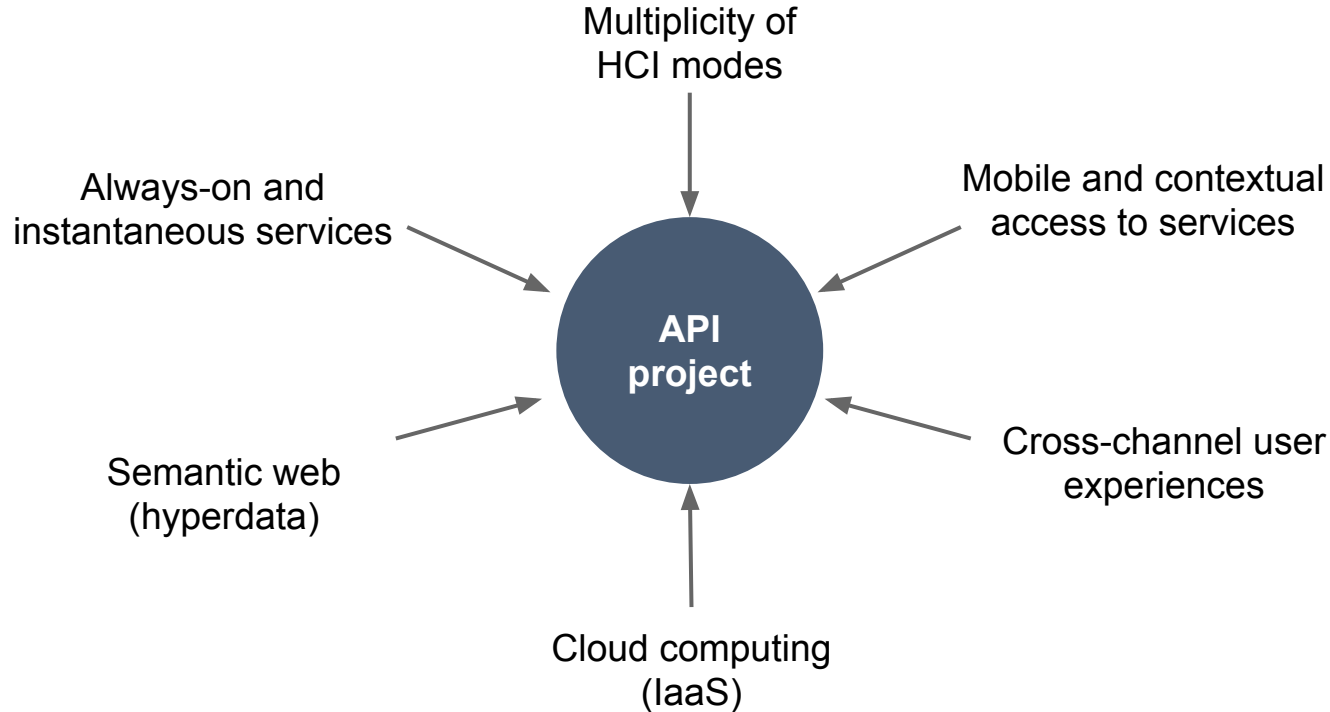QCon SF 2014

Create and Deploy APIs using Web IDEs, Open Source Frameworks and Cloud Platforms

