

Mutating Information Efficiently

Best Practices Series



The Google Ads API Developer Series



David Wihl

Developer Relations Engineer

Mutating Defined

- [Mutating](#) is the means of creating, updating or removing [resources](#)
- Most resources have a corresponding mutate service:
 - For example, [Campaign](#) has [CampaignService.MutateCampaigns](#)
 - A few services have special method signatures, for example
`RecommendationService.ApplyRecommendationRequest`

Anatomy of a mutate in a single service

CampaignService

Mutate Campaign Request

Campaign operations[]

Operation 1: create campaign 1

Operation 2: update campaign 2

Operation *n*: remove Campaign *n*

Google
Ads API

Mutate Campaign Response

<resource>results[]

Operation 1: resource1 name

Operation 2: resource2 name

Operation *n*: resource *n* name

Grouping / ungrouping of operations

- In a single request, all operations either complete successfully or fail together
- **Unless:**
 - [Partial failure](#) is set - the set of operations is now ungrouped
 - There is a backend process, like [conversion uploads](#)
 - The API may report success, but the backend process may fail later on
 - Therefore, requests are not atomic

Multiple services

What if there are dependencies?

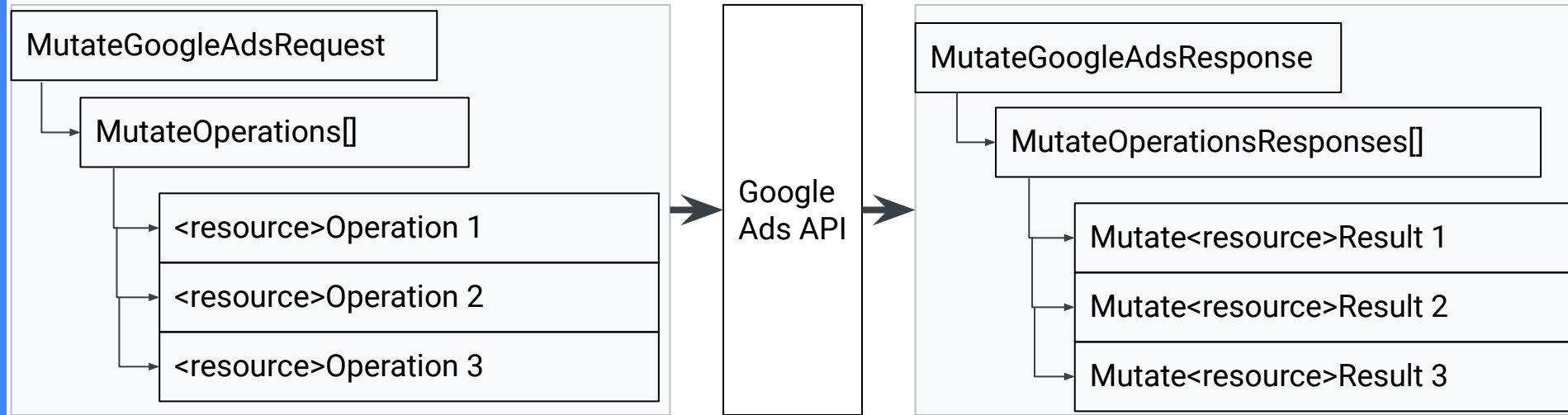
- **Scenario:** Adding an App ad campaign ([code example](#)) requires
 - Create a budget
 - Create a campaign
 - Set campaign targeting
 - Create an ad group
 - Create an App ad
- What if a step fails? How do you handle errors gracefully?
 - If not, some orphaned resources (like the budget) may exist

How to group operations across services

- The [GoogleAdsService.Mutate](#) endpoint can mutate all resources with two advantages:
 - group mutates *across multiple services* in one request
 - use temporary resource names

Anatomy of a mutate in the GoogleAdsService

GoogleAdsService



Temporary IDs are placeholders for forward references

REST

```
POST /v1/customers/1234567890/googleAds:mutate
```

```
mutate_operations: [  
  {  
    campaign_operation: {  
      create: {  
        resource_name: "customers/<YOUR_CUSTOMER_ID>/campaigns/-1",  
        ...  
      }  
    },  
    ad_group_operation: {  
      create: {  
        campaign: "customers/<YOUR_CUSTOMER_ID>/campaigns/-1"  
        ...  
      }  
    }  
  }  
]
```

New



Temporary IDs will be assigned the real IDs by the Google Ads API server

Reference



[Python example](#) to create unique Temporary IDs

Convenience of Grouping Operations

- In the vast majority of cases, grouping operations in a single request makes sense
 - No orphaned objects. Create a budget, campaign and ad group as one logical group of actions
- It is more efficient

Best Practice

Group operations whenever possible in a single request

By service: `<resource>Service`

Across services: `GoogleAdsService.Mutate`

Use temporary IDs for forward references

Thanks for watching

- We welcome your feedback at googleadsapi-support@google.com
- Check out the full [Best Practices Series](#) playlist for related content
- See more episodes on our [YouTube channel](#)

