Angular Installation

11/02/2023 6:32 am EDT

Introduction

To get your Autoaddress plugin up and running, do the following:

1. Follow the Account Center Sign Up documentation to get set up with an account
2. Generate an API key by following the instructions provided in the Creating an Integration documentation
3. Follow the guide below to seamlessly integrate the Autoaddress plugin into your Angular application

Setup

Start by installing the Autoaddress package in your project:

For more information can view our npm package here.

Installing the Package

If you are using npm: npm install @autoaddress.com/aa3-control

Or if you are using yarn: yarn @autoaddress.com/aa3-control

Importing the js & css files

In the build section of your angular.json file, add these references to the installed Autoaddress js and css files as shown below:

```
"styles": [      "node_modules/@autoaddress.com/aa3-control/autoaddress/autoaddress-style.css", ],
"scripts": [     "node_modules/@autoaddress.com/aa3-control/autoaddress/autoaddress.min.js", ]
```

Importing the asset files

Add the code block shown below to the asset section of the angular.json in your build script

```
"assets": [
  {
"glob": "**/*",
          "input": "./node_modules/@autoaddress.com/aa3-control/autoaddress/assets/images/",
          "output": "./assets/images/"
  },
  "src/favicon.ico",
  "src/assets"
],
```

Placing the Component

In your desired Angular component, declare the Autoaddress function in your component's typescript file (*.component.ts file) as shown in the example component below. The example demonstrates intializing the Autoaddress control, setting the options and using callbacks.

```
import { Component, OnInit } from '@angular/core';

declare function Autoaddress(options: any): void

@Component({
selector: 'app-custom',
templateUrl: './custom.component.html',
styleUrls: ['./custom.component.scss']
})

export class CustomComponent implements OnInit {
control: any;
constructor() { }
ngOnInit(): void {
this.control = Autoaddress({
apiKey: "YOUR_KEY",
elementId: "aa-control",
onPostLookup: function (result: any) {
console.log(result);
}
});
}
}
```

Once Autoaddress has been set up, got to the html part of your component (*.component.html). and add the following:

```
<div id="aa-control"></div>
```

As a minimum requirement, an **apiKey** and **elementId** will have to be provided to send requests to the Autoaddress control.

Configure

The Autoaddress component is event driven and we have provided the opportunity for you to hook into these

events and add your own logic with custom callback functions. These events range from when the component is initialised all the way to the end when your address has been found.

We also pass data back in key events so you can grab these data and perform custom actions to suit your own needs.

You can access data in key events and perform custom actions tailored to your needs.

For a comprehensive list of settings and configurations that you can pass as configurations, please refer to the following resources:

- Settings
- Callbacks
- Functions

Styling

The Autoaddress control comes with default styling provided using standard CSS.

If you need to customize the design, you can easily achieve this by adding your own custom CSS rules or even removing the Autoaddress styles entirely.

For more information around styling please see our styling section here.