Presented by Jerome Louvel, Chief Geek
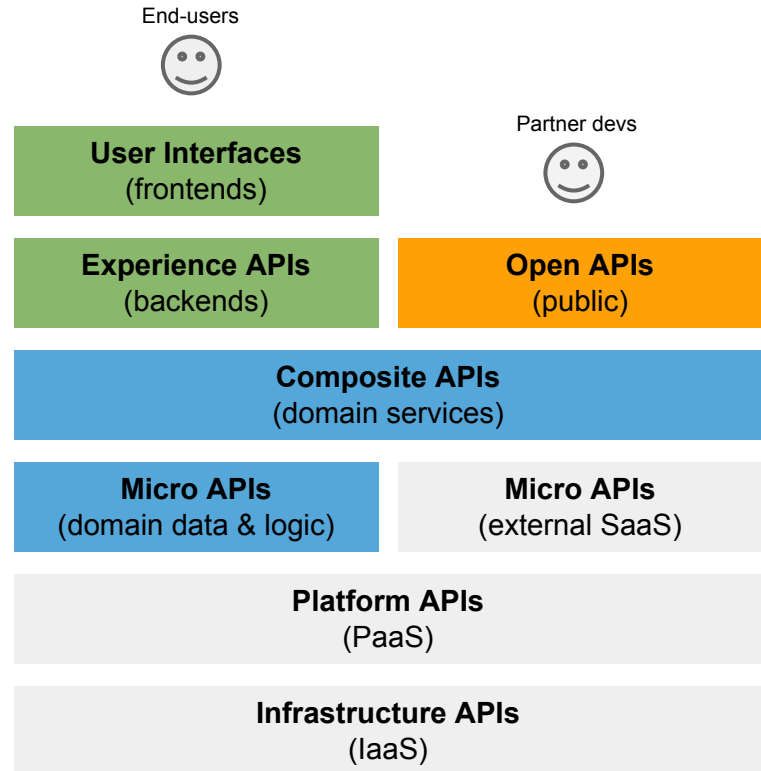
Restlet

New API Landscape

Impacts on API development

- **New types of APIs**
  - internal & external APIs
  - composite & micro APIs
  - experience & open APIs

- **Number of APIs increases**
  - channels growth
  - history of versions
  - micro services pattern
  - quality of service

- **→ Industrialization needed**
  - new development workflows

End-users

Partner devs

User Interfaces (frontends)	
Experience APIs (backends)	Open APIs (public)
Composite APIs (domain services)	
Micro APIs (domain data & logic)	Micro APIs (external SaaS)
Platform APIs (PaaS)	
Infrastructure APIs (IaaS)	

New API development workflows

- **API-driven approach benefits**
  - a pivot API descriptor
  - server skeletons & mock generation
  - up-to-date client SDKs & docs
  - rapid API crafting & implementation

- **Code-first or API-first approaches**
  - can be combined using
    - code introspectors to extract
    - code generators to resync

API source code

introspect   generate

API descriptor

generate   generate

API docs   Client SDKs
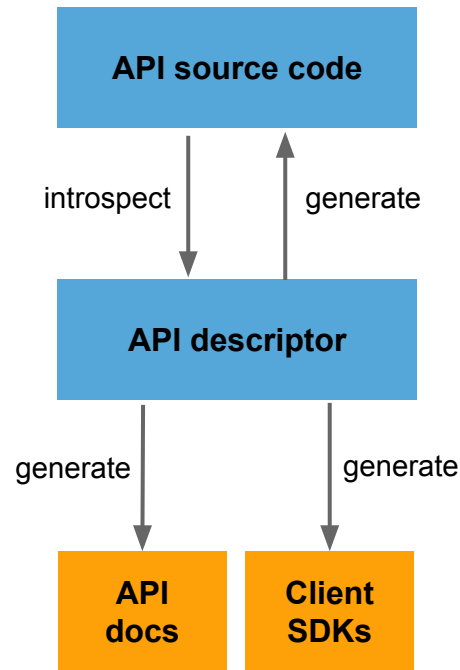
Crafting an API

- **Specialized API crafting tools**
  - code editors
  - visual designers
  - generation of
    - contract
    - client SDKs
    - skeletons

