# Do One Thing and Do It Well

In 2007 Eric Schmidt, the former Google CEO, was asked to summarize his vision for the future of app development and app distribution.

Schmidt answered that he expected apps to be pieced together from reusable components. He expected that apps would be relatively small. App data would be stored remotely. App development would have a low barrier to entry. App distribution would be viral, using email and social media.



'... applications will be pieced together ... applications will be small ...'

Eric Schmidt, 2007

In short, the vision was that apps would be *small* - the antithesis of application behemoths struggling under their own weight of feature bloat. Each app would do one thing, and do it well. If more functionality was needed - well, then there would be an app for that.

Fast forward to 2015.

Schmidt's vision for easily connecting small apps together is being realized with technology called web API automation platforms. Each app exposes a Web API and the automation platform joins the dots together.

Schmidt's vision of a low barrier to entry is also being achieved. The Web API automation platforms enable nontechnical users to assemble new functionality: Web APIs are integrated without writing a single line of code.

Web API automation platforms are finding use within digital learning products and recent entrants to the mature Learning Management System (LMS) markets distinguish themselves with Web API automation already on-board. No longer the bloated LMS products of the 20th Century, these lean 21st century offerings do one thing and do it well: course, student and workflow management is given pride of place. If more functionality is needed - then there are apps for that.

**Web API Automation Platforms**

Web API automation platforms are Web-based services that enable nontechnical users to create conditional statements: a change in one web service automatically triggers an action in a different web service when a condition is met.

Some examples: If I post a picture on Instagram, then the photo is saved to my Dropbox. If my company name is mentioned on Twitter, then a new row is added to my Google spreadsheet. If I tag an email, then my task list is updated.
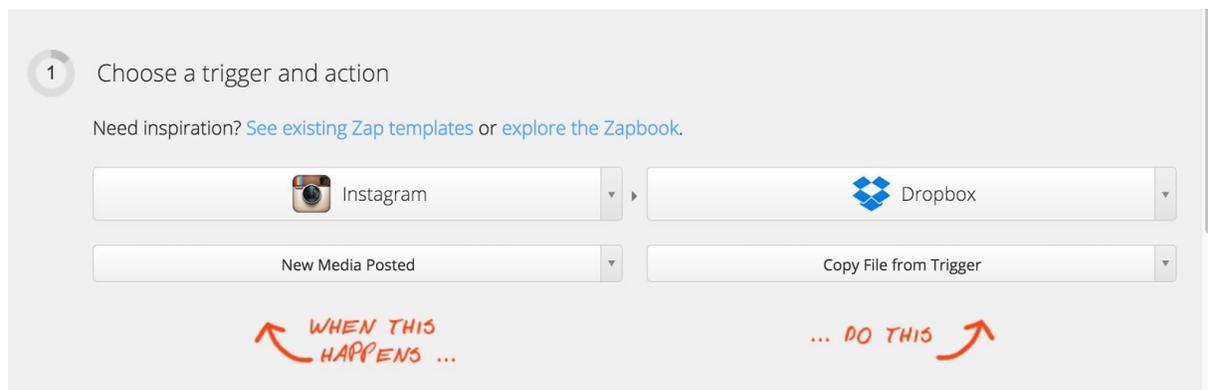


*A picture posted on Instagram becomes a trigger for an action to add the image file to Dropbox. The Web API automation platform makes the connection.*

If one Web API changes state, then a different Web API is updated. Or as one entrepreneur named his start-up project: "If this, then that".

Different Web API automation platforms use different vocabularies to express this conditional statement. For the Zapier platform the conditional statement is a "Zap". For the IFTTT platform the conditional statement is a "Recipe". For Elastic IO the statement is a "Connector" and for We Wired Web it is a "Task".

In all cases the intent is clear - to translate the vocabulary of the software developer, the conditional statement, into language familiar to the nontechnical user. The barrier to entry is lowered. Nontechnical users can start to piece together new behaviours without writing a line of code.

The conditional statement (the "Zap", the "Recipe", the "Connector" etc.) is composed of two parts: first, a trigger event (when the state of one API resource changes) and second, an action (a change to the state of a second API resource). The Web API Automation Platform is the system for creating, hosting and maintaining these conditional statements.

