

more <library mashups>

**Exploring New Ways
to Deliver Library Data**

blog

<subject>

code

photos

maps

<head>

<image>

<body>

<head>

web

service
"top" link

Edited by

Nicole C. Engard

Foreword by Michael P. Sauers

more
<library
mashups>

more
<library
mashups>

**Exploring New Ways
to Deliver Library Data**

Edited by
Nicole C. Engard



Information Today, Inc.
Medford, New Jersey

First printing, 2014

More Library Mashups: Exploring New Ways to Deliver Library Data

Copyright © 2014 by Nicole C. Engard

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without permission in writing from the publisher, except by a reviewer, who may quote brief passages in a review. Published by Information Today, Inc., 143 Old Marlton Pike, Medford, New Jersey 08055.

Publisher's Note: The editor and publisher have taken care in preparation of this book but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book and Information Today, Inc. was aware of a trademark claim, the designations have been printed with initial capital letters.

Library of Congress Cataloging-in-Publication Data

More library mashups : exploring new ways to deliver library data / edited by Nicole C. Engard.

p. cm.

Includes bibliographical references and index.

President and CEO: Thomas H. Hogan, Sr.
Editor-in-Chief and Publisher: John B. Bryans
Managing Editor: Amy M. Reeve
Production Manager: Norma Neimeister
Book Designer: Kara Mia Jalkowski
Cover Designer: Danielle Nicotra

To my sisters, Alissa and Kristen,
for being my very first students so many years ago

Contents

Acknowledgments	xi
About the Website.....	xii
Foreword.....	xv
<i>Michael P. Sauers</i>	
Introduction	xvii
<i>Nicole C. Engard</i>	

PART I: THE BASICS

Chapter 1: IFTTT Makes Data Play Easy.....	3
<i>Gary Green</i>	
Chapter 2: The Non-Developer's Guide to Creating Map Mashups	29
<i>Eva Dodsworth</i>	
Chapter 3: OpenRefine(ing) and Visualizing Library Data	43
<i>Martin Hawksey</i>	
Chapter 4: Umlaut: Mashing Up Delivery and Access	59
<i>Jonathan Rochkind</i>	

PART II: MASHING UP LIBRARY WEBSITES

Chapter 5: Building a Better Library Calendar With Drupal and Evanced Events.....	77
<i>Kara Reuter and Stefan Langer</i>	
Chapter 6: An API of APIs: A Content Silo Mashup for Library Websites	93
<i>Sean Hannan</i>	
Chapter 7: Curating API Feeds to Display Open Library Book Covers in Subject Guides	105
<i>Rowena McKernan</i>	

PART III: MASHING UP LIBRARY CATALOG DATA

- Chapter 8: Searching Library Databases Through Twitter** 119
Bianca Kramer
- Chapter 9: Putting Library Catalog Data on the Map** 131
Natalie Pollecutt
- Chapter 10: Mashups and Next Generation Catalog at Work... 143**
Anne-Lena Westrum and Asgeir Rekkavik
- Chapter 11: Delivering Catalog Records Using Wikipedia
Current Awareness..... 161**
Natalie Pollecutt

PART IV: VISUALIZING DATA WITH MASHUPS

- Chapter 12: Telling Stories With Google Maps Mashups** 181
Olga Buchel
- Chapter 13: Visualizing a Collection Using Interactive Maps... 197**
Francine Berish and Sarah Simpkin
- Chapter 14: Creating Computer Availability Maps** 209
Scott Bacon
- Chapter 15: Getting Digi With It: Using TimelineJS to
Transform Digital Archival Collections..... 221**
Jeanette Claire Sewell

PART V: MASHUPS FOR VALUE-ADDED SERVICES

- Chapter 16: BookMeUp: Using HTML5, Web Services,
and Location-Based Browsing to Build a
Book Suggestion App..... 235**
Jason Clark
- Chapter 17: Stanford's SearchWorks: Mashup Discovery
for Library Collections..... 247**
Bess Sadler
- Chapter 18: Libki and Koha: Leveraging Open Source
Software for Single Sign-on Integration** 261
Kyle M. Hall

Chapter 19: Disassembling the ILS: Using MarcEdit and Koha to Leverage System APIs to Develop Custom Workflows 273
Terry Reese

Chapter 20: Mashing Up Information to Stay on Top of News 297
Celine Kelly

Chapter 21: A Mashup in One Week: The Process Behind Serendip-o-matic 315
Meghan Frazer

Appendix A: Links 327

Appendix B: Glossary 339

About the Contributors..... 347

About the Editor 353

Index 355

Acknowledgments

I would like to thank many people for their support in bringing this second mashup of mashups together. First, I must thank all those of you who read the first edition of this book and kept asking me when we were going to provide you with more great mashup examples. Second, I must thank my colleagues at ByWater Solutions, in particular Brendan and Nate for always supporting me in whatever endeavor I choose to take on.

I'd also like to thank the many authors who contributed to this title, making a wide-reaching view of mashups for libraries. Without all of your contributions, this book wouldn't exist. Lastly, I thank my children (shelties, Coda and Beau) for their understanding while I spent more time with my computer than with them over the last year.

About the Website

mashups.web2learning.net

A website with links to resources in the book and stories from our readers will also be maintained at mashups.web2learning.net.

Disclaimer

Neither the publisher nor the editor make any claim as to the results that may be obtained through the use of this webpage or of any of the internet resources it references or links to. Neither publisher nor editor will be held liable for any results, or lack thereof, obtained by the use of this page or any of its links; for any third-party charges; or for any hardware, software, or other problems that may occur as the result of using it. This webpage is subject to change or discontinuation without notice at the discretion of the publisher and editor.

Foreword

Michael P. Sauers

I suspect that most people reading this had the same first encounter with mashups as I did, courtesy of a gentleman by the name of H.B. Reese. His name may not be immediately recognizable to you but the product he invented in 1928 probably is: the Reese's Peanut Butter Cup.

My next encounter with the world of mashups was in 2004 with the release of DJ Danger Mouse's *The Grey Album*. This masterpiece is the result of combining The Beatles' *The White Album* with Jay Z's *The Black Album*. It was with this that I learned of the term mashup, defined as the combining of two separate yet distinctive musical sources, to create a third, new musical composition. To this day, I still enjoy listening to musical mashups, especially those containing content that I'm already familiar with and content that I wouldn't normally listen to. Probably the best example of this would be 2008's *American Edit* by Dean Gray, a combination of Green Day's *American Idiot* album with other works, ranging from the Doctor Who theme music to Johnny Cash, just to name a couple.

