Land Use Permits II

Modify Land Use Permits I to download the live land use permit application data from Seattle's Department of Planning and Development. Download the JSON format, parse the data, and populate your Core Data datastore. The web interface to the data is https://data.seattle.gov/Permitting/Land-Use-Permits/uyyd-8gak. Retrieve the JSON formatted data at https://data.seattle.gov/api/views/uyyd-8gak/rows.json?accessType=DOWNLOAD.

Functionality and architecture

In addition to the requirements for Permits I, add a "Fetch It" button to your UI to trigger the fetch and parse.

Your download and parse method or methods can be synchronous (blocking the UI) or asynchronous (running in the background).

Notes

If you go the asynchronous route, you'll need an object to be the delegate of the NSURLRequest. You could use your NSPersistentDocument subclass for that, but that would lock you in to OS X and to a GUI. If you want your code to be portable to command line code or to iOS, you'll need a separate class, and an instance of that class, to manage the download and parsing.

I've broken the download and the parse/load into two separate methods. By doing this, I've made the parse/load testable (you can pass it a known NSData and test for predicted results), and I've made the API more amenable to asynchronous loading.

The textbook's example is a data source that returns only dictionaries and subdictionaries. The Seattle data source for this project is more complicated. As far as I have been able to tell, it looks like this:

- root node is a dictionary with two keys, "data" and "meta".
- "data" is an array of arrays. Each element of the "data" array corresponds to one row in the web-based display, and is an array of 25 elements. The first 7 of those elements seem to correspond to some internal server-side database information. The remaining 18 correspond to the visible columns.
- "meta" is a dictionary with only one key, "view".
- "view" is a dictionary with about 30 keys describing the database, some of it apparently internal.
- The "columns" element of "view" is a dictionary of dictionaries, and describes the individual columns presented in "data". But the "position" element in each "columns" dictionary is off by 6. It seems to indicate that column 2 is the Application/Permit Number. But actually, that's found in element 8 of each "data" member.

In class next week, we'll walk through ways to make the import operation seem faster, be faster, and be sparing of resources. We'll also take a look at better ways to present the information.

Optional Challenges

Provide some sort of download/parse/insert progress feedback.

Modify the file handling so that the Save dialog does not prompt for a file type, but instead saves to a file with an extension of your choosing.

Add a map to the detail view, showing the property's location. You could do this fairly simply with a WebView and your favorite online mapping service.

Add a button to open the application's status page in the default web browser.

Add a web view that shows the application's status page.

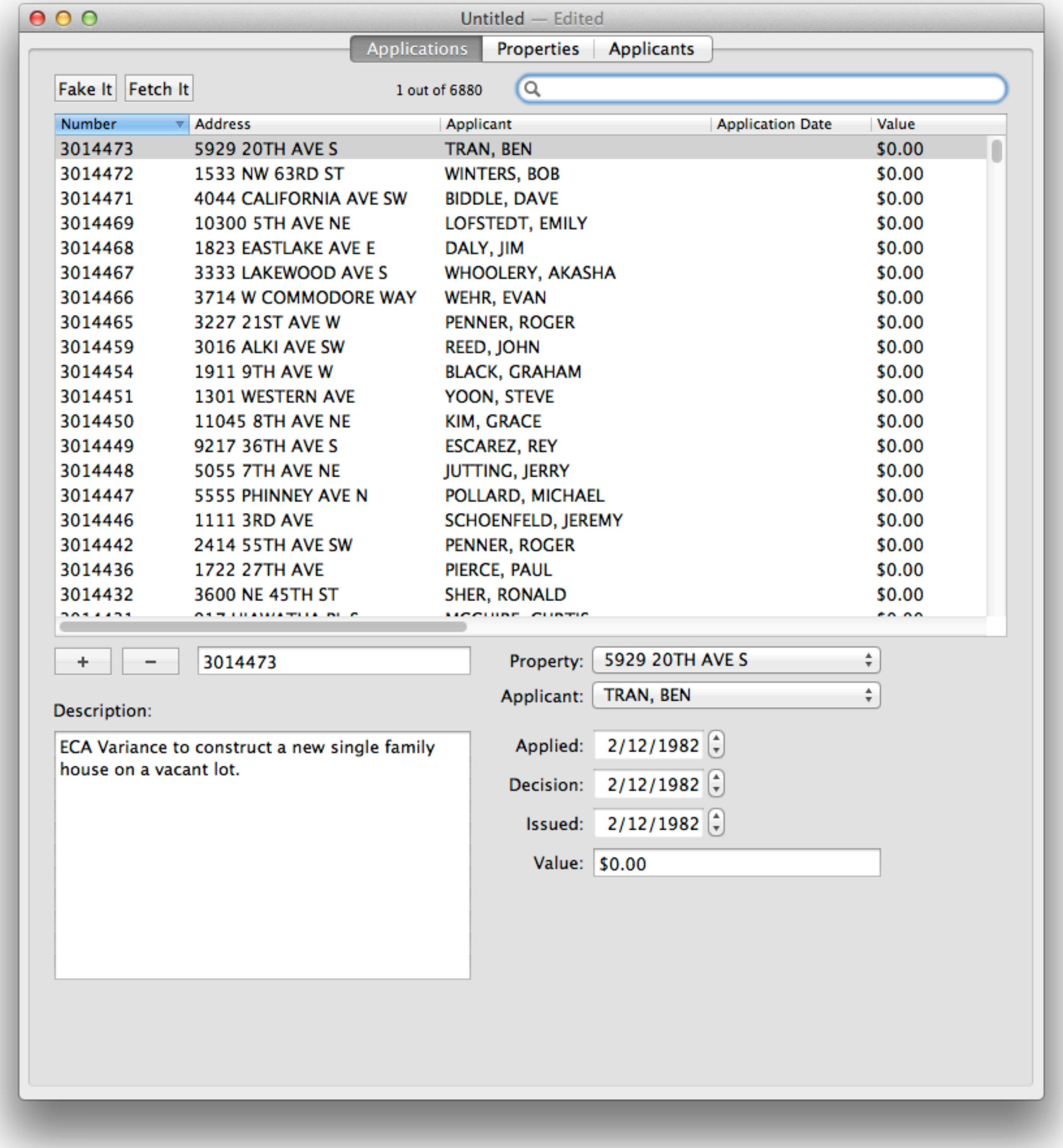
Modify the download request so that only data that has changed since your last request is downloaded.

