

DataHub

Getting Mock Done.

Mock 解决方案

张宇恒

2018.09.23

关于我

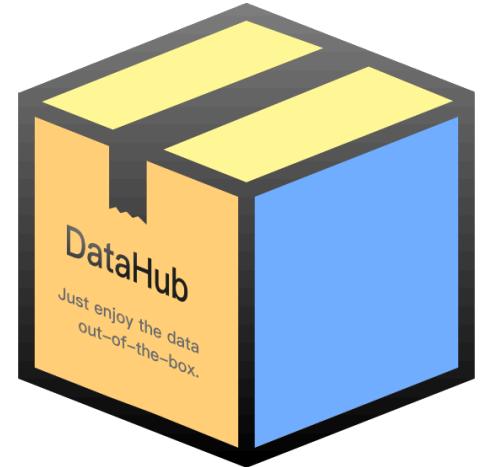
- web developer @Ant Financial
- twitter [@officer_wu](#)

目录

1. DataHub 是什么，能用来做什么
2. 业务实践，以及带来的价值
3. Q & A

01 | DataHub

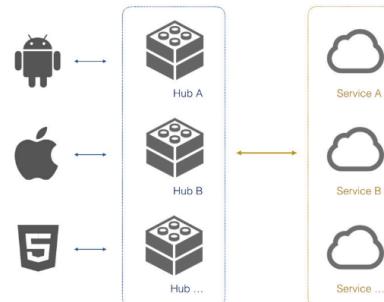
Just enjoy the data out-of-the-box



Home Page - DataHub [Not Secure | datahubjs.com](#)

 DataHub

Guide Issues



DataHub - Continuous data provider for development, testing, staging and production.

[GO NOW](#) [GITHUB](#)

server: v2.2.4, view: v2.1.32

 cloud base  teamwork  request snapshot  data control  quick start  scene manage

 continues  generate document  versioning  response setting  data migration  quick save

 decentralized  open API  cli tool  i18n support  open source

Macaca • Solution for Automati X +

https://macacajs.github.io/datahub#datahub

Home Document Support 中文

Macaca

Guides

- Introduction
- Environment Setup
- Quick Start
- Helpful Settings
- Versioning

API

- Node.js
- Java
- Python

Advanced

- Debugging
- Inspector
- Recorder
- DataHub ✓
- Coverage
- NoSmoke
- Monkey Testing
- Computer Vision

DataHub

DataHub is a multi-scenario data environment management solution provided by Macaca.

DataHub - Continuous data(testing/mock) provider for development, testing, staging and production.

A Comprehensive Solution

DataHub is born to solving the lifecycle needs of mock/testing data of software development, from development, testing, staging to final production. Software engineers and test engineers use DataHub to manage their mock/testing data.

```
graph LR; DeviceA[Android] <--> HubA[Hub A]; DeviceB[Apple] <--> HubA; HubA <--> ServiceA[Service A]
```

0. Preface

- There is **NO** prefect solution.
- 根本目的：用 Node.js 技术来更好地服务“web 应用”研发

Use data out-of-the-box



- Dev: for **local** development and debugging
- Debug: proxy to real server and save **snapshot**
- Test: test automatically, use the exactly **same** data from debugging stage
- Release: generate api docs include different scenes automatically

Use data out-of-the-box



- Stop writing code like this

Use data out-of-the-box



```
http://localhost:8080?scene=success
```

```
const { scene } = parseQuery(location.search)
```

```
request(`./mock/api/${scene}`)
```