Then, in 2009, mashups dropped into my work life with Nicole Engard's 2009 book *Library Mashups: Exploring New Ways to Deliver Library Data*. With her book, I learned that, beyond food and music, you could mashup data. And, since we're librarians, if we've got nothing else, we've got access to vast amounts of data. What I also learned is that data mashups don't have to be hard; you just need to know where the data is and what tools are available.

However, when it comes to mashups, you also have to have the "idea," and for many of us that's the hard part. Just because you have the tools and the data, that doesn't mean you automatically know what to do with them. Sure, before 2008, I'd listened to *American Idiot* many times and had previously heard lots of other music too—especially the Doctor Who theme. I also had a copy of the audio editing program Audacity. That doesn't mean I realized what I could create with that data and tool.

What I love about *More Library Mashups* is that not only are the contributors giving me more data and more tools to play with, they're giving me ideas and inspiring me to look at the resources at hand a certain way—to ask “If I took this and that, what could I turn it into?”

Keep that question in mind as you read the chapters of this wonderful book. Don't just focus on the exact projects as presented but at the end of each one ask yourself, “What can I do with this?” Take that inspiration, along with the tools and data, go forth, and start mashing!

Introduction

Nicole C. Engard

Whenever I give a talk about mashups, I like to start with a picture of s'mores. It's the most delicious type of mashup. The term mashup didn't originate with the s'more though; it started in the music industry, as a reference to mixing two or more songs together in various ways. For the purposes of this book, a mashup has to do with taking data from multiple sources and mixing the data together to provide better services for library patrons.

If you read *Library Mashups*, the precursor to this book, then you probably learned a lot of great ways to mash your data up. My favorite mashup from the first book—if an editor is allowed to choose a favorite—is that of Delicious (delicious.com) with your website. I used Delicious to bookmark all of the links found for this book (delicious.com/nengard/mash2nd) and for all the books I've written, so that I can generate easy-to-access pages of links for your reference. All it took was a consistent taxonomy and a little snippet of Javascript provided by Delicious. For example, to get the links from Chapter 1, I just pasted

```
<script type="text/javascript" src="http://  
feeds.delicious.com/v2/js/nengard/mash2ndch1?  
title=&count=200&sort=alpha"></script>
```

onto the links page. To learn about how you can use Delicious in this way, check out Chapter 5 by Brian Herzog in *Library Mashups*.

In *More Library Mashups*, I bring together some familiar names and many new ones to share all new tips, tutorials, and stories of how libraries are using mashups. Some are as simple as copying a snippet of code while others are a bit more complex, but all are fascinating reads and will kickstart your creativity for mashing up your own library resources.

Part I begins with a few introductory chapters—some basic mashups that anyone can do with just the knowledge of a few simple web applications. These will introduce you to IFTTT (ifttt.com), a tool that allows you to pull in data from many sources to make content delivery and publication easy; ArcGIS (arcgis.com), a mapping application that makes map mashups easy; OpenRefine (openrefine.org), a tool for working with messy data; and Umlaut (github.com/team-umlaut/umlaut), a just-in-time aggregator of “last mile” specific-citation services (you’ll have to read more to learn what that means).

Part II introduces you to various ways to enhance your library website with outside data. The authors in this section will help you integrate outside tools such as cover images, event calendars, subject guides, and social networks into your library website to give your patrons a one-stop shop for information. Some of these mashups require a bit of programming knowledge, while for others, it’s just a matter of finding the right widgets for the job. Whatever the case, the authors give you all the tools you need to get the job done.

In Part III, experts will walk you through ways to create mashups with your library catalog data. Five years ago, the hardest mashup to achieve was the catalog data mashup, and things have not really changed much, but that doesn’t stop us from trying. Learn how to answer reference questions on Twitter with a bot and present library information on a map. Explore an interactive physical device that lets users do anything from looking up and exploring library books to finding top stories on Wikipedia and delivering them alongside library holdings information.

Part IV takes you from the textual to the visual, through creating timelines and maps, and sharing pictures from your library. Visualizations are some of the most obvious types of mashups; in fact, the most common type of mashup listed on ProgrammableWeb (programmableweb.com) is the map mashup (Figure I.1).

Finally, Part V will cover general ways to add value to library services, including BookMeUp and Serendip-o-matic, two tools that help you find library resources; SearchWorks at Stanford for finding information in any library collection; and MarcEdit and Libki, each with Koha, for providing better data and services for your patrons. It also shows you how to use various data sources to generate a current awareness service.

Besides directing you to online tools to help you mash up data at your library, I hope *More Library Mashups* sparks your imagination.



Figure I.1 Top mashup tags on ProgrammableWeb

The contributors and I would love to hear from you after you've read the book, so contact information is provided in the About the Contributors and About the Editor sections at the end of the book. A website with links to resources in the book and stories from our readers will also be maintained at mashups.web2learning.net.

IFTTT Makes Data Play Easy

Gary Green

Surrey County Council Public Library Service

IFTTT (If This Then That; ifttt.com) is a free online service that can be used to automate the collection, manipulation, mashing, and sharing of data and information across a range of online sites and services without any knowledge of APIs or computer programming. The only thing you need to make use of the service is an IFTTT account. In addition to explaining how IFTTT works, this chapter will provide practical examples of how it can be used.

How IFTTT Works

IFTTT can be used to share data across more than 90 sites and services (referred to as channels). IFTTT channels (ifttt.com/channels) make the discovery and sharing of data easier and less time consuming, since users do not have to visit each site individually and log in to either pull out or share the data.

The types of services available in IFTTT include:

- Social networks
- Bookmarking sites
- Image, video, and audio sharing sites
- Blogging services
- Document storage and collaboration

4 More Library Mashups

- Email and messaging services
- Date, time, weather, and stock alerts

Using these channels, you can coordinate your data sharing from a single place, with options to: (1) perform the same or similar tasks in a variety of locations, (2) perform the same tasks in the same place on a regular basis, and (3) re-use information in different places.

Many of the channels offered by IFTTT are used by libraries, including Blogger, Delicious, Diigo, Dropbox, Evernote, Facebook, Flickr, Gmail, Google Calendar, Google Drive, HootSuite, Instagram, LinkedIn, Pinboard, Pocket, Tumblr, Twitter, Vimeo, WordPress, and YouTube. IFTTT can connect with SMS messaging, weather, date alerts, and remote controlled services. There is also a dedicated RSS channel, so users can access any service with an RSS output, even if it doesn't have a dedicated channel on IFTTT.

The main purpose of IFTTT is to allow users to push data from one channel to another. It gives the user some control over how that data is pushed to the receiving service. Users can tweak the data by defining which fields are shared, can leave out unwanted information, and can add their own data as part of the collection process.

