REST Best Practices

Claxton Correya
Dev@CBA

Agenda (or Why bother with this talk?)

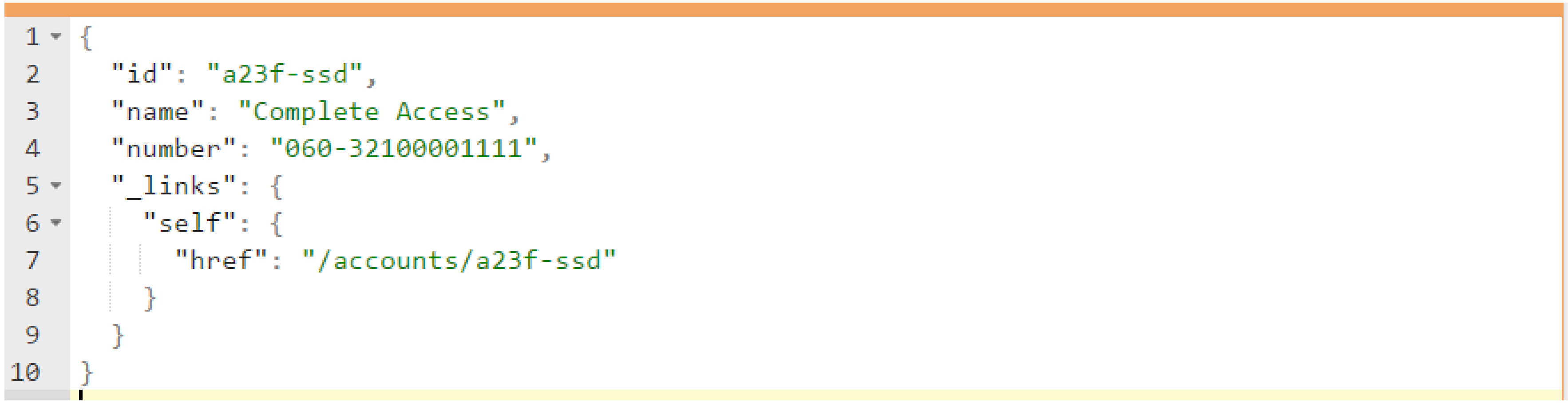
- Standards, guidelines and the rest of the noise
- How to deal with the problems that Dev’s & QA’s face
  - E.g. Versioning, Caching, Validation, Tracing
- Alternates like GraphQL
- Answer any questions

Why business requirements over standards?

- Unfortunately we have deadlines
- Security, Caching, Independence, Scalable etc. are hard
- Some guidelines are hard or might not work for us
  - Specifically HTTP + JSON
  - HATEOAS (Hypermedia As The Engine Of Application State)
- Make SSL mandatory

Examples

- GET /accounts/a23f-ssd



```json
{
  "id": "a23f-ssd",
  "name": "Complete Access",
  "number": "060-32100001111",
  "_links": {
    "self": {
      "href": "/accounts/a23f-ssd"
    }
  }
}
```

- But what if the response was an image?

How to deal with …?

Request & Response

- Verbs – GET PUT POST DELETE ~~HEAD PATCH CONNECT OPTIONS~~
  - Not really CRUD
- Headers – Accept, Content, Cache
- Status Codes
  - Success: 200 201 ~~202~~ 204
  - Redirection: ~~301 302~~
  - Client Error: 400 401 403 404 ~~409 413 415~~
  - Server Error: 500 503
- What about Facebook 200 OK with response body having error details?

Request & Response

- Nouns not Verbs
- Collections and instances
- CONTENT Header & ACCEPT Header
  - Accept application/json, text/plain
- Filter, Sort and Pagination