Use data out-of-the-box



- We **usually** maintain mock data like this

Use data out-of-the-box



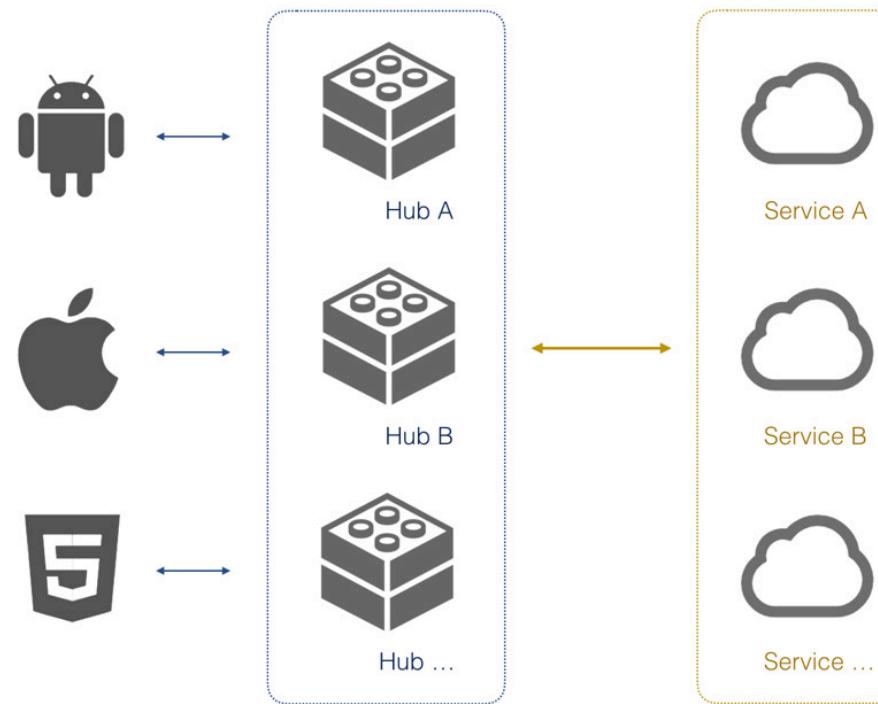
```
mock
  └── api_a
      └── error.js
      └── success.js
  └── api_b
      └── error.js
      └── success.js
```

Use data out-of-the-box



- Why not?
 - Remove hack code, for clean-code & maintainability
 - Store mock data in js or json file **manually** is neither **flexible** nor **powerful**

Use data out-of-the-box



<https://github.com/macacajs/macaca-datahub>

Project - owner - DataHub Not Secure | datahubjs.com/project/owner#/?

DataHub

产品介绍 问题反馈

接口列表 实时快照

所有项目 / owner / 项目配置

Rewrite response 响应延迟 0 秒 修改 接口文档

api/login
Login request method: ALL

场景管理 预览场景数据 (支持 GET|ALL 类型请求)

搜索场景 新增场景

fail default

点击场景进行切换:

代理模式 代理模式已关闭 添加代理 Url 地址 全局代理模式已关闭

请求字段描述

是否开启校验 编辑

字段	类型	是否必需	参数说明
No data			

Use data out-of-the-box



- Easy-use GUI dashboard
- One-key switch:
 - Proxy or mock mode
 - Return scenario (success or error) data

Use data out-of-the-box



- Inspect **real-time** network traffic
 - Save response
 - Rewrite response

Project - owner - DataHub datahubjs.com/data/owner/api/

Not Secure | datahubjs.com/project/owner#/?

接口列表 实时快照

所有项目 / owner / 实时快照

Request

保存为新场景

GET /owner/api/login 18-09-22 12:57:22
200 node_api

request header

```
host: node_api
connection: close
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
referer: http://datahubjs.com/project/owner
accept-encoding: gzip, deflate
accept-language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,zh-TW;q=0.6,zh-HK;q=0.5,ja;q=0.4
cookie: locale=en-us; io=1_RnbYTmm8BrYXCCAAAG
```

request body

Response

response header

response body

```
{
  "success": false
}
```

Data as test fixtures

- Provide **meaningful** data for UI testing
- Stable environment
- Run test cases fastly and efficiently
- DataHub **SDKs**

The screenshot shows a web browser window with the following details:

- Title Bar:** Home - Documentation
- Address Bar:** https://macacajs.github.io/datahub-nodejs-sdk/
- Page Content:** A sidebar on the left lists "Home", "Classes", and several methods: "DataHubSDK" (selected), "switchAllScenes", "switchMultiScenes", and "switchScene". The main content area displays a block of JavaScript code demonstrating how to use the "DataHubSDK" and its methods to switch scenes for different API endpoints.

```
import DataHubSDK from 'datahub-nodejs-sdk';

const sdkClient = new DataHubSDK();

// switch currentScene for 'POST api/create' to 'sucess'
await sdkClient.switchScene({
  hub: 'app',
  pathname: 'api/create',
  scene: 'success',
  method: 'POST', // method is optional, default method is 'ALL'
})

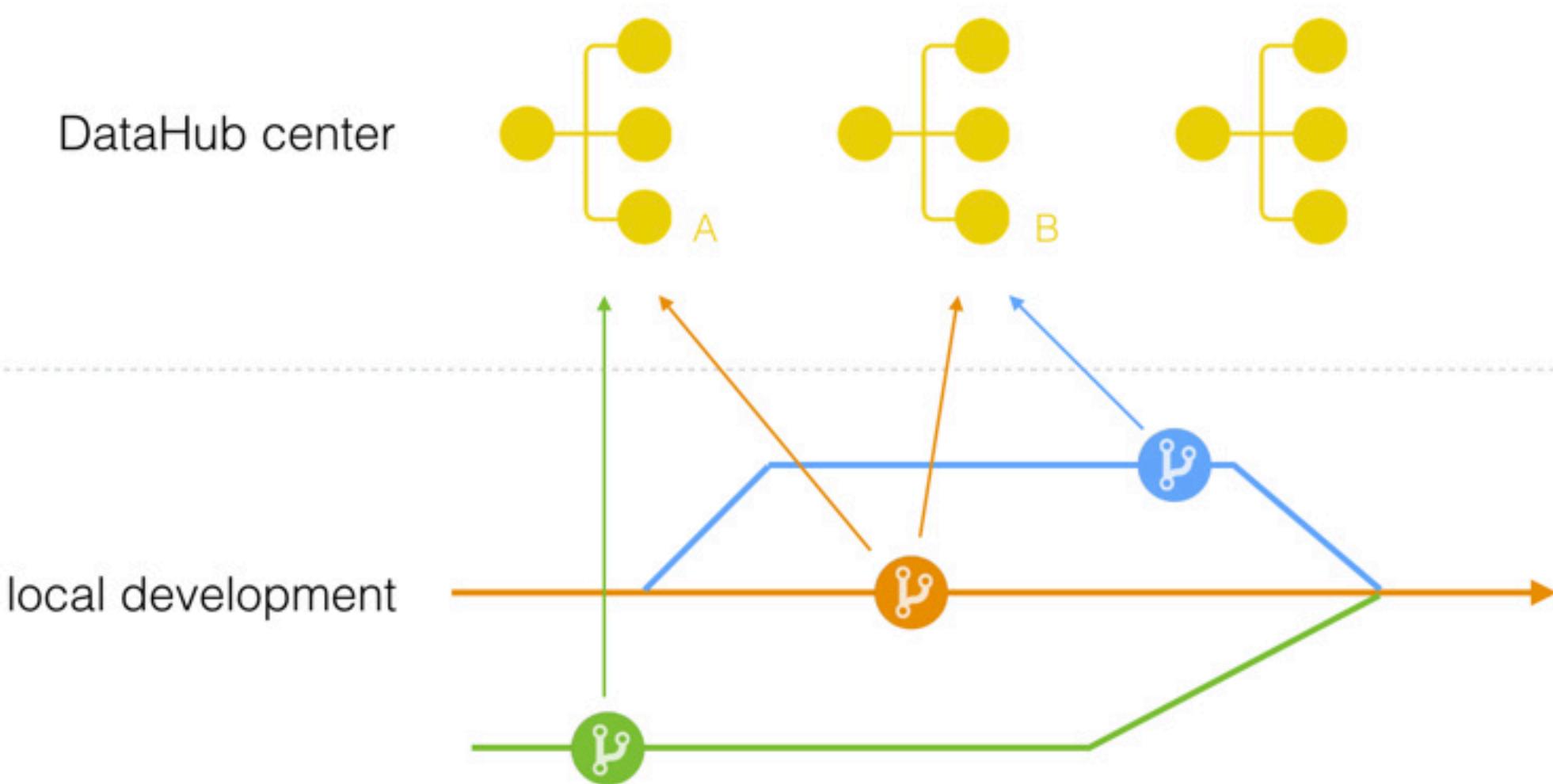
// switch currentScene for 'GET api/read' to 'success'
// switch currentScene for 'DELETE api/delete' to 'success'
await sdkClient.switchMultiScenes([
  {
    hub: 'app',
    pathname: 'api/read',
    scene: 'success',
    method: 'GET', // method is optional, default method is 'ALL'
  },
  {
    hub: 'app',
    pathname: 'api/delete',
    scene: 'success',
    method: 'DELETE', // method is optional, default method is 'ALL'
  }
])

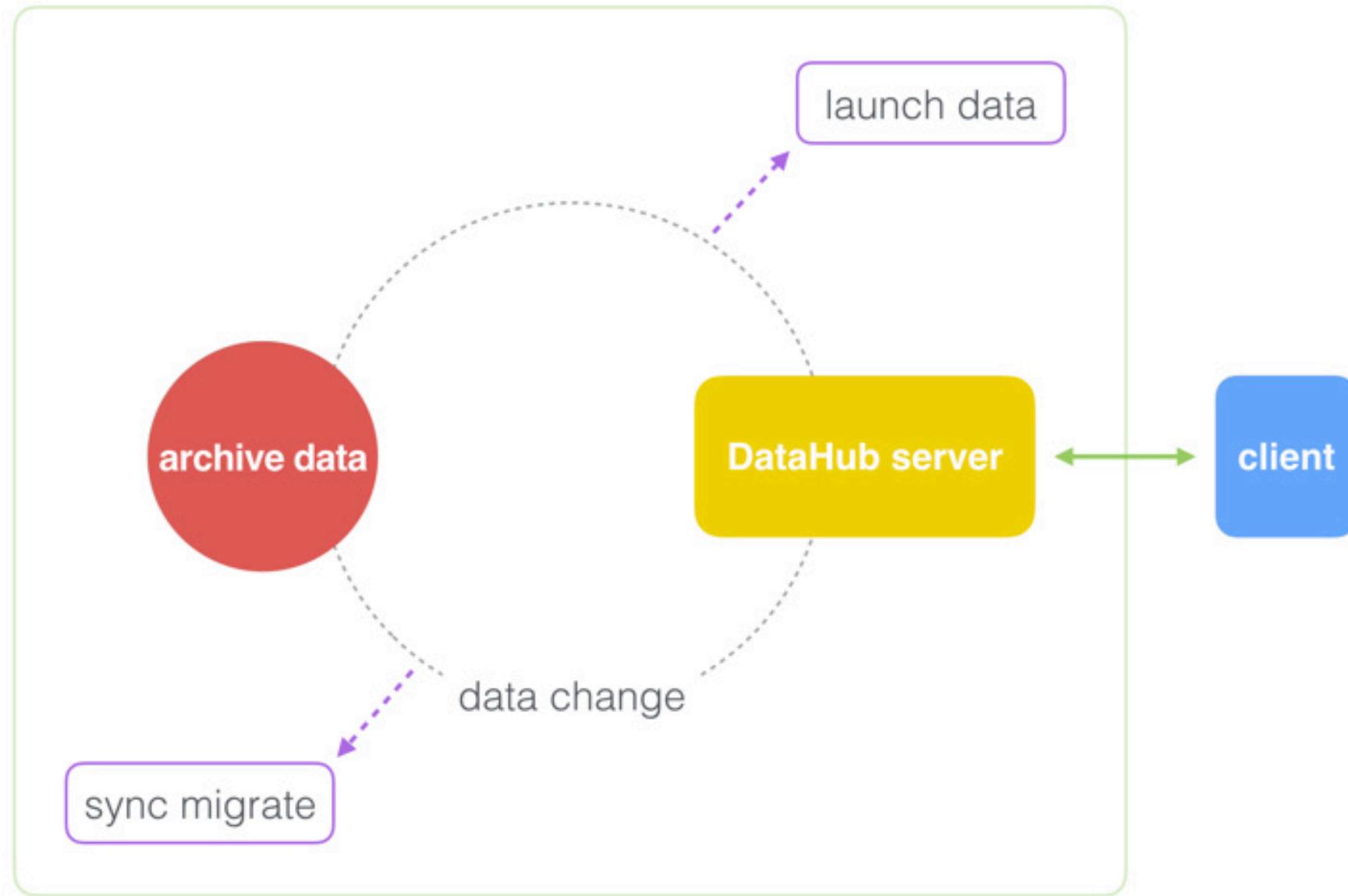
// switch all scenes for all pathnames under app to 'success'
await sdkClient.switchAllScenes({
  hub: 'app',
  scene: 'success',
})
```