In most cases, you will be required to authorize IFTTT to access your accounts on any channels you are using. This enables IFTTT to do things like identify when you've posted an update to your social networks and post information to your accounts based upon the actions you ask it to perform.

IFTTT uses some unique terminology, which is worth highlighting before getting into the intricacies of using the service:

- *Channels*: The services you connect together
- *Trigger channel*: The channel you are sharing the data from
- *Action channel*: The channel you are sending the data to
- *Recipe*: Any connection (process) you set up between two channels to share data
- *Ingredients*: The pieces of data you want to share from the trigger channel to the action channel

To create an IFTTT recipe you need to decide:

- The source of data, or the trigger channel (e.g., Flickr)
- The criteria for selecting specific data from the trigger channel (e.g., any public photo uploaded by you to Flickr)

- The action channel where you want the data to appear (e.g., Facebook)
- Which data from the trigger channel will appear in the action channel (e.g., the URL of the photo and its text description from Flickr)

IFTTT's unabbreviated name "If This Then That," describes this process: "IF the trigger channel does THIS, THEN the action channel does THAT."

Each channel has a unique range of triggers and actions. For example, when you use Facebook as the trigger channel, you can use any one of five triggers (Figure 1.1). Alternatively, if you use the social bookmarking service Diigo as the trigger channel, you only have two triggers to choose from (Figure 1.2).

However, when a service is used as an action channel, the options change. For example, as an action channel, Facebook only has three actions available (Figure 1.3), while Diigo still has two options (Figure 1.4).

In addition, some channels can only be used as a trigger channel or action channel, not as both. For example, the HootSuite channel has no trigger options (meaning that you can't get data directly out of HootSuite using IFTTT) and the RSS channel has no action options. The available triggers and actions for all of the channels can be viewed by visiting the Channels page on the IFTTT site and clicking on the appropriate channel icon.

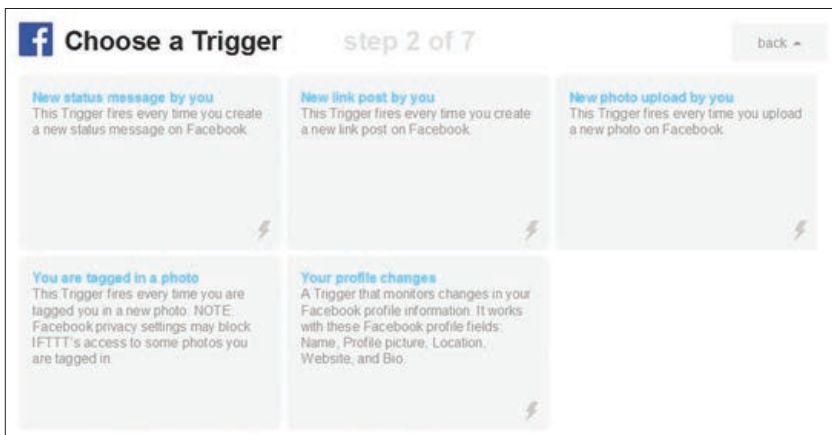


Figure 1.1 Trigger options for use with the Facebook channel

6 More Library Mashups



Figure 1.2 Trigger options for use with the Diigo channel



Figure 1.3 Action options for use with the Facebook channel



Figure 1.4 Action options for use with the Diigo channel

As well as having control over the triggers and actions, you also have some control over the data that is passed between them. For example, you might want to share a bookmark on Facebook, but not your own private description. Each channel offers you access to fields specific to that channel to share or not share.

Once you've set up a recipe, it is regularly checked by IFTTT and is automatically run whenever the criteria has been met by the trigger

channel. You can go back and edit or delete your recipes, and you can also turn them on or off as needed.

IFTTT also allows users to share recipes they have created with others. Shared recipes appear in the IFTTT public area (ifttt.com/recipes) as templates, which include the basic ingredients of the recipe, such as the trigger and action channels used, the specific triggers and actions, and a description. Other users can copy a shared recipe, modify it for their own purpose, and save it as a recipe in their own account.

IFTTT in Action

This section provides practical examples for using IFTTT, illustrating how recipes can be set up to perform various tasks. The first example provides the most detail, and the subsequent examples highlight the channels, triggers, and actions used, as well as the purpose of the recipe and how that specific recipe works.

Collecting Library News and Sharing Individual Items on Various Social Networks

This example uses IFTTT to pull news articles from various RSS feeds into a Pocket account (getpocket.com), where the user can assess each article, tag the articles worth sharing with any social networks, and then automatically send the tagged articles to specified social networks or save to read later. In this instance, I am using Pocket as an RSS feed reader—my chosen method of aggregating and sharing information. This example requires two sets of IFTTT recipes to be set up. The first set pulls RSS news feeds into Pocket, and the second set sends tagged news articles from Pocket to the social networks.

Feeding the RSS Feeds into Pocket

This recipe uses the following triggers and actions:

- Trigger channel: RSS feed
- Trigger: New feed item
- Action channel: Pocket
- Action: Save for later

8 More Library Mashups

First you decide on the RSS feed you are going to use as our source of information. In this instance, I'm going to use a Google news alert I have set up specifically for library news (news.google.co.uk/news/feeds?hl=en&gl=uk&q=library+OR+libraries&um=1&ie=UTF-8&output=rss).

Once you have logged into your IFTTT account, click the Create button at the top of the screen. Next, click on the word *this* to choose a trigger action (Figure 1.5). When the list of available trigger channels appears click on the RSS feed symbol (Figure 1.6).

