# **Steps for Migration from Angular5 to Angular6**

## **Before Updating**

* Switch from **HttpModule** and the **Http** service to **HttpClientModule** and the **HttpClientservice**. HttpClient simplifies the default ergonomics (You don't need to map to json anymore) and now supports typed return values and interceptors.

## **During the Update**

* Make sure you are using [**Node 8 or later**](http://www.hostingadvice.com/how-to/update-node-js-latest-version/)
* Update your Angular CLI globally and locally, and migrate the configuration to the **new** **angular.json** **format** by running the following:

**npm install -g @angular/cli  
npm install @angular/cli  
ng update @angular/cli**

* Update all of your Angular framework packages to v6, and the correct version of RxJS and TypeScript.

**ng update @angular/core**

After the update, TypeScript and RxJS will more accurately flow types across your application, which may expose existing errors in your application's typings

* Update Angular Material to the latest version.

**ng update @angular/material**

This will also automatically migrate deprecated APIs.

* Use **ng update** or your normal package manager tools to identify and update other dependencies.

## **After the Update**

* Remove deprecated RxJS 6 features using rxjs-tslint auto update rules.  
    
  For most applications this will mean running the following two commands:  
    
  **npm install -g rxjs-tslint  
  rxjs-5-to-6-migrate -p src/tsconfig.app.json**
* Once you and all of your dependencies have updated to RxJS 6 install rxjs-compat using the command :

**npm install rxjs@6 rxjs-compat@6 –save**

* Make changes to import **keyframes** **modules** from **“@angular/animations**” in “**app.modules.ts**”.
* Make sure all the imports for **HttpClientModule is** from **“@angular/common/http**”.
* Now you can deploy the code at local using :

**npm install**

**npm start**

## **References :**

<https://update.angular.io/>

<https://github.com/angular/angular-cli/issues/4756>