SOLID in Action: from Slack to Twilio

Micah Silverman | Principal Technical Platform Instructor


1515

Welcome!

- Agenda
  - SOLID in Action (35 mins)
  - Okta 101 (5 mins)
  - Q&A (10 mins)

S.O.L.I.D

S	SRP	Single Responsibility Principle
O	OCP	Open-Closed Principle
L	LSP	Liskov Substitution Principle
I	ISP	Interface Segregation Principle
D	DIP	Dependency Inversion Principle

https://join.slack.com/t/random-magic-card/shared_invite/enQtNDI1NDYyOTkwNTE2LWQ2ZTg5NTA1ODkxZGM1ZWNhNjUyNTJlZGY2ZDFhNDNmM2ZmYzY2Njk2YTY1NmZhN2RhYmIzMDhiZmM3ZWIxMTQ

1-929-236-9306

MMS

81% 2:14 PM

+19292369306

Sunday, October 1, 2017

4:57 PM Magic

MMS
4:58 PM

lrtf
Micah Silverman

All Threads

Channels
general
random

Direct Messages
slackbot
Micah Silverman (you)
MPS

+ Invite People

Apps

#general
2 | 0 | Company-wide announcements and work-based matters

Search

#general

You created this channel today. This is the very beginning of the #general channel. Purpose: *This channel is for workspace-wide communication and announcements. All members are in this channel.* (edit)

+ Add an app Invite people to lrtf

Today

Micah Silverman 1:19 PM
joined #general along with MPS.

Micah Silverman 1:34 PM
/magic

magic APP 1:34 PM
(23kB)

Message #general

Slack

D	DIP	Dependency Inversion Principle

High-level modules should not depend on low-level modules. Both should depend on abstractions.

Abstractions should not depend on details. Details should depend on abstractions.

tag: slack-first-pass

I	ISP	Interface Segregation Principle

Many client-specific interfaces are better than one general-purpose interface.

tag: slack-first-pass

S	SRP	Single Responsibility Principle

A class should have only a single responsibility.

“A class should have only one reason to change.” - Robert (Uncle Bob) Martin

tag: twilio-breaks-srp

The Magic TwiML Problem

http://.../Image.ashx?multiverseid=144276&type=card

```
<Response>
    <Message>
        <Body/>
        <Media>
            http://.../Image.ashx?multiverseid=144276&amp;type=card
        </Media>
    </Message>
</Response>
```

The Magic TwiML Problem

http://.../Image.ashx?multiverseid=144276&type=card

```
<Response>
    <Message>
        <Body/>
        <Media>
            <![CDATA[
             http://.../Image.ashx?multiverseid=144276&type=card
            ]]>
        </Media>
    </Message>
</Response>
```

The Magic TwiML Problem

http://.../Image.ashx?multiverseid=144276&type=card

```
<Response>
    <Message>
        <Body/>
        <Media>
            http://myserver/api/v1/magic_proxy/144276
        </Media>
    </Message>
</Response>
```

S	SRP	Single Responsibility Principle

A class should have only a single responsibility.

“A class should have only one reason to change.” - Robert (Uncle Bob) Martin

tag: twilio-breaks-srp

D	DIP	Dependency Inversion Principle

High-level modules should not depend on low-level modules. Both should depend on abstractions.

Abstractions should not depend on details. Details should depend on abstractions.

tag: modules-ftw

L	LSP	Liskov Substitution Principle

Objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.

tag: slack-violates-lsp

O	OCP	Open-Closed Principle

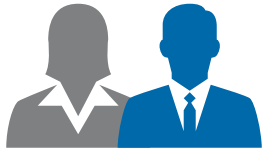
Software entities should be open for extension,
but closed for modification.

tag: master

S.O.L.I.D Recap

S	Single Responsibility Principle	Tag: twilio-fixes-srp - break TwilioController in two to keep each controller singly purposed.
O	Open-Closed Principle	Tag: master - SlackResponse is complete and does not need to be changed. It can be extended without changing existing service code.
L	Liskov Substitution Principle	Tag: master - none of the SlackResponse children returns null or has unneeded methods or annotations.
I	Interface Segregation Principle	Tag: slack-first-pass through master - MagicCardService and SlackResponseService perform different functions and are therefore separate services
D	Dependency Inversion Principle	Tag: slack-first-pass through master - Dependent services are autowired into controllers. Constructor injection is the “best practices” way to do dependency injection. Modules enforce DIP.

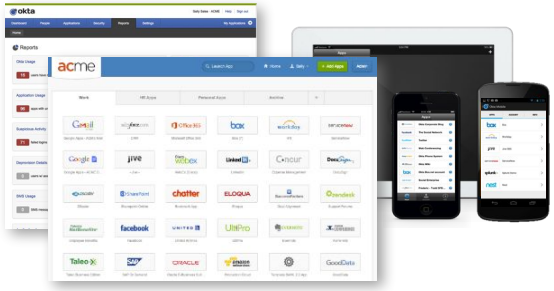
Okta Enables Companies to Do Both IT & API

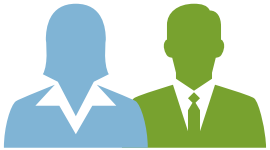


Employees,
Contractors

okta

Identity & Mobility Service:
Identity Management
Mobility Management
Strong Authentication

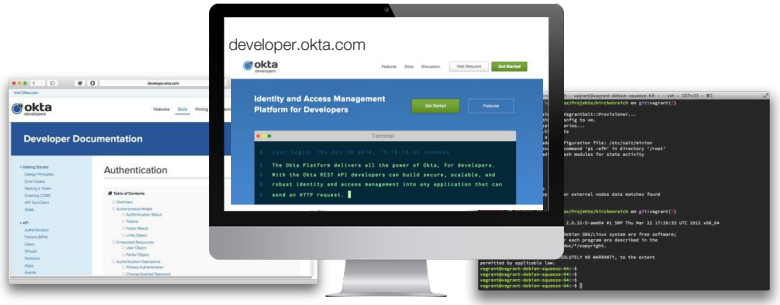




Partners,
Customers

okta

Identity API Products:
Identity API for Any App
Developer Tools
and Community



Resources

- https://github.com/dogeared
  - /solid-from-slack-to-twilio
  - /magic-with-slack-and-twilio
- @afitnerd @OktaDev
- https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)
- http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod
- OIDC Playground - https://okta-oidc-fun.herokuapp.com
- https://developer.okta.com/blog/2017/07/25/oidc-primer-part-1
- HTTPie - github.com/jkbrzt/httpie


okta
Thank You