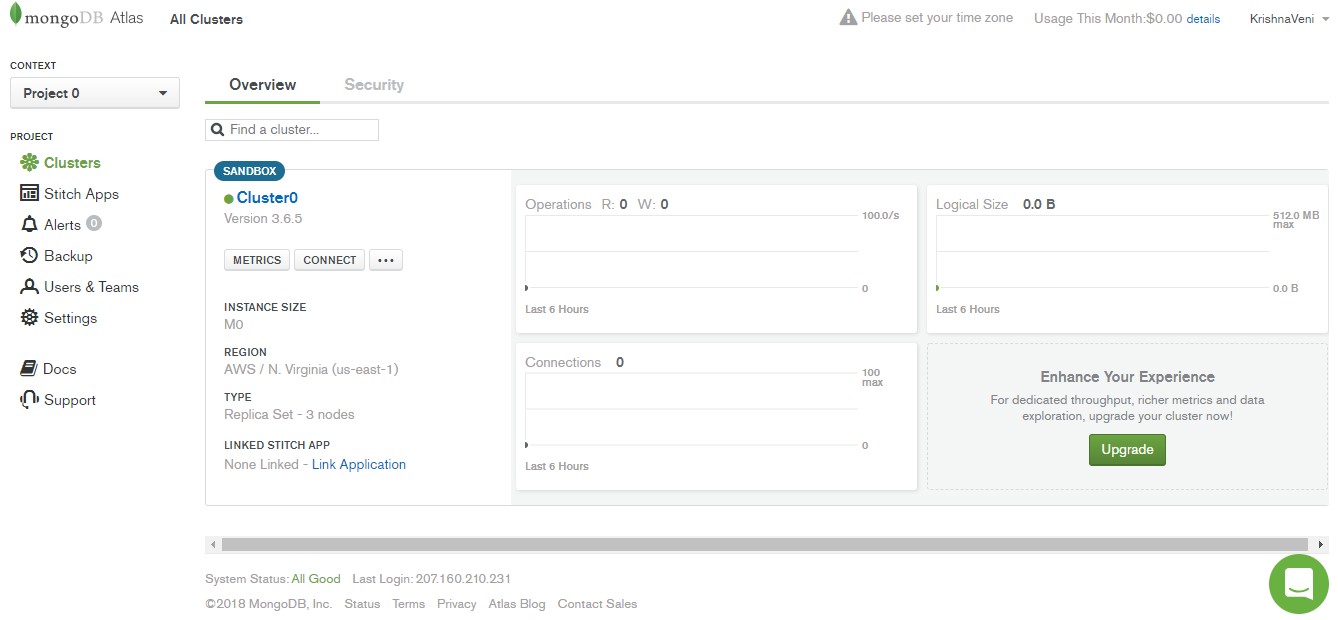
**NOTE: please make sure you are selecting free services. DONOT CHANGE ANY DEFAULT SELECTIONS.**



