



DEVELOPER SUMMIT

March 10–13

Building a Modern Geocoding Experience in your Applications using Suggestions and Categories

Christa Hash
Brad Niemand

Outline

- **What is geocoding?**
- **Modernizing the geocoding experience**
 - **Suggestions**
 - **Categories**
- **Putting it all together with the geocoding REST API**
- **Enhancing runtime SDKs with additional geocoding capabilities**
- **Additional resources**

- **NOTE: Sample app and demo will be iOS, but the same principles apply across the runtime platforms. Wherever you can leverage json requests in runtime, you can implement this functionality.**

What is geocoding?

- **Geocoding : Address/POI → Point on map**
- **Reverse geocoding: Point on map → Address/POI**

What makes a geocoding app effective?

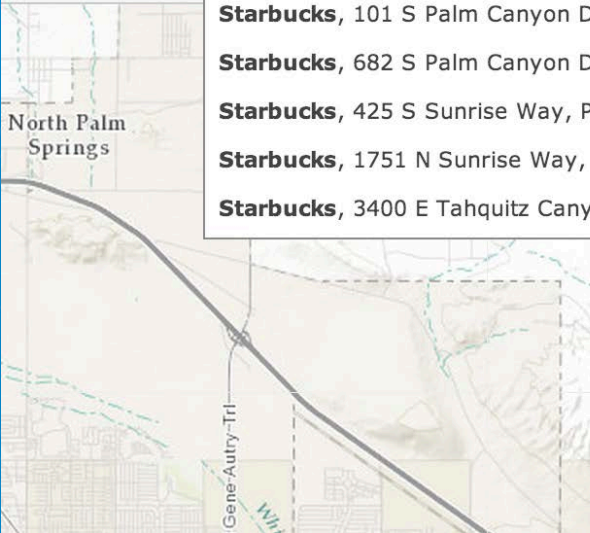
- **Speed**
- **Accuracy**
- **Autocomplete functionality (Suggestions)**
- **Narrowing down a search (Categories)**

MODIFY MAP [Sign In](#)



Measure

starbucks  

- Starbucks**, 101 S Palm Canyon Dr, Palm Springs, California
- Starbucks**, 682 S Palm Canyon Dr, Palm Springs, California
- Starbucks**, 425 S Sunrise Way, Palm Springs, California
- Starbucks**, 1751 N Sunrise Way, Palm Springs, California
- Starbucks**, 3400 E Tahquitz Canyon Way, Palm Springs, C...



Enter a place or address to Geocode

 starbucks  [Cancel](#)



DEMO

Suggest

- The geocoding REST API for the World Geocoding Service includes a method which allows character-by-character auto-complete suggestions to be generated for user input.
- **Output properties:**
 - `text`
 - `magicKey`
 - `isCollection`
- Can take an optional `category` parameter, among others

Categories

- The geocoding REST API for the World Geocoding Service provides the ability to filter out unwanted geocode results with the `category` parameter.
- Supported operations:
 - `findAddressCandidates`
 - `geocodeAddresses`
 - `suggest`

findAddressCandidates

- The geocoding REST API for the World Geocoding Service offers the `findAddressCandidates` operation in order to geocode one location per request.

- Request URL:

```
http://geocode.arcgis.com/arcgis/rest/services/World/GeocodeServer/findAddressCandidates?<PARAMETERS>
```

- Input parameters (relevant to today's demo):

- `SingleLine`
- `magicKey`
- `searchExtent`
- `location and distance`
- `category`
- `maxLocations`

Enter a place or address to Geocode

starbucks



Cancel

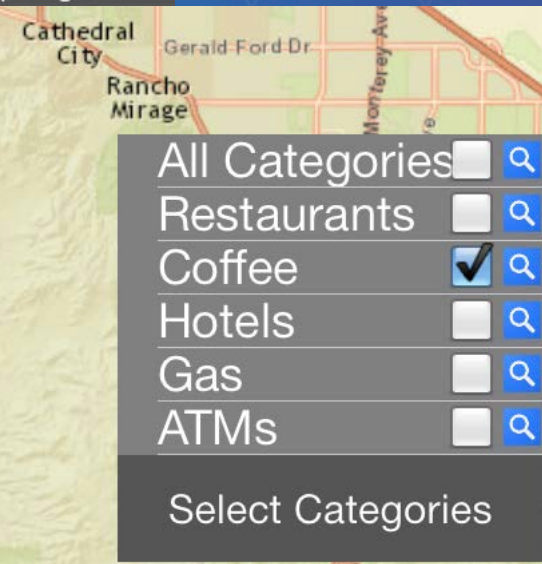
Starbucks, 101 S Palm Canyon Dr, Palm Springs,...

Starbucks, 425 S Sunrise Way, Palm Springs, Cali...

Starbucks, 682 S Palm Canyon Dr, Palm Springs,...

Starbucks, 1751 N Sunrise Way, Palm Spr

Starbucks, 3400 E Tahquitz Canyon Way,



DEMO

Geocoding in iOS (standard SDK)

- The iOS SDK provides methods to geocode against the world service without having to manually create the REST call.

```
//set values for AGSLocatorFindParameters
AGSLocatorFindParameters *parameters = [[AGSLocatorFindParameters alloc] init];
parameters.text = self.searchBar.text; //set the text to search for(user input)
parameters.outSpatialReference = self.mapView.spatialReference; //use map's SR
parameters.outFields = @"*"; //return all outFields
AGSPoint *location = [[AGSPoint alloc] initWithX:-116.538366 y:33.825678
    spatialReference:self.mapView.spatialReference]; //create point for desired location
parameters.location = location; //location to prioritize for search
parameters.distance = 30000; //search radius from location to prioritize for search

//perform search using the sdk method without any category or suggestions parameters
[self.locator findWithParameters:parameters];
```

- In order to generate suggestions or to include suggest/magicKey or category as parameters, you need to do something a little different...

