

# API-First Development with Swagger

[fred@willerup.com](mailto:fred@willerup.com)

# about:me

- .net, angularjs
- ruby on rails, ios
- nodejs, mongodb
- cd; azure, aws, heroku





## LeanLaw - software for lawyers

- .Net WebAPI with Angular front-end
- nodejs microservice
- Chrome app, and mobile
- Gmail extension and Outlook add-in
- Integrations: Google Apps, Quickbooks, ...
- CD using Azure and TeamCity

# API-First Development with Swagger

# API-First Development with Swagger

# API-First Development with Swagger

Tools for building systems  
with consistent, stable  
REST APIs



**REST+JSON**



**SOAP+XML**





**REST+JSON**



**GraphQL, jsData?**

Tools for building systems  
with consistent, stable  
REST APIs

## Specifically

- swagger the **language** for spec'ing APIs
- the swagger **editor**
- swagger for online **documentation** (UI)
- auto-generating **client libraries** (C# and js)
- generating **server-side** code
- **spec-first** vs code-first workflow

# Swagger ML and editor for spec'ing REST APIs

<http://editor.swagger.io/>

<http://editor.swagger.io/#/?import=http://localhost/start.yml>

## Example

GET /v1/customers - **list** customers

POST /v1/customers - **create** customer

GET /v1/customers/123 - **get** customer

DELETE /v1/customers/123 - **delete** customer

PUT /v1/customers/123 - **update** customer

# Resources

```
/customers:
  get:
    description: Get list of customers.
    responses:
      200:
        description: List of customers
        schema:
          type: array
          items:
            $ref: '#/definitions/Customer'
  post:
    description: Create a customer
    parameters:
      - name: customer
        in: body
        schema:
          $ref: '#/definitions/Customer'
    responses:
      201:
        description: Customer
        schema:
          $ref: '#/definitions/Customer'
```

# Parameters

- Path
- Query
- Body

```
parameters:  
  - name: id  
    in: path  
    required: true  
    type: string
```

```
name: offset  
in: query  
description: Index of first item to return  
type: integer  
format: int32
```

```
parameters:  
  - name: customer  
    in: body  
    schema:  
      $ref: '#/definitions/CustomerCreate'
```

# Data types

- String, date
- Bool
- Numbers
- Enum
- Array



# definitions

```
CustomerCreate:
  properties:
    name:
      type: string
    email:
      type: string
    active:
      type: boolean
CustomerSummary:
  properties:
    id:
      type: string
    href:
      type: string
    name:
      type: string
CustomerDetail:
  allOf:
    - $ref: '#/definitions/CustomerSummary'
    - properties:
        email:
          type: string
```

# Tags

```
POST http://localhost:3000/api/v1/operations/operationId:  
post:  
  tags:  
  - admin  
  description: Create a customer  
  operationId: createCustomer
```

# Examples

```
responses:
  200:
    description: Customer detail
    schema:
      $ref: '#/definitions/CustomerDetail'
    examples:
      application/json:
        id: abc123
        href: 'https://api.mysite.com/v1/customers/abc123'
        name: 'Customer Ltd'
        email: 'customer@company.com'
        active: true
        joined: '2015-09-11'
```

# Security

```
securityDefinitions:
  my_apikey:
    type: apiKey
    name: api_key
    in: header

  my_oauth:
    type: oauth2
    authorizationUrl: http://api.mysite/oauth2/auth
    flow: implicit
    scopes:
      write:customers: modify customers
      read:customers: read customers

  my_basic:
    type: basic

security:
  - my_apikey: []
```