1. You will see below page. The cluster is being created page. This will take up to 15 minutes.



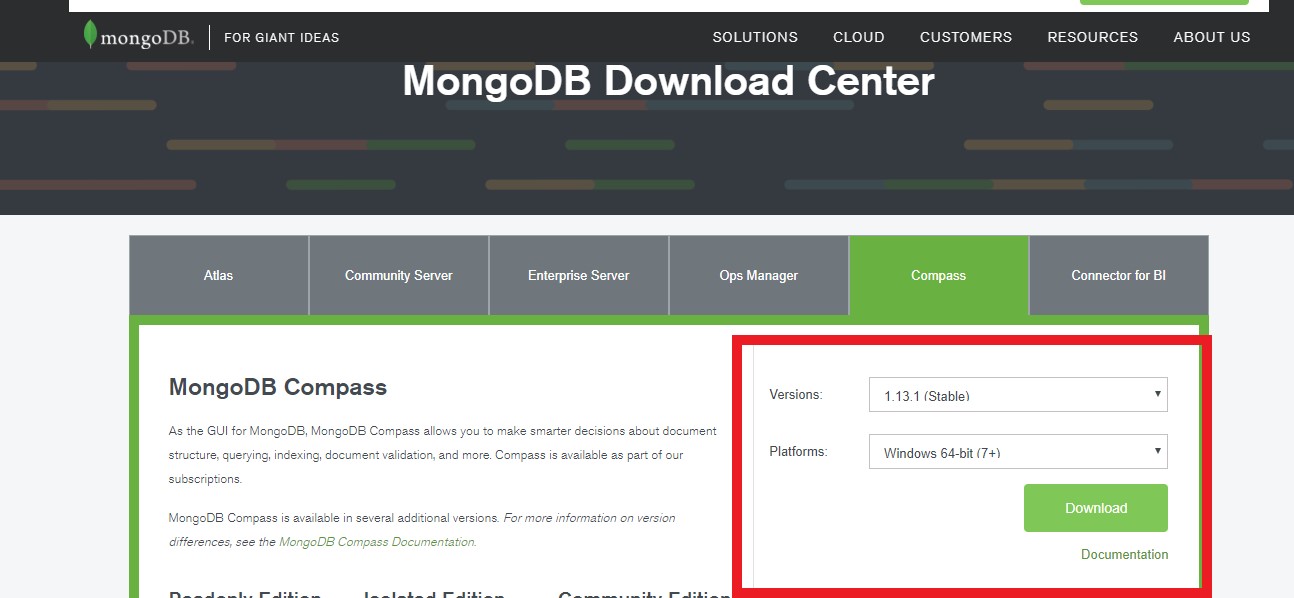
1. This will be the final screen



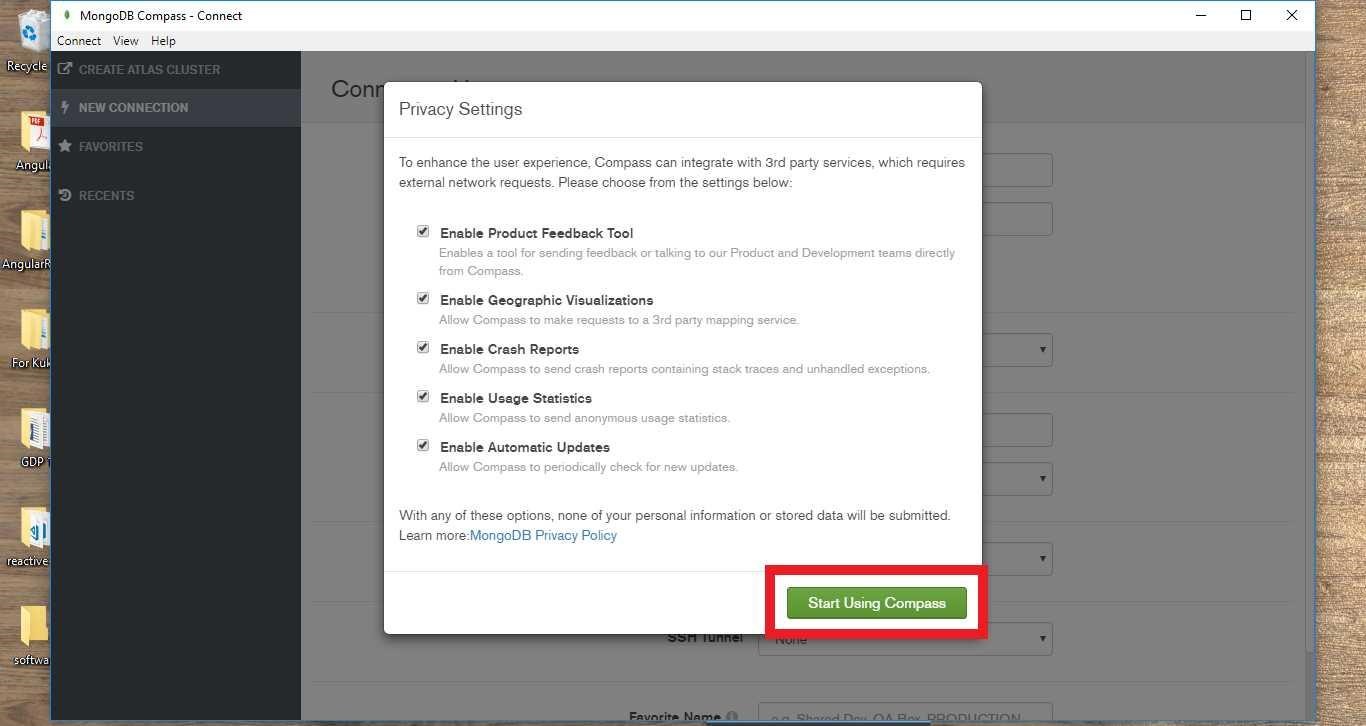
**Mongodb Compass installation guide**

1. Go to the link : <https://www.mongodb.com/download-center?jmp=hero#compass>
2. Click on download button.

**NOTE: PLEASE MAKE SURE TO INSTALL THE SAME VERSION (1.13.1) AS SHOWN BELOW PICTURE**

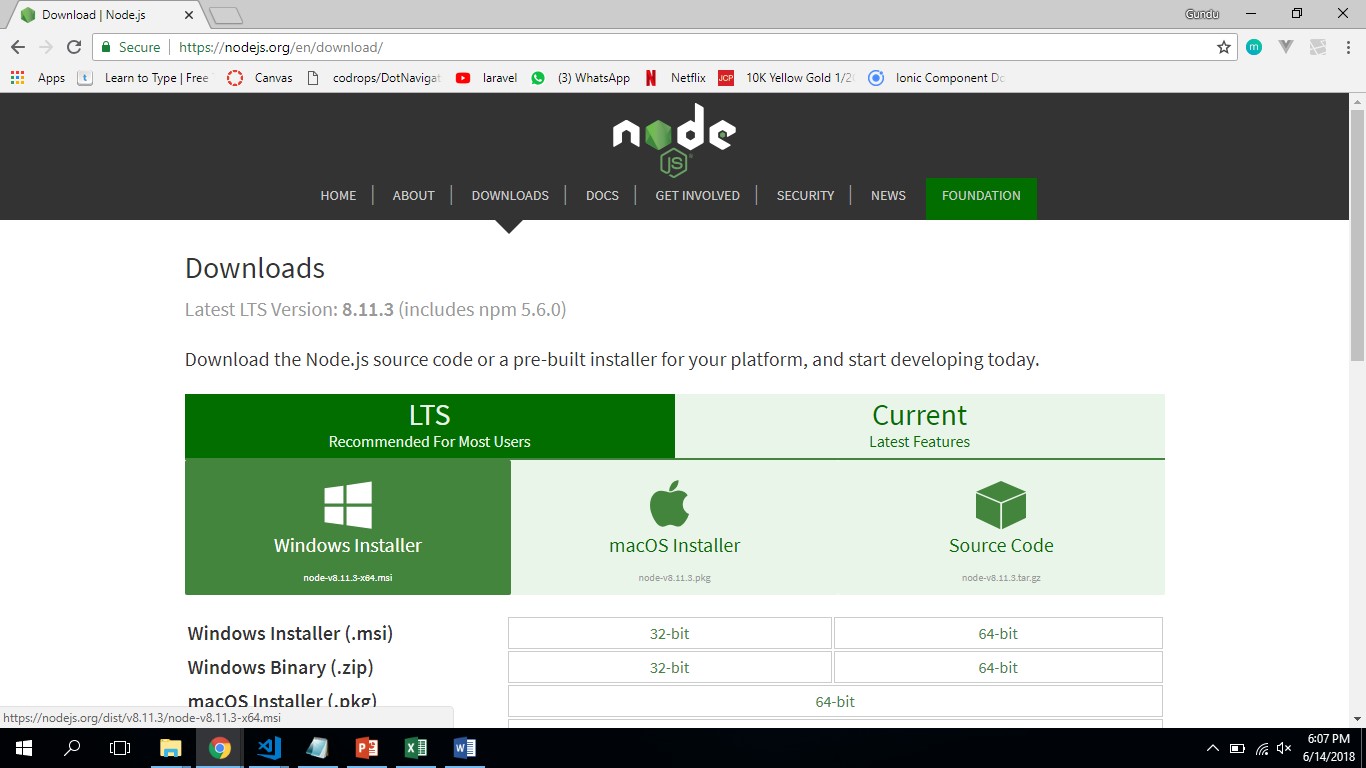