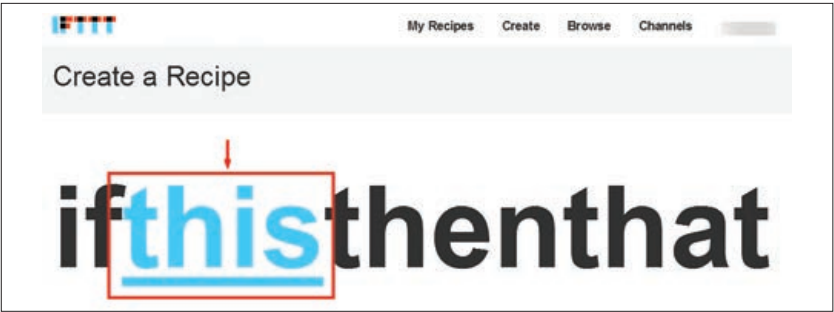


Figure 1.5 Click *this* to make your trigger selection

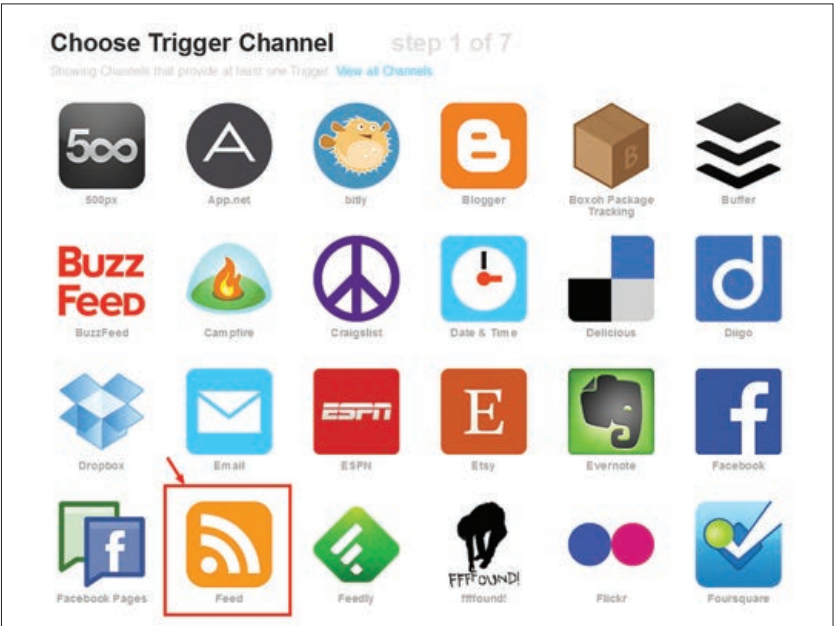


Figure 1.6 Choose your trigger channel

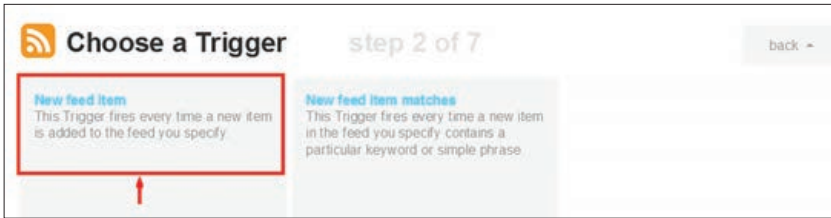


Figure 1.7 Choose the New Feed Item trigger



Figure 1.8 Enter the URL for your RSS feed and create your trigger channel

With RSS feeds, you have two options for triggers: New Feed Item (Figure 1.7) pulls any new items that are added to the RSS feed; New Feed Item Matches allows you to specify a keyword so that only matching items are pulled from the feed.

In Figure 1.7, notice that there is a small Back button on the top right side. If at any stage during the creation of a recipe you need to make a change, you can return to a previous stage by clicking this button. You can also scroll up the page and look for the link to Restart From Here.

Figure 1.8 shows the New Feed Item page, where you will add the URL for your RSS feed [1] and create your trigger channel [2].

Once you have created your trigger channel, the screen shown in Figure 1.9 will open. In the phrase “if this then that,” *this* is replaced by the trigger icon and feed URL information. Click on the word *that* to move to the list of available action channels. Click on the Pocket symbol, as shown in Figure 1.10.

The first time you use Pocket (and many of the other channels on IFTTT) you will be asked to activate the channel. This activation process authorizes IFTTT to access specific data from your account on this channel, as detailed on the authorization screen. At this point, log in with your Pocket username and password and click on Authorize and then Done.



Figure 1.9 Click *that* to make your action selection

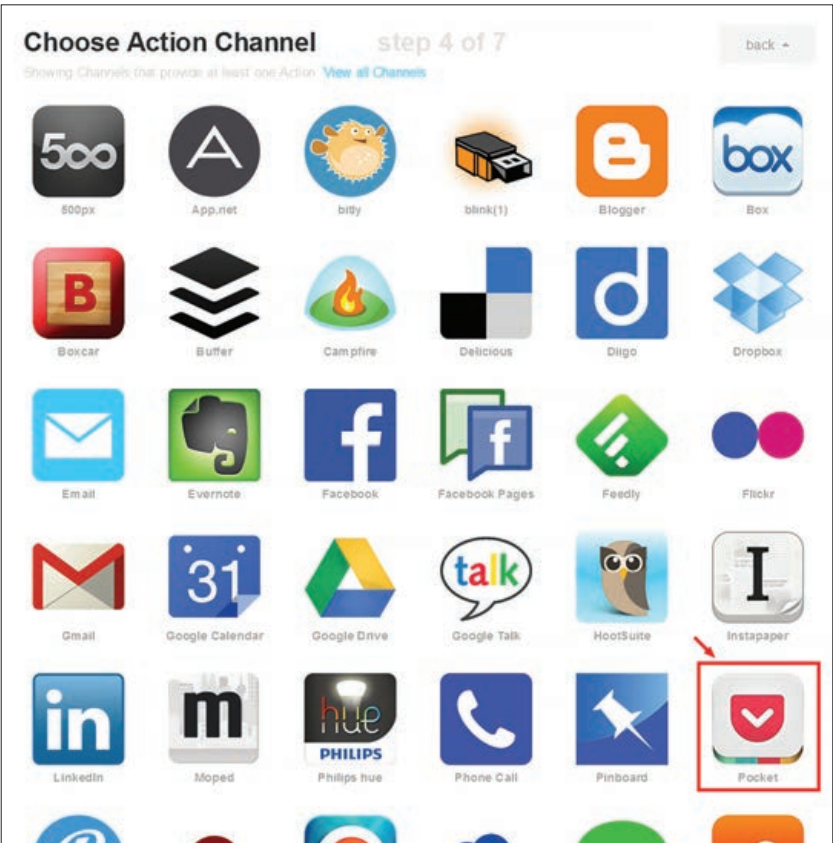


Figure 1.10 Choose your action channel

Now that you have authorized IFTTT to access your Pocket account, you will not need to go through this process again when you create further recipes using the Pocket channel. When the next screen appears, click on Continue to choose the desired action.

In this case, Pocket has only one action available: Save for Later (Figure 1.11). This action automatically posts any item published in the trigger RSS feed to Pocket.