Extract the column descriptions from the metadata, and use them to populate the Tool Tips for table columns and detail view elements.

Grading Focus

Core Data model and UI continue to work correctly.

Data is loaded over the net, parsed, and inserted into the database.



Number	Address	Applicant	Application Date	Value
3014473	5929 20TH AVE S	TRAN, BEN		$0.00
3014472	1533 NW 63RD ST	WINTERS, BOB		$0.00
3014471	4044 CALIFORNIA AVE SW	BIDDLE, DAVE		$0.00
3014469	10300 5TH AVE NE	LOFSTEDT, EMILY		$0.00
3014468	1823 EASTLAKE AVE E	DALY, JIM		$0.00
3014467	3333 LAKEWOOD AVE S	WHOOLERY, AKASHA		$0.00
3014466	3714 W COMMODORE WAY	WEHR, EVAN		$0.00
3014465	3227 21ST AVE W	PENNER, ROGER		$0.00
3014459	3016 ALKI AVE SW	REED, JOHN		$0.00
3014454	1911 9TH AVE W	BLACK, GRAHAM		$0.00
3014451	1301 WESTERN AVE	YOON, STEVE		$0.00
3014450	11045 8TH AVE NE	KIM, GRACE		$0.00
3014449	9217 36TH AVE S	ESCAREZ, REY		$0.00
3014448	5055 7TH AVE NE	JUTTING, JERRY		$0.00
3014447	5555 PHINNEY AVE N	POLLARD, MICHAEL		$0.00
3014446	1111 3RD AVE	SCHOENFELD, JEREMY		$0.00
3014442	2414 55TH AVE SW	PENNER, ROGER		$0.00
3014436	1722 27TH AVE	PIERCE, PAUL		$0.00
3014432	3600 NE 45TH ST	SHER, RONALD		$0.00

3014473

Property: 5929 20TH AVE S

Applicant: TRAN, BEN

Description:

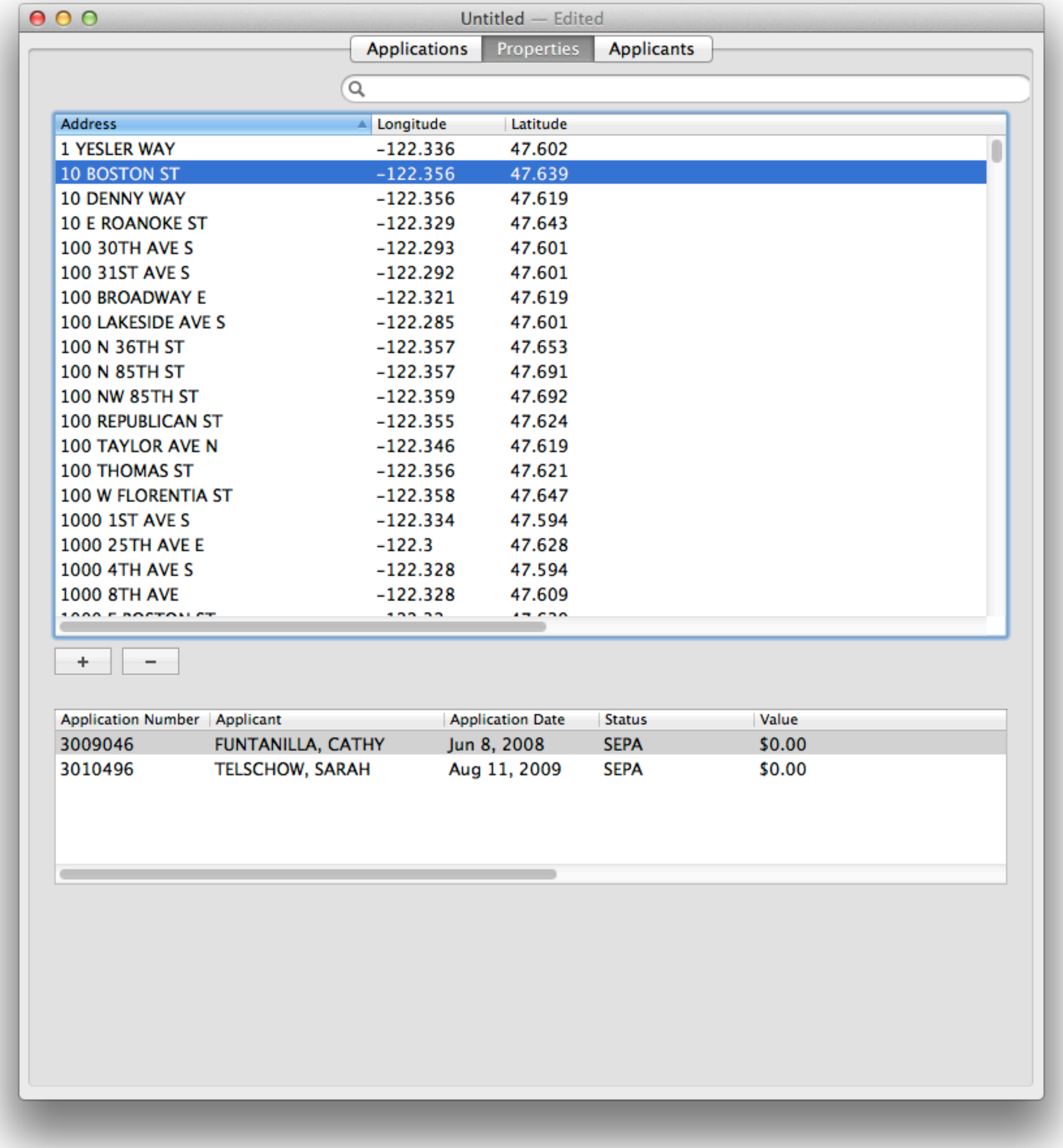
ECA Variance to construct a new single family house on a vacant lot.