1. Open the exe file you downloaded proceed with installation (it is pretty simple and obvious). In the end you can use the below screen.

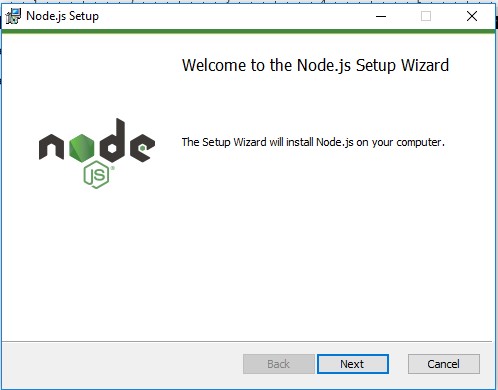


**NodeJS Installation guide**

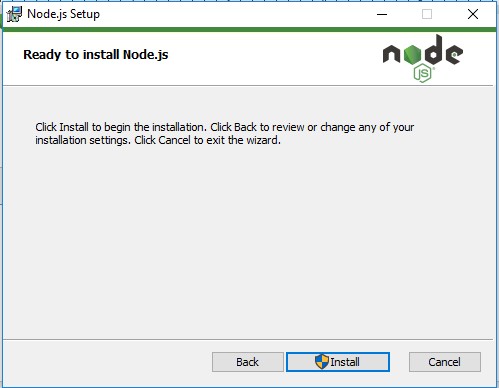
1. Go to Link: <https://nodejs.org/en/download/>

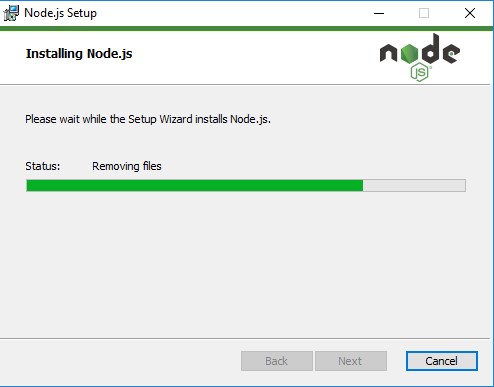


1. Select Windows Installer and you will be able to notice a file starts downloading.
2. Open and setup the downloaded file.