You now have the option to specify what data will be pulled through and how you want it to be presented or manipulated (Figure 1.12). In the case of Pocket, there are two data fields you can access: the URL of the item that is to be saved, and the keyword tags you associate with that item. (Different action channels will allow you



Figure 1.11 Choose the Save for Later action



Figure 1.12 Choose your data presentation

to access different data fields, as you will see in later examples.) The text in both of the fields appears with or without a grey background (e.g., IFTTT and FeedTitle in the Tags field). Any text with a grey background indicates that it is a data field taken from the trigger feed, whereas data without a background is data you are adding via IFTTT (e.g., enter the keyword *IFTTT* and the feed's title in the Tags field). You can remove any field by clicking in the field and pressing the Delete key. When you are editing, the data fields from the trigger feed appear in `{{ }}` brackets. Additional data fields from the RSS trigger feed can be added by the plus sign next to an action field.

Once you click on a plus sign, you will be presented with a list of the data fields available within the RSS trigger channel: EntryTitle (title of news story in the feed), EntryUrl (the URL for that particular news story), EntryAuthor (the author of the article), EntryContent (the text of the entry; either a summary or full text of the news article), EntryImageUrl (the URL for an image used in this story), EntryPublished (the date the news story was published), FeedTitle (the title of the RSS feed used as the trigger), and FeedUrl (the URL of the RSS feed used as the trigger). To add a data field from the trigger channel, click on the field name in the list displayed and click Add Ingredient. (Pocket will automatically pull EntryTitle or EntryContent through when it pulls through the URL.)

In this example, I decided to delete the existing information in the Tags field and replace it with my own text to indicate the topic and source: LibraryNews and GoogleNews (Figure 1.13). I didn't pull through any other data fields apart from the EntryUrl. Once the action fields have been edited, click on Create Action.

The next screen provides an overview of the recipe you have created, which in this example is *if* a new item is added to this RSS feed *then* automatically post it to my Pocket account (Figure 1.14). To remind yourself of its purpose at a later date, add a summary of this recipe to the Description box (Figure 1.15). You can add hashtags to the description, which can be useful for organizing your recipes at a later date. Finally, click on Create Recipe.

You have now created your first recipe using IFTTT. To view your new recipe, click My Recipes at the top of the screen.

From your list of recipes (Figure 1.16), you are able to do the following:

- Switch the recipe on and off [1]; it is automatically set to “on”
- Delete the recipe [2]

- Share your recipe with others [3]
- View the recipe in more detail and edit it [4]

A summary of when the recipe was created, how many times it has been triggered, and when it was last triggered appears below these buttons.

Click on the view/edit button [4] to make any changes to the recipe. You can now see the recipe in more detail (Figure 1.17) and can

Complete Action Fields step 6 of 7

Save for later

URL
EntryUrl

Tags
IFTTT, FeedTitle
comma separated

Create Action

Figure 1.13 Customize RSS data fields

Create and activate step 7 of 7

if  then 

New feed item from
<https://news.google.co.uk/news/feeds?hl=en&gl=uk&q=library+OR+libraries&um=1&ie=UTF-8&output=rss>

Save later for

Figure 1.14 Recipe summary



Figure 1.15 Recipe description

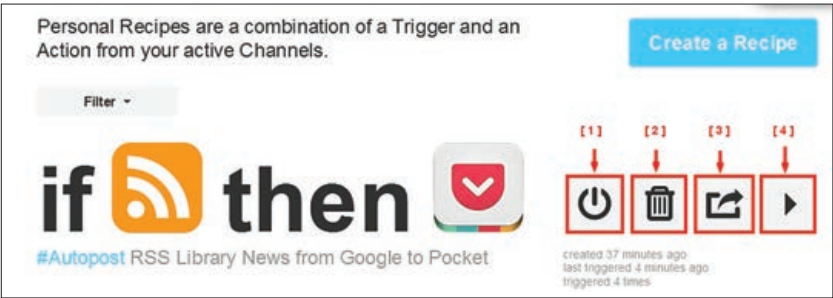


Figure 1.16 List of recipes

edit the Description [1], the RSS feed URL used for the trigger channel [2], and the data to be passed from the trigger channel to Pocket [3]. To save any changes you make to the recipe, scroll to the bottom of the page and click on Update.

When this recipe runs, it will pick up any new items from the specified Google news search (items that mention *library* or *libraries*) and save them to Pocket along with the tags you specified in Figure 1.13 so that they can be read there and subsequently shared.

If you have a number of sources of information you want to feed into Pocket in the same way, you can create more recipes like this example and just change the RSS feed you use as the trigger.

Feeding the News Items from Pocket to LinkedIn

This recipe uses the following triggers and actions:



- Trigger channel: Pocket
- Trigger: New item tagged

- Action channel: LinkedIn
- Action: Share a link

Once you have set up your recipes to pull news information feeds into Pocket, you will need to set up the second stage recipes that

Personal Recipe ID 4818348

4 My Recipes

if  then 

New feed item from
<https://news.google.co.uk/news/feeds?hl=en&gl=uk&q=library+OR+libraries&ump=1&ie=UTF-8&output=rss>

Save later for

Turn off

Share

Check

Activity

Delete

created about 1 hour ago

last triggered 15 minutes ago

triggered 4 times

Description

#Autopost RSS Library News from Google to Pocket

use "if" to add tags

Trigger

New feed item

This Trigger fires every time a new item is added to the feed you specify

Feed URL

[https://news.google.co.uk/news/feeds?](https://news.google.co.uk/news/feeds?hl=en&gl=uk&q=library+OR+libraries&ump=1&ie=UTF-8&output=rss)

Action

Save for later

This Action will add a new item to your Pocket queue

URL

EntryUrl

Tags

LibraryNews, GoogleNews

comma separated

Update



Figure 1.17 Recipe detail

enable you to share the individual news items from your Pocket account to your social networks. This stage of the process relies on Pocket's tagging feature. You can add tags manually to specific items, and the tags you add will trigger the items to be posted to various sites and services (e.g., add the tag *tolinkedin* to an item to share it on LinkedIn (linkedin.com) or add the tag *totwitter* to send it to Twitter). If you use Feedly (feedly.com) instead of Pocket, you can set up similar recipes using the same principals in this example, as Feedly also provides similar trigger functionality. Figure 1.18 shows the process of sharing a link from a news article in a Pocket account to a LinkedIn account [1]. The recipe is triggered when the tag *gglinkedin* [2] (a tag I defined for this purpose) is added to an article in Pocket. IFTTT looks for any article with that tag and pulls through the article's URL, a brief excerpt of the article, and an image from Pocket if the article contains one. This information then appears as a post on LinkedIn [3].

The previous example is focused on LinkedIn, but similar recipes can be set up to send items from Pocket to channels such as Twitter, Facebook, Buffer, or indeed anywhere you want to share those links. To share information from Pocket to these other channels, you will need to set up a new recipe for each one, indicating the tag you want to use as the trigger. Remember when setting up your recipes that different social networks have different conventions and etiquette for sharing information. For example, if you are sharing a link from Pocket to Twitter, you may want to set up a recipe that includes an appropriate hashtag.