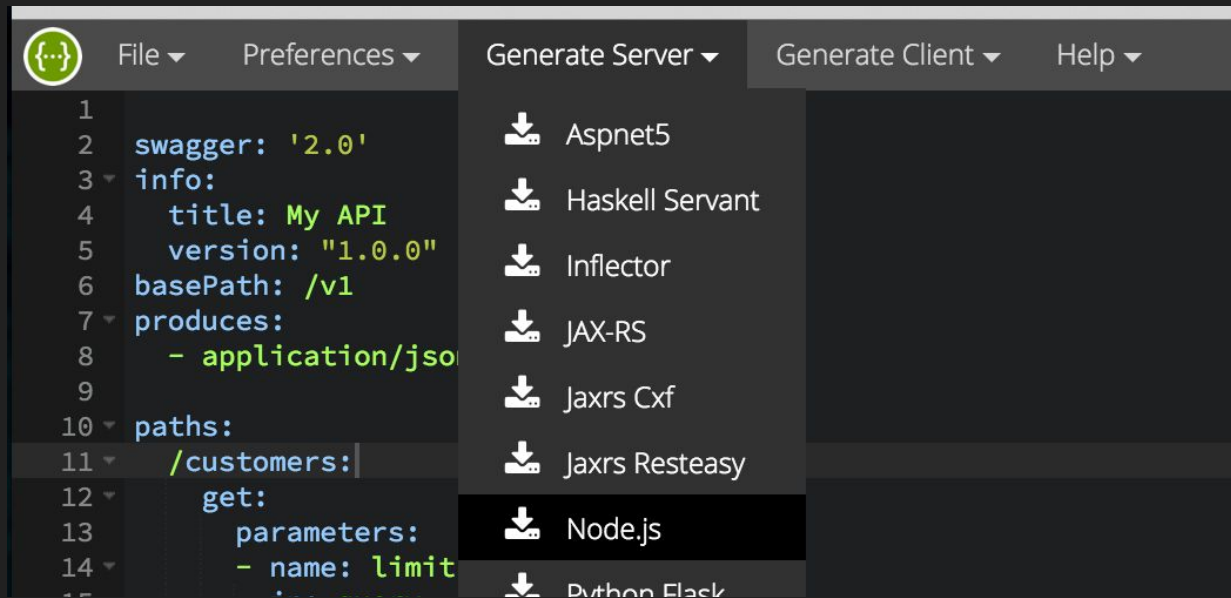
# Swagger issues

- Duplication
  - Nested resources
  - parameter groups
  - Limitations on data model
- Microservice limitations
  - Pieces from individual services combined into one?
- Bizarre error reporting

# Tools

- Server
  - swagger-node
- Codegen
  - Javascript client library
  - C# client library
- Swashbuckle
  - adding Swagger to existing WebAPI project
- API gateways
  - Azure, aws

# Server



<https://github.com/swagger-api/swagger-node>

# Javascript Client Libraries

- generate dynamically from URL
  - swagger-js
- or use
  - swagger-js-codegen
- or just use \$http



# C# Client Libraries: Swagger Codegen

- Generate client libraries from Swagger spec
- Invoke from editor or run locally (java app)
- Needs
  - Packages: RestSharp and Newtonsoft.Json
  - Reference: System.Runtime.Serialization
- Project file issues
  - <https://github.com/willerup/swagger-demo/tree/master/WebAPI/scripts>

# Swashbuckle

- Generates Swagger spec (and UI) from existing WebAPI
- <https://github.com/domaindrivendev/Swashbuckle>

