About endpoint proposal

This is a proposal to change the way FT's web applications are documented for support purposes.

Abstract

Web services are required to expose `/__about` endpoints to describe themselves. The response given at this URL must be a JSON document conforming to the following format. The data published at this address should also be copied to an independent system to enable it to be referenced if the service (and therefore the runbook endpoint) is down.

Format

Imp.	Property	Type	Description	Why do we need this
	`{`			
🟥	`schemaVersion`	integer	Set to 1	Allows client applications to consume runbook info, eg the Chrome extension will not nicely format runbooks unless they declare a schemaVersion
🟥	`name`	string	The name of the service. Should not be unnecessarily different from: ● the name of the code repository ● the public subdomain ● the system code	To find the correct runbook for a particular service
🟥	`purpose`	string	A description of the service in terms of how it will be used/seen by the end user.	So that we know what it's for
🟧	`audience`	string	One of: ● 'public': Service is available to and offers value to anyone ● 'public-selected': Service is available publicly but is only useful to a selected group (eg. an admin tool for advertising clients) ● 'internal': Service is available only to employees	So we can all understand who will be impacted and how, if there was a failure or outage'
🟥	`systemCode`	string	The system code of the service as defined in the CMDB. Required if the service has a CMDB entry, which should be created as soon as practical.	To uniquely identify a service and to allow runbooks to be automatically associated with data in other tools (eg monitoring)

🟧	`primaryUrl`	string	URL that gains access to the service	To be able to quickly check that the site is up and allow support staff to know where to find it.
🟥	`serviceTier`	string	A service tier as defined by the [Product Tier Selection guide]()	So that we know what level of availability to consider normal, and how to prioritise problems
🟧	`appVersion`	string	The version number of the revision of the code that is checked out into this running instance. Could be the version number associated with the build in forge, or the output of `git describe --tags`.	Generally useful information to provide to engineering teams when reporting bugs
▨	`apiVersion`	integer	Version number of the current stable API supported by this service, if the service exposes an API	Allows other services that act as clients of this one to verify whether their integration is using the current API
▨	`apiVersions: [{`	array	Array of API versions that are or have ever been supported by this service	Ensures that users are able to be aware of upcoming versions of the API, and when versions will be terminated well in advance of the termination date.
▨	`path`	string	Path component of URL to base endpoint for this version	
▨	`supportStatus`	string	Origami support status of this API version. Must match the options defined in the [origami.json spec]().	
▨	`dateTerminated`	string	ISO8601-format date when this version stopped being supported (or will stop being supported, for future dates)	
	`}, ...]`			
🟧	`_hostname`	string	The local machine hostname of the machine from which the runbook is being served. This is expected to change on refresh if multiple origin backends are behind a load balancer, and that should be considered normal.	Helps developers verify or debug deployments and load balancing, and verify that information is coming from the expected origin node.
▨	`dateCreated`	string	ISO8601-format date when the app was first created in its current form	Understanding how old the system is gives insight into its likely technical architecture, and how maintainable it is.
🟧	`dateDeployed`	string	ISO8601-format date when the app was last deployed	Helps to identify or eliminate recent deploys as the cause of problems with the service.
🟥	`contacts: [{`	array	Array of contacts for the application, who may be consulted for expert knowledge about it. At least one up to date contact should be provided.	So that we have a defined person/group who takes responsibility / accountability for services
🟥	`name`	string	Name of person or team (with a preference for a team rather than an individual)	
🟥	`email`	string	Email address	
🟧	`extension`	string	Phone extension on internal phone network	