So the current breed of Web API Automation Platforms are best described as an Integration Platform as a Service (iPaaS)  layered with a user interface that enables nontechnical users to wire together disparate Web APIs.

**Digital Learning Platforms**

Digital learning platforms, cloud applications for education and training, trace their origins to Learning Management System (LMS) products: Web-based services for managing courses, learners, content and assessments. LMS products now exist in a mature market, having been available since the 1990s.

A legacy of this [two decade history](#) is that the LMS products accrued feature bloat and a low quality user experience, the antithesis of the simplicity and a consumer grade user experience that 21st century users expect.

In response to market saturation by established products, recent disruptive entrants have sought to distinguish themselves by doing one thing and doing it well:  by paring down the product offering to a set of core features - and then delegating other requirements to third-party apps.

The poster child for this new approach is a stripped down, LMS-lite product called [Classroom](#) released in 2014 by Google that handles basic functionality for student cohorts and course management. Third-party integrations are supported, but most functionality is delegated to Google apps, such as Google Drive, via Web APIs.

Other recent entrants have sought to use Web API automation platforms to similar effect: TalentLMS, an eLearning tool released in 2012, places emphasis on usability and workflow, not feature breadth. TalentLMS  uses the [Zapier Platform](#) to integrate with Google Mail and Google Calendar applications to extend the feature breadth of the LMS.

The result of this new approach is a consumer grade user experience for core functionality integrated with a bespoke collection of third-party, best-of-breed tools. Features, such as push notifications for learners, or dashboards for course leaders, are just apps connected to the core platform using Web API automation.

**Notifications for Learners**

Course notifications are a required feature of digital learning platforms: a teacher needs to broadcast notifications to the students of their course. Traditional LMS products implemented such a feature more, or less, successfully - but Web API automation platforms make the entire implementation redundant.

The digital learning platforms simply expose the notification trigger as a Web API; apps expose an update action as a Web API; and the Web API automation platforms make the connection.

The screenshot below shows that this web service orchestration is achieved without programming - using familiar web form controls instead: a course announcement event in a digital learning platform becomes a trigger for an action to create a new reminder in Evernote.
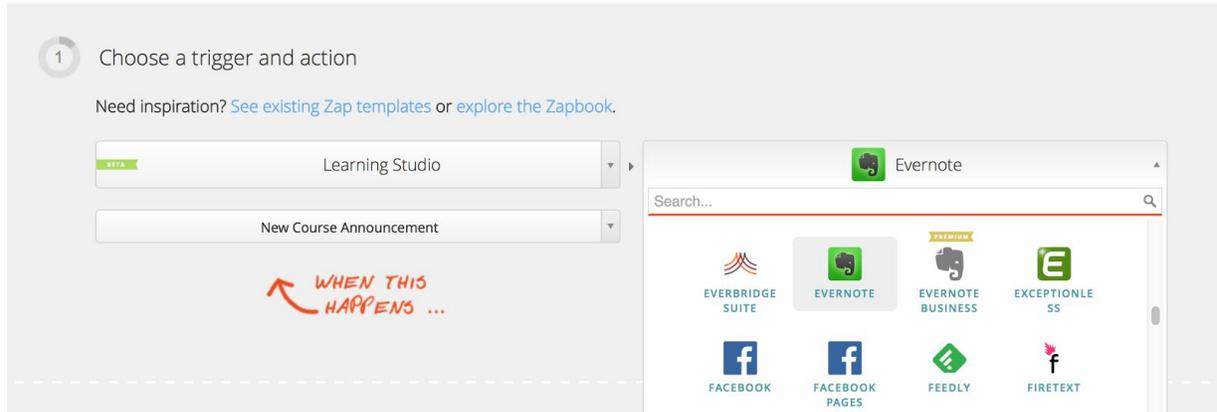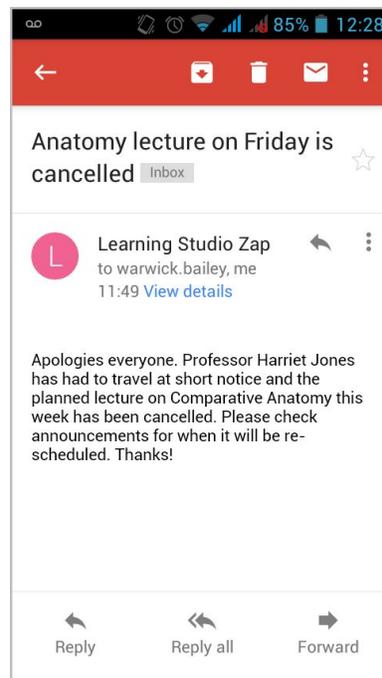