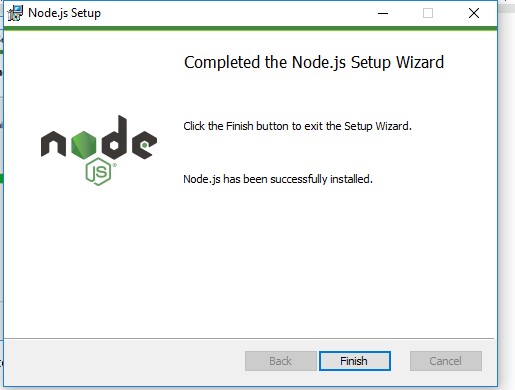


1. Proceed till you install the NodeJS

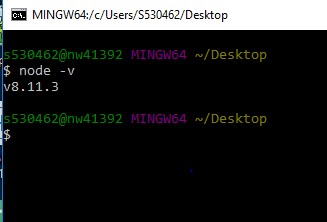




1. After setting up the NodeJS you will be able to see the below screen

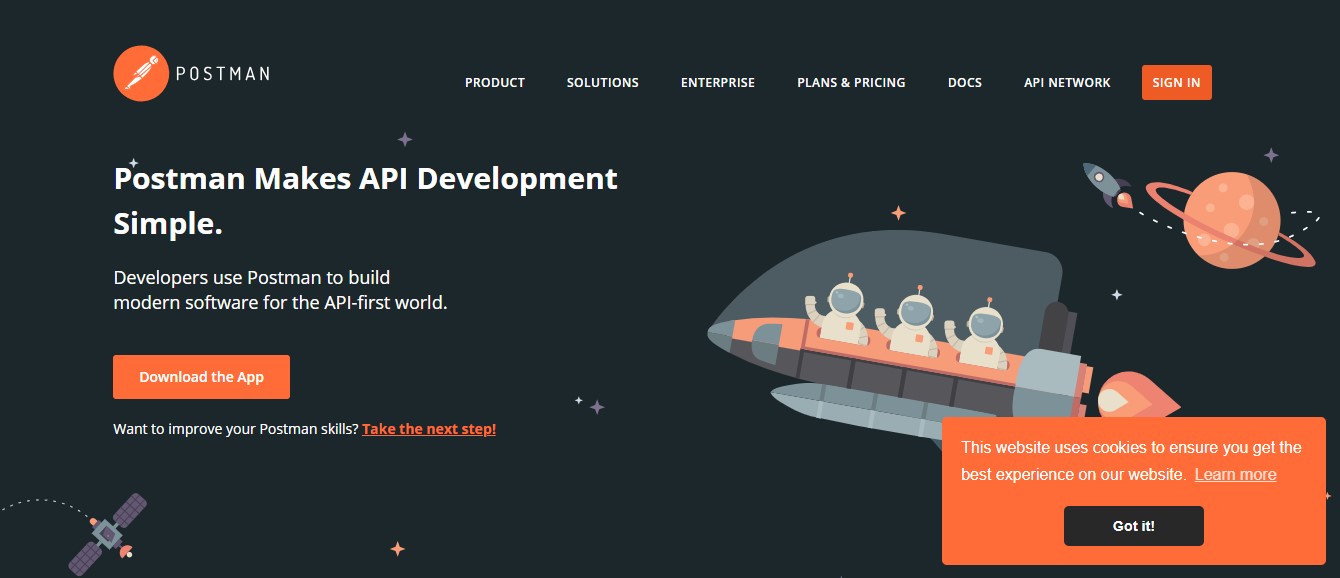


1. Open GitBash terminal or Command prompt anywhere on desktop and check the version by entering command $node -v

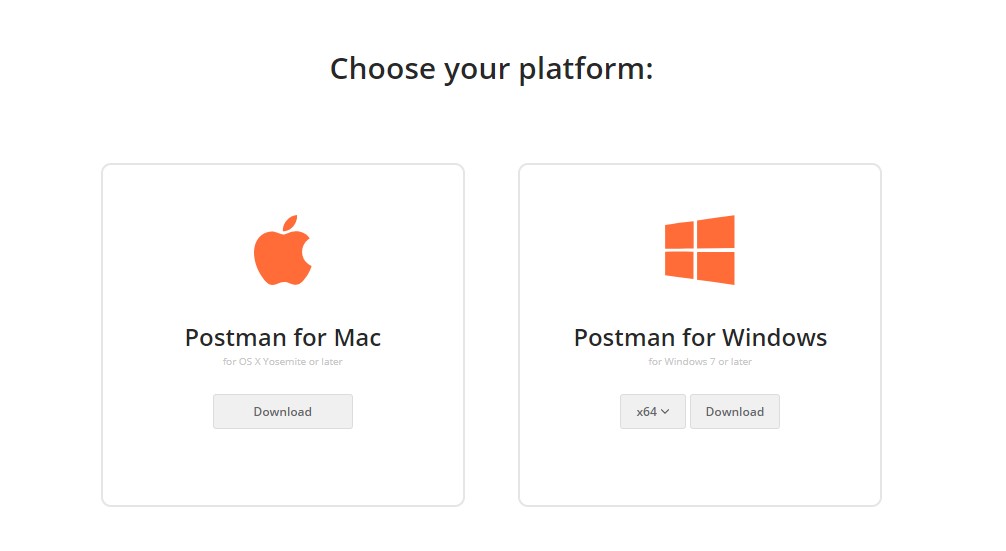


**Installation steps for PostMan**

1. Go to Link: <https://www.getpostman.com/>



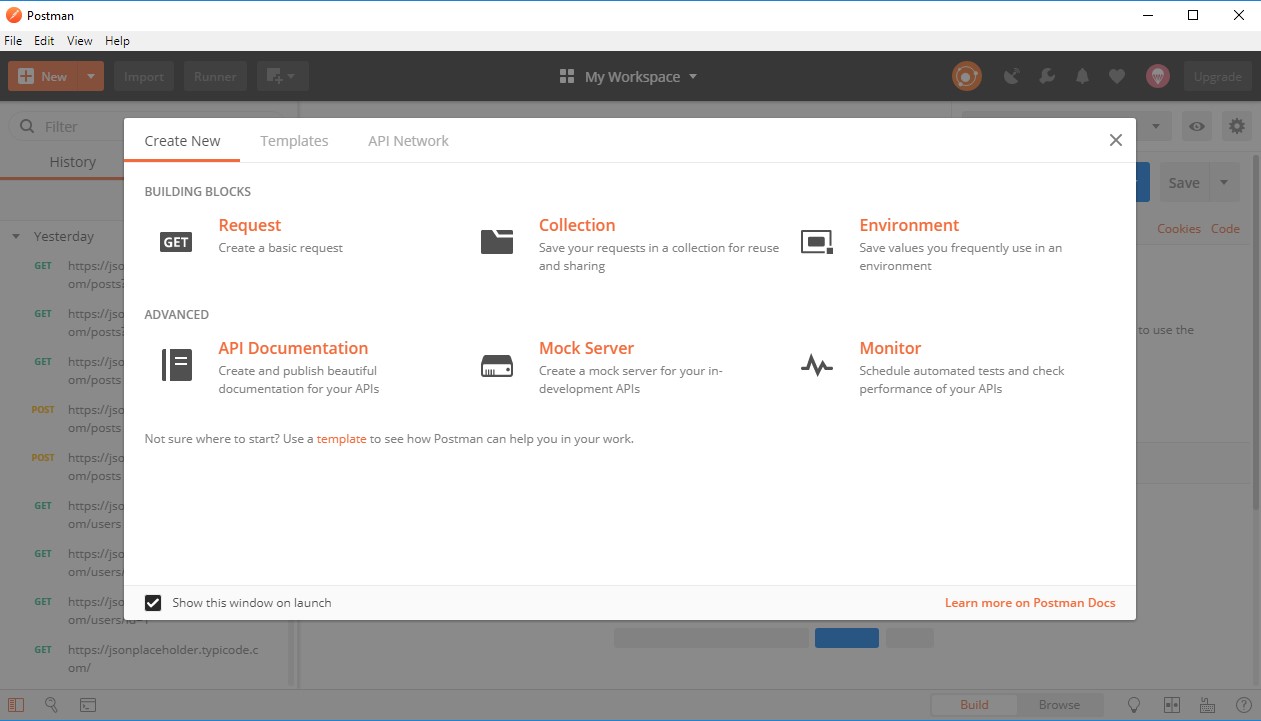
1. Click on the Download the App button



1. Select Windows platform
2. A file starts downloading
3. Open the downloaded file



1. Then a popup window as show below will appear



Dependencies(npm libraries) runs on top of NodeJS so it isn't installed directly on Windows, it's added via NPM (Node Package Manager) when you run "npm install" for an application, "npm install" looks at the dependencies section of a MEAN stack application's package.json file and downloads everything required.

# **Set up the Development Environment**

1. Install the Angular CLI globally.

npm install -g @angular/cli

1. Install Ionic globally

npm install -g ionic

# **Steps to run application**

**Web Application**

1. Clone the repository  
   GitHub Repo Link: <https://github.com/Kishan-Kalburgi/DisasterResponseAndReportingSystem-Backend/tree/Working_Version>
2. Install dependency  