Applied: 2/12/1982

Decision: 2/12/1982

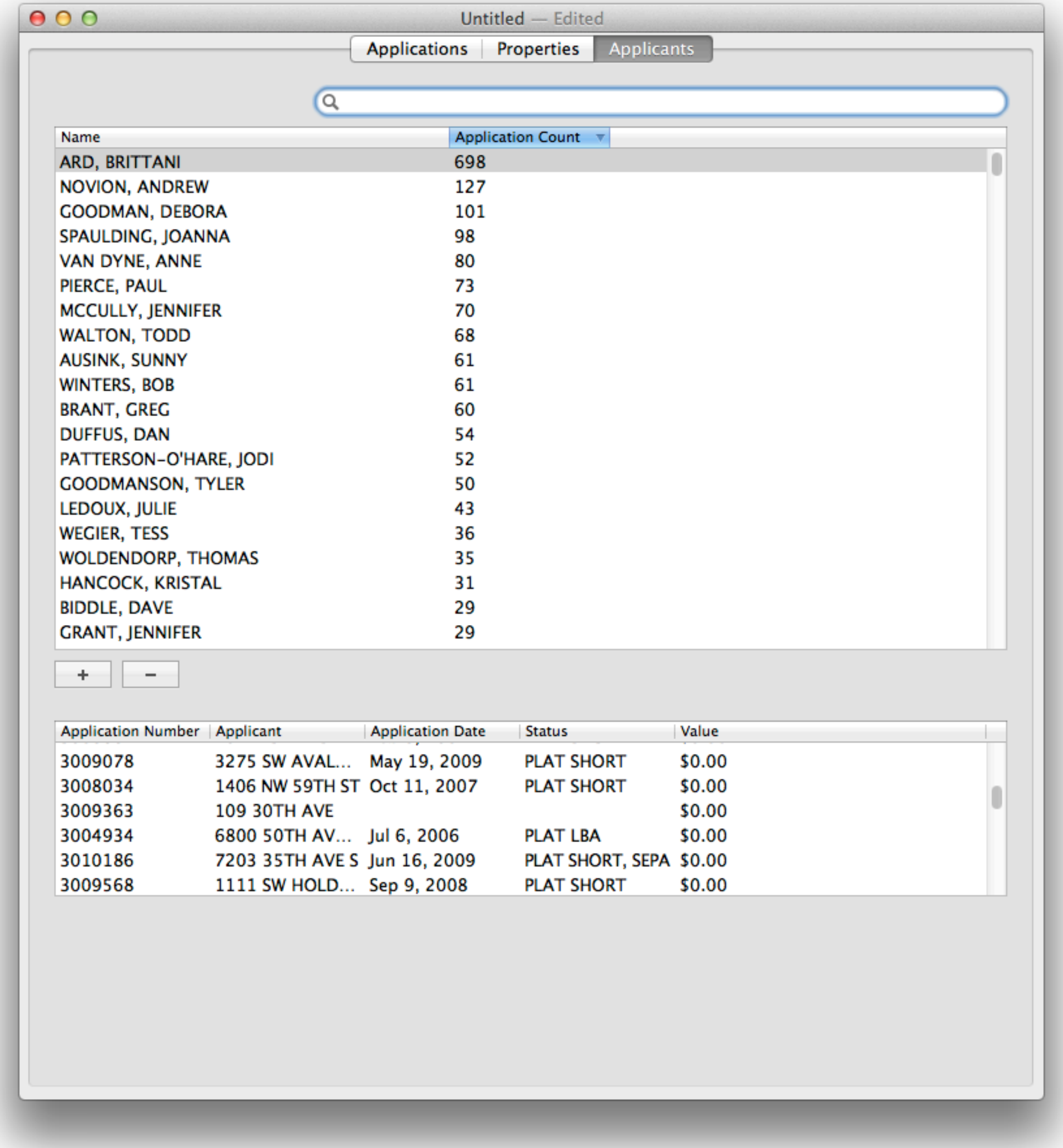
Issued: 2/12/1982

Value: $0.00



Address	Longitude	Latitude
1 YESLER WAY	-122.336	47.602
10 BOSTON ST	-122.356	47.639
10 DENNY WAY	-122.356	47.619
10 E ROANOKE ST	-122.329	47.643
100 30TH AVE S	-122.293	47.601
100 31ST AVE S	-122.292	47.601
100 BROADWAY E	-122.321	47.619
100 LAKESIDE AVE S	-122.285	47.601
100 N 36TH ST	-122.357	47.653
100 N 85TH ST	-122.357	47.691
100 NW 85TH ST	-122.359	47.692
100 REPUBLICAN ST	-122.355	47.624
100 TAYLOR AVE N	-122.346	47.619
100 THOMAS ST	-122.356	47.621
100 W FLORENTIA ST	-122.358	47.647
1000 1ST AVE S	-122.334	47.594
1000 25TH AVE E	-122.3	47.628
1000 4TH AVE S	-122.328	47.594
1000 8TH AVE	-122.328	47.609
1000 E BOSTON ST	-122.32	47.639

Application Number	Applicant	Application Date	Status	Value
3009046	FUNTANILLA, CATHY	Jun 8, 2008	SEPA	$0.00
3010496	TELSCHOW, SARAH	Aug 11, 2009	SEPA	$0.00



Name	Application Count
ARD, BRITTANI	698
NOVION, ANDREW	127
GOODMAN, DEBORA	101
SPAULDING, JOANNA	98
VAN DYNE, ANNE	80
PIERCE, PAUL	73
MCCULLY, JENNIFER	70
WALTON, TODD	68
AUSINK, SUNNY	61
WINTERS, BOB	61
BRANT, GREG	60
DUFFUS, DAN	54
PATTERSON-O'HARE, JODI	52
GOODMANSON, TYLER	50
LEDOUX, JULIE	43
WEGIER, TESS	36
WOLDENDORP, THOMAS	35
HANCOCK, KRISTAL	31
BIDDLE, DAVE	29
GRANT, JENNIFER	29

