### business.json Contains business data including location data, attributes, and categories. { // string, 22 character unique string business id

**"business\_id": "tnhfDv5Il8EaGSXZGiuQGg",**

// string, the business's name

**"name": "Garaje",**

// string, the full address of the business

**"address": "475 3rd St",**

// string, the city

**"city": "San Francisco",**

// string, 2 character state code, if applicable

**"state": "CA",**

// string, the postal code

**"postal code": "94107",**

// float, latitude

**"latitude": 37.7817529521,**

// float, longitude

**"longitude": -122.39612197,**

// float, star rating, rounded to half-stars

**"stars": 4.5,**

// integer, number of reviews

**"review\_count": 1198,**

// integer, 0 or 1 for closed or open, respectively

**"is\_open": 1,**

// object, business attributes to values. note: some attribute values might be objects

**"attributes": {**

**"RestaurantsTakeOut": true,**

**"BusinessParking": {**

**"garage": false,**

**"street": true,**

**"validated": false,**

**"lot": false,**

**"valet": false**

**},**

**},**

// an array of strings of business categories

**"categories": [**

**"Mexican",**

**"Burgers",**

**"Gastropubs"**

**],**

// an object of key day to value hours, hours are using a 24hr clock

**"hours": {**

**"Monday": "10:00-21:00",**

**"Tuesday": "10:00-21:00",**

**"Friday": "10:00-21:00",**

**"Wednesday": "10:00-21:00",**

**"Thursday": "10:00-21:00",**

**"Sunday": "11:00-18:00",**

**"Saturday": "10:00-21:00"**

**}**

}

### review.json

Contains full review text data including the user\_id that wrote the review and the business\_id the review is written for.

{

// string, 22 character unique review id

**"review\_id": "zdSx\_SD6obEhz9VrW9uAWA",**

// string, 22 character unique user id, maps to the user in user.json

**"user\_id": "Ha3iJu77CxlrFm-vQRs\_8g",**

// string, 22 character business id, maps to business in business.json

**"business\_id": "tnhfDv5Il8EaGSXZGiuQGg",**

// integer, star rating

**"stars": 4,**

// string, date formatted YYYY-MM-DD

**"date": "2016-03-09",**

// string, the review itself

**"text": "Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks.",**

// integer, number of useful votes received

**"useful": 0,**

// integer, number of funny votes received

**"funny": 0,**

// integer, number of cool votes received

**"cool": 0**

}

### user.json

User data including the user's friend mapping and all the metadata associated with the user.

{

// string, 22 character unique user id, maps to the user in user.json

**"user\_id": "Ha3iJu77CxlrFm-vQRs\_8g",**

// string, the user's first name

**"name": "Sebastien",**

// integer, the number of reviews they've written

**"review\_count": 56,**

// string, when the user joined Yelp, formatted like YYYY-MM-DD

**"yelping\_since": "2011-01-01",**

// array of strings, an array of the user's friend as user\_ids

**"friends": [**

**"wqoXYLWmpkEH0YvTmHBsJQ",**

**"KUXLLiJGrjtSsapmxmpvTA",**

**"6e9rJKQC3n0RSKyHLViL-Q"**

**],**

// integer, number of useful votes sent by the user

**"useful": 21,**

// integer, number of funny votes sent by the user

**"funny": 88,**

// integer, number of cool votes sent by the user

**"cool": 15,**

// integer, number of fans the user has

**"fans": 1032,**

// array of integers, the years the user was elite

**"elite": [**

**2012,**

**2013**

**],**

// float, average rating of all reviews

**"average\_stars": 4.31,**

// integer, number of hot compliments received by the user

**"compliment\_hot": 339,**

// integer, number of more compliments received by the user

**"compliment\_more": 668,**

// integer, number of profile compliments received by the user

**"compliment\_profile": 42,**

// integer, number of cute compliments received by the user

**"compliment\_cute": 62,**

// integer, number of list compliments received by the user

**"compliment\_list": 37,**

// integer, number of note compliments received by the user

**"compliment\_note": 356,**

// integer, number of plain compliments received by the user

**"compliment\_plain": 68,**

// integer, number of cool compliments received by the user

**"compliment\_cool": 91,**

// integer, number of funny compliments received by the user

**"compliment\_funny": 99,**

// integer, number of writer compliments received by the user

**"compliment\_writer": 95,**

// integer, number of photo compliments received by the user

**"compliment\_photos": 50**

}

### checkin.json

Checkins on a business.

{

// string, 22 character business id, maps to business in business.json

**"business\_id": "tnhfDv5Il8EaGSXZGiuQGg"**

// string which is a comma-separated list of timestamps for each checkin, each with format YYYY-MM-DD HH:MM:SS

**"date": "2016-04-26 19:49:16, 2016-08-30 18:36:57, 2016-10-15 02:45:18, 2016-11-18 01:54:50, 2017-04-20 18:39:06, 2017-05-03 17:58:02"**

}

### tip.json

Tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions.

{

// string, text of the tip

**"text": "Secret menu - fried chicken sando is da bombbbbbb Their zapatos are good too.",**

// string, when the tip was written, formatted like YYYY-MM-DD

**"date": "2013-09-20",**

// integer, how many compliments it has

**"compliment\_count": 172,**

// string, 22 character business id, maps to business in business.json

**"business\_id": "tnhfDv5Il8EaGSXZGiuQGg",**

// string, 22 character unique user id, maps to the user in user.json

**"user\_id": "49JhAJh8vSQ-vM4Aourl0g"**

}

### photo.json

Contains photo data including the caption and classification (one of "food", "drink", "menu", "inside" or "outside").

{

// string, 22 character unique photo id

**"photo\_id": "\_nN\_DhLXkfwEkwPNxne9hw",**

// string, 22 character business id, maps to business in business.json

**"business\_id" : "tnhfDv5Il8EaGSXZGiuQGg",**

// string, the photo caption, if any

**"caption" : "carne asada fries",**

// string, the category the photo belongs to, if any

**"label" : "food"**

}