Born for Productivity

from Hackthon to Elite

- 一些设计理念
 - 数据可持续
 - 服务去中心化
 - 内部实现单向数据流





CI/CD with Marmot

The screenshot shows a web browser window titled "Marmot Platform - Marmot" with the URL "localhost:9900/?jobName=android-app-bootstrap". The left sidebar has a dark theme with icons for "Build Info" (selected), "Mock Service", "Insight", and "Setting". The main content area shows a breadcrumb navigation: "Home Page / All Builds / android-app-bootstrap". Below this, there are two tabs: "All Builds" and "android-app-bootstrap" (which is selected). A table displays build details:

Job Name	Build Log	Build Number	Platform	Build End Time	Detail Info
android-app-bootstrap	Build Log	② 12	android	2018-07-08 19:42:15	Detail Info

At the bottom right of the content area are navigation buttons: <, 1, and >.

<https://github.com/macacajs/marmot>

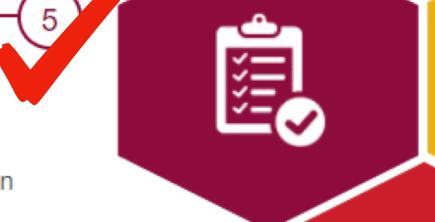
- Platform independent Web iOS Android etc.
- <https://github.com/eggjs/examples#usage>
- <https://zhuanlan.zhihu.com/p/37424566>

02 | 业务实践

WEB APPLICATION DEVELOPMENT LIFECYCLE

Release & Evolution Process

- Change Requests
- Final Delivery
- Deployment on Live Server
- Client Training & Documentation
- Maintenance & Support



Development Process

- Back-end
- Source Code
- Database
- Third Party Integrations
- Deployment on Staging Server
- Unit Tests



1

Requirement Process

- Project Analysis & Consultation
- Requirement Gathering
- Project Budget
- Contract Acceptance

2

Planning Process

- Resource Planning
- Project Timeline Planning
- Functional Specification Documentation