- **New generation of tools**
  - IDE-type
  - Web-based

swagger editor

```
1   apiVersion: 1.0.0
2   swaggerVersion: 1.2
3   basePath: http://petstore.swagger.wordnik.com/api
4   authorizations:
5     oauth2:
6       type: 'oauth2'
7       scopes:
8         scope: 'write'
9         description: 'write to your albums'
10      grantTypes:
11        implicit:
12          loginEndpoint:
13            url: 'http://petstore.swagger.wordnik.com/oauth/dialog'
14            tokenName: 'access_token'
15  info:
16    title: Petstore example API
17    description: This is a sample server Petstore server.  You can find out more about Swagger \n   at <a
      href=\http://swagger.wordnik.com>http://swagger.wordnik.com</a> or on irc.freenode.net, #swagger.  For
      this sample,\n   you can use the api key \special-key\ to test the authorization filters
18    termsOfServiceUrl: http://helloreverb.com/terms/
19    contact: apiteam@helloreverb.com
20    license: Creative Commons 4.0 International
21    licenseUrl: http://creativecommons.org/licenses/by/4.0/
22
23  apiDeclarations:
24    -
25      apiVersion: 1
26      swaggerVersion: 1.2
27      basePath: "http://localhost:8002/api"
28      description: "Operations for Albums"
29      resourcePath: "/albums"
30      produces:
31        - "application/json"
32        - "application/xml"
33      consumes:
34        - "application/json"
35        - "application/xml"
36      apis:
37        -
38          path: "/albums"
39          operations:
40            -
41              method: "GET"
42              summary: "Retrieve albums of the user."
43              notes: "Returns a list of all albums for the user."
44              type: "array"
45              items:
46                $ref: "Album"
47              nickname: "getAlbums"
48              parameters:
49                -
```

albums : Operations for Albums

Hide | List Operations

GET /albums — Retrieve albums of the user.

Implementation Notes
Returns a list of all albums for the user.

Response Class
array

Response Content Type [application/json ▾]

Parameters

Parameter	Value	Description	Parameter Type	Data Type
quality	(required)	The quality of the album	query	string
limit		The maximum number of records to return. Depending on the data, less values may be returned	query	integer
skip		An offset for the first record to retrieve. If there are less records than the skip, an empty result will be returned	query	integer

POST /albums — Create a new album.

Implementation Notes
Create a new album for the logged in user and returns its ID.

Response Class
ID

Response Content Type [application/json ▾]

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body		The new album to be created.	body	InputAlbum

API Blueprint Syntax Tutorial

✔ Valid API Blueprint   A   Hide Preview   Save API Blueprint

```
1    FORMAT: 1A
2    HOST: http://www.google.com
3
4    # FasterLap
5    Notes API is a *short texts saving* service similar to its physical paper presence o
6
7    # Group Notes
8    Notes related resources of the **Notes API**
9
10   ## Notes Collection [/notes]
11   ### List all Notes [GET]
12   + Response 200 (application/json)
13
14           [{
15             "id": 1, "title": "Jogging in park"
16           }, {
17             "id": 2, "title": "Pick-up posters from post-office"
18           }]
19
20   ### Create a Note [POST]
21   + Request (application/json)
22
23           { "title": "Buy cheese and bread for breakfast." }
24
25   + Response 201 (application/json)
26
27           { "id": 3, "title": "Buy cheese and bread for breakfast." }
28
29   ## Note [/notes/{id}]
30   A single Note object with all its details
31
32   + Parameters
33       + id (required, number, `1`) ... Numeric `id` of the Note to perform action with
34
35   ### Retrieve a Note [GET]
36   + Response 200 (application/json)
37
38
```

FasterLap

Notes API is a *short texts saving* service similar to its physical paper presence on your table.

Notes

Notes related resources of the **Notes API**

Notes Collection

List all Notes

GET   /notes

Create a Note

POST   /notes

Note

A single Note object with all its details

Retrieve a Note

GET   /notes/{id}

Parameters

Name	Description	Details

This can be done via the side menu, Endpoints > New Endpoint. For our example, the details could be as follows:

Endpoint Settings ?

Name	Weather at Location
Route	/WeatherAtLocation
Description	Returns the weather for a particular longitude and latittude
Method	GET	**Group**	e.g., Location Group

Then we add the query parameters as follows:

Parameters ➕ ?

