Addison-Wesley Professional Ruby Series

SERVICE-ORIENTED DESIGN WITH RUBY AND RAILS

Foreword by **Obie Fernandez,** *Series Editor*

PAUL DIX

with TROTTER CASHION ▪ BRYAN HELMKAMP ▪ JAKE HOWERTON

Service-Oriented Design with Ruby and Rails

Service-Oriented Design with Ruby and Rails

Table of Contents

Contents

Foreword

Preface

Acknowledgments

About the Author

1 Implementing and Consuming Your First Service

- Whats a Service?
- Service Requirements
- The Ruby Tool Set
  - Sinatra
  - ActiveRecord
  - JSON
  - Typhoeus
  - Rspec
- The User Service Implementation
  - Using GET
  - POSTing a User
  - PUTing a User
  - Deleting a User
  - Verifying a User

Pearson

Table of Contents

Implementing the Client Library

Finding a User

Creating a User

Updating a User

Destroying a User

Verifying a User

Putting It All Together

Conclusion

2 An Introduction to Service-Oriented Design

Use of Service-Oriented Design in the Wild

Service-Oriented Design Versus Service-Oriented Architecture Versus RESTful-Oriented Architecture

Making the Case for Service-Oriented Design

Isolation

Robustness

Scalability

Agility

Interoperability

Reuse

Conclusion

3 Case Study: Social Feed Reader

A Typical Rails Application

The Rails Social Feed Reader Application

Features

Current Setup

Pearson

Table of Contents

Converting to Services

Segmenting into Services

Breaking Up the Application into Services

Conclusion

4 Service and API Design

Partitioning Functionality into Separate Services

Partitioning on Iteration Speed

Partitioning on Logical Function

Partitioning on Read/Write Frequencies

Partitioning on Join Frequency

Versioning Services

Including a Version in URIs

Using Accept Headers for Versioning

URIs and Interface Design

Successful Responses

HTTP Status Codes

HTTP Caching

Successful Response Bodies

Error Responses

HTTP Status Codes

Error Response Bodies

Handling Joins

Storing References

Joining at the Highest Level

Beware of Call Depth

API Complexity

Pearson

Table of Contents

Atomic APIs

Multi-Gets

Multiple Models

Conclusion

5 Implementing Services

The Vote Service

A Multi-Get Interface

The Vote Interface

API Design Guidelines

Models

Rails

Rails 2.3 Routes

Rails 3 Routes

The Rails Controller

Sinatra

Rack

Conclusion

6 Connecting to Services

Blocking I/O, Threading, and Parallelism

Asynchronous I/O

Multi-threading

Typhoeus

Making Single Requests

Making Simultaneous Requests

Multi-threaded Requests

Table of Contents

JRuby

Logging for Performance

Handling Error Conditions

Testing and Mocking Service Calls

Requests in Development Environments

Conclusion

7 Developing Service Client Libraries

Packaging

Jeweler

Building and Deploying a Library

Parsing Logic

The JSON Gem

YAJL Ruby

Wrapping Parsed Results

ActiveModel

Validations

Serialization

Connection and Request Logic

Data Reads

Data Writes

Mocks, Stubs, and Tests

Conclusion

8 Load Balancing and Caching

Latency and Throughput

Load Balancing

Pearson

Table of Contents

Load Balancing Algorithms

Implementing Load Balancing

Caching with Memcached

The Memcached Client and ActiveRecord

Time-Based Expiration

Manual Expiration

Generational Cache Keys

HTTP Caching

Expiration-Based Caching

Validation-Based Caching

Implementing HTTP Caching

Conclusion

9 Parsing XML for Legacy Services

XML

REXML

Nokogiri

SOAP

Exploring Web Services with a WSDL File

Making Requests

Conclusion

10 Security

Authentication

HTTP Authentication

Signing Requests

SSL for Authentication

Authentication for Rails Applications

Pearson

Table of Contents

Authorization

Firewalls

An RBAC Authorization Service

Encryption

SSL for Encryption

Public/Private Key Pairs for Encryption

Conclusion

11 Messaging

What Is Messaging?

Synchronous Versus Asynchronous Messaging

Queues

Message Formats

RabbitMQ and AMQP

Queues in RabbitMQ

Exchanges and Bindings

Durability and Persistence

Client Libraries

Synchronous Reads, Asynchronous Writes

HTTP-Based Reads

Messaging-Based Writes

The CAP Theorem

Eventual Consistency

Designing Around Consistency

Data Is the API

Operations on Fields

Modifications to Field Operations

Pearson

Table of Contents

Conclusion

12 Web Hooks and External Services

Web Hooks

PubSubHubbub

Receiving Web Hooks

Providing Web Hooks

Strategies for Dealing with Failure

OAuth

Implementing an OAuth Consumer

Implementing an OAuth Provider

Integrating with External Services

Consuming Data

Pushing Data

The Request Lifecycle

Worker Processes

Ensuring Performance and Reliability

Segregating Queues

Metrics

Throttling and Quotas

Conclusion

Appendix RESTful Primer

Roy Fieldings REST

Constraints

Architectural Elements

Architectural Views

REST and Resources

Pearson

Table of Contents

- URIs and Addressability
- Representations

HTTP and the Uniform Interface

- HTTP Methods
- HTTP Headers
- HTTP Status Codes

Conclusion

Index