Versioning

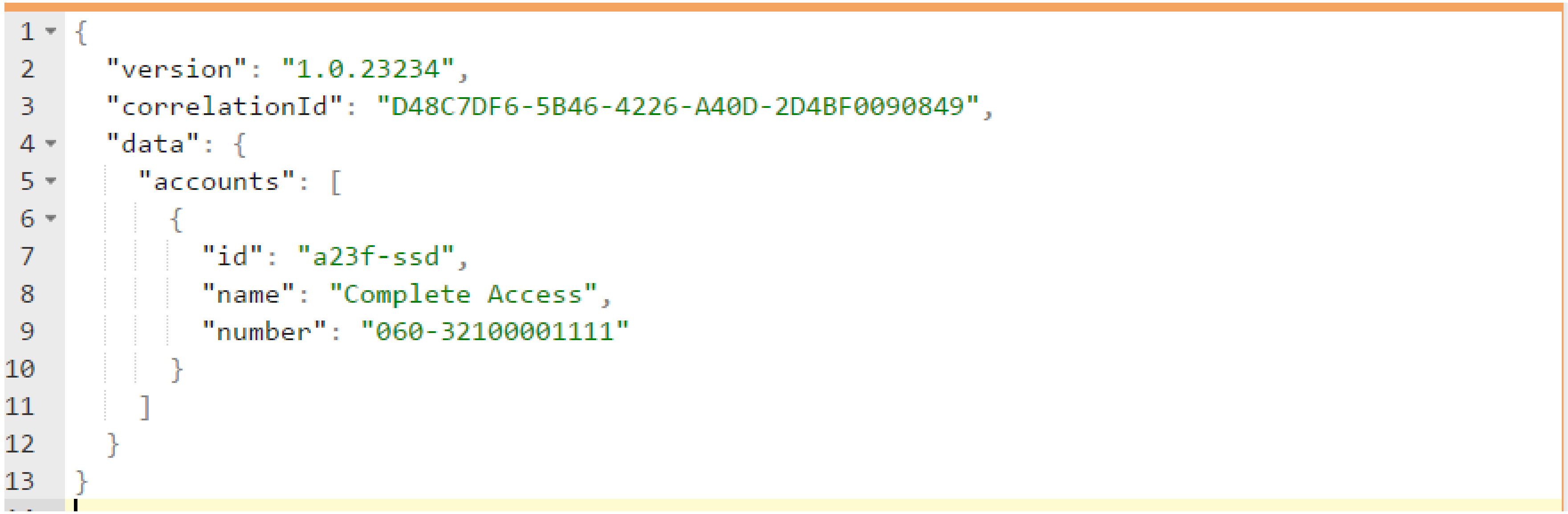
- Headers vs URL vs Query String
- Treat version as feature toggle (branching by abstraction)
- Start with single deployment for multiple versions
- Separate deployments for separate versions (SLA based)
- GIT: master branch vs separate version branches vs separate forks
- Semantic versioning of APIs

Tracing

- Start with sending back a Correlation-Id
- Honour any Correlation-Id
- Pass this to calling systems (APIs, DBs etc)
- Read about Dapper (Distributed Tracing Google White Paper)

Examples

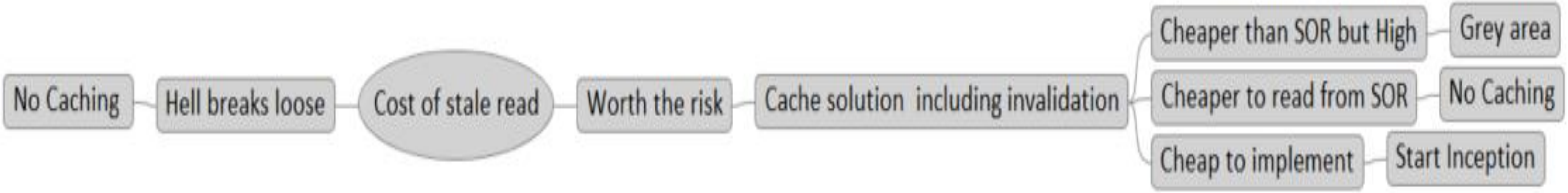
- GET /accounts/a23f-ssd



```
{
  "version": "1.0.23234",
  "correlationId": "D48C7DF6-5B46-4226-A40D-2D4BF0090849",
  "data": {
    "accounts": [
      {
        "id": "a23f-ssd",
        "name": "Complete Access",
        "number": "060-32100001111"
      }
    ]
  }
}
```

Caching

- Complex problem to solve
- Set to no-cache to start of with
- This takes out any unknown behaviour from day one
- ETag & Last Modified



No Caching

Hell breaks loose

Cost of stale read

Worth the risk

Cache solution including invalidation

Cheaper than SOR but High

Grey area

Cheaper to read from SOR

No Caching

Cheap to implement

Start Inception

Different views of the same Resource

- GET /accounts/060-32100001111/views/balance
- GET /accounts/060-32100001111/views/tplus1
- GET /accounts/060-32100001111?view=balance&expand=agent
- GET /accounts?view=agentdetails
- GET /views/accounts/realtimebalance
- GET /views/accounts/agentdetails
- PREFER header (ODATA - Open Data Procotol)