Name	Description	Default Value	Type	Format
Longitude	Description	Default Value	Precisi ▼	Query ▼

Is Array ☐   Is Optional ☐   Is Constant ☐   ✖ Delete

Name	Description	Default Value	Type	Format
Latitude	Description	Default Value	Precisi ▼	Query ▼

Is Array ☐   Is Optional ☐   Is Constant ☐   ✖ Delete

Crafting an API with Restlet

- **Browser-based IDE for API crafting**
  - accelerate the design of APIs
  - Chrome extension
  - compatible with other modern browsers

- **Supports**
  - visual design of APIs
  - source code views
  - APIs with large # of resources (sections)
  - skeleton and SDK generation
  - adherence to REST
  - multiple API languages (Swagger & RAML initially)

RESTLET
Studio

Preview Launch
November 18th

API definition ▾   Server skeleton ▾   Client SDK ▾   Help ▾

🏳 General information

Endpoints   [Add]

https://mycontactsapi.apispark.net/yeah

Sections   [Add]

All

Default

Section 1

Resources   [Add]

▸ /contacts/

▾ /contacts/{contactid}

GET	Loads a Contact
PUT	Stores a Contact
DELETE	Deletes a Contact

Representations   [Add]

▸ Contact

▸ ContactList

Resource

Relative path	/contacts/{contactid}
Name	Contact
Description	
Sections	Section 2 ▾

Path variables

▾ contactid

Name	contactid
Type	String ▾
Description	Identifier of the Contact

▸ test

⊕ **Add a path variable**

Web API   Swagger   RAML

Rest**let**

API definition ▾   Server skeleton ▾   Client SDK ▾   Help ▾

🏁 General information

Endpoints   [Add]

https://mycontactsapi.apispark.net/yeah

Sections   [Add]

All

Default

Section 1

Resources   [Add]

▸ /contacts/

▾ /contacts/{contactid}

GET	Loads a Contact
PUT	Stores a Contact
DELETE	Deletes a Contact

Representations   [Add]

▸ Contact

▸ ContactList

```
1    {
2        "swagger": "2.0",
3        "info": {
4            "description": "A tutorial for Restlet Framework",
5            "version": "1.0.0",
6            "title": "WebAPITutorial",
7            "contact": {
8                "name": "Cyprien QUILICI",
9                "url": "https://apispark.com",
10               "email": "cquilici@restlet.com"
11           },
12           "license": {
13               "name": "Apache 2.0",
14               "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
15           }
16       },
17       "host": "mycontactsapi.apispark.net:443",
18       "basePath": "yeah",
19       "schemes": [
20           "https"
21       ],
22       "paths": {
23           "/contacts/": {
24               "get": {
25                   "tags": [
26                       "Section 1"
27                   ],
28                   "summary": "Loads a list of Contact",
29                   "description": "Loads a list of Contact",
30                   "consumes": [],
31                   "produces": [
32                       "application/json",
33                       "application/msword",
34                       "text/css"
35                   ],
36                   "parameters": [
37                       {
38                           "in": "query",
39                           "name": "strategy",
40                           "description": "Configures the way to retrieve content in case of complex representations. Possible values are: 'load' to
     all content and 'reference' to only load root content and references to other elements.",
41                           "required": false,
42                           "type": "string"
43                       }
44                   ],
45                   "responses": {
46                       "200": {
47                           "description": "good",
48                           "schema": {
```

