

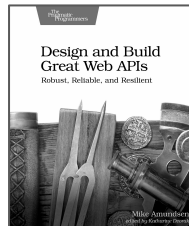
Design and Build Great Web APIs

Summary

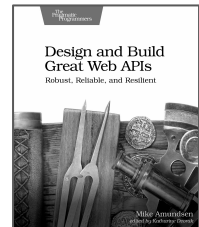
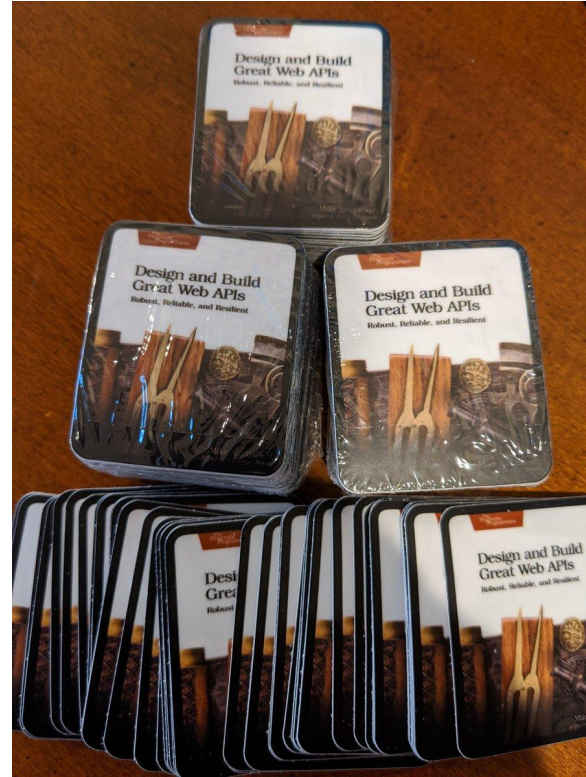
@mamund

Mike Amundsen

training.amundsen.com



Don't forget your 25% discount & your laptop stickers!



Rate today's session

Cyberconflict: A new era of war, sabotage, and fear [See passes & pricing](#)

David Sanger (The New York Times)
9:55am-10:10am Wednesday, March 27, 2019
Location: Ballroom
[Add to Your Schedule](#)
[Add Comment or Question](#)

[Secondary topics: Security and Privacy](#)

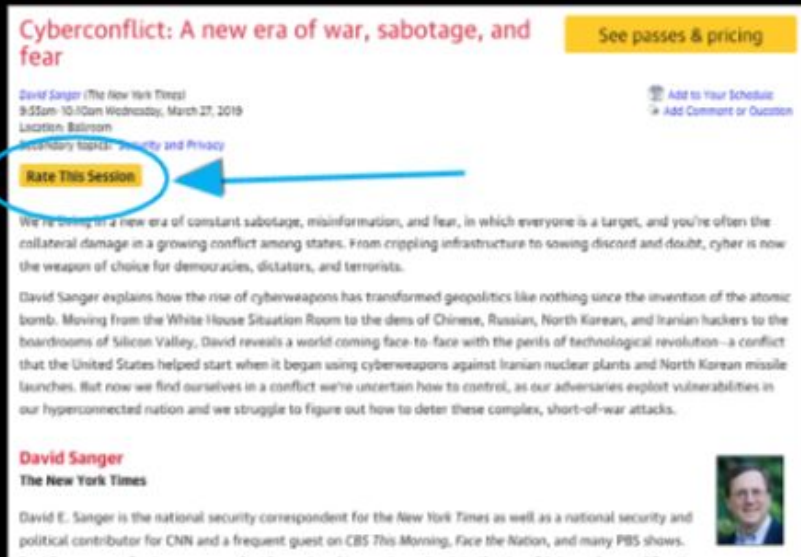
Rate This Session

We're living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you're often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we're uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David E. Sanger is the national security correspondent for the New York Times as well as a national security and political contributor for CNN and a frequent guest on CBS This Morning, Face the Nation, and many PBS shows.



Session page on conference website

✓ **Attending** [Notes](#) [Remove](#)

Cyberconflict: A new era of war, sabotage, and fear

9:55 AM - 10:10 AM, Wed, Mar 27, 2019

Speakers


 **David Sanger**
National Security Correspondent
The New York Times

📍 Ballroom

Keynotes

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

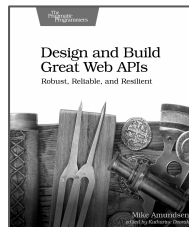
[SESSION EVALUATION](#)



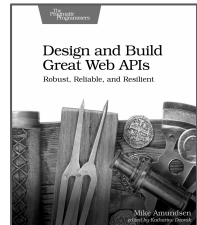
O'Reilly Events App

Design and Build Great Web APIs

- Principles
 - *Install & Setup*
- Designing
 - *Profile Exercise (ALPS yaml file)*
- Building
 - *DARRT Exercise (NodeJS & npm)*
- Releasing
 - *Deployment Exercise (Heroku CLI)*



Principles

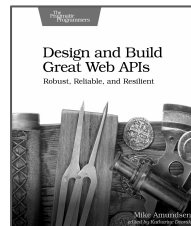


Principles : API-First

"API-first design means identifying and/or defining key actors and personas, determining what those actors and personas expect to be able to do with APIs."