Creating a Continuing Professional Development Log from Your Social Networks, Web Presences, and Google Calendar

The purpose of the following recipes is to pull together information about your work or study-related activities from a variety of sources into a single place to build up a continuing professional development (CPD) log, which can be referred back to for formal qualifications (e.g., chartership) or annual work appraisals. If you already share information about your work activities via social networks, blog about your work activities, and add work-related meetings and events to an online calendar, then you are already recording this information in a variety of places. However, it will be much easier to refer back to all of this information if you are also compiling it in a single place.

if  then 

New item tagged `gglinkedin`

Share a link on `profile`

Description


#Autopost Pocket to LinkedIn

use '#' to add tags

Trigger

New item tagged

This Trigger fires every time you tag an item with a specific tag in Pocket.


 Single tag

gglinkedin

Action


Share a link

This Action will share a link on your LinkedIn profile.

 Link URL


Url

+

 Comment

Excerpt

+

 Image URL

ImageUrl

+

optional, will attempt to use the best image if left blank

Figure 1.18 Feed news from Pocket to LinkedIn

18 More Library Mashups

As sources of information, the following services can be useful:

- LinkedIn: A social network used for posting work related status updates
- Google Calendar (calendar.google.com): An online calendar used to record work-related meetings and events
- SlideShare (slideshare.net): A place to share your presentations from training sessions and conferences
- Blogs: Somewhere you write about your work or training activities in more detail

All of these resources can be used as triggers. LinkedIn and Google Calendar have dedicated channels on IFTTT. SlideShare and blogs provide RSS feed as output, so RSS channels can be used for both of them.

To turn these various feeds into a continuing professional development log, we need to use an IFTTT channel that will enable us to generate either a text file or spreadsheet of information in a single place. A number of IFTTT channels can be used in this way, including Google Drive (drive.google.com), Dropbox (dropbox.com), and Evernote (evernote.com). In this example, I will be using Google Drive to build up a CPD spreadsheet. Whenever one of the channel triggers is activated, a new row is added to the spreadsheet indicating the activity that was undertaken, any links to relevant URLs, and the date on which the activity happened (e.g., “July 23, 2013, attended equalities in public libraries training” with a link to a reflective writing blog post about the training). I’ve aimed to standardize the information that is recorded from the different triggers, although date formats may not always be the same.

To automate the process of building a CPD log on Google Drive, it’s necessary to set up a recipe for each source of data to be fed into the log.

Pulling Through a LinkedIn Status Update to Google Drive

This recipe uses the following triggers and actions:

- Trigger channel: LinkedIn
- Trigger: Any new shared update
- Action channel: Google Drive
- Action: Add row to spreadsheet

This recipe pulls through any updates publicly posted to LinkedIn [1] (Figure 1.19). It records the LinkedIn CreatedDate (when the update was posted to LinkedIn) and StatusText (text of the update) in data fields on a new row in the Google Drive spreadsheet. The two data fields are saved in separate cells. In the Formatted Row box [2], “|||” appears between the names of the two data fields. This is used to indicate that the data fields pulled through should be saved in adjacent spreadsheet cells. As well as indicating how the data should be recorded in the spreadsheet, the recipe specifies the Spreadsheet Name (where the data should be saved), in this instance, Personal

The image shows an IFTTT recipe configuration interface. At the top, the 'if' part of the recipe is configured with the trigger 'Any new shared update by' and the action 'Add row to spreadsheet in Google Drive'. The description is '#log LinkedIn update to personal log'. Below this, the 'Action' section is titled 'Add row to spreadsheet'. It includes a note: 'This Action will add a single row to the bottom of the first worksheet of the spreadsheet you specify. NOT: A new row's.' The configuration fields are: 'Spreadsheet name' set to 'Personal Log' (with a note 'will create a new spreadsheet if one with this title doesn't exist'), 'Drive folder path' set to 'IFTTT/Personal Log' (with a note 'format: some/folder/path (defaults to 'IFTTT')'), and 'Formatted row' set to 'CreatedDate ||| StatusText' (with a note 'use "|||" to separate cells'). Red boxes and arrows highlight the description field [1] and the formatted row field [2].

Figure 1.19 Post LinkedIn updates to spreadsheet

Log. You are also required to specify the file path in the Drive Folder Path field, in this instance, IFTTT/Personal Log. Any status update posted to LinkedIn would then be pulled through to the Google Drive spreadsheet.

Pulling Through Events from Google Calendar to Google Drive

This recipe uses the following triggers and actions:

- Trigger channel: Google Calendar
- Trigger: Event from search starts
- Action channel: Google Drive
- Action: Add row to spreadsheet



This recipe pulls through any events on my Google Calendar attended for work purposes (Figure 1.20) into the same spreadsheet in Google Drive that we used in the previous LinkedIn example. I have used the trigger Event From Search Starts as this allows me to filter out any events that are not work related [1]. To do this, I need to tell it that I only want events to activate the recipe if I have included the tag *sccwork* in the description (my own personal tag to work-related events in the calendar) [2]. It records the Starts date (when the event is happening), the Title (event name as it appears in the calendar), and the EventURL (link to the event in my calendar). You can see in Figure 1.20 in the Formatted Row box [3] that I have placed the data fields in Starts, Title, and EventURL order, so that the date and event text appear in the same columns as the LinkedIn date and status text information indicated in the previous example. You will also see that it uses the same file path and file name as the previous example.

Uploading Presentations From SlideShare to Google Drive

This recipe uses the following triggers and actions:

- Trigger channel: RSS
- Trigger: New feed item
- Action channel: Google Drive
- Action: Add row to spreadsheet

This recipe records any presentations uploaded to SlideShare (Figure 1.21) in the same spreadsheet as the two previous examples [1]. It uses an RSS feed that is available from my SlideShare account [2]. All of my

if  then 

Event from search for 'sccwork' starts on

Add row to spreadsheet in Google Drive

Description


#log Work event to personal Log

use '#' to add tags

Trigger

Event from search starts

This Trigger fires within 15 minutes of the starting time of an event on your Google Calendar that contains a specific keyword event's Title, Description, and Location.