   npm install
3. Start Application  
   npm start
4. Application will be running in hhtp://localhost:4200
5. Login with Admin credentials  
   Email: admin@drrs.com  
   Password: Drrs@1234

**Mobile Application**

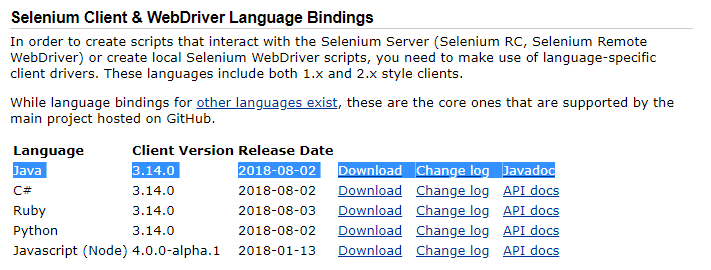
1. Clone the repository  
   GitHub Repo Link: <https://github.com/HemanthNarne/DisasterResponseAndReportingSystem-MobileApp>
2. Install dependency  
   npm install
3. Start Application  
   ionic lab
4. Application will be running in hhtp://localhost:8200
5. Signup and Login

# **Application Testing**

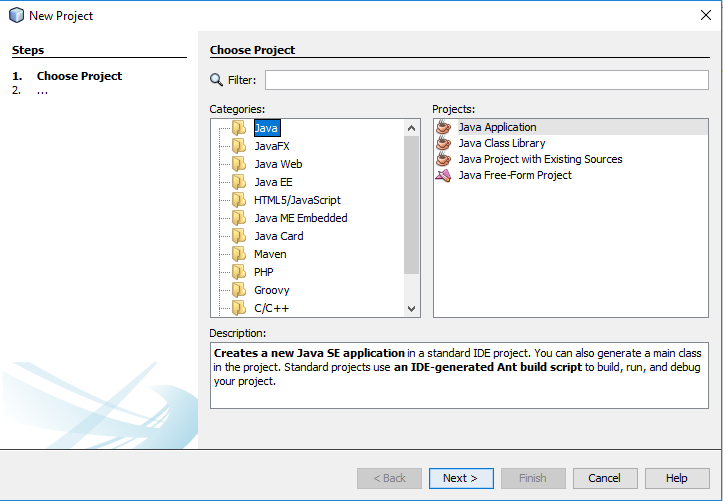
**Tool used: Selenium**

Instruction for using and executing selenium

1. Go to Selenium official website (<https://www.seleniumhq.org/>)
2. Click on downloads tab. Scroll down to Selenium Client and WebDriver Language Bindings.
3. Click on the download link Java programming language and download the highlighted link as shown in the below screenshot.



1. Extract the zipped folder in a specific location.
2. Open NetBeans 🡪Create a new java project 🡪 Create a new class within the package as follows:



A screenshot of a cell phone

Description generated with very high confidence

1. Now include the jar files

Right click on Project name 🡪 go to properties 🡪 Click on libraries 🡪 Click on Add JAR/folder 🡪 add all the jar files from the location where you saved (Point 4).