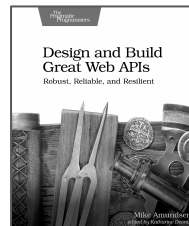
-- Kas Thomas, 2009

<http://asserttrue.blogspot.com/2009/04/api-first-design.html>



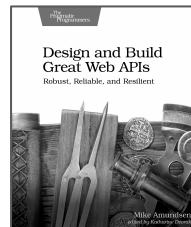
Principles : HTTP, the Web, and REST

- HTTP is a protocol
- The Web is a set of common practices
- REST is a specific style



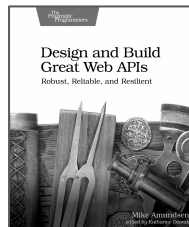
Principles : Interactions

- Norman's Action Lifecycle
- Request - Parse - Wait (RPW Loop)
- Modeling Interactions

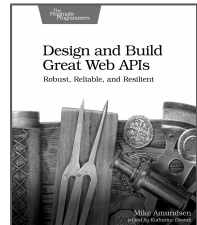


Principles

- API-First
 - Solve your API consumer's problem, not yours
- HTTP, the Web, and REST
 - Rely upon standards, common practices, & style
- Modeling Interactions
 - Use loops to allow API consumers to control interactions



Designing APIs



Document the Story

- Interview each stakeholder
 - Record their references to data, actions, processing, purpose
- Use plain language
 - Speak in the language of th stakeholder, not the developer
- Document each story **separately**
 - Share w/ each stakeholder for validation
- Write composite story to cover all stakeholders
 - Share w/ the group and work out details

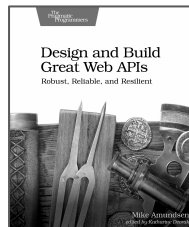
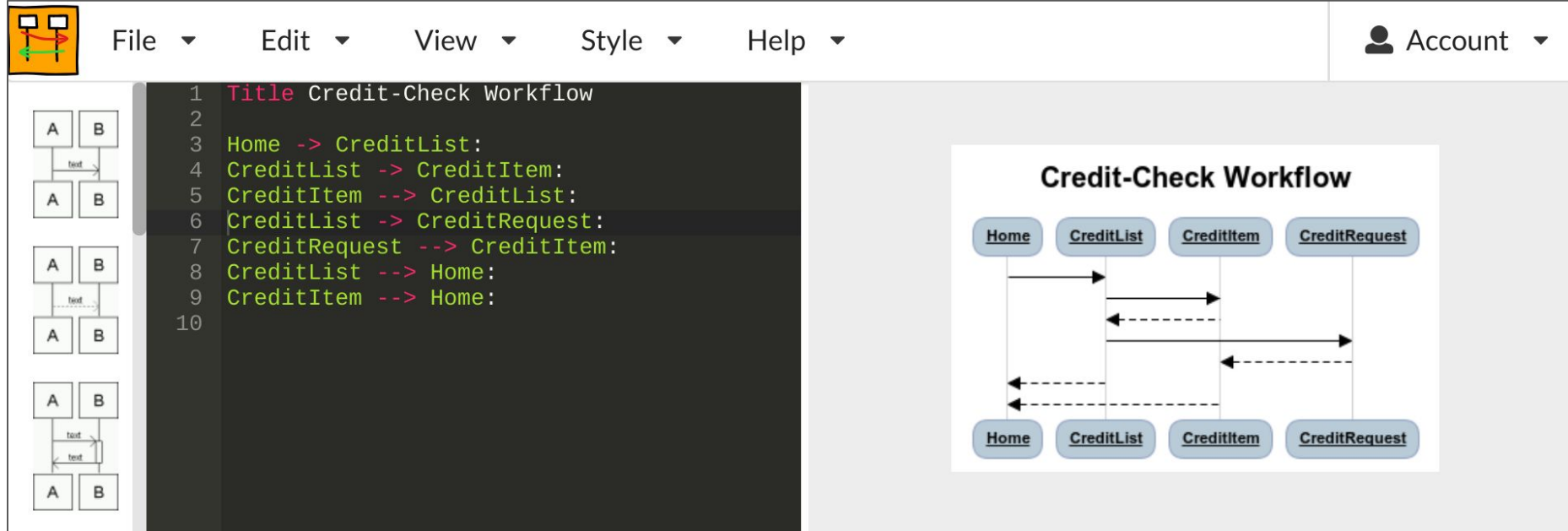