 Keyword or phrase

sccwork

Action


Add row to spreadsheet

This Action will add a single row to the bottom of the first worksheet of the spreadsheet you specify. NOTE: A new spreadsheet

 Spreadsheet name


Personal Log

will create a new spreadsheet if one with this title doesn't exist

 Drive folder path

IFTTT/Personal Log

format: some/folder/path (defaults to "IFTTT")

 Formatted row

Starts ||| Title ||| EventUrl


use "|||" to separate cells

Figure 1.20 Add Google Calendar event to Google Drive


22 More Library Mashups

presentations on SlideShare are work-related and so are all relevant. Again, the recipe records the information in the Google spreadsheet, using EntryPublished (date the presentation was uploaded), EntryTitle (name of presentation), and EntryURL (link to the presentation on

if



then



New feed item from
<http://www.slideshare.net/rss/user>

Add row to spreadsheet in
[Google Drive](#)

Description


#log Slideshare link to personal log

use "w" to add tags

Trigger

New feed item

This Trigger fires every time a new item is added to the feed you specify.


 Feed URL

<http://www.slideshare.net/rss/user/garys>


Action

Add row to spreadsheet


This Action will add a single row to the bottom of the first worksheet of the spreadsheet you specify. NOTE: A new spreadsheet will create a new spreadsheet if one with this title doesn't exist.

 Spreadsheet name


Personal Log




will create a new spreadsheet if one with this title doesn't exist

 Drive folder path

IFTTT/Personal Log



format: some/folder/path (defaults to "IFTTT")

 Formatted row

EntryPublished ||| EntryTitle ||| EntryUrl




Figure 1.21 SlideShare presentations posted to spreadsheet

SlideShare) [3]. Again, I provide a standard order that the information is recorded in—date, title, link—so that the information appears in the correct columns in the spreadsheet.

If you want to record links to blog posts about work activities or reflective CPD writing, you can create another recipe using your blog's RSS feed. Some of the fields in your RSS feed may be different to the fields in the SlideShare feed, so make sure to include the fields that pull through the date the blog post was written, its title, and a link to the blog post itself in the Formatted Row box.

These are just a few examples of how I built my own CPD log. You may use other services and channels that suit your needs better.

Forewarning People of Possible Library Closures Due to Snow via Twitter

This recipe uses the following triggers and actions:

- Trigger channel: Weather
- Trigger: Tomorrow's forecast calls for
- Action channel: Twitter
- Action: Post a tweet

With the IFTTT Weather channel, users are able to enter a location and create an action based on the current weather or tomorrow's forecast in that location (Figure 1.22). Using this channel, the following recipe [1] automatically sends out a tweet to forewarn library users of possible heavy snowfall the following day. This is one way of alerting library users to possible closures due to bad weather, with the weather channel trigger set to look for a condition of snow [2]. Even though this recipe makes use of the Weather channel, it does not pull through any data fields into the Twitter channel, as you will see in the Formatted Row text box in Figure 1.22 [3]. We added the text for the tweet ourselves ("Possible snow in Surrey tomorrow. Please check website for library closures if heavy snowfall occurs."). This is a useful illustration of how you can make use of valuable data from the trigger channel in the recipe without passing any of the actual data fields into the action channel.

The weather channel also provides triggers based on other weather conditions (e.g., clear, rain, cloudy, high and low temperature ranges). Using some of these other triggers, you could set up more recipes to

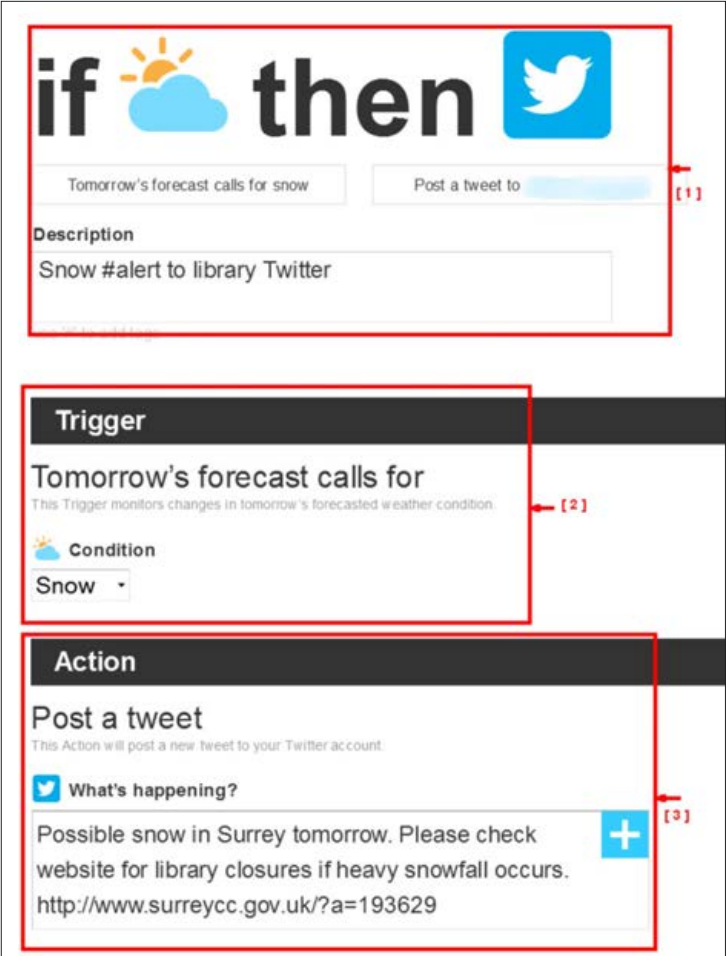


Figure 1.22 Send weather alerts to Twitter

creatively promote library stock (e.g., *if* the temperature tomorrow is forecast to reach at least 23C *then* send out a tweet promoting our walking guides for the county.)

Building an Archive of Shared Links from a Number of Resources

This recipe uses the following triggers and actions:

- Trigger channel: Pocket
- Trigger: New item tagged

- Action channel: Diigo
- Action: Add a public bookmark

Expanding on the idea of sharing links to a range of networks, library news articles could also be pooled into a single social bookmarking service to build an archive of useful links for all to access. The idea is not only to share the information, but to keep an organized record of those links for future use. Many social networks experience a flood of information, and a link you shared months ago might be lost in that flood no matter how hard you try to track it down. I like to make sure I can find those old links I've shared elsewhere and provide a way for others to find them too. With IFTTT, it is easy to organise these shared links into a useful resource. In this instance, the recipe connects Pocket to the social bookmarking service Diigo (diigo.com) (Figure 1.23). Again, we would use the same Pocket New Item Tagged trigger [1], but this time we Add a Public Bookmark to Diigo [2].