1. Download the ChromeDriver-WebDriver for Chrome by going through the following link: <https://chromedriver.storage.googleapis.com/index.html?path=2.42/> and download the windows version.
2. Add an input and output files in the location C:\Users\s530945\Documents\NetBeansProjects\SeleniumPHP and input field with data validations and add the below data to the input file and output will be displayed in the output file automatically when program runs.

admin@drrs.com

Drrs@1234

Floods

KansasOverlandPark

There are Floods in KansasCity

James

andrew

admin@drrs.com

Drrs@1234

Fire

SaintJoe

There are Fire accident in SaintJoe

andrew

James

admin@drrs.com

Drrs@1234

Fire

SaintJoe

There are Fire accident in SaintJoe

andrew

James

admin

Drrs@1234

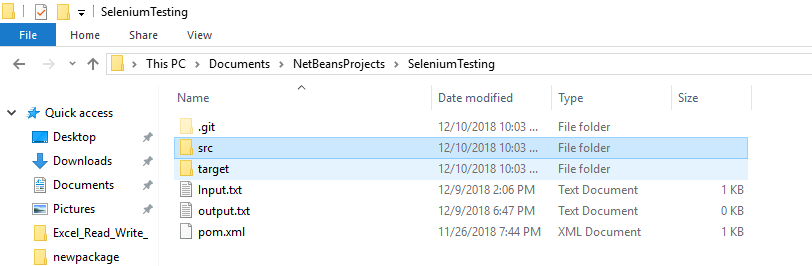
Snow Strom

Savannah

There are Snow Strom in Savannah

andrew

James



1. The input file should consists of the input data
2. Now copy the following code in SeleniumTesting.java

<https://github.com/HemanthNarne/SeleniumProject>

1. Run the program

**Test cases**

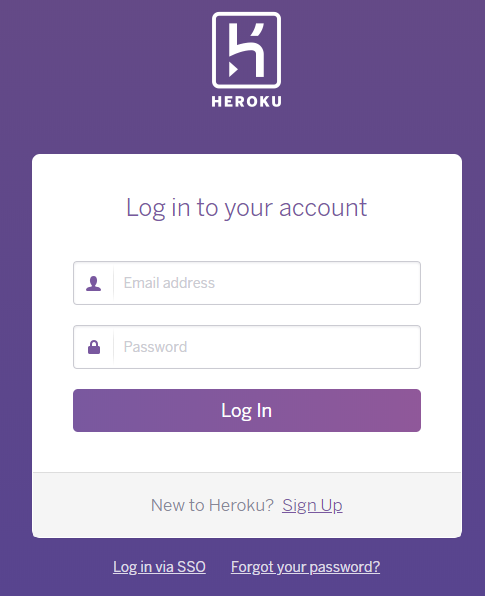
1. User should be able to login in https://drrs.herokuapp.com/#/login website with Active Credentials
2. User should not be able to login in https://drrs.herokuapp.com/#/login website with inactive Credentials
3. User should be able to navigate to dashboard webpage by logging in
4. User should be able to see create incident webpage by clicking on create incident button in dashboard page
5. Webpage should be able to create incident with details entered by the user in the respective textbox when create button is clicked
6. Webpage should be able to navigate to the dashboard page when incident is created.
7. Webpage should be able to display created incident in the dashboard webpage and an popup with an message that incident is created.
8. Website should be able to navigate to the report page of the specific incident when that specific incident report button is clicked
9. User should be able to search for submitted reports by any text in reports page in the filter text box in created report page
10. Website should be able to navigate to the common operating picture page when that specific common operating picture button is clicked in the dashboard
11. Website should be able to navigate to the archived incidents page when archived button is clicked in the dashboard
12. User should be able to get the downloaded file when download button is clicked in the archived field
13. User should be able logoff when logoff button is clicked
14. User should be able to navigate to the login page https://drrs.herokuapp.com/#/login website once user logged off.

# **Deployment**

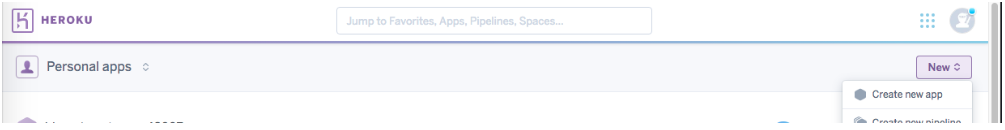
The project involves two different applications: a web application, accessed by EOC and a mobile application, accessed by CERT members. The deployment of web application is done on heroku. Heroku serves as a “Cloud Platfrom as a Service”, generally used to deploy any application so that the developers can access the application from anywhere and run it.

**Procedure to deploy and run the application:**

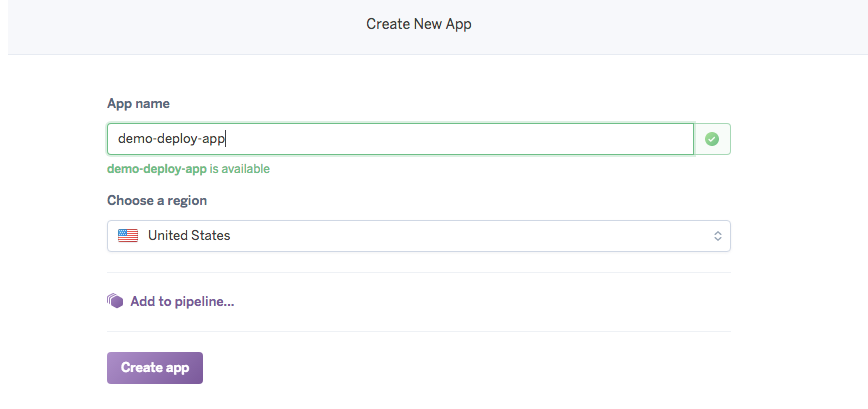
1. GitHub Repo link: <https://github.com/Kishan-Kalburgi/DisasterResponseAndReportingSystem-Backend/tree/Working_Version>
2. Login to your Heroku account with your respective credentials, if not create an account and then login.