3

Prototyping

- User Experience Design
- HTML
- Front-end
- UI/UX Testing

② API design process

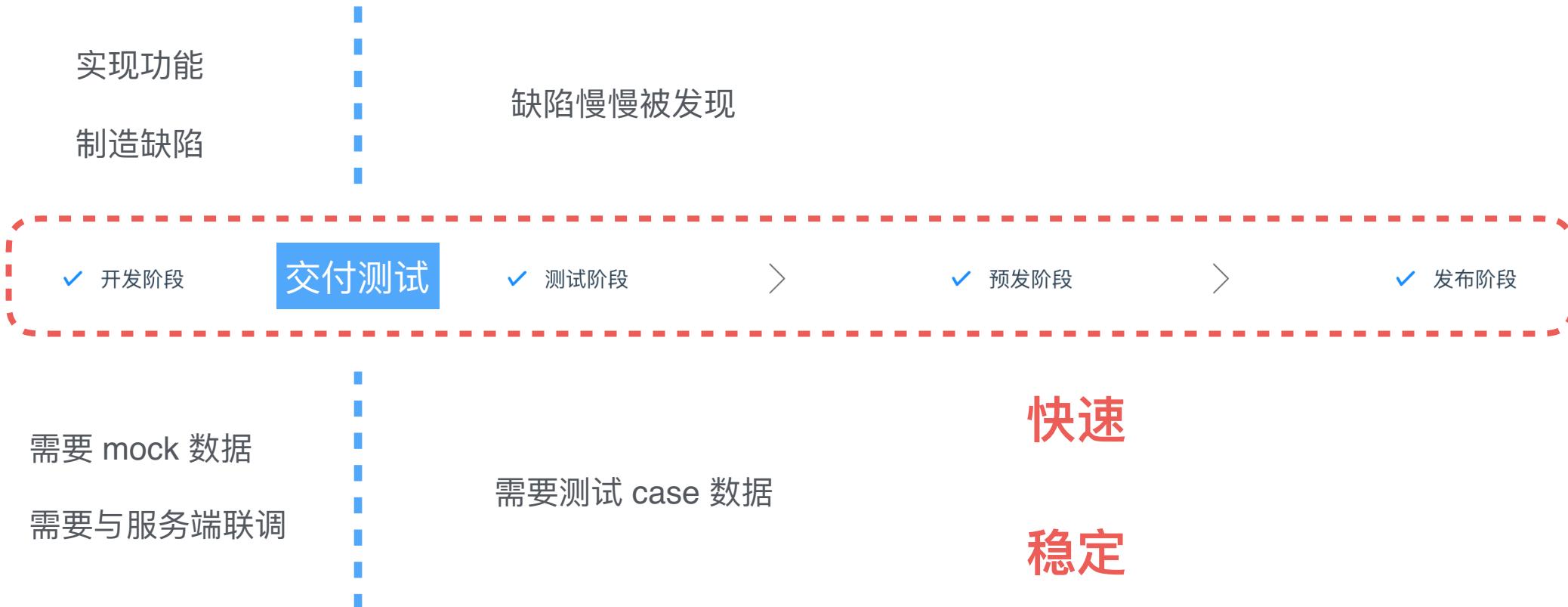
- 接口字段约定，字段统一管理，并行开发
- 还可以开启接口字段校验功能

③ Prototyping

- 快速制作一些原型，不依赖硬编码即可证明可行性

④ Development Process

What we do & what we **really** want



④ Development Process

Mock 数据怎么来?

1. 自己造

Rewrite response

响应延迟 0 秒

修改

场景管理

预览场景数据 (支持 GET|ALL 类型请求)

搜索场景

+ 新增场景

点击场景进行切换:

fail

default

更新场景

* 场景名称

default

```
1 {  
2   "success": true  
3 }
```

④ Development Process

Mock 数据怎么来?

1. 自己造 #hard mode
2. 代理模式抓请求 #easy mode 🤙

代理模式



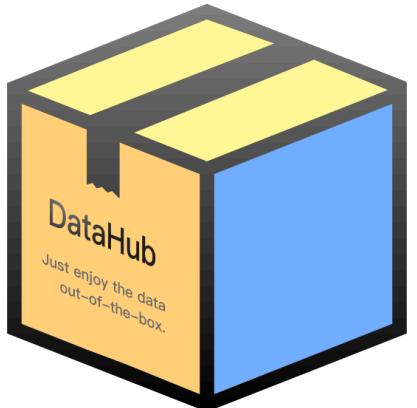
请求字段描述

A screenshot of a browser window displaying a JSON response from datahubjs.com/data/owner/api/. The URL in the address bar is datahubjs.com/data/owner/api/login. The page content shows a JSON object representing a user profile, with fields like login, id, node_id, avatar_url, etc., for two users: "xudafeng" and "zhangyuheng".

