

# Query Examples with the Content Delivery API

## Query by URL

```
const getContentByUrl = async ({ contentType, slug, refUids, locale, filter }) => {
  try {
    const Query = global.Stack
      .ContentType(contentType)
      .Query()
      .language(locale)
    if (refUids) {
      let refs = refUids.replace(/ /g, "").split(',')
      Query.includeReference(refs)
    }
    let result = await Query
      .includeFallback()
      .where('url', `/${slug}`)
      .toJSON()
      .includeEmbeddedItems()
      .addParam('include_metadata', 'true')
      .find()
    if (result && result.length > 0 && result[0].length <= 0) { // for trailing slash match
      result = await Query
        .includeFallback()
        .where('url', `/${slug}/`)
        .toJSON()
        .includeEmbeddedItems()
        .addParam('include_metadata', 'true')
        .find()
    }

    (global.env === 'preview' || global.env === 'staging-preview' || global.env === 'development') ?
    Contentstack.Utils.addEditableTags(result[0][0], contentType, true, locale) : "
    //metadata key required for _metadata in cms entry data
    if(result.length > 0 && result[0].length > 0) {
      result[0][0] = {...result[0][0], metadata: result[0][0]}
    }
    if (result) return result[0][0]
  }
  catch (error) {
    throw new Error(error)
  }
}
```

```
}  
}
```

## Query by UID

```
const getContentByUID= async ({ contentType, entryUid, refUids, locale }) => {  
  try {  
    const Query = global.Stack  
      .ContentType(contentType)  
      .Entry(entryUid)  
      .language(locale)  
      .includeFallback()  
      .includeEmbeddedItems()  
    if (refUids) {  
      let refs = refUids.replace(/ /g, "").split(',')  
      Query.includeReference(refs)  
    }  
    let result = await Query.toJSON().fetch()  
  
    if (result) return result  
  } catch (error) {  
    return error  
  }  
}
```

## Example to Get All Entries

```
const getEntries = async ({ contentType, refUids, locale }) => {  
  let collectionData  
  try {  
    const Query = global.Stack.ContentType(contentType)  
      .Query()  
      .language(locale)  
    if (refUids) {  
      let refs = refUids.replace(/ /g, "").split(',')  
      Query.includeReference(refs)  
    }  
    let result = await Query  
      .includeFallback()  
      .toJSON()  
      .includeEmbeddedItems()  
      .addParam('include_metadata', 'true')  
      .find()  
  
    if (result && result.length > 0 && result[0].length <= 1) { //for entry type = single  
      (global.env === 'preview' || global.env === 'staging-preview' || global.env === 'development') ?  
      Contentstack.Utils.addEditableTags(result[0][0], contentType, true, locale) : "  
    }  
  }  
}
```

```
//metadata key required for _metadata in cms entry data
result[0][0] = {...result[0][0], metadata: result[0][0]}
return result[0][0]
}

//for entry type = multiple
(global.env === 'preview' || global.env === 'staging-preview' || global.env === 'development') ?
Contentstack.Utils.addEditableTags(result[0][0], contentType, true, locale) : "
  if (result) return result
}
catch (error) {
  res.status(500).send(`Internal Server Error\n: ${error}`)
  // throw new Error(error)
}
}
```