*A course announcement event in a digital learning platform becomes a trigger for an action to create new reminder in Evernote. The Web API automation platform makes the connection.*

The result: students personalize the core digital learning platform product with their choice of notification, on their choice of best-of-breed, third-party app, on their choice of device.
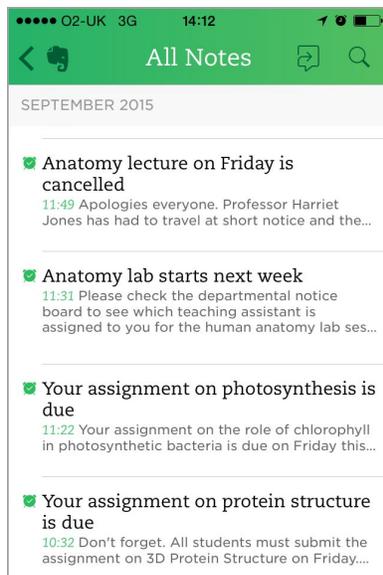


*Course announcement notification as a to-do list reminder.*

*Course announcement notification as email.*

*Course announcement as comment in a social network activity feed.*

**Dashboards for Course Leaders**

Dashboards are a required feature of digital learning platforms: an academic administrator needs a window into course analytics. What metrics identify students at risk of failure? Which courses have the lowest dropout rates? Traditional LMS products may have such a dashboard, or not - but Web API automation platforms make it a moot point.

The digital learning platforms simply expose course events as a trigger in a Web API; big data and dashboard services expose an update action as a Web API; and the Web API automation platforms join the dots.

The screenshot below shows that this connection is achieved with zero coding: a course event in a digital learning platform becomes a trigger for an action to create new data in a learning record store.
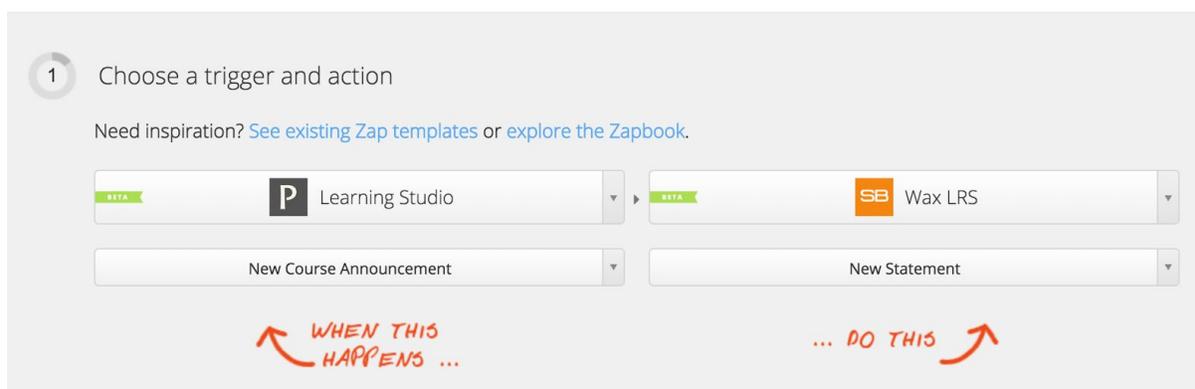


*A course announcement event in a digital learning platform becomes a trigger to create a new data point in a learning record store. The Web API automation platform makes the connection.*

