MageINIC

City Region Postcode GraphQl

FEATURES

- The City Region Postcode module offers GraphQL to efficiently manage and display customer City Region Postcode on websites.

How to install the extension

1. Install via composer (recommend)

- Enable maintenance mode **php bin/magento maintenance:enable**
- Run the command **composer require mageinic/city-region-postcode-graphql**
- Run the command **php bin/magento module:enable MageINIC_Core MageINIC_CityRegionPostcode**
- Run the command **php bin/magento setup:upgrade** to install the extension
- Run the command **php bin/magento cache:clean** to clean the cache
- Disable maintenance mode **php bin/magento maintenance:disable**
- Check the store environment if everything is working properly. If, at any point, the store displays an error, check the Magento logger for more details.

2. Upload via ZIP/Archive

- Back up your store's database and web directory
- Download the extension zip file and extract the zip file
- Transfer the **MageINIC** folder to the Magento 2 **app/code/** directory on the server
- Enable maintenance mode **php bin/magento maintenance:enable**
- Run the command **php bin/magento module:enable MageINIC_Core MageINIC_CityRegionPostcodeGraphQl**
- Run the command **php bin/magento setup:upgrade** to install the extension
- Run the command **php bin/magento cache:clean** to clean the cache
- Disable maintenance mode **php bin/magento maintenance:disable**
- Check the store environment if everything is working properly. If, at any point, the store displays an error, check the Magento logger for more details.

Store Configuration

- Please include the following query in the body section to fetch the module store configuration values.

  End Point : {{baseurl}}/graphql
  Method: POST

  Request:

  ```
  {
    storeConfig {
      city_region_postcode_module_enabled
      country_searchable
      city_enable
      city_searchable
      postcode_enable
      postcode_searchable
      region_searchable
    }
  }
  ```

  Response :

  ```
  {
    "data": {
      "storeConfig": {
        "city_region_postcode_module_enabled": 1,
        "country_searchable": 0,
        "city_enable": 0,
        "city_searchable": 0,
        "postcode_enable": 0,
        "postcode_searchable": 0,
        "region_searchable": 0
      }
    }
  }
  ```

You receive store configuration data set on the backend, including:

- **city_region_postcode_module_enabled** - Enable the CityRegionPostcode module functionality in the Website using the Yes/No option.
- **country_searchable** - you can easily search the country on the dropdown list using the Yes/No option.
- **city_enable** - if you want to show a list of cities showing on the dropdown on frontend you need to select Yes option and if you don't want then need to select No option.
- **city_searchable** - you can easily search the country on the dropdown list using the Yes/No option.
- **postcode_enable** - if you want to show a list of cities showing on the dropdown on frontend you need to select Yes option and if you don't want then need to select No option.
- **postcode_searchable** - you can easily search the country on the dropdown list using the Yes/No option.
- **region_searchable** - you can easily search the country on the dropdown list using the Yes/No option.

Note 💬

- In the Yes/no dropdown, the value of "Yes" corresponds to "1," while the value of "No" corresponds to "0."

City GraphQl Query

1. CityFiltered List

- To implement store-specific city filtering, it is essential to configure the Header with the **Key:** "Store" and **Value:** "Default" (using your Store Code). This setup ensures effective segregation of the city based on individual store contexts.

Request:

```
{
  CityInfo (filter: {region_id:{eq:"580"}}, pageSize: 5, currentPage: 1, sort: {}) {
    items {
      id
      code
      country_id
      default_name
      region_id
    }
    total_count
  }
}
```

Response :

```
{
  "data": {
    "CityInfo": {
      "items": [
        {
          "id": 1,
          "code": "AH",
          "country_id": "IN",
          "default_name": "Ahmedabad",
          "region_id": "580"
        }
      ],
      "total_count": 3
    }
  }
}
```

Postcode GraphQl Query

2. PostCode Filtered List

- Get List Of Postcode By City id you can easy to filter postcode data as well as your able To set sort order as per your requirement.

Request:

```
{
  PostcodeInfo(filter: {}, pageSize: 5, currentPage: 1, sort: {}) {
    items {
      id
      city_id
      country_id
      postcode
      region_id
    }
    total_count
  }
}
```

Response :

```
{
  "data": {
    "PostcodeInfo": {
      "items": [
        {
          "id": 1,
          "city_id": "1",
          "country_id": "IN",
          "postcode": "382440",
          "region_id": "580"
        }
      ],
      "total_count": 2
    }
  }
}
```