



Snapchat Auto-Bidding Solution Proposal

Problem: Nike's Social team must periodically update Ad Set/Squad bids in the Snapchat platform in accordance with their current campaign and platform performance and objectives.

Current Solution: The team decides how to adjust bids in [this spreadsheet](#), but then must manually go into the platform to adjust bids.

Opportunity:

- Save 0.5 hours per week
- Develop an automation scalable to other clients and platforms

Proposed Solution: Using Snapchat API, automatically update new bids from the spreadsheet into the Snapchat platform. This solution doesn't replace the human aspect of *deciding* new bids, but merely makes it easier to auto-update them in the platform.

Solution Architecture:

- **Data Source:** Use the existing bid spreadsheet, tab `Commerce: Seasonless/Always On [Bid Monitoring]` with columns: Ad Set Name (`ad_squad_id`), Bidding Type (`bid_strategy`), Effective Date (no matching parameter in API call but this indicates *when* we will run the update), and Bid (`bid_micro`).
- **Authentication:**
 - Do we have API access yet?
 - If not: create a Business account → Ads Manager → Business Dashboard → +OAuth App
 - Then, use Bearer access tokens in `Authorization: Bearer <token>`
 - Scope = `snapchat-marketing-api`
- **Update Flow:**
 1. **Initial Setup:**
 - a. Pull all Ad Squads in account and build a mapping. Set to refresh daily.
 - b. Set update script to run on a weekly Apps Script.
 2. **Read the Spreadsheet:** Filter rows where Effective Date ≤ today and status != applied.
 3. **Get Current Parameters:** Run a GET call to the API to get current parameters that will remain unchanged and we will need to pass into our PUT call.
 - a. GET /v1/adsquads/{ad_squad_id}
 4. **Update an Ad Squad:** body contains an adsquads array with objects including: `ad_squad_id`, `campaign_id`, `name`, `type`, `placement_v2`, `optimization_goal`, `bid_strategy` (`TARGET_COST`), `bid_micro`, `daily_budget_micro`/`lifetime_budget_micro`, `targeting`, etc.
 - a. PUT /v1/campaigns/{campaign_id}/adsquads
 5. **Logging:** write back "Applied", timestamp, and exact payload sent.

- a. Raise error if Ad Squad isn't found.
- **Rate Limits:** ~20 requests/sec per app and ~10 requests/sec per token (average).
 - Batch ~50–100 squads per call (payload size dependent), backoff on 429, and cap to ~8–9 rps per token to be safe.
 - NB: Snapchat uses **micro-currency**; 1.00 USD → 1,000,000 micros.

Estimated Cost of Proposed Solution:

- **Build time:** ~2–3 days to prototype
- **Cost:** <\$10/month at low weekly volume

References & Documentation:

- Ads API Intro & setup:
<https://developers.snap.com/api/marketing-api/Ads-API/introduction>
- OAuth overview: <https://marketingapi.snapchat.com/docs/profile.html> Snap for Developers Snapchat Marketing API
- Bid Estimate: <https://developers.snap.com/api/marketing-api/Ads-API/bid-estimate> Snap for Developers
- Strategy definitions:
https://businesshelp.snapchat.com/s/article/bidding-strategies?language=en_US
- Auto-bid overview:
https://businesshelp.snapchat.com/s/article/auto-bidding?language=en_US .
- API note: [businesshelp.snapchat.com+1Snap for Developers](https://businesshelp.snapchat.com+1Snap+for+Developers)
- Rate limits: <https://developers.snap.com/api/marketing-api/Ads-API/rate-limits>
- Endpoint & field reference (with example):
<https://developers.snap.com/api/marketing-api/Ads-API/ad-squads#bid-strategy>