# API Gateways

- aws

- <https://github.com/aws-labs/aws-apigateway-importer>

- Azure

Import API

From clipboard

From file

From URL

Specification document URL

<http://calcapicloudapp.net/calcapicloudapi.json>

Specification format

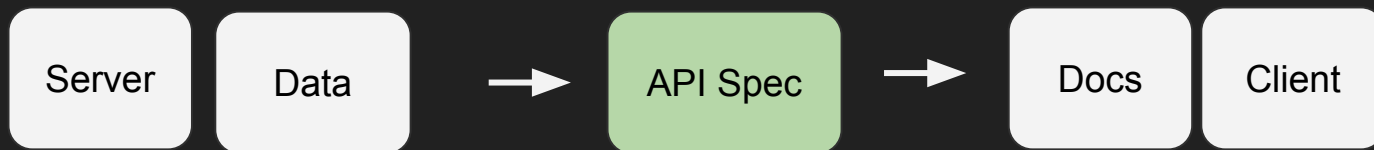
☐ WADL

☒ Swagger

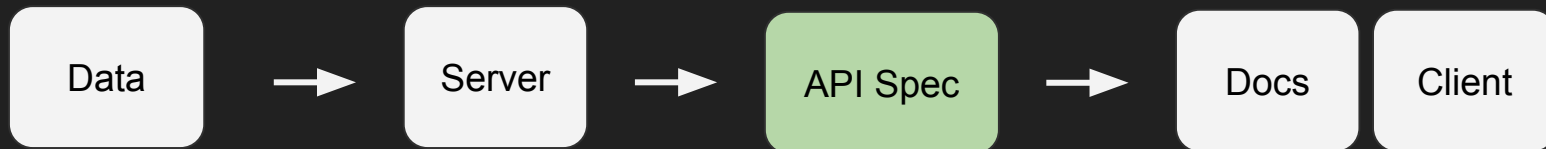
# API-First Development with Swagger

# API-First Development with Swagger

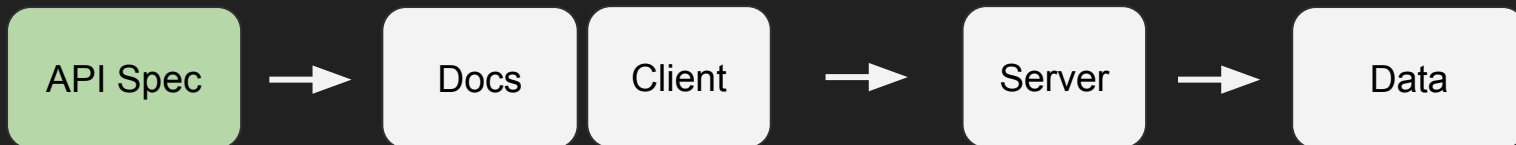
## Code-First



## Database-First



## Spec-First



# Why API first (or “spec first”)?

- APIs are the center of a micro-service architecture
  - Especially a heterogeneous one
- Enables third-party integration / eco-system
- Prevents leakage of internal data
- Design - mock - test - build

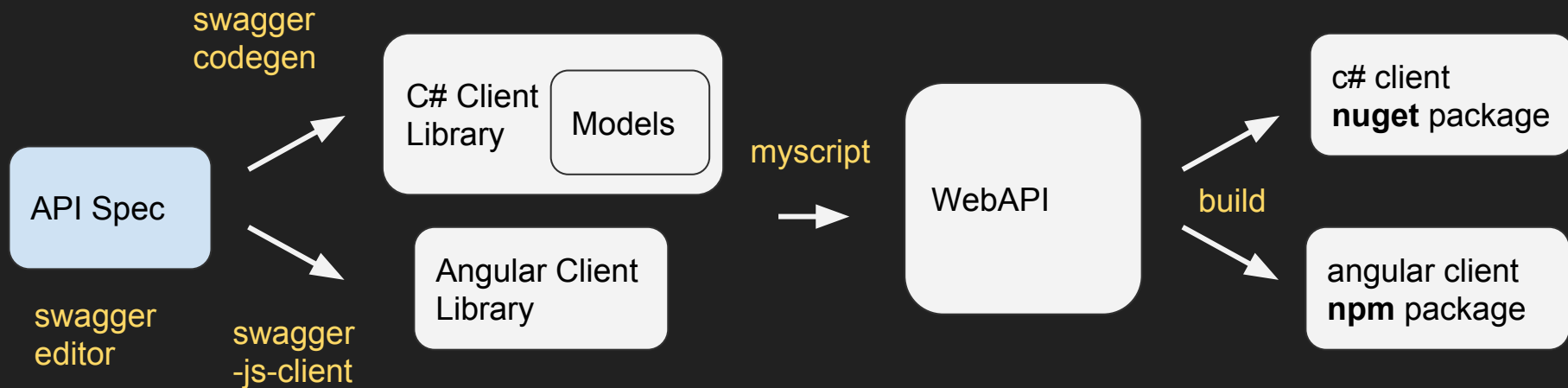
Why not? In many cases it is more work

“Architecture is  
making decisions about  
*what is hard to change*”

*Martin Fowler, sort of*



# Our (current) spec-first flow in more detail



MAKE  
YOUR  
LOOK  
FUN  
CARE

Questions, Comments, Ridicule?

# Contact

[fred@willerup.com](mailto:fred@willerup.com) / [fwillerup@leanlaw.co](mailto:fwillerup@leanlaw.co)

@fwillerup