Validation, Errors & Warnings

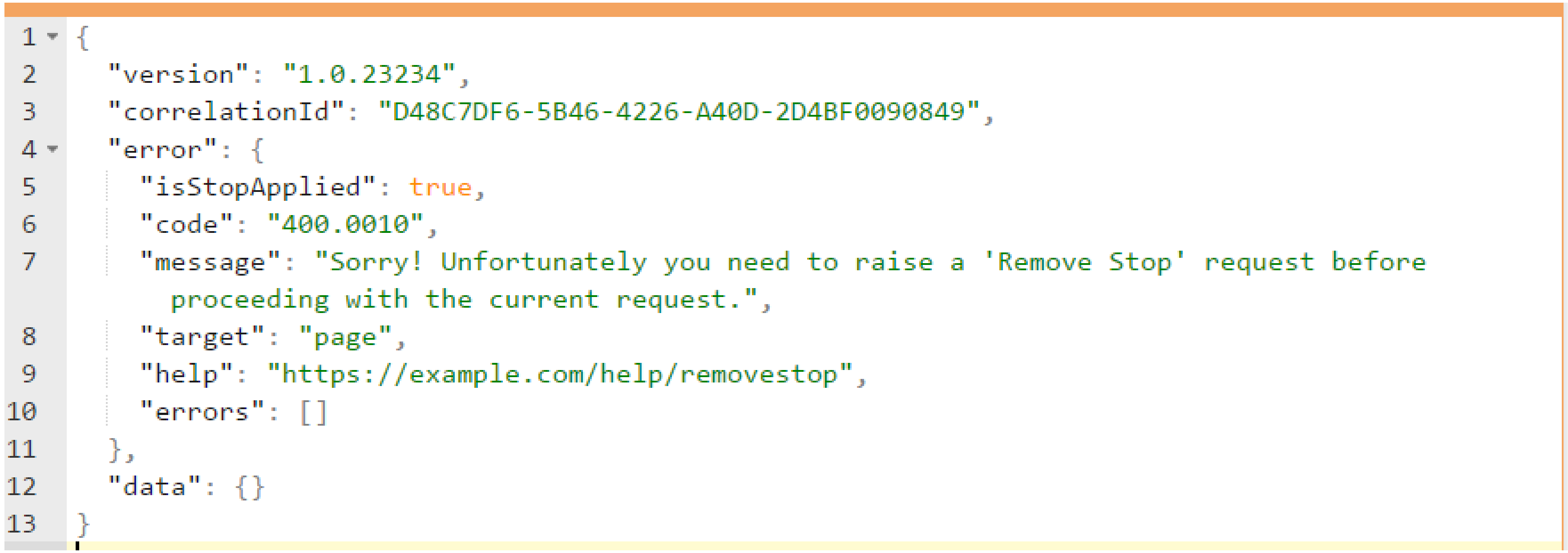
- Validating a request before submitting
- Structure to the response
- Many servers in between
- 4XX or 5XX with a known structure makes it easier to differntiate
- Link to documentation if possible
- Auto mappers & Fluent Validators – Good or Bad?
- Read about Problem Details for HTTP APIs (application/problem+json)

Examples

- POST /accounts/actions/validate
  - 200 OK
  - 400 Bad Request
- POST /validators/accounts
  - 200 OK
  - 400 Bad Request
- POST /accounts followed by POST /accounts/a23f-ssd/states
  - 2xx (internally stored as initiated and then created)
  - 4xx