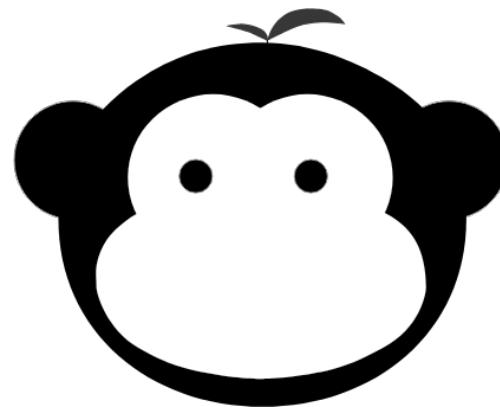
```
[{"login": "xudafeng", "id": 1011681, "node_id": "MDQ6VXNlcjEwMTE2ODE=", "avatar_url": "https://avatars1.githubusercontent.com/u/1011681?v=4", "gravatar_id": "", "url": "https://api.github.com/users/xudafeng", "html_url": "https://github.com/xudafeng", "followers_url": "https://api.github.com/users/xudafeng/followers", "following_url": "https://api.github.com/users/xudafeng/following{/other_user}", "gists_url": "https://api.github.com/users/xudafeng/gists{/gist_id}", "starred_url": "https://api.github.com/users/xudafeng/starred{/owner}{/repo}", "subscriptions_url": "https://api.github.com/users/xudafeng/subscriptions", "organizations_url": "https://api.github.com/users/xudafeng/orgs", "repos_url": "https://api.github.com/users/xudafeng/repos", "events_url": "https://api.github.com/users/xudafeng/events{/privacy}", "received_events_url": "https://api.github.com/users/xudafeng/received_events", "type": "User", "site_admin": false, "contributions": 155}, {"login": "zhangyuheng", "id": 2139038, "node_id": "MDQ6VXNlcjIxMzkwMzg=", "avatar_url": "https://avatars1.githubusercontent.com/u/2139038?v=4", "gravatar_id": "", "url": "https://api.github.com/users/zhangyuheng", "html_url": "https://github.com/zhangyuheng", "followers_url": "https://api.github.com/users/zhangyuheng/followers", "following_url": "https://api.github.com/users/zhangyuheng/following{/other_user}", "gists_url": "https://api.github.com/users/zhangyuheng/gists{/gist_id}", "starred_url": "https://api.github.com/users/zhangyuheng/starred{/owner}{/repo}", "subscriptions_url": "https://api.github.com/users/zhangyuheng/subscriptions", "organizations_url": "https://api.github.com/users/zhangyuheng/orgs", "repos_url": "https://api.github.com/users/zhangyuheng/repos", "events_url": "https://api.github.com/users/zhangyuheng/events{/privacy}", "received_events_url": "https://api.github.com/users/zhangyuheng/received_events", "type": "User", "site_admin": false, "contributions": 155} ]
```

④ Development Process

轻松拿到 mock 数据之后，写测试更方便



DataHub



Macaca

④ Development Process

轻松拿到 mock 数据之后，写测试更方便。

先切换场景再跑用例

```
import DataHubSDK from 'datahub-nodejs-sdk';

const sdkClient = new DataHubSDK();

// switch currentScene for 'POST api/create' to 'success'
await sdkClient.switchScene({
  hub: 'app',
  pathname: 'api/create',
  scene: 'success',
  method: 'POST', // method is optional, default method is 'ALL'
})

// switch currentScene for 'GET api/read' to 'success'
// switch currentScene for 'DELETE api/delete' to 'success'
await sdkClient.switchMultiScenes([
  {
    hub: 'app',
    pathname: 'api/read',
    scene: 'success',
    method: 'GET', // method is optional, default method is 'ALL'
  },
  {
    hub: 'app',
    pathname: 'api/delete',
    scene: 'success',
    method: 'DELETE', // method is optional, default method is 'ALL'
  }
])

// switch all scenes for all pathnames under app to 'success'
await sdkClient.switchAllScenes({
  hub: 'app',
  scene: 'success',
})
```

<https://macacajs.github.io/datahub-nodejs-sdk/>

DataHub

Getting Mock Done.

Open source

- Welcome to join in and feel free to contribute.
- <https://github.com/macacajs/macaca-datahub#contributors>

欢迎加入蚂蚁金服

- <https://github.com/xudafeng/job-model>
- 联系  zhangyuheng91[at]gmail.com

03 | Q & A

谢谢

THANK YOU