****

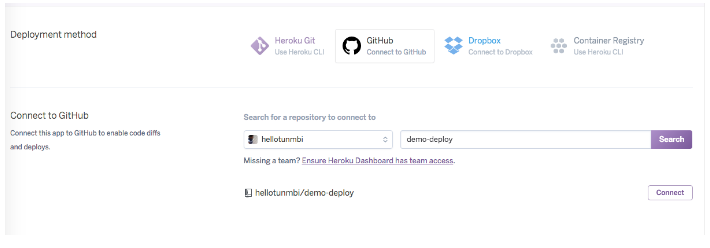
1. Click on “New” present on the top right corner.
2. Select “Create new app”.

****

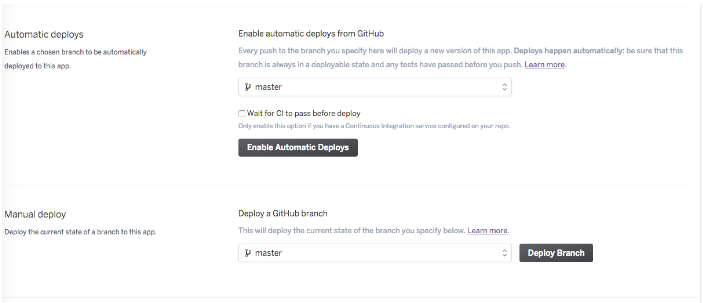
1. Enter the name of the application (Criteria: should contain lower letters, numbers and dashes).

****

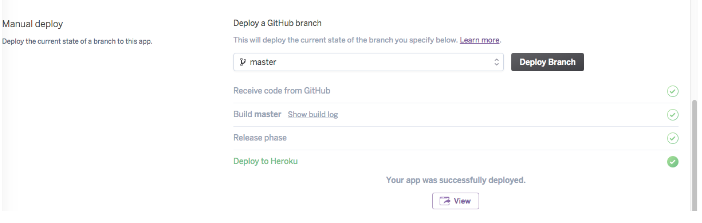
1. Click on “Create app”.
2. Select “GitHub” option and click on the search button in order to connect to the specific repository.

****

1. Under “Automatic Deploys”, select the master branch and click on “Enable Automatic Deploys”.
2. Under “Manual Deploys”, click on “Deploy Branch” in order to push in the fresh code to heroku.

****

1. It takes a little time to deploy the application. Once done a feedback message will be displayed at the bottom.

****

1. In order to run the deployed application, run the “Heroku Git URL”, present in settings, in any browser.

# **Known Incomplete Functionality**

**Web Application**

1. **Access Privileges**

Web Application should be able to provide login access privileges based on three main categories. They are admin, operator and the viewer. Admin is the person who manages the entire application. The operator is the person who will be taking the responsibility of updating, deleting and making necessary modifications. Viewer is the person who will be dealing with Common Operating Picture (COP).

1. **Restore option for archived incidents**

Web application should be able to provide an option to restore the archived incidents. On restoring an archived incident, it has to be displayed back on the dashboard.

1. **Update the archived incidents**

Web application should be able to provide an option to update the archived incidents. Updating the archived incidents would include making necessary changes to that particular incident so that it could be helpful for further use.

1. **Downloading of archived incidents**

The Emergency Operation Center will be able to download the incidents that have been archived, on clicking on the download option present in archived incident page. Here in this situation, we are able to download static excel file when the download option is selected

**Mobile Application**

1. **Update the reports**

Mobile application should be able to provide an option to update the reports. The CERT members generate reports regarding an affected disaster location. On providing an update option in the reports, makes it easy for the CERT members to perform any required necessary changes.

1. **Camera Functionality**

Mobile application should be able to save the images captured. The users who want to be a part of CERT require camera functionality during registration in order to capture their respective certificates and the CERT requires camera functionality to capture the images of affected disaster location. Here in this situation, we are not able to save the captured images so that they can be uploaded previously clicked images

1. **Push Notification**

The Emergency Operation Center is responsible to send notifications to the CERT members once an incident is created on the dashboard, asking about their availability to visit affected disaster location. Here in this situation, we are able to send the notifications from the web application to mobile application which we have manually registered in the backend

1. **Fetching the coordinates of longitude and latitude**

The Mobile application is responsible to fetch the longitude and latitude coordinates. Here in this situation we are able to fetch the coordinates only in Samsung galaxy S9 plus, the issue with other android devices is permission to access location and the browser which supports this functionality is Mozilla Firefox

# **Methodologies**

The project involves agile software development methodology. We attained this methodology through

1. **Daily standup meetings**

The standup meetings are meetings in which all the team members discuss about “what they have done yesterday, what they are going to do today and what impediments did they go through”. By doing this, the team gets to know till what extent the project has been accomplished and what are the remaining aspects to be completed in the project. We have done standup meeting every day in the class in order to discuss the complete and incomplete aspects of the project. For better and clear understanding, we have also used google docs to maintain the details of standup meeting.

1. **Weekly client meetings**

The client meetings are the meetings in which team discusses their progress regarding the project with the client. By doing this, the client gets to know till what extent the project has been accomplished and provides suggestions regarding any required necessary changes. We have been meeting the client every Wednesday in order to discuss our progress and took suggestions and feedback from the client so that we can further work on the project and make necessary required improvements.