When setting up the recipes, any tags you have added to the item in Pocket can be exported to the bookmarking site and are stored along with the link itself. In the following example, I use the tag *librarynews* to save the news article to the social bookmarking account along with a number of standard tags (including *library news* and *libraries*) and subject-specific tags (for example, *technology in libraries*) that I have added in Pocket. In addition to the URL of the original article and the tags, the recipe pulls through a description of the article to be stored as part of the process. Once the item has been bookmarked, users of the Diigo site will be able to find the item via a tag search.

Building a Source of Data for Mashups

A number of the examples I have provided use channels that not only share data, but store it too. For example, the CPD log information is stored as a spreadsheet and the shared link resource stores the information as bookmarks. With this in mind, it is easy to see how data could be collected to be used as a source in a data mashup. For example, instead of collecting details of your CPD activities via Google Drive, you could pull in mentions of the library from a variety of resources, either with a dedicated channel on IFTTT or with an RSS feed as output (e.g., news articles, social network updates, photographs, videos). You've then got a source of data you can make use of.

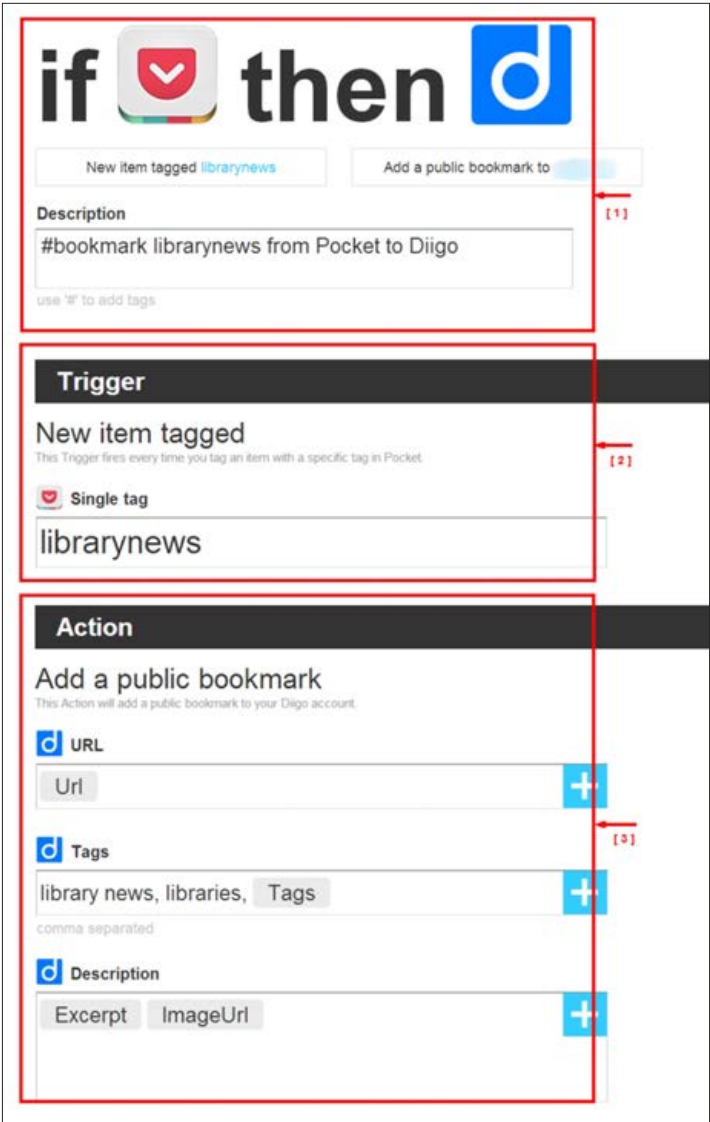


Figure 1.23 Share links from Pocket to Diigo

Google Drive spreadsheets provide users with the capability to easily publish the documents in various formats, including RSS and CSV output, and it provides a unique URL that can be accessed by data mashup tools. Diigo provides a RSS, which is also a useful output method. Again, this can be used as a data feed for a mashup.

The Human Touch

While IFTTT does most of the work for us, it is still important to include human input in the process when setting up the recipes, and also once they have been set up. When setting up recipes, we need to be aware of how data is structured in the trigger channel as its structure will impact what data is pulled through into the action channel. Once the recipes are up and running, you will need to regularly check that they are still active and that the data being pulled through is correct. Changes to a channel's authentication procedure may require you to re-activate that channel in IFTTT. Or changes to a channel's data structure could affect how that data is passed through to IFTTT. Such changes may be particularly relevant if you are using RSS feeds rather than dedicated channels.

Other Services

Throughout this chapter I have talked about using IFTTT to automate information sharing, as it is the service I'm most familiar with. However, there are other services available that work along very similar lines that you may want to try. If IFTTT doesn't quite suit your circumstances, information needs, or the way you work, it's worth taking a look at one of these other services:

- CloudWork (cloudwork.com)
- Elasti.io (elastic.io)
- We-Wired Web (wewiredweb.com)
- Zapier (zapier.com)

Like IFTTT, each of these services shares the common aim of passing data easily between different channels. However, the way they do this varies, as does the range of channels they work with. Of the services on this list, We-Wired Web and Zapier are most similar to IFTTT.

Summary

IFTTT is a way to connect different internet and communication services together, to move and share data and information between them. The examples in this chapter show just a small number of ways

you could use IFTTT. With over 90 channels, numerous triggers and actions within each channel, and access to untold numbers of RSS feeds and data sources, IFTTT offers countless opportunities for library and information services and anyone working in these sectors. If you are looking for more inspiration, browse through the thousands of the recipes IFTTT users have shared on the site (ifttt.com/recipes).

IFTTT can be a great time saver, especially if you are trying to cope with data sharing between a number of networks. If you are looking for a tool that will help you manipulate and share data across popular online services without the need to write code or understand APIs, it is worth investigating IFTTT.

About the Editor

Nicole C. Engard is the vice president of education at ByWater Solutions. Her primary role at ByWater Solutions is to educate librarians about open source software with a focus on the Koha ILS. In addition to her daily responsibilities, Nicole keeps the library community up to date on web technologies via her website “What I Learned Today ...” (www.web2learning.net) and volunteers as a community moderator on opensource.com. In 2007, Nicole was named one of *Library Journal*’s Movers & Shakers. She is the editor of *Library Mashups* (2009, Information Today, Inc.) and the author of *Practical Open Source Software for Libraries* (2010, Chandos Inc.). She also worked on the second edition of *The Accidental Systems Librarian* with Rachel Singer Gordon (2012, Information Today, Inc.).

If you enjoyed reading this chapter of *More Library Mashups*, please visit the ITI bookstore to pre-order your copy.