Geocoding in iOS (using JSON request)

- The runtime SDKs do not currently support passing suggestion/magicKey or category directly as parameters because this functionality is so new (They will support it in the next release).
- BUT...You can customize your code to use the functionality now!

```
//First generate the string for the json request
NSString *categoriesString = @"coffee shop,gas station,food";
NSString *suggestionText = @" Grill A Burger of the Desert, 166 N Palm Canyon Dr, Palm
    Springs, California"; //generated from suggest operation
NSString *magicKey =
    @"GST7YMc0AM9UOsE9HhFtGTyVGST7YMc0AM9UOsE9DbTVHgA9HhB0Zcp0OhNtGMytaikZQskKQsk
    EUBdtIS9GOghnYnwZJQTcJN49IjbuGZpm7PWcAFcpOh9bZgKZQoc3YSyaagDIZhkZO1FF"; //generated
    from suggest operation
NSString *searchString = [NSString stringWithFormat:@"http://geocode.arcgis.com/arcgis/
    rest/services/World/GeocodeServer/findAddressCandidates?SingleLine=%@&category=
    %@&outFields=*&location=-116.538366,33.825678&distance=100&magicKey=%@&f=json",
    suggestionText, categoriesString, magicKey];
NSString *encodedSearchString = [searchString
    stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
NSURL *searchUrl = [[NSURL alloc] initWithString:encodedSearchString];
```

Geocoding in iOS (using JSON request)

- The runtime SDKs do not currently support passing suggestion/magicKey or category directly as parameters because this functionality is so new (They will support it in the next release).
- BUT...You can customize your code to use the functionality now!

```
//Then create the request operation and add selectors for success and failure
NSMutableDictionary *params = [NSMutableDictionary dictionaryWithObjectsAndKeys:
                               @"json", @"f",
                               nil];

NSAGSJSONRequestOperation *requestOp = [[AGSJSONRequestOperation
    alloc]initWithURL:searchUrl queryParameters:params];
requestOp.target = self;
requestOp.action = @selector(requestOp:completedWithResultsGeocoder:);
requestOp.errorAction = @selector(requestOp:failedNoResultsGeocoder:);
[[AGSRequestOperation sharedOperationQueue] addOperation:requestOp];

-(void) requestOp:(NSOperation*)op completedWithResultsGeocoder:(NSDictionary*)results {
    ... //parse results and display on map }
-(void) requestOp:(NSOperation*)op failedNoResultsGeocoder:(NSDictionary*)results {
    ... //get error description to display for user }
```

Generating Suggestions in iOS (using JSON request)

- The runtime SDKs do not currently support generating suggestions because this functionality is so new (They will support it in the next release).
- BUT...You can customize your code to use the functionality now!

```
//First generate the string for the json request
NSString *categoriesString = @"coffee shop,gas station,food";
NSString *currentSearchInput = @self.searchBar.text; //set the text to get suggestions
for(user input)
NSString *suggestString =[NSString stringWithFormat:@"http://geocode.arcgis.com/arcgis/
rest/services/World/GeocodeServer/suggest?text=%@
%&location=-116.538366,33.825678&distance=10000&f=json",currentSearchInput,
categoriesString]; // categories are optional, but they can enhance the results
NSString *encodedSuggestString = [suggestString
stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
NSURL *suggestUrl = [[NSURL alloc] initWithString:encodedSuggestString];
```

Generating Suggestions in iOS (using JSON request)

- The runtime SDKs do not currently support generating suggestions because this functionality is so new (They will support it in the next release).
- BUT...You can customize your code to use the functionality now!

```
//Then create the request operation and add selectors for success and failure
NSMutableDictionary *params = [NSMutableDictionary dictionaryWithObjectsAndKeys:
                               @"json", @"f",
                               nil];

NSAGSJSONRequestOperation *requestOp = [[AGSJSONRequestOperation
    alloc]initWithURL:searchUrl queryParameters:params];
requestOp.target = self;
requestOp.action = @selector(requestOp:completedWithResultsGeocoder:);
requestOp.errorAction = @selector(requestOp:failedNoResultsGeocoder:);
[[AGSRequestOperation sharedOperationQueue] addOperation:requestOp];

-(void) requestOp:(NSOperation*)op completedWithResultsGeocoder:(NSDictionary*)results {
    ... //parse results and display on map }
-(void) requestOp:(NSOperation*)op failedNoResultsGeocoder:(NSDictionary*)results {
    ... //get error description to display for user }
```

Additional resources

- **Geocoding REST API doc:**
 - <https://developers.arcgis.com/rest/geocode/api-reference/overview-world-geocoding-service.htm>
- **Suggest doc:**
 - <https://developers.arcgis.com/rest/geocode/api-reference/geocoding-suggest.htm>
- **Categories doc:**
 - <https://developers.arcgis.com/rest/geocode/api-reference/geocoding-category-filtering.htm>
- **iOS SDK info:**
 - <https://developers.arcgis.com/ios/>
- **iOS code samples:**
 - <https://github.com/Esri/arcgis-runtime-samples-ios>
- **Sample project from demo:**
 - <https://github.com/Esri/geocoding-with-suggestions-and-categories>

Rate This Session

www.esri.com/RateMyDevSummitSession