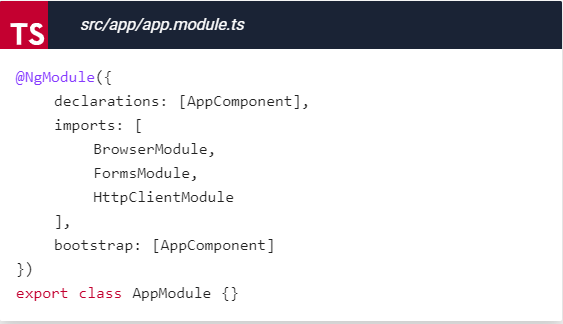
# **Glossary of errors noticed during application development**

**Angular Application (Web Application) common errors and mistakes**

* **Import required Angular Modules**  
    
  **Error:** Can't bind to 'ngModel' since it isn't a known property of 'input'  
    
  This error indicates, that the angular Forms Module has not been imported into your module.

**Error:** Unhandled Promise rejection: No provider for HttpClient!  
  
This error tells you, that you have not imported the angular HttpClient Module into your (root) module.

**Solution:** To resolve the problem, you need to import the missing module into your module. In our case, that module would be the AppModule in our app directory

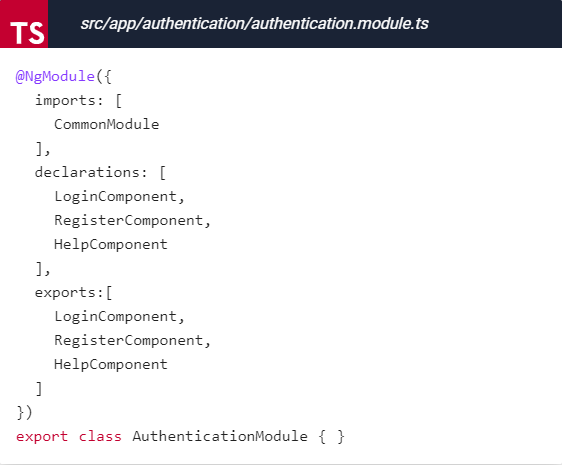


**Note:** Only import the modules you really need! Importing unnecessarily module bloats your application size significantly.

* **Declare Components only once**

**Solution:** Wrap your component into a module. Maybe a module per component is a bit too much, so why don't we create a components module? That module can then be imported into other modules and you can use your components there.

When you do so, make sure that you don't only declare your components in that components module, but to also export them. Otherwise, they can only be accessed from within the module itself.



* **Avoid maintainability issues by wrapping in Services**

It is always good practice, to extract your core business logic into services. That way, it becomes much easier to maintain, as it can be swapped out and replaced by a new implementation in just a few seconds. The same goes for testing. Often times you need services that fetch external data, to fake the results in a test environment. If you fetch your data in services, that is an easy one. If not, make all the lines that need to be changed for that.

When using angular HttpClient. It should always be wrapped inside of a centralized service. That way, it not only stays testable, it is also easy to make changes to it. Imagine, your backend requires a new header to be passed with every request after a recent update. Without a centralized service, you now have to find all the lines across your whole app that are affected. Needless to say, that this would be far from optimal.

Instead, you should always wrap your HTTP-requests into services.

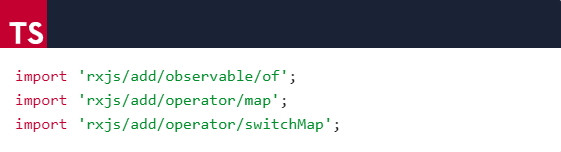
* **Keep your Application Size small by only importing what you need**

**Problem:** Every import statement you use increases the size of your bundle. We are adding more code, so the size goes up. The problem here is, that some libraries are quite huge. When using the wrong import statement, you can end up with the whole library in your application.



This tiny statement almost doubles the size of our application.

**Solution:** If your library offers sub-modules, make sure to only import the stuff you need. Regularly check the resulting bundle using the bundle-analyzer. For example, you need to import every operator you want to use



**Ionic Application (Hybrid Mobile Application) common errors and mistakes**

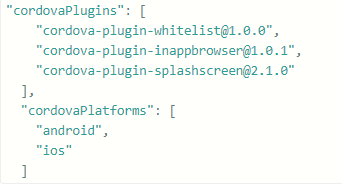
* **Not Using the Ionic CLI to Install Platforms and Plugins**

Ionic CLI adds features to the Cordova CLI. Platforms and plugins persistence is a great feature that Ionic CLI adds.

The problem with Cordova CLI is that the platforms and plugins you install are installed on your machine only. When working on a team, to avoid bugs you want to share the same environment, platforms, and plugins. With Cordova CLI, it’s harder to keep the project in sync between developer’s machines. Yes, you could commit the platforms and plugins folders, but it is not recommended.

When using Ionic CLI to install platforms ionic platform add ios and plugins ionic plugin add camera, the package.json file is edited appropriately.

Platforms and plugins are stored in cordovaPlatforms and cordovaPlugins properties:



It is now easy for other developers to get in sync when pulling a new code, simply by running ionic state restore when necessary (addition, deletion or version update).

* **Prototyping Ionic Applications Manually**

Ionic has a specific design, almost a visual language. Especially with prototypes and early-stage products, a lot of time and expenses can be saved by utilizing the available components and styles. They are actually rather minimal and have a good aesthetic.

Presenting wireframes and mockups with basic functionality has become an industry standard. To see a picture and to see an actual app with dynamic components on a mobile device are two very different cups of tea. Many designers, and also UX developers, use tools like Adobe XD, or Axure, which allow quickly making wireframes with minimal functionality.

Now, the creators of Ionic released a similar tool made exclusively for Ionic developers. It is called Ionic Creator. It has a drag and drop web interface, and supports close to everything core Ionic provides. What’s great about it is that it allows to export the prototype into several formats, with standard working Ionic code, and even build the application and share it. The tool is proprietary, but many of the options are free to use.