Web API   **Swagger**   RAML

Restlet

🏳 General information

Endpoints   Add

https://mycontactsapi.apispark.net/yeah

Sections   Add

All

Default

Section 1

Resources   Add

▸ /contacts/

▾ /contacts/{contactid}

GET   Loads a Contact

PUT   Stores a Contact

DELETE   Deletes a Contact

Representations   Add

▸ Contact

▸ ContactList

```
1   #%RAML 0.8
2   title: "WebAPITutorial"
3   version: "1.0.0"
4   baseUri: "https://mycontactsapi.apispark.net:443yeah"
5   schemas:
6     -
7       "ContactList": '{"type":"object","title":"ContactList"}'
8       "Contact": '{"type":"object","title":"Contact","properties":{"lastName":{"type":"string","required":true,"title":"lastName"},"_id":{"type"
    :"string","required":true,"title":"_id"},"age":{"type":"integer","required":true,"title":"age"},"firstName":{"type":"string","required":true,"title"
    :"firstName"}}}'
9   securitySchemes:
10    -
11  "/contacts/":
12    displayName: "ContactList"
13    get:
14      description: "Loads a list of Contact"
15      queryParameters:
16        "strategy":
17          displayName: "strategy"
18          description: "Configures the way to retrieve content in case of complex representations. Possible values are: 'load' to all content
    and 'reference' to only load root content and references to other elements."
19          type: string
20          required: false
21          repeat: false
22      responses:
23        "200":
24          description: "good"
25          body:
26            "text/css":
27              schema: "ContactList"
28            "application/msword":
29              schema: "ContactList"
30            "application/json":
31              schema: "ContactList"
32    post:
33      description: "Adds a Contact"
34      body:
35        "application/xml":
36          schema: "Contact"
37        "application/x-yaml":
38          schema: "Contact"
39        "text/x-yaml":
40          schema: "Contact"
41        "text/xml":
42          schema: "Contact"
43        "application/json":
44          schema: "Contact"
45  "/contacts/{contactid}":
46    displayName: "Contact"
```

Web API   Swagger   **RAML**

Restlet

Implementing an API

- **Implementation using a RESTful API framework**
  - Benefits
    - less mental gap compared to MVC & RPC frameworks
    - higher level than raw HTTP libraries (Servlet, Netty)
    - easier to use HTTP protocol features
    - open source (no lock-in)

  - Key features of an API framework
    - API descriptor/contract definition
      - in Java, typically uses annotations
    - both client and server side support
    - make it easy or transparent to use HTTP features
    - performance and scalability

Custom web API
REST framework
HTTP semantics
HTTP / WebSocket transport

API Frameworks Landscape

- **JAX-RS centric**
  - Oracle Jersey
  - JBoss RESTeasy
  - DropWizard
  - Apache Wink
  - Apache CXF

- **Alternative Java APIs**
  - Restlet Framework
    - favor Restlet API
    - support JAX-RS API
  - REST.li
  - RESTx
  - REST Express

