# Build your bot army

***Joe Kokenge** has worked on the data side of the news business at a number of publications including ProPublica, The Dallas Morning News and The Des Moines Register. Currently, he is Data Editor at the San Antonio Express-News.*

***Abraham Epton** is a developer on the Chicago Tribune's News Apps team, and previously worked on Google News publisher support for a few years before deciding he liked Chicago more than California after all. He's a fan of Python, open government and tweeting at @aepton.*

***Brian Abelson** is a statistician, hacker, and internet troll. He's currently a data scientist at Enigma where he's working on making sense of the world's public data. Before that he was a OpenNews Fellow at the New York Times. Brian is the co-founder of csv soundsystem, a collective or journalists, artists, and hackers in New York City.*

Joe walks us through some basic terms: a Twitter bot is something that manipulates the Twitter API, essentially letting you talk to Twitter data. To create one, you have to authenticate your application and use a library for your language of choice. Joe advises us to "start small," showing us a Hello World-type program that uses the Twitter API. From these humble beginnings, you can build some more sophisticated bots, like Joe's @newsapps_jobs, which pulls information from a spreadsheet. Two weeks ago, he discovered @SourceJobs, which validated his idea — this is a good way to get the word out about jobs.

From the audience, there's a point about rate limiting and Twitter's policies on bots. Brian tells us about some workarounds, like the one used by @StealthMountain. This bot doesn't continuously spam tweets, but instead programmatically responds to anyone who tweets the misspelled phrase "sneak peak."

Abraham introduces us to his project on campaign contributions in Illinois. "Twitter bots in general are not necessarily that complex." His bot uses a cron job that scrapes recent postings on the campaign finance website every five minutes and then tweets out any new contributions. Even though this information was being disclosed publicly, it wasn't necessarily visible, but by broadcasting it on Twitter, more people interested in campaign contributions could see them.

In the newsroom, there was concern that the Twitter bot would tip off the competition. Some editors were concerned this would take away a competitive advantage the Chicago Tribune might have. Abraham reiterates that the information was public anyway; any reporter could go and get this data. Even though there's no financial benefit to the Twitter bot, it gets people thinking that the Tribune is doing something interesting.

One of the things Abraham learned in building the bot is that private Twitter accounts are great for beta-testing and may enable some interesting internal tools in the newsroom, like a deadline bot.

“You can let other people tweet to the outside world when it’s ready, but not necessarily automatically.”

Abraham says it’s difficult to get personality in 140 characters in a way that doesn’t lose meaning. “You really don’t have a lot of room for creativity and expression.” He shows us the line of code that builds the tweet:

tweet = ‘%s %s from %s to %s.’

This is composed from four fields: commentary, amount, donor, recipient. The danger is that names can be arbitrarily long, and information towards the end of the tweet can get cut off because of the character limit — you should preempt this when you’re creating your bot so you lose the least essential content when a tweet gets truncated.

“You don’t want to be too jokey about stuff, especially stuff that’s too serious.” Writing automated jokes can go really wrong if you’re working with something like murder data.

Now Brian takes the mic and walks us through some bots he’s written in the past. First is @YourRepsOnGuns, which looks for tweets from representatives on guns and shootings and retweets them. Next, he shows us Treasury.io, the first API of the government’s daily expenditure, which parses through a cryptic form and turns that into clean data. Based on this project, Brian created a Twitter bot that tweets random facts generated from this API. “Sometimes it tweets something really interesting and a lot of people retweet it, sometimes it tweets something that’s really nonsensical and doesn’t really add to the understanding of government finance... it’s a visibility tool.”

Brian shows us some less-serious bots, in the vein of “weird Twitter.” For example, @haikugrams looks for haikus in tweets, with a particular emphasis on profanity. The New York Times latched onto this idea — Jacob Harris created Times Haiku, which finds haikus in comments on Times articles. It’s meant to be whimsical and provides another way of drawing people into stories. Next, Brian shows us Jeremy Merrill’s @ProPubSunset, which used a series of computer vision algorithms to identify and take photos of sunsets.

There’s also bots like @stopandfrisk, which communicates the volume of stop and frisks with a little bit of commentary as well. Josh Begley’s @dronestream tweets every reported US drone strike. Recently, he tried to build an iPhone app based on the same API that would send you a push notification every time there was a drone strike, but it was rejected by Apple. @_FloridaMan takes stories with “Florida Man” in their headlines, turning them into the stories of a composite character — “the world’s worst superhero.” Brian notes, “The best bots do one thing, and do it very well.”

Now we move into Q&A — a question is asked about @YourRepsOnGuns and accuracy. Brian says the logic behind the Twitter bot is strict rather than greedy. He also points to issues faced with Times Haiku. “The challenge is, how are we not going to make a haiku of the Syrian civil war, how are we not going to make a haiku of something that’s serious... that’s why it’s easier to do some of these

funny artistic ones rather than something you can put the name of a newsroom on."

Once again, rate limiting is brought up — Abraham says you can write the logic of your bot to avoid having your account get deleted. "Use common sense," he says. The more you avoid behaviors that make your bot seem like a spam bot, the safer your account will be. Joe and Brian agree — the rate limit is high enough that you can get away with a tweet every 5 minutes without hitting it.

Abraham says that Twitter bots are a good way of getting visibility for stories, once you figure your way around the challenge of tagging. There's a technical question about how to avoid duplicating tweets. Abraham says to use a log file with some identifiable information about what you're tweeting out, like a link. Each time the bot tweets, it can check the log file first to avoid duplication. With Treasury.io, the tweets are based on SQL queries and can therefore be controlled to avoid tweeting the same thing.

Another question is asked about low-cost hosting — if you can't host your bot on your newsroom server, what are some ways around this? Twitter bots are simple enough that you can run it straight from your laptop, but you can also use services like Rackspace, EC2, or Iron.io to do this. Brian references Jacob Harris's ebooks account generator, the reason for the surge of ebooks accounts for news apps developers.

**Feel free to add text/images/links if you are at the talk. Add your Twitter handle if you contributed:**
@s2tephen