Application Number	Applicant	Application Date	Status	Value
3009078	3275 SW AVAL...	May 19, 2009	PLAT SHORT	$0.00
3008034	1406 NW 59TH ST	Oct 11, 2007	PLAT SHORT	$0.00
3009363	109 30TH AVE			$0.00
3004934	6800 50TH AV...	Jul 6, 2006	PLAT LBA	$0.00
3010186	7203 35TH AVE S	Jun 16, 2009	PLAT SHORT, SEPA	$0.00
3009568	1111 SW HOLD...	Sep 9, 2008	PLAT SHORT	$0.00

Starting Code

```
+ (NSString *)stringForJSONURL
{
    return @"https://data.seattle.gov/api/views/uyyd-8gak/rows.json?accessType=DOWNLOAD";
}

+ (void)exploreDatabase:(NSDictionary *)database
{
    NSArray *data = database[@"data"];
    NSDictionary *meta = database[@"meta"];
    NSLog(@"'meta' keys: %@", meta.allKeys);
    NSDictionary *view = meta[@"view"];
    NSLog(@"'view' keys: %@", view.allKeys);
    //NSObject *columns = view[@"columns"];
    NSArray *columns = view[@"columns"];
    NSLog(@"columns are of class %@", columns.class);
    NSLog(@"column elements are of class %@", [columns[0] class]);
    NSDictionary *aColumnsElement = columns[0];
    NSLog(@"columns keys are %@", aColumnsElement.allKeys);

    int columnCount = 0;
    NSLog(@"column values: position name description dataTypeName fieldName");
    for (NSDictionary *obj in columns) {
        columnCount++;
        //NSLog(@"column %d: %@", columnCount, obj);
        NSLog(@"column %d: %@ %@ %@ %@ %@", columnCount,
              obj[@"position"], obj[@"name"], obj[@"description"],
              obj[@"dataTypeName"], obj[@"fieldName"]);
    }
    NSLog(@"meta %@ %@", meta.class, meta.allKeys);
    NSLog(@"view %@ %@", view.class, view.allKeys);

    int j = 0;
    for (NSArray *row in data) {
        int i = 0;
        for (id col in row) {
            NSLog(@"%d %@", i++, col);
        }
        if (j++ > 20)
            break;
    }
}
```

Here’s a start on the parsing and loading code:

```
+ (BOOL)repopulateFromScratch:(NSManagedObjectContext *)moc
{
    NSData *data = [NSData dataWithContentsOfURL:
                    [NSURL URLWithString:[self stringForJSONURL]]];
    return [self repopulateFromScratchWithData:data context:moc];
}

+ (BOOL)repopulateFromScratchWithData:(NSData *)inputData
                              context:(NSManagedObjectContext *)moc
{
    NSLog(@"parsing data of size %ld bytes", inputData.length);
    NSDictionary *database;
    NSError *error;
    database = [NSJSONSerialization
                JSONObjectWithData:inputData
                options:kNilOptions
                error:&error];
    if (!database) {
        NSLog(@"%@ %@", error.localizedDescription,
error.localizedFailureReason);
        return NO;
    }
    else {
        [self exploreDatabase:database];
        NSLog(@"data downloaded");

        NSDateFormatter *rfc3339DateFormatter = [[NSDateFormatter alloc] init];
        NSLocale *enUSPOSIXLocale = [[[NSLocale alloc]
initWithLocaleIdentifier:@"en_US_POSIX"] autorelease];

        rfc3339DateFormatter.locale = enUSPOSIXLocale;
        rfc3339DateFormatter.dateFormat = @"yyyy'-'MM'-'dd'T'HH':'mm':'ss'Z'";
        rfc3339DateFormatter.timeZone = [NSTimeZone timeZoneForSecondsFromGMT:
0];


        for (NSArray *row in database[@"data"]) {
            NSString *permitNumberString = nil;
            NSString *addressString = nil;
            NSString *applicantString = nil;
            if (row[8] != [NSNull null]) {
                permitNumberString = row[8];
            }
            if (row[10] != [NSNull null]) {
                addressString = row[10];
            }
            if (row[16] != [NSNull null]) {
                applicantString = row[16];
            }

            Application *application = [Application
findOrCreateApplicationWithPermitNumber:permitNumberString

context:moc];
            application.applicationPermitNumber = permitNumberString;
            if (row[9] != [NSNull null]) {
                application.permitTypes = row[9];
            }
            if (applicantString) {
                Applicant *applicant = [Applicant
findOrCreateApplicantWithName:applicantString

context:moc];
                application.applicant = applicant;
            }
```