- **JavaScript**
  - Express.js

Implementing an API with Restlet

- **All HTTP features supported**
    - caching, content ranges & compression
    - content negotiation & conditional requests
    - confidentiality & authentication
    - all HTTP headers mapped to Java classes
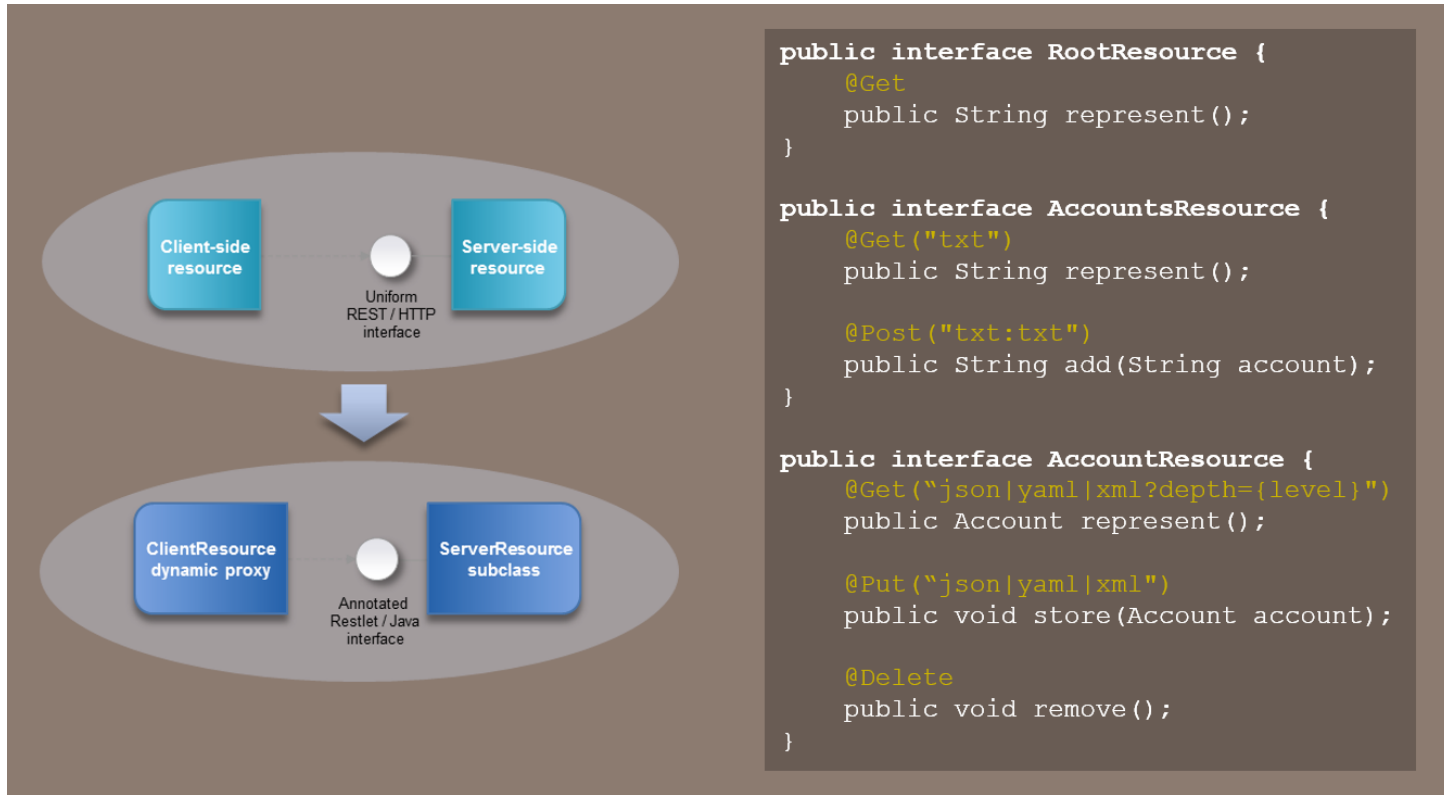
RESTLET
FRAMEWORK

- **What's New in v2.3?**
    - API introspector (Swagger, RAML, WADL)
    - API management (API firewall, APISpark integration)
    - @Status annotation (map Java exception & HTTP error)
    - CORS support (cross-origin access)
    - easier access to raw HTTP headers
    - upgrade to Jetty 9.2 with client-side support
    - compatible with Java 7, GWT 2.7, Android, GAE, Java EE, OSGi

Version 2.3.0 launch
November 18th

Restlet-annotated Java interfaces


```java
public interface RootResource {
    @Get
    public String represent();
}

public interface AccountsResource {
    @Get("txt")
    public String represent();

    @Post("txt:txt")
    public String add(String account);
}

public interface AccountResource {
    @Get("json|yaml|xml?depth={level}")
    public Account represent();

    @Put("json|yaml|xml")
    public void store(Account account);

    @Delete
    public void remove();
}
```

JAX-RS and Restlet APIs | Server-Side

JAX-RS API v2

```
@POST
@Path("withdrawal")
@Consumes("text/plain")
@Produces("application/json")
public Money withdraw(
@PathParam("card") String card,
@QueryParam("pin") String pin,
String amount){
    return getMoney(card,
pin, amount);
}
```

Restlet API v2

```
@Post("txt:json")
public Money withdraw(){
    String card = getPathValue("card");
    String pin = getQueryValue("pin");
    String amount = getQueryValue("amount");
    return getMoney(card, pin, amount);
}
```

