Algolia

Build Unique Search Experiences

AUTO-COMPLETE

INSTANT RESULTS PAGE

MOBILE SEARCH

What is Algolia

Algolia is a hosted search engine ... you provide the data and it provides an api to for your app to use

Why would you want to use it?

Auto-complete

Instant Results

Spell Checking

It’s fast

Easy ranking and tuning

Multiple Languages

Geo-Search

Automatic Indexing

Schema-less data objects

Trending Results

Advanced Query filters

Personalization

Synonyms/Natural Language

Don’t re-invent the wheel

Set it and forget it :)

Monthly  Annually

COMMUNITY	STARTER	GROWTH	PRO	ENTERPRISE ∞
FREE	$59 /month	$299 /month	$999 /month	Contact us Billed annually
10,000 RECORDS	100,000 * RECORDS	1,000,000 * RECORDS	5,000,000 * RECORDS	Dedicated infrastructure
100,000 OPERATIONS /month	1,000,000 * OPERATIONS /month	10,000,000 * OPERATIONS /month	50,000,000 * OPERATIONS /month	Dedicated Solutions Engineer & Customer Success Manager
SUPPORT Forum	SUPPORT Forum + Email	SUPPORT Forum + Email	SUPPORT Forum + Email Live Chat	Premium onboarding
	* Overquota Allowed compare plans	ANALYTICS 30 days	ANALYTICS 90 days	Contextual search
		INFRASTRUCTURE Distributed Search Network	INFRASTRUCTURE Distributed Search Network	Sandbox environment
Community plans must include the logo: Search by algolia				Monitoring API
				Analytics API
				99.99% SLA
				Custom terms of use

APIs and Integrations

API Clients

PHP | Ruby | JavaScript | Python | iOS | Android | C# | Java | Go | Scala

Frameworks

Integrate with your favorite framework

Rails | Symfony | Django | Laravel

Search UI Libraries

Projects to help you build great search UI

instantsearch.js | React InstantSea... | autocomplete.js | Helper JS

Integrations

Open source & 3rd party integrations

Magento | Wordpress | Zendesk | Shopify | Firebase

Tools

Pre-made search

DocSearch | Places

REST API

Full API reference

Monitoring | Analytics | Search

How to get data into Algolia

- You have to import your data into algolia (everything is run on their backend)
- This means you are responsible for keeping your data up to date
  - Remove items that are no longer relevant
  - Update items if their description changes
  - Insert items if they are new
- You can do this through their REST API or web interface

DEMO: https://github.com/b-nroths/algolia-firebase-cloud-functions/blob/master/README.md

Query Settings (set on each “table”)

searchableAttribures - what items in the json do you want to search on

unretrieveableAttributes - don’t return these to customer (ex. # sales)

attributesToRetrieve - list of attributes to return

attributesForFaceting - kind of like categories

customRanking - control how results are ranked ex [desc(popularity), asc(price)]

typoTolerance

ignorePlurals

AND MORE....

SHOW: https://www.algolia.com/explorer#?index=items&tab=ranking

Adding Searching in your App

CocoaPods

- SearcherCore - Swift wrapper for Algolia Rest API
  - Add data
  - Set Index Settings
  - Query
- InstantSearcher - Extra utils to make library easier to use such as
  - Session Management
  - Pagination
  - Continuous scrolling (pre-fetching)
  - Highlighting
  - “Debouncing”

DEMO: iOS APP

```swift
import AlgoliaSearch
import InstantSearchCore
import AFNetworking
import UIKit


class SearchTableViewController: UITableViewController, UISearchBarDelegate, UISearchResultsUpdating, SearchProgressDelegate {

    var searchController: UISearchController!
    var searchProgressController: SearchProgressController!

    var itemSearcher: Searcher!
    var itemHits: [JSONObject] = []
    var originIsLocal: Bool = false

    override func viewDidLoad() {
        super.viewDidLoad()

        // Algolia Search
        itemSearcher = Searcher(index: AlgoliaManager.sharedInstance.itemsIndex, resultHandler: self.handleSearchResults)
        itemSearcher.params.hitsPerPage = 15
        itemSearcher.params.attributesToRetrieve = ["name", "yelp_rating", "merchant_logo", "review_count", "description"]
        itemSearcher.params.attributesToHighlight = ["name", "description"]

        // Search controller
        searchController = UISearchController(searchResultsController: nil)
        searchController.searchResultsUpdater = self
        searchController.dimsBackgroundDuringPresentation = false
        searchController.searchBar.delegate = self
        searchController.searchBar.placeholder = NSLocalizedString("Search for food", comment: "")

        // Add the search bar
        tableView.tableHeaderView = self.searchController!.searchBar
        definesPresentationContext = true
        searchController!.searchBar.sizeToFit()

        // Configure search progress monitoring.
        searchProgressController = SearchProgressController(searcher: itemSearcher)
        searchProgressController.delegate = self

        // First load
        updateSearchResults(for: searchController)
    }
```

```swift
// MARK: - Search
func updateSearchResults(for searchController: UISearchController) {
    print("new search")
    print(searchController.searchBar.text ?? "hi")
    itemSearcher.params.query = searchController.searchBar.text
    itemSearcher.search()
}

private func handleSearchResults(results: SearchResults?, error: Error?) {

    guard let results = results else { return }
    if results.page == 0 {
        itemHits = results.hits
    } else {
        itemHits.append(contentsOf: results.hits)
    }
    print(itemHits)
    originIsLocal = results.content["origin"] as? String == "local"
    self.tableView.reloadData()
}


override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "resultsCell", for: indexPath) as! ResultsCell

    // Load more?
    if indexPath.row + 5 >= itemHits.count {
        itemSearcher.loadMore()
    }

    // Configure the cell...
    let item = Record(json: itemHits[indexPath.row])
    print(item)
    cell.nameLabel?.text = item.name
    cell.yelpLabel?.text = "\(String(describing: item.yelp_rating))"
    cell.descriptionLabel?.text = item.description
    cell.imageView?.cancelImageDownloadTask()
    if let url = item.merchant_logo {
        cell.imageView?.setImageWith(url)
    }
    else {
        cell.imageView?.image = nil
    }
    cell.backgroundColor = originIsLocal ? AppDelegate.colorForLocalOrigin : UIColor.white

    return cell
}
```