Diagram the Flow : websequencediagrams.com



The screenshot shows the websequencediagrams.com interface. On the left is a toolbar with icons for creating lifelines, messages, and other diagram elements. The main editor area displays a sequence diagram titled "Credit-Check Workflow" with the following code:

```
1 Title Credit-Check Workflow
2
3 Home -> CreditList:
4 CreditList -> CreditItem:
5 CreditItem --> CreditList:
6 CreditList -> CreditRequest:
7 CreditRequest --> CreditItem:
8 CreditList --> Home:
9 CreditItem --> Home:
10
```

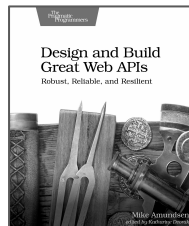
On the right, a preview of the diagram is shown. It features four lifelines: Home, CreditList, CreditItem, and CreditRequest. The messages are as follows:

- Home to CreditList (solid arrow)
- CreditList to CreditItem (solid arrow)
- CreditItem to CreditList (dashed return arrow)
- CreditList to CreditRequest (solid arrow)
- CreditRequest to CreditItem (dashed return arrow)
- CreditList to Home (dashed return arrow)
- CreditItem to Home (dashed return arrow)



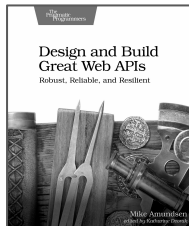
Describe the API : ALPS

- Application-Level Profile Semantics
 - Amundsen-Richardson-Foster (2011)
- Identifies all interface properties
 - Id, familyName, givenName, telephone, etc.
- Identifies all interface actions
 - saveCompany, setStatus, approvePayroll, etc.
- Does not include implementation details
 - URLs, schemas, methods, response codes, etc.

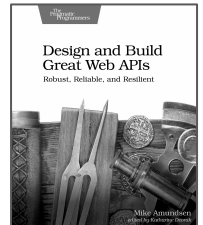


Designing

- Document the Story
 - Purpose, Actions, & Data
- Diagram the Flow
 - Sequence diagrams are not HTTP
- Describe the API
 - Before OpenAPI, AsyncAPI, protobuf and SDL



Building APIs



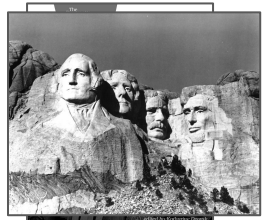
Sketching APIs

- Sketches are terse, rough drawings
- They give the general idea of a thing but lack important details.
- Usually, one can glean the basics from a sketch but
- Sketches usually are just explorations of ideas, not fully-formed items.



Prototyping APIs

- Prototypes look like the real thing, but are not. They're "fakes."
- They let you work up something with all the details of a real API, but without the actual functionality behind it.
- They're an inexpensive way to work out the details
- Use them to discover challenges before you go into production.



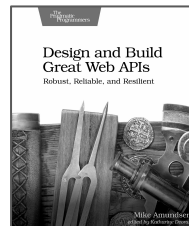
Building APIs

- API builds are the real thing
- Production-ready, access-controlled, resilient, scalable.
- Building the production implementation means
 - Working out all the kinks
 - Supporting all the use-cases identified during the sketch and prototype phases.

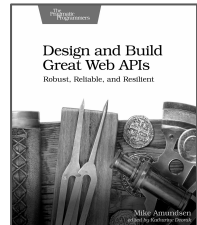


Building

- Sketching your API
 - Sketches are made to be thrown away
- Prototyping your API
 - Prototypes are made to be tested
- Building your API
 - Builds are forever



Releasing APIs



Testing APIs : BDD

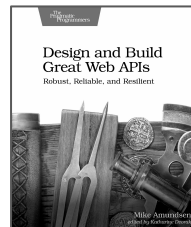
- Behavior-Driven Development (2006)
- Dan North, Thoughtworks
- Outside-in

Dan North, the developer of the BDD, described it as: "...a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters."



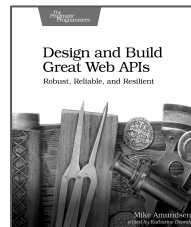
Securing APIs - Security Basics

- Authentication (Identity)
- Authorization (Access Control)
- TLS/HTTPS (message encoding)
- Encryption (field-level encoding)



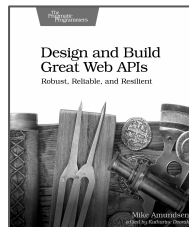
Deploying APIs - Challenges

- Deploying your app can be complicated
- Compatibility
 - Hardware
 - OS
 - Platform
 - Framework
- Dependencies

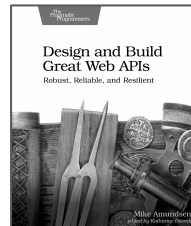


Releasing

- Testing
 - From request lists to BDD
- Securing
 - Identity and Access Control
- Deploying
 - Automation is your friend



Thanks!



Design and Build Great Web APIs

Summary

@mamund

Mike Amundsen

training.amundsen.com