JAX-RS and Restlet APIs | Client-Side

JAX-RS API v2

```
// Get instance of Client
Client client =
ClientFactory.newClient();

// Get account balance
String bal =
client.target("http://.../atm/{cardI
d}/balance")
.pathParam("cardId", "112233")
.queryParam("pin", "9876")
.request("text/plain").get(String.cl
ass);
```

Restlet API v2

```
// Get instance of Client
ClientResource client = new
ClientResource(""http://.../atm/{cardId}
/balance");

// Get account balance
client.setPathValue("cardId", "112233");
client.setQueryValue("pin", "9876");
client.accept(MediaType.TXT_PLAIN);
String bal = client.get(String.class);
```

Deploying an API

- **PaaS make it easier to deploy**
  - less operational burden
  - cost and time effective to start then scale
  - more availability and lower latency
    - necessary for a global reach

- **Generic Infrastructure PaaS**
  - hosting and auto-scaling
  - good density thanks to VMs
  - better density thanks to containers (Docker)

- **Specialized PaaS for APIs**
  - add API management on top of the hosted API
  - integrate with Infrastructure PaaS
    - remote agent close to API or API wrapper

Managing an API

- **Typical Benefits**
  - documentation
  - access control
  - firewall
  - adaptation
  - monetization
  - analytics

- **APISpark Supports**
  - APIs with large # of resources (sections)
  - skeleton and SDK generation
  - integration with Restlet Framework

apigee

3SCALE

mashape

RESTLET APISpark

General Availability
November 18th

APISpark

Your cells

Web API Entity Store File Store

Filter by name

Value

Type All ▾

State All ▾

Role All ▾

☑ All versions in one cell

Transaction ES ⚙
Contract
12/02/2014 Version created

Entity Store

Version 1 Draft

Contacts DB Personal ⚙
Contract
12/02/2014 Version created
14/02/2014 Successfully deployed

Web API

Version 1 Draft

Analysis platform Alpha ⚙
Descriptor
12/02/2014 Successfully deployed
14/02/2014 Successfully deployed
16/02/2014 Successfully published

File Store

Version 1 Published

Transaction ES ⚙
Descriptor
12/02/2014 Version created

Entity Store

Version 1 Draft

Contacts DB Personal ⚙
Descriptor
12/02/2014 Version created
14/02/2014 Successfully deployed

Web API

Version 1 Draft

Contacts DB Personal ⚙
Template

Web API

Version 1 Draft

Transaction ES ⚙
Template

Entity Store

Version 1 Draft

Analysis platform Alpha ⚙
Contract
12/02/2014 Successfully deployed
14/02/2014 Successfully deployed
16/02/2014 Successfully published

File Store

Version 1 Published

APISpark

Contacts DB Personal

Entity Store - Instance of Contacts DB Template

Version 1 Draft

Deploy

Overview | Settings | Downloads | Backups | Members | Messages

General informations

Endpoints   Add

https://contactsdbpersonal.apispark.net/v1

Resources   Search   Add

/contacts/

| GET | Loads a list of Contact |
| POST | Adds a list of Contact |

/contacts/ {contactid}

GET	Loads a Contact
PUT	Stores a Contact
DELETE	Deletes a Contact

/css/
/css/{file}
/img/
/img/{file}
/html/
/html/{file}
/js/
/js/{file}

Representations   Add

FolderListRepresentation

| JSON | Loads a list of Contacteuh ... |
| YAML | Adds a list of Contact |

Overview | Actions

Name   GET

Description   Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vestibulum nulla lorem, ultrices non laoreet vel, tempus
in dolor. Curabitur ante risus, pharetra non magna nec,
dictum aliquam orci. Maecenas pulvinar

QUERY PARAMETERS

▸ Strategy

▾ Another QueryParameter

Description   My third QueryParameter description lorem ipsum
dolor sit amet.

⊕ Add a query parameter

RESPONSES

▸ success

⊕ Add a response

Conclusion

- **APIs enable new abstracting layers**

  - IaaS became a game changer thanks to APIs
    - S3, EC2, Route53, etc.

  - PaaS is the next game changer
    - thanks to APIs as well!

Thank you

Restlet