▨	`slack`	string	Slack handle	
🟥	`rel`	string	Relationship to the service. One of: ● 'owner': The person or team which has ultimate responsibility for the service. Prefer owner over other more descriptive categories if both apply. ● 'product': A person or team that has expertise in the non-technical aspects of the service, eg customer relations, business and data analytics ● 'technical': A person or team with expertise in the technical implementation of the service.	
🟥	`domain`	string	Description of expertise, resources/help available, or circumstances in which this contact should be used.	
	`}, ...]`			
🟥	`links: [{`	array	Array of links to resources created to support or describe the application. All resources that exist should be listed here.	So that people can find all the monitoring and other important documents that support the service
🟥	`url`	string	URL of the resource	
🟥	`category`	string	One of: ● 'monitoring': A resource that exposes reliability information about the service, eg logging, pingdom or a healthcheck ● 'testing': A resource where tests can be run on the service, eg CI like Jenkins or Travis, or a test tool like Mocha built into the service. ● 'deployment': A resource that exposes information and/or tools relating to deploying the service ● 'documentation': A resource that documents the service, eg a architecture diagram, troubleshooting guide or the readme ● 'issues': The URL of an issue tracker where issues can be filed for this service. ● 'repo': The URL of the service's code repository ● 'dependency': The URL of another service on which this service depends	
🟧	`description`	string	Description of the purpose of the resource.	
	`}, ...]`			
	`support`	string	Deprecated (Instead see the 'issues' category within `links`).	

	}			

Key to importance field:

- 🟥 **Red**: Without this data the service may not be supportable, may accumulate costs without oversight, or might be liable to unintentional deletion. If this data applies to your service, include it as soon as possible.
- 🟧 **Orange**: This data is extremely useful, and might be crucial in some scenarios. Consider providing it if it's known and applicable.
- ⬜ **Grey**: This type of data can be useful, but there's no need to provide it if it is not at hand.

Additionally, any property not defined in the list above can be added to the root object. When adding custom properties, developers should observe some general naming conventions:

- Use camelCase
- For dates, use ISO8601
- If the information is dynamic (ie it may change at runtime without a deployment, prefix the key name with an underscore, eg `_hostname`, `_lastScanTime`)

Example

```
{
    "name": "polyfill-service",
    "purpose": "Stores a library of FT-approved polyfills and serves them to FT websites that need them in older browsers.",
    "audience": "public",
    "primaryUrl": "https://cdn.polyfill.io",
    "serviceTier": "silver",
    "appVersion": "2.1.0",
    "apiVersion": 2,
    "apiVersions": [
        { "path": "/v1", "supportStatus": "deprecated", "dateTerminated": "2016-01-01T00:00:00Z" },
        { "path": "/v2", "supportStatus": "active" }
    ],
    "hostname": "16ac42bb9513da",
    "dateCreated": "2014-07-14T10:28:45Z",
    "dateDeployed": "2015-11-06T14:32:44Z",
    "contacts": [
        { "name": "Origami team", "email": "origami-support@ft.com", "rel": "owner", "domain": "All support enquiries" }
    ],
    "links": [
        {"url": "https://github.com/Financial-Times/polyfill-service/issues", "category": "issues"},
        {"url": "https://github.com/Financial-Times/polyfill-service", "category": "repo"},
        {"url": "https://dashboard.heroku.com/apps/ft-polyfill-service", "category": "deployment", "description": "Production Heroku app"},
        {"url": "https://dashboard.heroku.com/apps/ft-polyfill-service-qa", "category": "deployment", "description": "QA Heroku app"},
        {"url": "http://grafana.ft.com/dashboard/db/origami-polyfill-service", "category": "monitoring", "description": "Grafana dashboard"},
        {"url": "https://app.fastly.com/#stats/service/4E1GeTez3EFH3cnwfyMAog", "category": "deployment", "description": "Fastly CDN app"},
        {"url": "https://my.pingdom.com/reports/uptime#check=1338405", "category": "monitoring", "description": "Pingdom check"},
        {"url": "https://docs.google.com/drawings/d/1eA_sYaSRkvOqIxdkN6LRpyHeOzv8Mxr51WMfXM1sS3Q/edit", "category": "documentation", "description": "Architecture diagram"},
        {"url": "https://github.com/Financial-Times/polyfill-service/blob/master/README.md", "category": "documentation", "description": "README"}
    ]
}
```