Examples

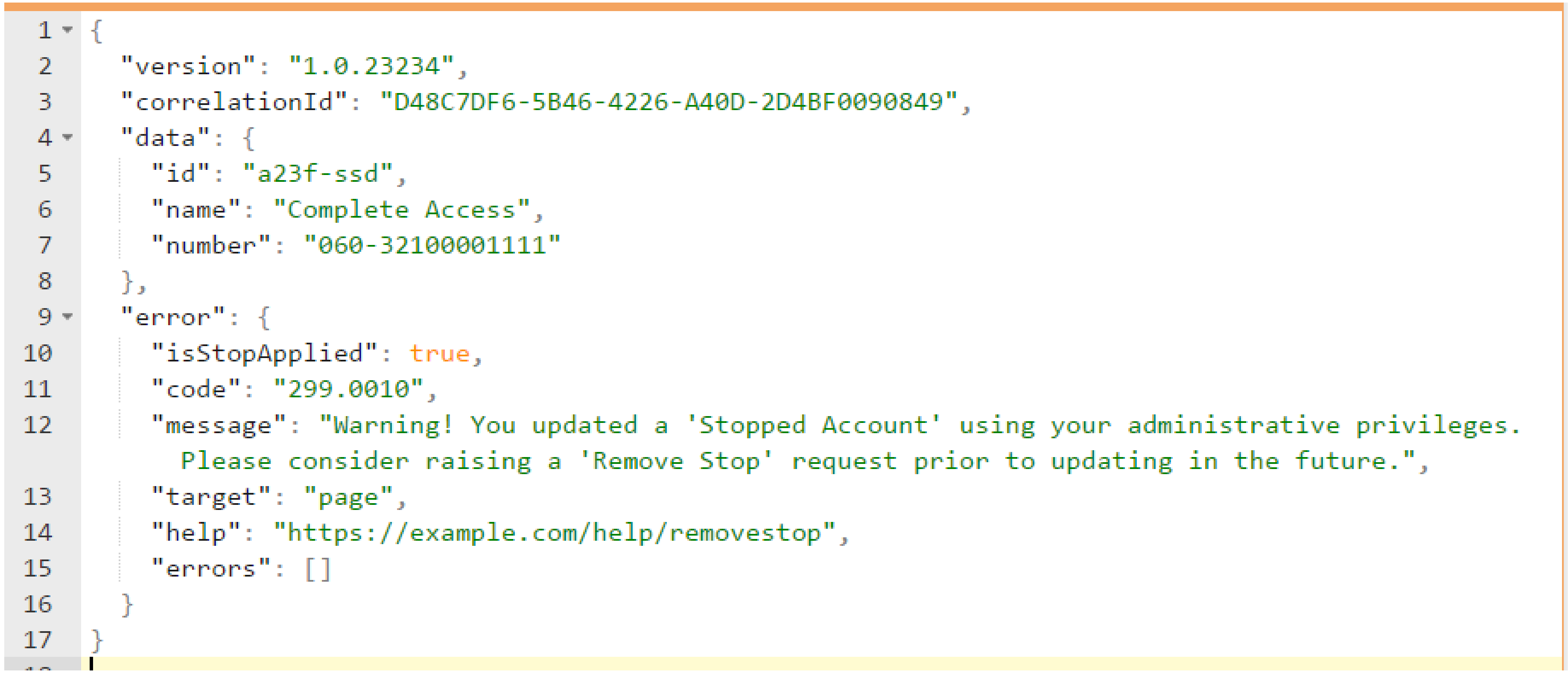
- PUT /accounts/a23f-ssd
- 400 Bad Request



```
{
  "version": "1.0.23234",
  "correlationId": "D48C7DF6-5B46-4226-A40D-2D4BF0090849",
  "error": {
    "isStopApplied": true,
    "code": "400.0010",
    "message": "Sorry! Unfortunately you need to raise a 'Remove Stop' request before proceeding with the current request.",
    "target": "page",
    "help": "https://example.com/help/removestop",
    "errors": []
  },
  "data": {}
}
```

Examples

- PUT /accounts/a23f-ssd
- 200 OK



```json
{
  "version": "1.0.23234",
  "correlationId": "D48C7DF6-5B46-4226-A40D-2D4BF0090849",
  "data": {
    "id": "a23f-ssd",
    "name": "Complete Access",
    "number": "060-32100001111"
  },
  "error": {
    "isStopApplied": true,
    "code": "299.0010",
    "message": "Warning! You updated a 'Stopped Account' using your administrative privileges. Please consider raising a 'Remove Stop' request prior to updating in the future.",
    "target": "page",
    "help": "https://example.com/help/removestop",
    "errors": []
  }
}
```

Maintenance & Deprecation

- APIs have a life cycle
- Months to Years
- Give clients option to test and stub
- Have contract tests to honour client calls
- Read about PACT
- Start with a readme file
- Swagger vs RAML vs API Blueprint vs …

Security

- Basic Authentication
- OAuth
- SASL
- Certificates

Rate Limitation

- Rate limiter throttling
- Rate limiter response headers
  - X-Rate-Limit vs X-RateLimit
    - Handle Limit
    - Handle Remaining
    - Handle Reset

Alternates & Future

- GraphQL
- REST for IoT devices

Summary

- Why pragmatism helps
  - Respecting Dev’s & QA’s
- Opinionated: How to deal with common situations
- Future of REST APIs

Questions?