The result: academic administrators can use state of the art analytics dashboards against data stores populated with records from their choice of digital learning platforms.
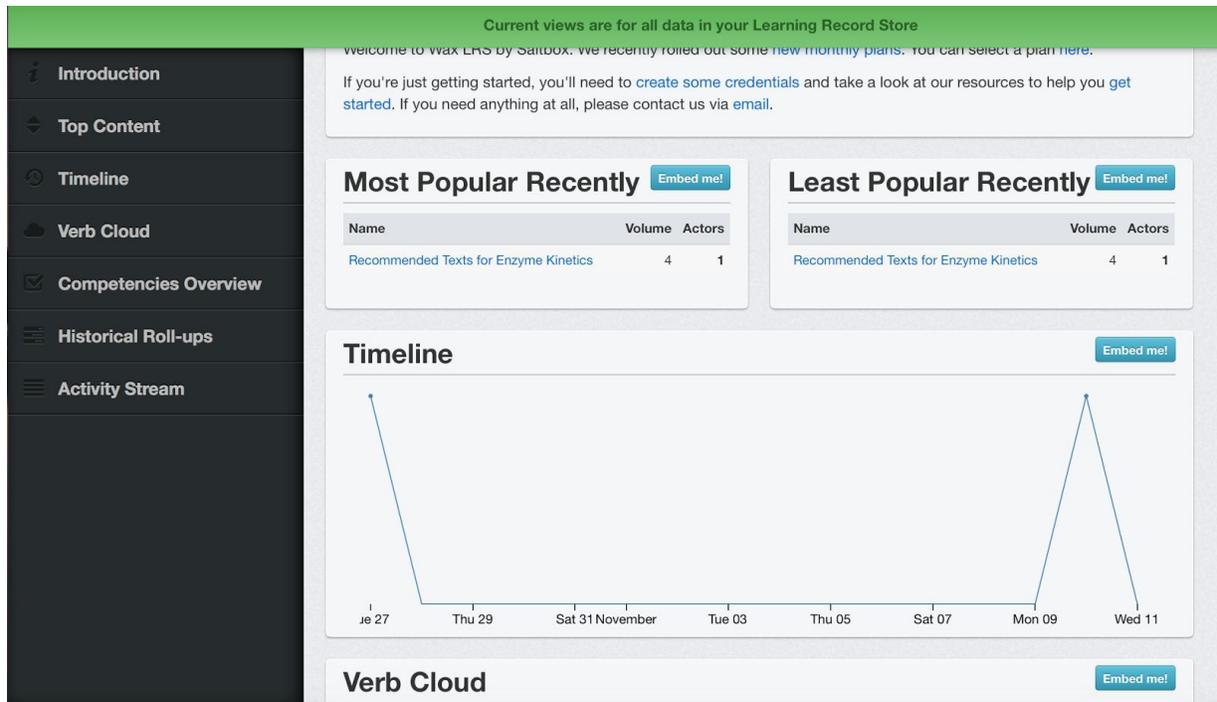
*A dashboard for an academic administrator provides a window into a learning record store populated with records from their choice of digital learning platforms.*

**Do One Thing and Do It Well**

This article has described how Web API automation platforms, such as Zapier, enable different software-as-a-service applications to be integrated without coding and at low cost.

Digital learning platforms are applications for managing courses, and can be extended using Web API Automation. Students, for example, may personalize the core product with their choice of best-of-breed, third-party app, on their choice of device to receive course notifications.

Web API automation platforms reinforce the 2007 prediction from Eric Schmidt that "apps should be small" - that software-as-a-service applications need to do one thing only, but do it well. There is no need for feature bloat, because features from best-of-breed, third-party applications can be added, as needed, to a core product using Web API automation.

The digital learning platform needs to do one thing - course, student and workflow management - and do it well. Other features, whether notifications, data analytics, threaded discussion or assessment, can be delegated to third party applications - each of which do their one thing-and it do it well.

Web API automation platforms provide the low-cost, no-code solution for the movement of data - integrating and extending the digital learning platform with a bespoke collection of tools.

**Warwick Bailey**
European Director, Pearson Developers Network
warwick.bailey@pearson.com
@techwarwick